IMPROVED MAPPING ACCURACY OF PLANETARY SURFACES USING SUPER-RESOLUTION OF THERMAL INFRARED DATA

by

Christopher Gerald Hughes

B.S. Computer Science, Dickinson College, 1996

Submitted to the Graduate Faculty of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

ARTS AND SCIENCES

This dissertation was presented

by

Christopher Gerald Hughes

It was defended on

November 18, 2010

and approved by

Thomas H Anderson, Professor, University of Pittsburgh

Daniel J Bain, Assistant Professor, University of Pittsburgh

Joshua L Bandfield, Research Assistant Professor, University of Washington

William Harbert, Professor, University of Pittsburgh

Charles E Jones, Lecturer, University of Pittsburgh

Dissertation Advisor: Michael S Ramsey, Associate Professor, University of Pittsburgh

IMPROVED MAPPING ACCURACY OF PLANETARY SURFACES USING SUPER-RESOLUTION OF THERMAL INFRARED DATA

Christopher Gerald Hughes, PhD

University of Pittsburgh, 2011

Super-Resolution is the process of obtaining a spatial resolution greater than that of the original resolution of a data source. This can be done through the fusion of original data with an additional source that has the desired resolution. These approaches can either be qualitative for visual appeal, quantitative for data accuracy, or some combination of both. The super-resolution approach offers an alternative to traditional sub-pixel deconvolution identification and provides higher resolution TIR data for Earth and Mars.

The Thermal Emission Imaging System (THEMIS) has provided the highest spatial resolution (100 meter / pixel) thermal infrared (TIR) data of the Mars surface to date. These data have enabled the discovery of small-scale compositional units and helped to constrain surface processes operating at these scales. Higher resolution visible instruments have revealed smaller-scale differences, creating a need to detect compositional variability using TIR data at scales below 100 meters. Putative chloride deposits identified on Mars are one such area. These deposits have a unique spectral signature in the TIR and are present within topographic lows. The super-resolution algorithm helped constrain the local mineral assemblages and stratigraphic order. This data reveals that associated phyllosilicate-rich units may be part of a common lithostratigraphic unit with a phyllosilicate-poor ST-2 material.

Lunar Lake playa, located ~100 km northeast of Tonopah, Nevada, has been used as an analog site for multiple planetary surfaces and as a vicarious calibration site for Earth-orbiting satellites. As such, the ability to obtain higher resolution data through super-resolution has the potential to improve Earth data and give to insight into the formation of similar environments on other planetary surfaces. Super-resolved data show Lunar Lake playa to be more compositionally heterogeneous than previously thought. A gradation of mineralogy exists within the playa, seen in both super-resolved data and in samples collected during fieldwork. The composition of the playa is influenced by the immediate surroundings, with variation existing between the western side of the playa, bounded by basaltic units, and the eastern, bounded by rhyolitic tuff. As the surrounding material weather, different clasts are transported onto the playa, and weather into different mineral assemblies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

Catch that light
It falls in subtle patterns
("Dollskin", The Toadies)

Late in the year 2000, I decided that it would be a good idea to quit my job and go hike the Appalachian Trail. On April 4, 2001 I stood on top of Springer Mountain, Georgia, clean-shaven and bald like a cue-ball. Six months later, I was in Maine, with hair falling in my eyes and a beard down to my chest. That time inspired me to make a career change, from the computer security I had loved to something outside. My first thanks must go to my fellow Walking Wounded for inspiring this life changing decision.

Somehow, I stumbled into Geology and Planetary Science, despite my initial plans of Forestry. I owe special thanks to three faculty at Pitt for this. Charlie Jones and Tom Anderson have inspired me from my first semester back in college. Both TA and Charlie have inspired a deep and abiding love for the mysteries of the Earth, and its beauty as a system. I owe an even greater thanks to Michael Ramsey. Mike took me in as an undergrad and when I applied to graduate school, Mike was kind enough to write a letter of recommendation for me…to himself. Whatever my successes as a Scientist, I owe them to Mike's mentorship.

My committee helped me get through this process as well. They deserve an award for accepting a 417 page draft document from me, including one chapter of 127 pages. Their edits greatly improved this dissertation AND greatly reduced its heft. My external advisor, Josh Bandfield, gave me advice numerous times, and was always helpful with all things THEMIS. I'd also like to thank Hideyuki Tonooka, the originator of this super-resolution method, and Micki Osterloo for the discovery of the "putative" chlorides – I don't think that word is needed.

In science, we get paid to go to school! I'd like to thank Phil Christensen and the THEMIS Science Team for funding me eight months of the year, every year. The other four months, I have been fortunate enough to be supported as a Teaching Assistant. TAing can be good or bad, but I've been extraordinarily fortunate in my TAing. My two years of TAing Mineralogy and three years of Structure labs have shown me that I want to teach. My students have been wonderful; they have tolerated, and perhaps even embraced and encouraged, my insanity in the classroom. I have loved every moment I have spent with them. I've also been fortunate enough to be employed as an Adjunct by the department for Remote Sensing, as well as by Duquesne University for Core Astronomy and Core Geology. Duquesne truly went above and beyond in their treatment of me while I worked there, and in their support of me afterword as well. Amanda and Simonetta are two of the best people I know.

It is impossible to get through graduate school without the love and understanding of your family. Dad, Mom, Jim, and Laura have supported me in any way they could. They listened when I needed to rant, nodded at the appropriate times when I started dricking out, and they

made sure I was sleeping and eating. My little brother Cory may have been subjected to it even more, having been my housemate too. Cory is one hell of an inspiration; that kid can find the hardest way to do difficult things, and then accomplishes what he sets his mind on.

Between two years as a post-bac in the department and six years working on this Ph.D, I have made a large number of friends. I have known Katie and Liz, two of my current officemates, since my start at Pitt. Katie was the very first person I ever met in Pittsburgh. My IVIS cohort of Adam, Steve, Shellie, and Rachel are all graduated now, all with Ph.Ds. Kevin, the "new kid" seems well prepared to carry on our traditions. Despite graduate school's habit of eating lives, I have friends outside of the department too: Veronica, Chris, Bob, Patrick, and the rest have worked (or played!) to keep me semi-sane through all of this. Other friends – Bobby, Daniel, Holly, James, Lindsay, and everyone else – have done the same, particularly my lunchmates. There is only so much space for Thank You's, and I owe many more than can fit.

Alison Graettinger deserves immense thanks. She was crazy enough to start dating me as I began the writing process, and has supported me when I needed it. I am constantly in awe of her, and cannot even begin to explain how amazing she is to me. May all your gargoyles be comfortable, and your middle-class / nobility wars always involve marriage-breaking treaties.

This dissertation brought to you by Pizza Sola, Coca-Cola, and all manufacturers of Oatmeal Creme Pies and Circus Peanuts. It has also been said that without music, life has no meaning. Girl Talk is the official soundtrack of edits, and The Pixies and The Toadies of writing.

1.0 SUPER-RESOLUTION OF THEMIS THERMAL INFRARED DATA: COMPOSITIONAL RELATIONSHIPS OF SURFACE UNITS BELOW THE 100 METER SCALE ON MARS

1.1 INTRODUCTION

Compositional and/or thermophysical mapping of the martian surface using multispectral or hyperspectral thermal infrared (TIR) data has been ongoing since the earliest TIR instruments sent to Mars (e.g., Herr et al., 1970; Kieffer et al., 1976; Peterfreund et al., 1977). However, it was not until the higher spectral resolution of the Thermal Emission Spectrometer (TES) and, later, the higher spatial resolution of THEMIS that quantitative mineral mapping was performed. Previous workers have used: TES data to positively identify numerous minerals on the surface (e.g., Bandfield, 2006; Bandfield et al., 2004a; Christensen et al., 2003; Hamilton and Christensen, 2005), THEMIS data to detect smaller scale outcrops (e.g., Christensen et al., 2003; Rogers et al., 2005), or a combination of both instruments to detect a compositional difference with THEMIS and then accurately identify those minerals with TES (e.g., Bandfield et al., 2004a).

At the native IR resolution of THEMIS, there have been numerous important scientific discoveries about the surface composition and processes of Mars. These have included the existence of thermally-distinct layered units at the tens of meters scale (Christensen et al., 2003); the occurrence of kilometer-scale exposures and outcrops of bedrock (Christensen et al., 2003; Rogers et al., 2005); particle size differences between dune and interdune regions based on thermal inertia (Christensen et al., 2003); the discovery of olivine rich basalts within Ganges Chasma (Christensen et al., 2003) and Nili Fossae (Hamilton and Christensen, 2005), confirmation of water ice deposits at the southern polar cap (Christensen et al., 2003); an aqueous origin for some Martian deposits (Glotch and Christensen, 2005; Glotch and Rogers, 2007); the cause and formation of polar "dark spots" (Kieffer et al., 2006); gullies formed from the melting of snow deposits (Christensen, 2003); evolved compositions (Bandfield, 2006; Bandfield et al., 2004a; Christensen et al., 2005); properties of Martian dunes (Fenton, 2005; Fenton and Mellon, 2006; Hayward et al., 2007); generation of higher resolution thermal inertia maps (Fergason et al., 2006); and identification of putative chloride deposits (Osterloo et al., 2008). A majority of these studies have relied on spectral deconvolution techniques (e.g., Ramsey and Christensen, 1998) using a spectral library to map mineral percentages at or below the spatial scale of the data. This approach generally works well, but is limited by the breadth of the spectral library (e.g., the number and particle size of the minerals used to compose the library), the scale of the areal mixing versus pixel size of the instrument, the presence of errors/noise in the data, and the ground conditions (such as sub-pixel temperature heterogeneities or dust cover) at the time of data collection.

In particular, one interesting study that combined spectral deconvolution with data from both THEMIS and TES was that of Bandfield et al. (2004a). The authors presented a study of the identification of a spectrally-anomalous region within two craters in Syrtis Major. This area, initially identified using THEMIS decorrelation stretch (DCS) images (Gillespie et al., 1986), had an absorption band at a shorter wavelength than the surrounding material, indicating it was likely a more evolved composition. Subsequent analysis using deconvolution of TES data identified this material to be quartzofeldspathic with a composition most akin to a quartz monzonite. Bandfield et al. (2004a) showed that this unit was confined to the intercrater materials surrounding the central peak, and explained that this quartz-bearing unit could have formed as an intrusion at depth, and possibly brought to the surface during the uplift of the central peak. Ehlmann et al. (2009) alternatively proposed the presence of hydrated silica and other alteration products using CRISM data in this same location. This region was chosen as the focus for this work because of the contrasting spectral variation of the surface units, the presence of only small-scale outcrops, the predominance of eolian mantling, and the uncertainty of a source for the spectrally-anomalous material. The application of the new super-resolution approach using THEMIS VIS data as the higher spatial resolution source improved TIR spatial data resolution by a factor of three. Using these new data, small-scale outcrops and compositional unit contact boundaries have been explored in an effort to better constrain the source and emplacement processes of this unusual compositional unit on Mars.

1.1.1 Super-Resolution

Satellite data collection is frequently limited by numerous trade-offs between the spatial, spectral, and temporal resolution of the instrument. In certain circumstances, satellites with

multiple instrument payloads may require a higher orbit for some of the instruments, thus reducing the pixel size or requiring a larger telescope for imaging spectrometers. Furthermore, instruments on satellites orbiting other planets may be data limited due to the bandwidth dedicated to that mission by the Deep Space Network. Finally, the goals of higher temporal resolution and/or complete planetary coverage will always conflict with the goal of higher spatial/spectral resolution data. As a result of these trade-offs, it is rare to have an instrument that collects data at an ideal spatial resolution for a particular analysis of the surface, and that also satisfies most or all of the previous constraints. This has lead to a series of creative image processing techniques, commonly grouped under the general term of "super-resolution", which enhance the spatial resolution of data in some way (e.g., Tonooka, 2005; Zhukov et al., 1999).

Improved IR spatial resolution on Mars is needed in order to better describe processes on the martian surface that operate below the 100 meter scale. These include small-scale sediment mixing in eolian and crater ejecta deposits, igneous and eolian layered deposits, as well as thermal heterogeneities that lead to anomalous thermal inertia and emissivity recovery. Super-resolution can provide a means to improve the spatial resolution by incorporating the spectral information of data with a higher spatial resolution. These data can be collected by the same instrument or in conjunction with other higher-resolution Mars-orbiting instruments such as the Compact Reconnaissance Imaging Spectrometer for Mars (CRISM) or the High Resolution Stereo Camera (HRSC), provided that the fundamental assumptions, such as similar surfaces exist under similar atmospheric conditions and illumination, of the technique are not violated (Tonooka, 2005). However, super-resolution using data from instruments on different satellites is technically possible, but would require using data that are not widely separated in time in order

to mitigate significant surface / atmospheric changes. Data would need to be collected during similar seasons and without any intervening dust storms or other major events. Differing heating conditions between datasets may also have the potential to cause undesirable results when using non-contemporaneous data.

Super-resolution is the process of creating a higher spatial resolution than that of the original (or native) data source. This can be done through techniques that fuse the original data with an additional higher resolution data source. The most commonly-applied technique is pan-sharpening, in which a single high-spatial resolution channel is used to enhance multi-spectral lower resolution data (Figure 1-1). This can be done in a variety of ways, including transforming the lower spatial resolution image to Intensity-Hue-Saturation space, and substituting in the single higher resolution channel for the image intensity (Garzelli et al., 2004; Wang et al., 2005; Zhang, 2004; Zhukov et al., 1999) or through Principle Component Analysis (PCA), in which the first principle component of the lower resolution multi-spectral data is replaced with the higher-resolution channel (Garzelli et al., 2004; Wang et al., 2005; Zhang, 2004; Zhukov et al., 1999). The Brovey Transform (Pohl, 1999; Wang et al., 2005; Zhang, 2004) is an arithmetic approach, in which each channel of the lower resolution data is multiplied by the higher resolution channel, and the result is divided by the sum of the lower-resolution channels. Newer methods to fuse a single high-resolution channel to lower resolution multi-spectral data are based on wavelet transforms, such as the Mallat or à trous algorithms (Aiazzi et al., 2002). The work of Burt and Adelson (1983) used a Laplacian Pyramid to fuse multi-spectral data from lower and higher resolution channels together. The Multisensor Multiresolution Technique (MMT) of Zhukov et al. (1999) also works with multispectral data at both higher and lower spatial resolutions.

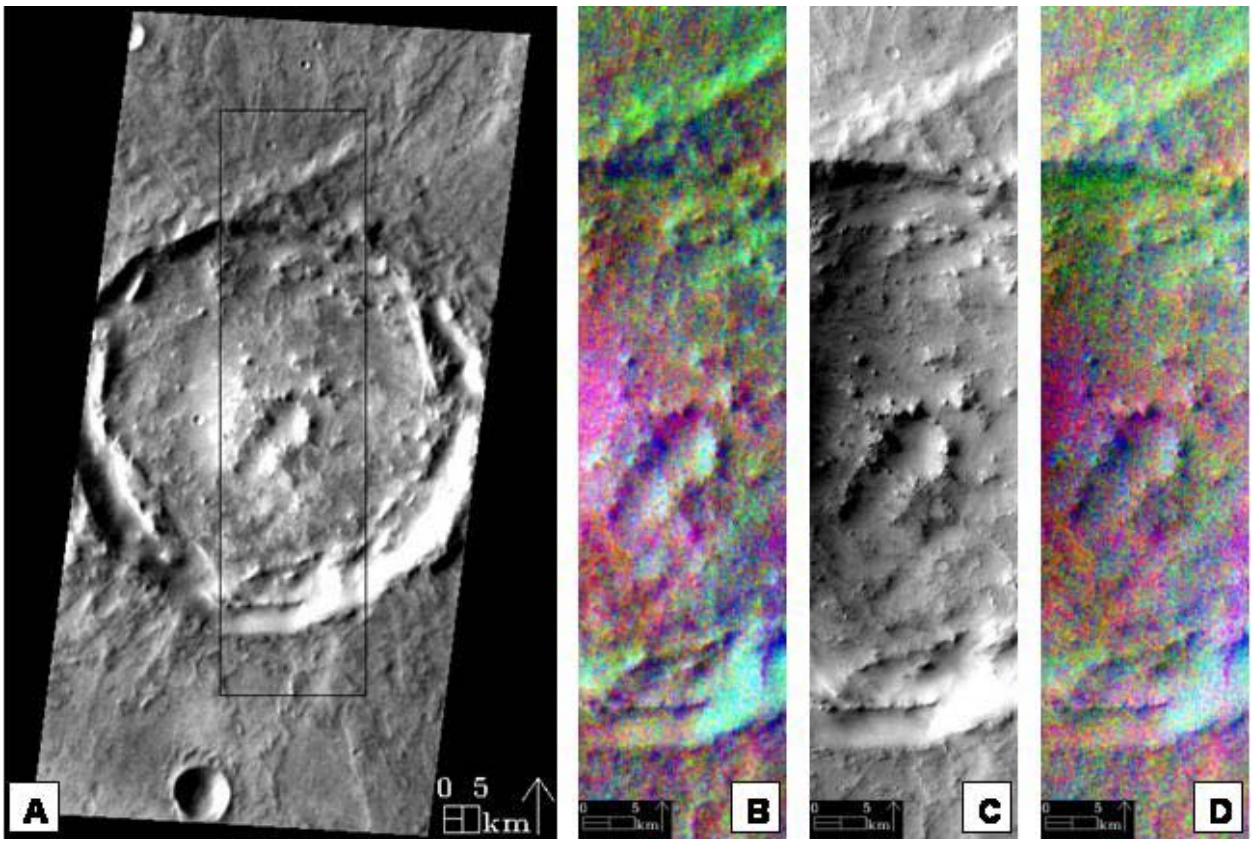

Figure 1-1 Comparison of the original TIR data (THEMIS image I12929010) with other pan-resolved techniques. (A) Daytime TIR channel 9 radiance data with the rectangle denoting the area shown in (B), (C), and (D). (B) Decorrelation stretch of TIR radiance data (bands 9, 6, and 4), from the original resolution data. (C) Channel 9 radiance data after principal component (PC) transform and replacement of PC band 1. This image shows significant improvement in spatial resolution compared to (A), and has strong visual appeal. (D) Decorrelation stretch of TIR bands 9, 6, and 4 radiance data from PC-transformed data. The data in (D) differ significantly from the data in (B), reflecting the fact that the PC-transform pan-resolution technique creates images with good visual appeal but also with loss of radiometric accuracy.

However, in all of these techniques there is a trade-off between visual appeal and ones that are most radiometric accuracy (Zhukov et al., 1999). It is rare to find a super-resolution methodology that is both quantitatively accurate and qualitatively acceptable. The super-resolution technique presented here is a significant modification of an algorithm first described by Tonooka (2005) and applied to data from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) of an urban target in Japan.

1.1.2 Instrument Datasets

The ASTER instrument is composed of three separate wavelength sub-systems, all using separate telescopes. ASTER has 3 channels between 0.52 and 0.86 microns at 15 m/pixel spatial resolution, 6 channels in the 1.6 to 2.43 micron short wave infrared (SWIR) region with at a spatial resolution of 30 m/pixel, and 5 channels between 8.13 and 11.65 microns at 90 m/pixel spatial resolution (Fujisada et al., 1998). The Signal to Noise Ratio (SNR) values range between 44 – 368 for low gain radiance values, and 156 – 466 for high gain radiance values (Fujisada et al., 1998). The VIS instrument uses a 5000 x 4 element array of silicon detectors; the SWIR instrument uses a 2048 x 6 element array of PtSi detectors. Both form images with a pushbroom configuration. The TIR instrument is a 10 x 5 array of HgCdTe elements used in a whiskbroom configuration (Yamaguchi et al., 1998). The FOV for all three instruments is 6.09° square. Since January 2009, ASTER has been acquiring multispectral / multispatial data in only two wavelength regions. Data from the SWIR instrument are no longer usable due to a failed cryo-cooler. Therefore, super-resolution of ASTER data after this date (as well as all THEMIS data) utilizes only the two datasets – the lower resolution IR and the higher VIS.

The Mars-orbiting THEMIS instrument (Christensen et al., 2004) has a similar spatial scale and spectral range as the Earth-orbiting ASTER instrument (Yamaguchi et al., 1998). THEMIS utilizes the same telescope for the VIS and IR instruments, with 5 VIS channels between 0.40 and 0.88 microns at 18 m/pixel spatial resolution, although pixel summing is commonly used to acquire images over larger areas at 36 m/pixel resolution (Christensen et al., 2004). THEMIS also has 9 IR channels at 100 m/pixel spatial resolution between 6.27 and 12.98 microns (the first two of which are duplicated for better signal to noise). An additional tenth channel at the same spatial resolution is centered at 14.88 microns and used for detection of atmospheric opacity (Christensen et al., 2004). The SNR varies from 50 (in the shortest TIR wavelengths, band 1 / 2) to 143 (TIR bands 3 through 9) at 245 K; the VIS instrument has a SNR intermediate to these values (Christensen et al., 2004). The VIS instrument contains a 1024 x 1024 array of silicon elements with a 2.66° by 2.44° field of view (FOV) , whereas the IR instrument has a 320 x 240 array of uncooled microbolometers elements with a 4.6° (cross track) by 3.5° FOV (Christensen et al., 2004). The VIS instrument has a 5 stripe filter, with all elements exposed simultaneously (McConnochie et al., 2006). Groundtrack motion is also used to expand spatial coverage (McConnochie et al., 2006), making it both a framing and pushbroom system. The IR instrument has a ten stripe filters, representing nine separate wavelength ranges (bands 1 – 10) with an image being formed in a pushbroom fashion (Christensen et al., 2004).

1.2 METHODOLOGY

1.2.1 Current Approach

The super-resolution process (Hughes et al., 2007; Tonooka, 2005) is organized into five stages (Figure 1-2), which are repeated if more than two datasets in different wavelength regions and spatial resolutions are available. For example, ASTER from 2000-2008 had three distinct subsystems and therefore the five stages of the model would first operate to super-resolve the 30 m/pixel SWIR data to the 15 m/pixel VIS data. The stages are then repeated to super-resolve the 90 m/pixel IR data to those 15 m/pixel VIS and SWIR images. Tonooka (2005) applied this methodology to the multi-resolution, multi-spectral ASTER data acquired in 2002. THEMIS has slightly better spectral resolution than ASTER in both wavelength regions, which allows the modified super resolution technique to more readily identify surface compositional variations and thus produce better results.

In the first stage of the super-resolution approach, the higher resolution dataset is convolved with the Point Spread Function (PSF) of the lower resolution instrument. This allows the image resolution to be degraded to match that of the lower resolution data. These degraded pixels are tested for homogeneity by comparison with the original co-located higher resolution data, and homogeneous pixels are selected. A homogeneous pixel is one in which the standard deviations, calculated on a per-band basis, of all the higher-resolution pixels co-located within it are less than a single threshold value. The threshold value is the average of the standard deviations for each band for the whole image, resulting in a quick method to determine if the co-located pixels are sufficiently similar to be considered homogeneous. The homogeneous pixels in

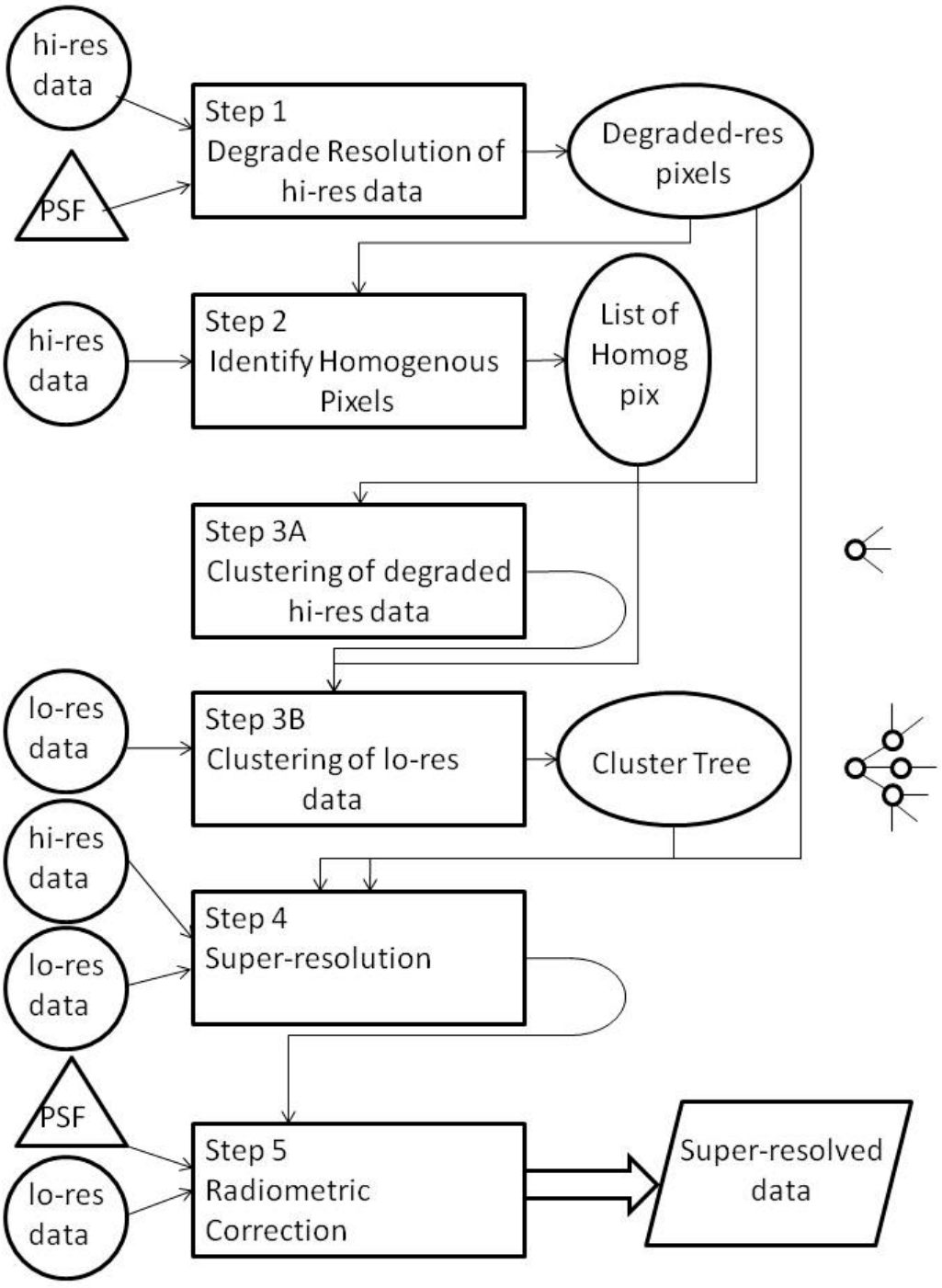

Figure 1-2 The super-resolution algorithm showing the relationship between the five stages as well as the input and output datasets. The ball and stick figures to the right of steps 3A and 3B represent the building of the cluster tree. The initial branches represent VIS clusters form in step 3A. Secondary branches represent co-located IR clusters from step 3B.

stage three are clustered together based on spectral similarity, as measured by Mahalanobis Distance (MD). Mahalanobis distance is analogous to Euclidian distance, but unit length is not constant in all dimensions, rather a single unit is equal to a standard deviation in that dimension. This permits the measurement of differences relative to the data being measured, rather than to an arbitrary fixed scale. Once these groups are created, the third stage continues by selecting and clustering co-located lower-resolution data in a stepwise fashion, creating a tree structure of pixels with similar spectral shapes. The fourth stage leverages this tree structure to assign a spectrum to a new super-resolved pixel with the same channels as the original lower resolution data. The new IR spectrum comes from the one closest in MD space from either the tree or immediate spatial surroundings. Finally, the fifth stage compares the new super-resolved data to the original data by convolving it again with the PSF, resulting in data at the original resolution. Radiometric differences are identified between the two datasets and a correction factor based on this difference is allocated across the new super-resolved pixels. Rather than allocating this correction factor evenly, the correction is scaled based on their MD from their source values; pixels with higher MD values receive a greater percentage of the correction factor. One interesting side-effect of this radiometric correction is that it is possible to generate super-resolved pixels whose spectra are closer to “pure” end-member spectra than are present anywhere within the original resolution data.

1.2.2 Modifications to the Super-Resolution Methodology

The approach originally described by Tonooka (2005) has been significantly modified, adapted and tested to operate on data with only two wavelength regions rather than three. The algorithm was initially examined for performance and optimized where possible to decrease the run-time

and address other deficiencies. For example, in the original approach, the k-means clustering algorithm was used in the third stage. This technique has been replaced by the ISODATA clustering algorithm (Ball and Hall, 1967). Both algorithms are iterative classification algorithms, in which data are randomly clustered, with pixels subsequently moved to clusters whose center (the average of all cluster members) is closer to their value. After each set of moves, cluster centers are recalculated based on their member pixels, and the process repeats until no such moves remain. The k-means clustering technique is limited by the fact that the user is required to make a number of a priori assumptions, including the final number of clusters. ISODATA will create and merge clusters during its run, and thus does not require this assumption.

With the use of any clustering approach for classifying similar groupings of pixels, a limitation arises in that repeated iterations are not strictly reproducible. The final result of the clustering will depend upon its initial starting conditions. These conditions include the number of clusters, the distances to use before either splitting or joining clusters, the number of iterations for clustering, how many pixels must change for the process to continue, and the initially seeded cluster centers. The clustering techniques use a random choice for the very first cluster center that could lead to different results at the end of the third stage of the super-resolution process, which could produce varying results. However, any differences created in cluster center values by this potential randomness are removed in the final radiometric correction stage. Varying results would lead to varying Mahalanobis Distances, which would cause the radiometric correction to be divided among the sub-pixels differently, and thus arrive at the same final product.

A second alteration to the algorithm in its adaptation to using THEMIS data was the approximation of the Point Spread Function (PSF). The PSF describes the blur inherent within a system, and models how the radiance from a point source is actually imaged spatially by the instrument in two dimensions (Townshend et al., 2000). Super-resolution makes use of the PSF during the degradation of the higher-resolution channels, so that the mapping of surface to image space is identical between the spectral regions. The PSF of the ASTER instrument was calculated prior to launch (Fujisada et al., 1998); however this was not the case during the pre-launch testing of the THEMIS instrument. Furthermore, it is relatively easy to calculate the PSF from Earth orbit by imaging spatially-small features that have a large contrast in albedo to their surroundings (e.g., a small bright island in a large dark water body) or by looking at long linear features (e.g., buildings and bridges) in the two different directions. These in-scene calibration targets are obviously not present on Mars. Therefore, the THEMIS PSF used in this implementation of the super-resolution algorithm represents an approximation. This was based on what is known about the instrument and thorough iterative testing of the algorithm using different values of the PSF as input parameters. This approximation results in THEMIS data having slightly more pixel to pixel blurring than ASTER data.

1.2.3 Identification of Acceptable Data

Potential test locations on the surface of Mars with interesting and spectrally-diverse features were first identified based on a thorough literature review. These included the first quartzofeldspathic site (Bandfield et al., 2004a), quartzofeldspathic deposits elsewhere (Bandfield, 2006), a dacite lava flow (Christensen et al., 2005), olivine-rich deposits near Nili Fossae (Hamilton and Christensen, 2005; Hoefen et al., 2003) and regions of phyllosilicate

detection using VSWIR data (Poulet et al., 2005), as well as the landing sites for the Spirit and Opportunity rovers. Images over some of these regions were identified using the THEMIS website (http://themis.asu.edu/) and the JMARS visualization / GIS tool (Weiss-Malik et al., 2005). Initially, images were deemed acceptable if both the VIS and IR wavelength data were acquired contemporaneously and with complete or near-complete spectral resolution. Image searches were constrained to pairs having at least channels 2 through 5 in the VIS and channels 3 through 9 in the IR. Pairs that included band 1 (VIS), or channels 1, 2 and 10 (IR) were also acceptable, although data from these channels were rarely used due to their relatively poor quality or due to high atmospheric opacity. All of these constraints were important in order to maximize the fidelity of the algorithm, obtain the most reliable results, and increase the methodological similarity to the ASTER results.

Nearly 1,400 contemporaneous image pairs fitting the criteria were identified, including one pair (V12929011 and I12929010) over the quartzofeldspathic site in Syrtis Major. The area of spatial overlap of the two data sets does not correspond to the location of the strongest spectral signature in the region (Bandfield et al., 2004a). However, a region with the quartzofeldspathic spectral signature is imaged in both scenes and located near the center of the images, at the western edge of the area of overlap (Figure 1-1). TIR emissivity and radiance data were both super-resolved, in order to first assess if there is a difference between super-resolved emissivity data, and emissivity data separated from super-resolved radiance data. In order to correlate quartzofeldspathic detection with surface morphology, the super-resolved IR data were then compared to an image acquired by the High Resolution Imaging Science Experiment (HiRISE).

1.2.4 Data Preprocessing Prior to Super-Resolution

Both the VIS and IR data were obtained from the PDS data archive maintained by the THEMIS science team at Arizona State University. Geometrically corrected VIS data cubes (both the CUB and LBL file) were initially processed using the JENVI toolkit (Piatek and Moersch, 2006), which converts the data into floating point values and exports the images in a format easily used by the super-resolution program, which is written in the interactive data language (IDL).

THEMIS IR data were preprocessed and downloaded using the THMPROC tool with all but two of default processing flags used. The *unrectify* flag was selected, because JENVI requires it, and the spatial resolution was changed. The standard 100 m/pixel IR resolution was degraded slightly to 108 m/pixel in order to make each IR pixel an integer multiple of the VIS data, which are either 18 or 36 m/pixel. This ensures that there are 9 (or 36) unique VIS pixels per every one unique IR pixel without the need for higher level image resampling later.

THMPROC was also used to place constraints on the data volume by eliminating the area not covered by both the TIR and VIS datasets, in order to limit the processing time. The spatial extent of the IR data was limited to 0.25 degrees beyond the edges of the THEMIS VIS data. The resulting 32-bit image cubes containing all channels were imported into JENVI. Channels 1 and 2 (6.78 μm) were averaged together for better signal to noise and channel 10 (14.88 μm) was removed due to the high atmospheric opacity, resulting in eight usable IR channels. Pixel-integrated brightness temperature and emissivity data products were created using the normalized emissivity approach with an assumed maximum emissivity of 1.0 (Christensen, 1982; Realmuto, 1990).

Images obtained from THMPROC are not fully atmospherically corrected. The attenuation from the atmosphere must be removed, and was done using the in-scene approximation approach (Bandfield et al., 2004b). This was done by choosing a morphologically bland and spectrally homogeneous region (e.g. basaltic plains) of a few hundred pixels within the IR radiance scene at approximately the same surface elevation as the region of interest. This same area was also located for the emissivity data, and the mean emissivity value extracted in each IR channel. Atmospherically corrected TES data (bands 9 – 35 and 65 – 110) over that same region were also extracted, averaged together, and convolved to the THEMIS spectral response functions. The mean emissivity value from THEMIS was divided by the TES-derived emissivity in order to derive an atmospheric correction factor, which was then applied to the entire THEMIS emissivity image. Some unusual spectral shapes were noted in the TIR image (I12929010) both before and after this correction, possibly due to slightly miscalibrated data or an issue during data collection; however, these unusual spectra do not overlap with the area containing the quartzofeldspathic signature and therefore can be ignored.

Finally, the VIS radiance and IR emissivity images must be spatially aligned, and a mask defined, which encompasses the overlapping region of interest and removes data at the image borders as well as data having errors in one or more channels. Both images are subset to the extent of the mask, aligned to one another with co-located corners, and are thus an integer multiple apart in size. It is commonly necessary at this stage to visually inspect the image pairs in greater detail using ground features (e.g., small impact craters) to identify any possible remaining

misalignment between the VIS and IR data. If one is found, it is corrected by applying a further manual correction until the data files are aligned to within one VIS pixel. These two datasets are used by the super-resolution algorithm.

Previous application of the super-resolution approach (Tonooka, 2005) discovered that an alignment within 0.2 of the larger pixel size was sufficient for valid results. Given the relative resolution of the VIS and IR data, this algorithm is limited to an alignment within 0.33 of the larger pixel. As the resampling of the IR data to 108 m / pixel within THMPROC occurs before this alignment, and the alignment is performed using the smallest resolvable surface features within this resampled data, it is assumed that this resampling does not introduce further misalignment beyond the one VIS pixel limit. A significant misalignment between the datasets would result in a failure of the super-resolution algorithm during the clustering stage. In such a case, it would be unlikely that any IR cluster could achieve the minimum cluster size due to the misalignment causing an excess number of dissimilar spectra.

1.3 RESULTS

For the quartzofeldspathic region of interest, the VIS data was subset to 312 samples by 1254 lines and included 4 wavelength channels with a spatial resolution of 36 m/pixel. The same region imaged by the IR sensor was 104 lines by 418 samples with a spatial resolution of 100 m/pixel. These pixels were resampled to 108 m/pixel, in order to improve the alignment with the VIS data. The number of IR wavelength channels decreased from ten to eight after channels 1

and 2 were averaged together due to low SNIR values at Mars temperatures and channel 10 was discarded due to high atmospheric opacity. Channel 3, typically used for temperature / emissivity separation, was included within the super-resolved data.

The initial stage of the super-resolution algorithm uses ISODATA clustering with an assumed PSF, which is based on a similar resolution and response of the ASTER instrument. In both instruments, the PSF is modeled as a two dimensional Gaussian. The ASTER instrument is assumed to have a narrower peak, with a slightly greater percentage of the energy in any given pixel being contributed by that pixel versus neighboring pixels. The measured PSF of the ASTER instrument indicates 75.5% of the energy within any given pixel originates in that pixel. The PSF of THEMIS instrument was found by varying its shape until one was found that produced the best results, based on the radiometric accuracy and visual appearance of the final super-resolved product across multiple co-located image pairs. This best case PSF of the THEMIS indicates that only 56% of the energy originates in that pixel. Experimentation with altering the assumed PSF to increase the percentage of energy originating from the pixel to match or exceed that of ASTER did not produce significantly different results.

During the clustering stage of homogeneous VIS pixels, the ISODATA algorithm was configured to run no more than 500 times and with a default change limit of 0.5%. With the latter limit, clustering ends if fewer than 212 pixels change their cluster assignment during any single iteration. If a cluster was created with less than 0.01% of all pixels in the scene (e.g., 5 pixels for this case), it was deemed to fall below the minimum cluster size and deleted. The pixels from this deleted cluster are then assigned to the next closest cluster to that individual pixel based on

the MD. Clusters with pixel members whose value exceeded four standard deviations in any band were marked as eligible to be split, and clusters whose centers were closer than two standard deviations were eligible to be joined. Only the closest four pairs of clusters could be joined in any one iteration, and cluster splitting was subject to further limitations based on the size and number of clusters in that iteration. These same values were used during clustering of the IR pixels co-located with each visible pixel cluster, although actual pixel counts would differ for the IR data. A maximum value for the VIS and IR radiance data was specified as being slightly higher than the highest pre-existing value; similarly, a maximum value of 1.0 would be used for emissivity. During the splitting of clusters, the value of each band was not allowed to exceed these amounts. This permitted cluster centers to be used later in the algorithm as spectra for newly created super-resolved pixels with values that are sane for the image area, even before radiometric correction. These values are based upon previous performance of the algorithm on both THEMIS and ASTER data, and are designed to minimize the algorithm's runtime and maximize the chance that clustering before the iteration limit.

The algorithm identified 42,241 homogeneous pixels in the VIS data, from which an initial 50 data cluster centers were chosen. The first homogeneous pixel / cluster center was chosen at random, with subsequent pixels clustered based on maximizing the total Mahalanobis distance (MD) to all other chosen pixels. Pixels that had a MD of zero to any previously chosen center (i.e., having the same values in all channels as that existing center) were prevented from being selected, and the pixel with the next greatest sum of MD to all existing centers would be chosen instead. By doing this, even the initially chosen cluster centers should represent different spectral units.

After all homogeneous pixels are clustered, the most likely spectrum is assigned to each newly created super-resolved pixel. This is done by finding the spectrum closest to the VIS pixel associated with the super-resolved pixel. The pixel spectrum is first compared to the cluster centers created in the previous step. Any homogenous pixels are within a radius of 10 low resolution pixels, they are also compared. The closet spectrum is used for the new pixel whether it originates from the clusters or from neighboring pixels. The MD distance between the VIS pixel and the spectrum assigned to the created pixel is also recorded for use in the next step.

The image may not be radiometrically accurate, however, after each new super-resolved pixel has a spectrum associated to it. To determine where any correction factors are needed, the newly super-resolved data are convolved with the PSF and degraded to match the original spatial resolution, as in step 1. These data are then subtracted from the original data on a per-band basis. Any differences represent a correction factor, which is applied to the original data on a per-pixel basis. There are a number of different ways to divide this correction factor across the super-resolved pixels such that they would convolve back to the original pixel value and maintain radiometric accuracy. In this algorithm, the correction factor is allocated among the super-resolved pixels based on their MD distance from the original VIS spectrum, rather than being evenly divided across the sub-pixels of the degraded data. This distribution of the correction factor represents an assumption that pixels that are more dissimilar than their source are the major contributors to any radiometric inaccuracies and should receive a greater share of the correction. Following this, the super-resolved data should be radiometrically accurate and a match to the original data (see Table 1-1), which can be verified by again convolving the corrected pixels with the PSF.

Upon completion of the super-resolution program, a number of files are created, including the new super-resolved data, cluster maps, source maps, maps of homogeneous pixel locations in the higher resolution data, maps of the distance from each pixel to its source in Mahalanobis space, and maps of the size of correction necessary to make each pixel radiometrically match the original source data. These data files provide insight into how the final product (i.e., the super-resolved data of the lower resolution channels) was created, and can be used to compare the results from multiple runs in order to identify larger spatial patterns within the data.

In order to verify the source of each of the newly super-resolved pixels, source maps for both the emissivity and the radiance data are examined (Figure 1-3). These maps indicate whether the pre-corrected value of the super-resolved pixel came from a cluster in the ISODATA tree or from a nearby homogeneous pixel. A super-resolved pixel that is sourced from the tree is more likely to represent a widely-distributed, yet sparse, end-member; whereas a pixel that is sourced from a neighbor is more likely to represent a continuation of the surrounding compositional unit. However, these are not strict rules for the super-resolution model results. The important test in examining the source maps is whether there is minimal to no recognizable spatial patterns in the distribution of choices. Such a coherent spatial pattern with one choice being more strongly preferred over the other would likely indicate a problem with the data in areas that are expected to be heterogeneous and compositionally diverse. There is no regularly

Table 1-1 Comparison of the original data to that of the super-resolved data over the area denoted by the white box in Figure 1-1A. (A) The super-resolved radiance data shows a greater dynamic range than the original, but, both sets of data have the same mean values over the region, indicating that radiometric accuracy has been preserved. All values shown are in W/cm^2 str μm, and are multiplied by 10^4 for easier viewing. (B) The super-resolved emissivity data shows the same enhanced dynamic range as the radiance data. Some of the original emissivity data had emissivity values exceeding 1.0 following atmospheric correction. This issue is also seen in the super-resolved data.

A	**Minimum**		**Maximum**		**Mean**		**Standard Deviation**	
	orig	super-res	orig	super-res	orig	super-res	orig	super-res
Band 1 / 2	0.56	0.1	1.69	2.25	0.93	0.93	0.12	0.13
Band 3	1.15	0.58	2.72	3.47	1.65	1.65	0.17	0.18
Band 4	1.33	0.58	2.95	3.69	1.86	1.86	0.18	0.19
Band 5	1.61	0.82	3.32	4.07	2.19	2.19	0.19	0.2
Band 6	2.05	1.19	3.88	4.72	2.68	2.68	0.2	0.22
Band 7	2.15	1.4	3.92	4.7	2.77	2.77	0.2	0.21
Band 8	2.17	1.29	3.86	4.49	2.77	2.77	0.19	0.2
Band 9	2.06	1.29	3.56	4.17	2.61	2.61	0.17	0.18

B	**Minimum**		**Maximum**		**Mean**		**Standard Deviation**	
	orig	super-res	orig	super-res	orig	super-res	orig	super-res
Band 1 / 2	0.882	0.868	1.140	1.151	1.021	1.021	0.033	0.032
Band 3	0.947	0.943	1.027	1.042	1.000	1.001	0.013	0.013
Band 4	0.930	0.926	1.006	1.016	0.967	0.967	0.009	0.009
Band 5	0.900	0.898	0.964	0.967	0.933	0.933	0.007	0.007
Band 6	0.913	0.912	0.932	0.935	0.932	0.932	0.001	0.001
Band 7	0.922	0.920	0.974	0.976	0.948	0.948	0.006	0.006
Band 8	0.938	0.935	0.999	1.001	0.969	0.969	0.007	0.007
Band 9	0.942	0.941	1.016	1.024	0.980	0.980	0.009	0.009

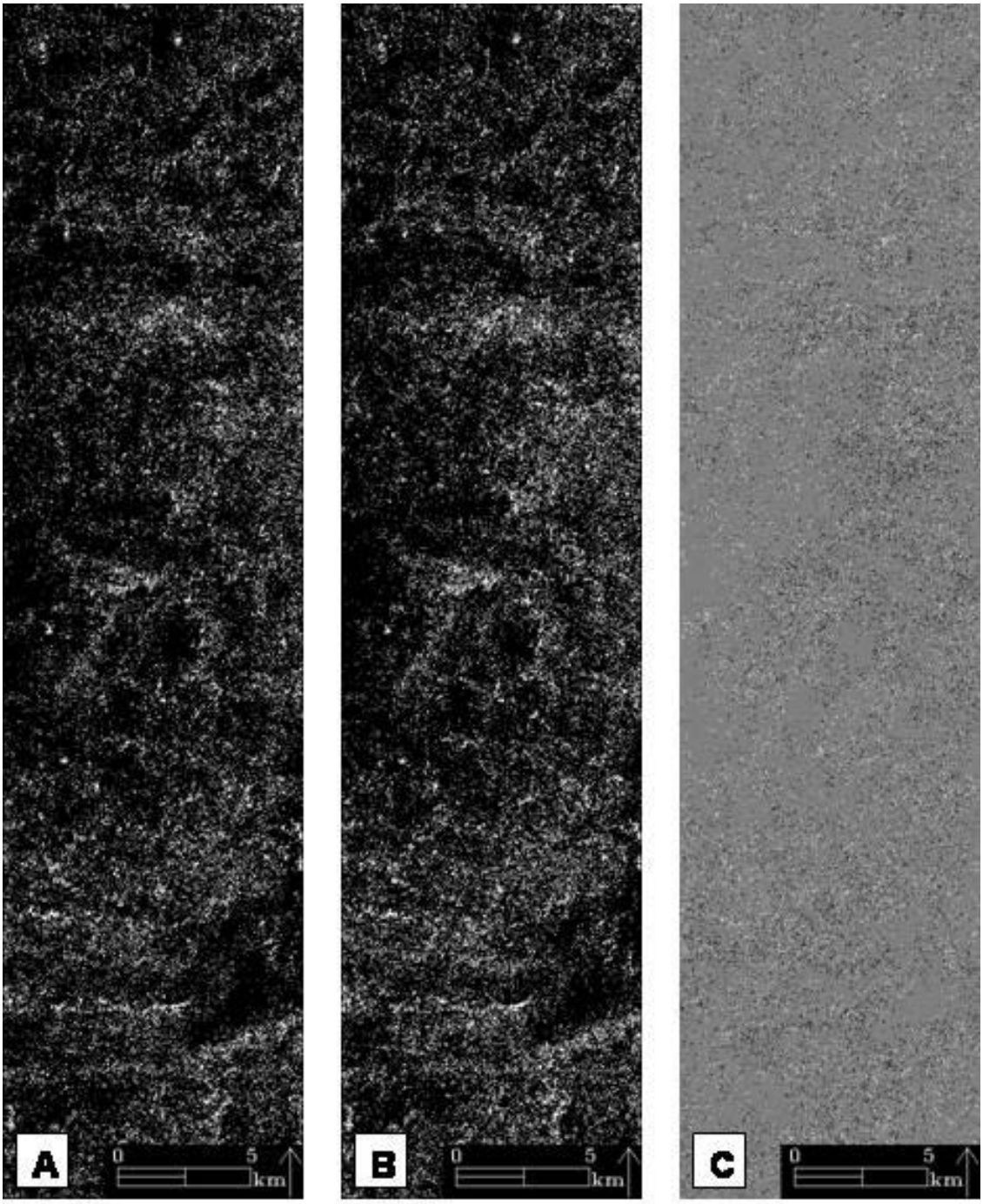

Figure 1-3 Source map results produced by the super-resolution approach. Bright pixels were sourced from the tree created by the ISODATA clustering process, whereas dark pixels were sourced from adjacent homogeneous pixels. (A) Source map for the emissivity data. (B) Source map for the radiance data. (C) Difference image: (A) minus (B). Light grey pixels indicate that both the emissivity and radiance were sourced from the same location; dark pixels indicate a source from adjacent homogenous pixels (in the radiance data) and from the tree (in the emissivity data). Bright pixels indicate a source from the tree (in the radiance data) and from adjacent homogeneous pixels (in the emissivity data).

discernible pattern in the source maps of either the super-resolved emissivity or radiance data, indicating that the spectra derived from the cluster tree are not from spatially limited areas. An isolated pixel can have a spectrum sourced from a spatially remote area, through the cluster tree, rather than rely only upon nearby homogenous pixels. There is a slightly greater occurrence of pixels sourced from the ISODATA tree distributed around the crater rims (Figure 1-3). These rim edge pixels are distributed throughout the image, rather than being allocated in any contiguous spatial region. During clustering, these pixels are clustered together and result in a center whose value is a better spectral match to their members than any spatially adjacent pixels. There is also no apparent pattern in the difference between these two images (Figure 1-3C), although the majority of pixels in the final super-resolved products were sourced from the same location (i.e., the ISODATA cluster created tree or the adjacent homogeneous pixels) in the two data sets.

Because the super-resolved data represents two distinct runs of the program, with two different low resolution inputs (IR radiance and emissivity), seeing a pattern here would indicate an issue with the data. In both the emissivity and radiance source maps, the majority of pixels take their pre-radiometric correction values from adjacent homogeneous pixels, but a significant percentage are sourced from the ISODATA tree. The emissivity source map shows 84.4% of the super-resolved pixels were sourced from the adjacent homogeneous pixels (Figure 1-4A), whereas the radiance source map shows that 85.9% of the super-resolved pixels were sourced from adjacent pixels (Figure 1-3B). The difference between the two images (Figure 1-4C) shows that 92% of the super-resolved pixels came from the same source, with 4.8% sourced from the

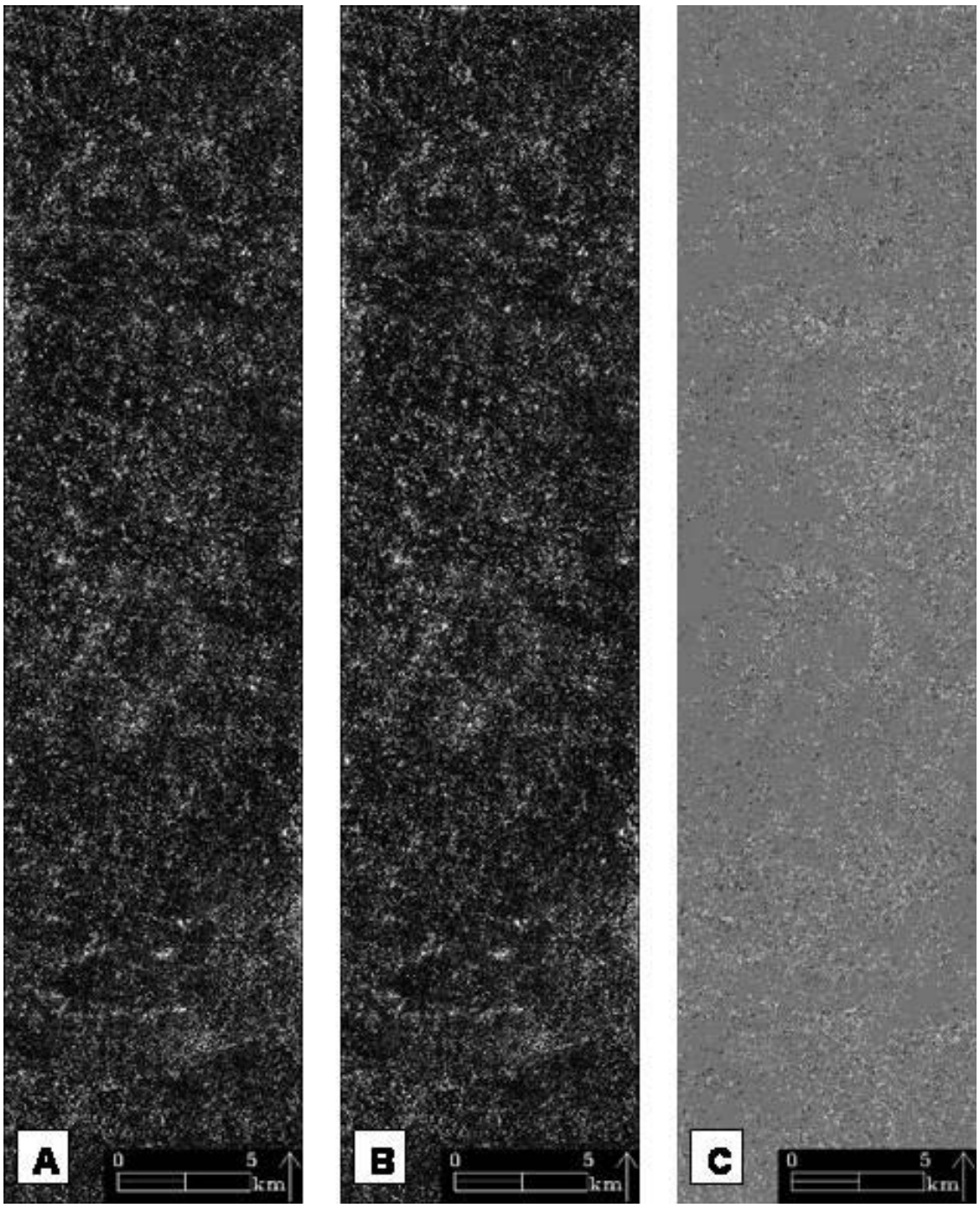

Figure 1-4 Mahalanobis distance (MD) map results produced by the super-resolution approach. Bright pixels have a greater MD between the original VIS pixel and the VIS portion of the super-resolved spectrum inserted at that location. As such, these pixels will require greater radiometric correction than adjacent pixels. (A) Distance map for the emissivity data. (B) Distance map for the radiance data. (C) Difference image: (A) minus (B). Dark pixels in (C) are closer to their source in the super-resolved radiance data, bright pixels are closer to their source in the super-resolved emissivity data, and the light grey pixels have nearly equal distance between radiance and emissivity.

ISODATA tree in the emissivity data and from the adjacent homogeneous pixels in the radiance data. The remaining 3.2% of the super-resolved pixels were oppositely-sourced (i.e., from adjacent pixels in emissivity data and from the ISODATA tree in radiance data).

The Mahalanobis distance maps used during the final radiometric correction step are shown in Figure 1-4. During the radiometric correction, an assumption is made that any correction should be weighted by the distance between the super-resolved pixel and its source. As with the source maps, a sufficient number of pixels with a high distance fall along the outline of crater rims and they can be easily discerned from the random distribution of distances in the remainder of the image. However, the overlap between crater rim pixels and pixels with high distance to their source is not complete. Statistics for these maps indicate that the pixels created by the super-resolution approach had spectra that matched well with their source (Table 1-2). Pixels whose match is perfect (an MD of zero) would receive none of the necessary radiometric correction in the final step, whereas those pixels with a large MD would receive a greater correction. This is due to the assumption that poor matches at this stage are the most likely cause of poor radiometric accuracy after convolving the data to the original resolution. A comparison (Figure 1-4C) of the two maps, by subtracting the emissivity distance map from the radiance distance map, shows that the two runs have very similar products. Statistics for this comparison can be found in Table 1-3. In this case, a majority of the pixels had a difference of 0.0, indicating that they derived their pre-radiometrically corrected spectra the same way. Those pixels which had a difference of 0.0 and were both originally sourced from the cluster tree indicate that pixels are clustered together similarly between runs. This is despite the initial random start and the difference between emissivity and radiance data. Although there was randomness in the initial

Table 1-2 Mahalanobis Distance between super-resolved pixels and their source, as used during radiometric correction.

	Emissivity Data	Radiance Data
Average MD	0.27	0.28
Maximum MD	33.2	34.5
Standard Deviation of MD	0.46	0.48
% with MD = 0	50.3	51.1
% with MD < 1.0	95.5	95.5
% with MD < 2.0	99.0	99.0

Table 1-3 Difference between Emissivity and Radiance MD maps

Minimum Difference	Maximum Difference	Mean Difference	Standard Deviation	% with no difference	% with absolute difference < 1.0
-2.9	5.1	0.01	0.11	83.2	99.9

choice of clusters, ultimately the ISODATA process terminates in the same state. The vast majority of both radiance and emissivity had MD values of less than 1.0, which also indicates a very good fit of the modeled data. The multiple runs of the algorithm (i.e., emissivity and radiance input data) also ensure extreme cases are not being generated. The algorithm has reproducible results even before the radiometric correction is applied.

Super-resolved IR emissivity and radiance data were created with a spatial sampling of 36 m per pixel (Figure 1-5C). Features such as the rim of the inner crater are more clearly defined, and small craters that were not visible in the original IR image are now more apparent. Also of note in the output product is the minimization of thermal shadowing, which is common in TIR data collected over topographic relief in the late afternoon (local time). These temperature variations, especially where they occur at the subpixel scale, can negatively impact the compositional interpretation of the surface (Bandfield et al., 2004b; Zhang et al., 2004). The decorrelation stretch (DCS) approach can minimize this impact by highlighting similar composition surfaces in the same color and temperature differences within a particular unit as variation in the intensity of that color (Gillespie et al., 1986). However, DCS images can be difficult to interpret quantitatively and subpixel mixing is not easily recognized. The DCS images (Figure 1-6) of the original data show the effects of thermal shadowing as well as prominent pixel-to-pixel noise and line striping, making the interpretation of the color patterns more challenging for small features. These effects can be clearly seen in the DCS of IR bands 6, 4, 2 in R, G, B, respectively (Figure 1-6E). In this color combination, the region of increased quartzofeldspathic material is denoted by the reddish-purple color (white arrow in the lower left

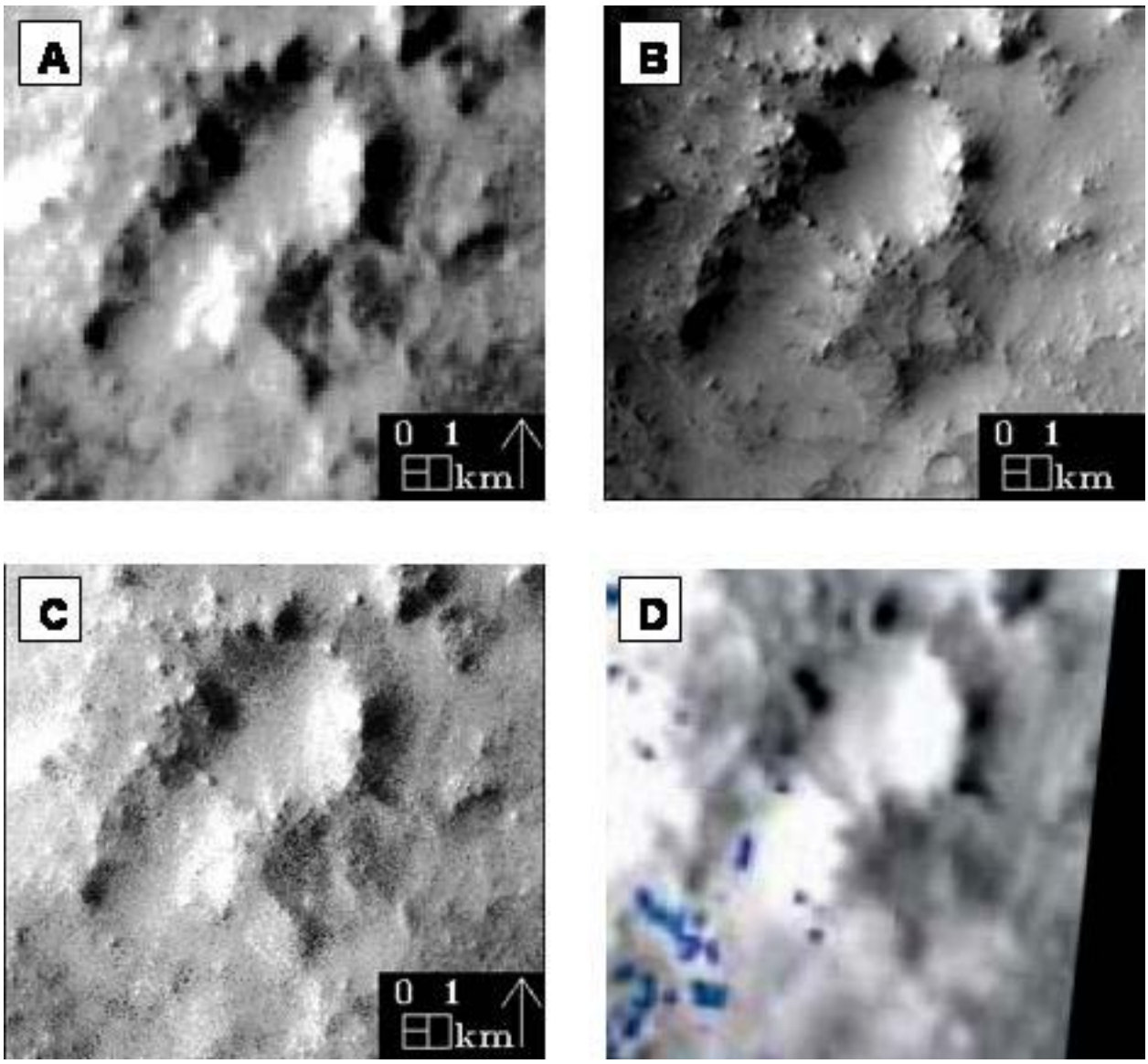

Figure 1-5 THEMIS radiance images and the super-resolution result for the crater central peak area. (A) IR radiance (channel 9) at 108 m/pixel. (B) VIS radiance (channel 3) at 36 m / pixel resolution. (C) Super-resolved IR radiance (channel 9) now at 36 m / pixel. Figures (B) and (C) have been stretched using histogram matching to Figure (A) for display purposes. (D) Concentration map of quartzofeldspathic material with a range of 15 to 20 (cyan to green) areal percent (Bandfield et al., 2004a).

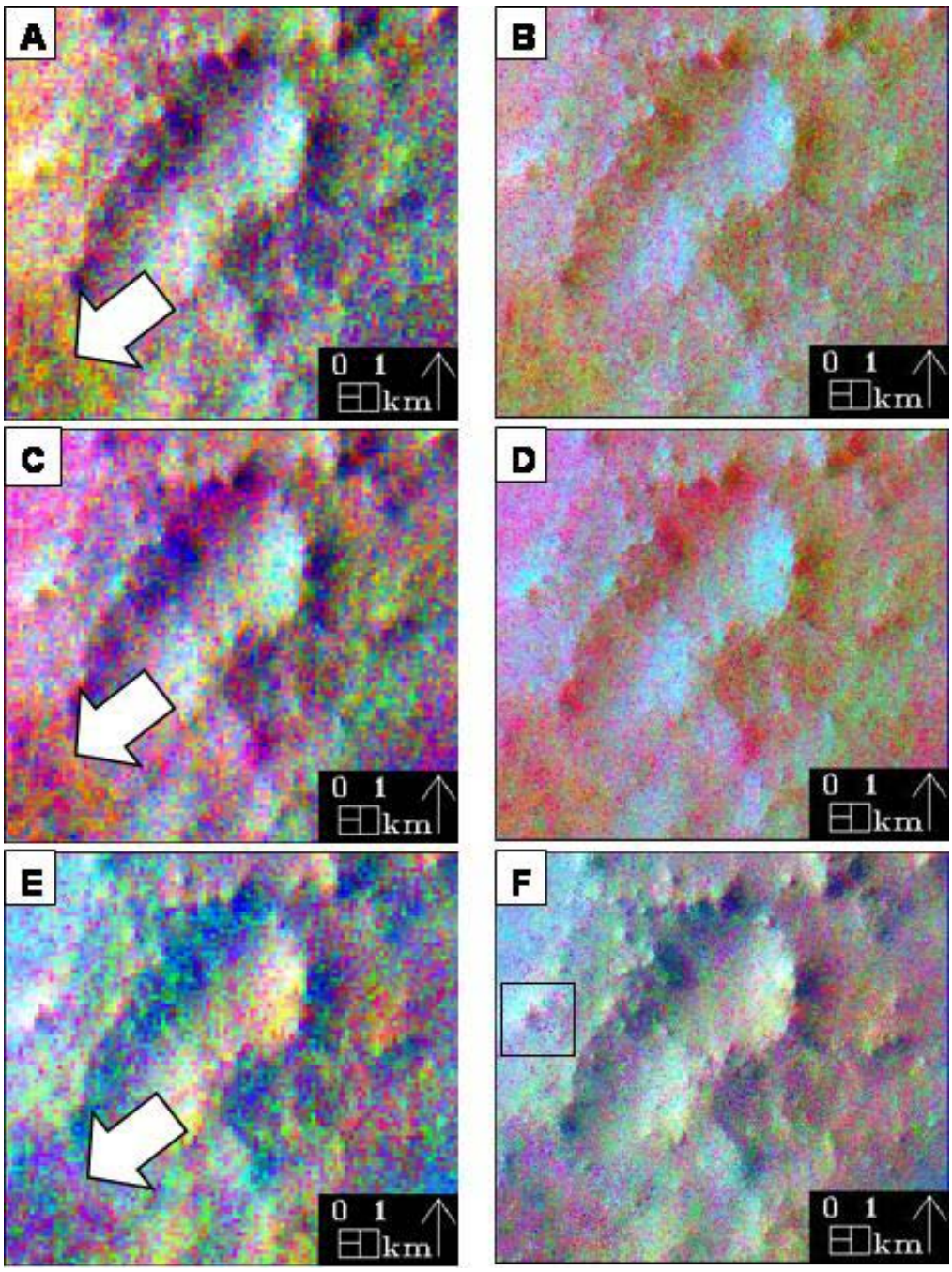

Figure 1-6 DCS results of the same region shown in Figure 1-5 with the original resolution shown on the left and super-resolved result shown on the right. The higher concentration of quartzofeldspathic material is denoted by the arrow (see previous figure). The super-resolved data show less thermal shadowing and pixel noise than the original data. These images have more muted colors because of the larger number of pixels, which directly affect the statistics of the DCS approach. However the general color patterns are preserved. (A) Bands 8, 7, 5 in R, G, B. (B) Super-resolved bands 8, 7, 5 in R, G, B. (C) Bands 9, 6, 4 in R, G, B. (D) Super-resolved bands 9, 6, 4 in R, G, B. (E) Bands 6, 4, 2 in R, G, B. (F) Super-resolved bands 6, 4, 2 in R, G, B. The box highlights a small cluster of quartzofeldspathic pixels not evident in the original resolution data.

corner of the image), whereas the basaltic plains unit is blue to cyan in color. This reddish-purple region is along the edge of an area previously identified in Bandfield (2004a), and represents the strongest signature within the co-located and super-resolvable area used in this study. Other smaller reddish-purple areas can be seen elsewhere within the image. The dominant absorption band of the quartzofeldspathic material is shifted to a shorter wavelength (Figure 1-7) and therefore has a higher emissivity in IR band 6 (9.66 microns) resulting in the dominant reddish coloration of this unit. This is visible in the original resolution data; however, it is subtle and overprinted by pixel noise in band 4 (7.98 microns), which shows as bright green in the DCS image. The super-resolution version of this same DCS color combination clearly shows the coloration indicative of this unit without the pixel noise and with a minimum of thermal shadowing. In addition, the super-resolved data also reveals spatial patterns at scales not apparent in the original image. These were checked against a Context Camera (CTX) image of the region (Figure 1-8A) and the strongest quartzofeldspathic detections do not appear to correlate with any of the large rock outcrops/boulders. Rather this unit lies in a topographic low area, marked by the rectangle in the lower left of Figure 1-8A. This area can be seen in slightly more detail in the segment of a Mars Orbiter Camera Narrow Angle (MOC-na) image shown in Figure 1-8B. The ridges surrounding this material may extend into the area shown in the HiRISE image in Figure 1-8C, in which high albedo material can be seen weathering out of two isolated peaks.

Emissivity spectra were extracted from 108 m/pixel and the 36 m/pixel IR data (Figure 1-7). Of interest was whether spectral shifts indicative of the quartzofeldspathic material would be detectable in the super-resolved data. In particular, did super-resolved pixels created from an

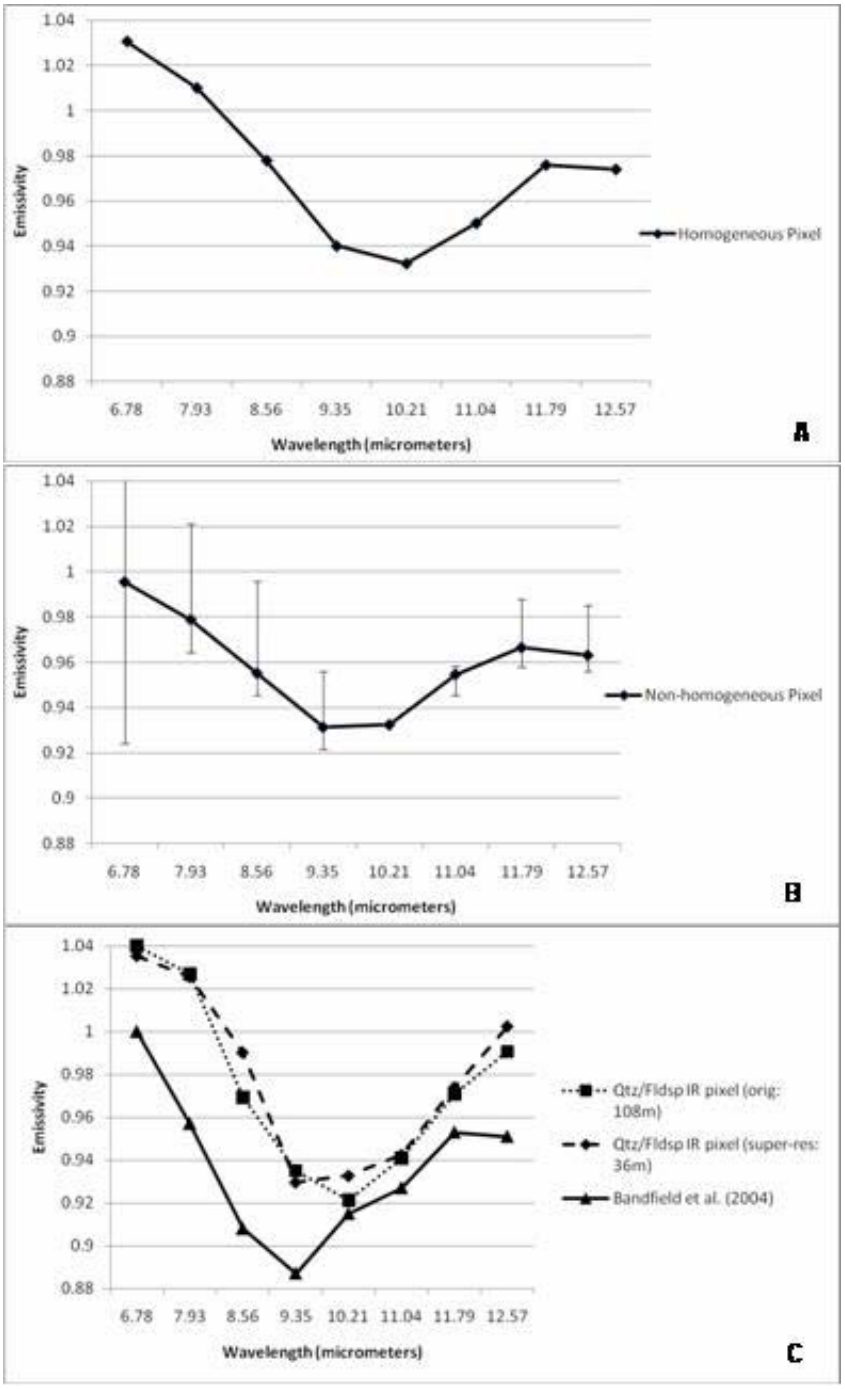

Figure 1-7 Sample spectra from super-resolved emissivity data. (A) Spectrum of a homogenous pixel. Error bars are less than the width of the line showing the spectrum, with a maximum difference from the original spectrum of less than 0.001. (B) Spectrum of a non-homogenous pixel. Error bars are used to indicate the extent of values seen in the nine super-resolved pixels derived from this original spectrum. (C) Spectra of both an original resolution IR pixel, and one of the co-located super-resolved pixels located within it. The super-resolved spectrum shows a shift in the minima, making it more similar to a mixture of the original spectrum plus the quartzofeldspathic spectrum used in Bandfield et al. (2004).

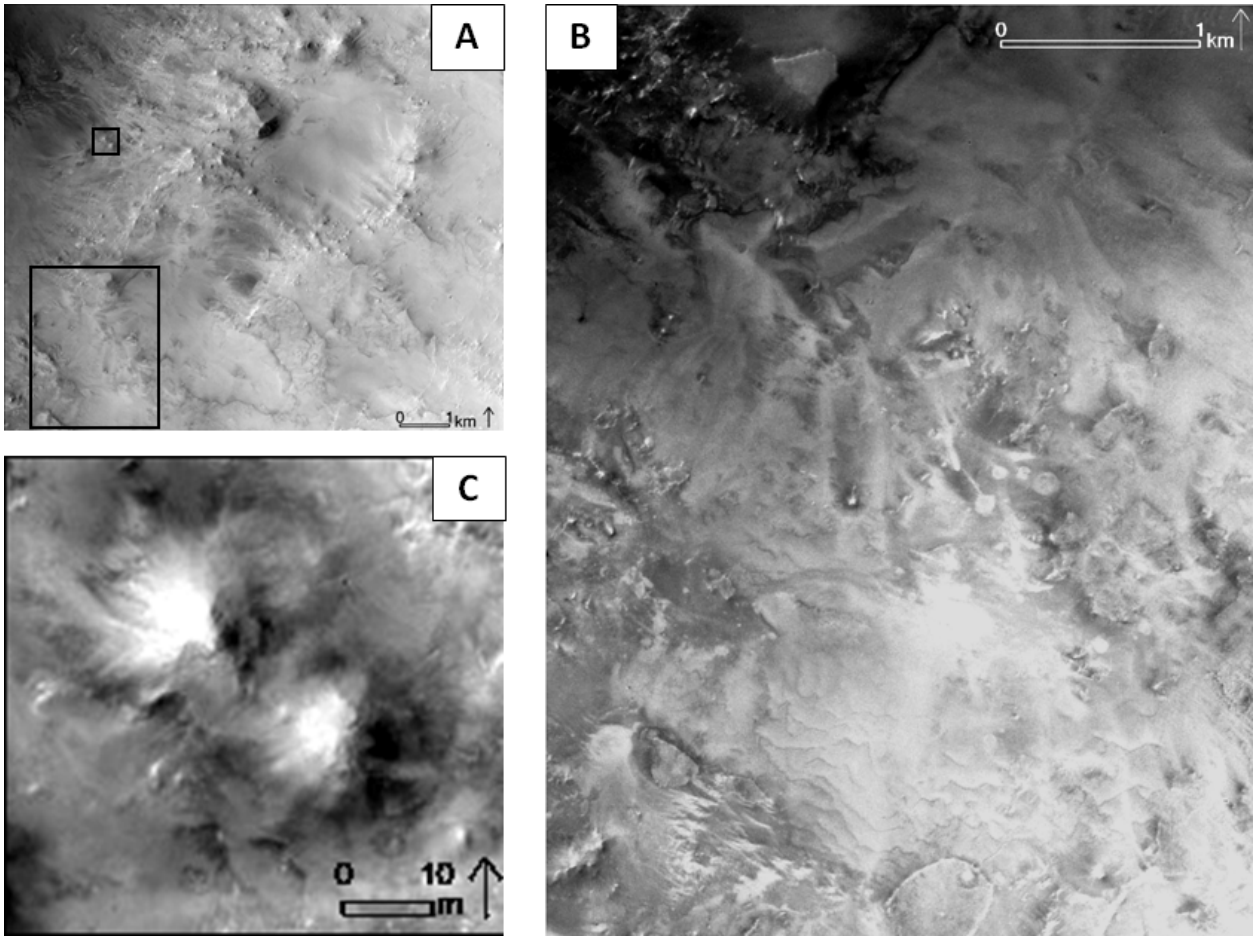

Figure 1-8 Higher spatial resolution data of the same spatial area shown in Figures 1-5 and 1-6. (A) A subset of CTX image P16_007108_1988, showing the central crater and surrounding area at 5.76 m / pixel (~ 40 times greater than the super-resolved data). The central crater is located slightly off-center, allowing the regions of interest to the west and southwest to be shown. The large box indicates the area shown in (B) and the smaller box denotes the area shown in (C). (B) A segment of MOC image R22-00941 with a spatial resolution of 3.62 m / pixel, showing the arcuate ridge that extends from the central crater, as well as the depression to the south and west of this ridge. This basin is co-located with a number of quartzofeldspathic-rich pixels identified in Bandfield et al. (2004a). (C) Two isolated high albedo peaks surrounded by weathered low-albedo material transported from the northeast. This image is a subset of HiRISE image PSP_001385_1985_RED, with a spatial resolution of 0.25 m / pixel. The higher albedo along the western slopes of these two peaks is shown in this image, but is also present in CTX, MRO, and THEMIS data.

original resolution IR pixel that had a basaltic signature show quartzofeldspathic signatures? This would confirm that the new higher resolution pixels created by this process were different than the original data and more importantly, were contributing to the average pixel-integrated spectrum of that original pixel. In other words, the super-resolution data would provide an independent means of identifying sub-pixel mixing of two compositional units, confirm the presence of a more areally-extensive deposit, and potentially identify the small-scale sources for this deposit. Example spectra from a pixel bordering on the quartzofeldspathic region are shown in Figure 1-7. These super-resolved pixels showed a distinct shift in the location of the absorption minima from longer to shorter wavelengths, and are associated with the smaller-scale spatial patterns seen in Figure 1-6. These spectra do not perfectly match the spectra from Bandfield (2004a), because they are located along the edge of the deposit and not extracted from the same region. A mixture between the spectra identified by Bandfield (2004a) and the surrounding material is evident in the spectrum, indicating that there is a local mixing between the two end-members in this region.

A ratio of band 6 (10.21 μm) to band 5 (9.35 μm) shows these same shifts in the absorption minima location (Figure 1-9A). This ratio measures the lever arm between the location of the quartzofeldspathic minima (common in band 5) and the typical mafic minimum (common in band 6). Greater values would indicate a surface more enriched in quartzofeldspathic material. Values over the entire super-resolved area (Figure 1-9A), range from 0.82 (mafic-rich) to 1.35 (relatively quartzofeldspathic-rich) with a mean of 0.98 (slightly mafic). A very strong mafic signature accounts for 1.4% of the pixels, with an additional 44.4% having a definite absorption minimum in band 6. Nearly a third pixels (31.8%) show roughly

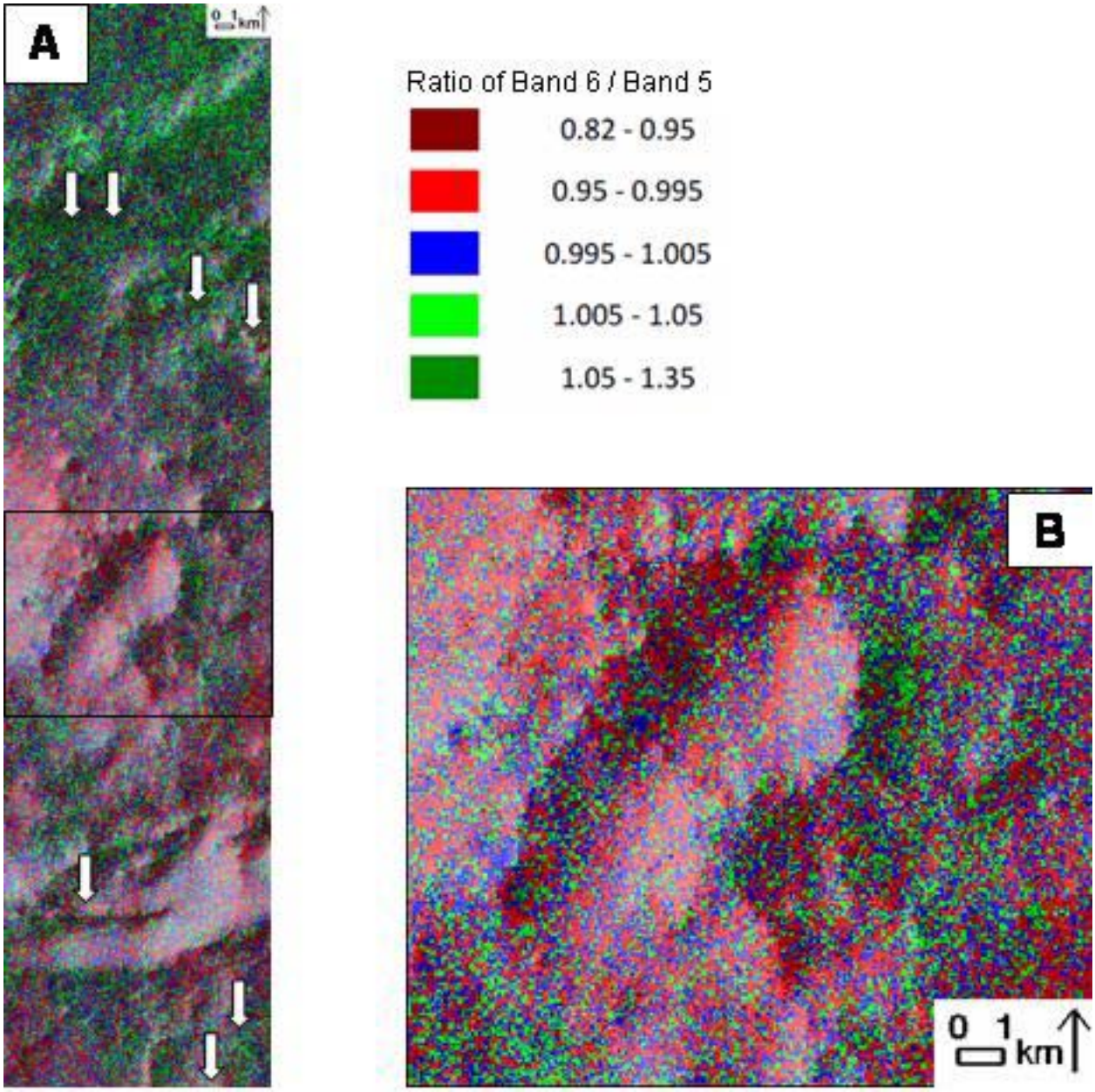

Figure 1-9 Band ratio image of band 6 (10.21 µm) divided by band 5 (9.35 µm) color coded and draped over the super-resolved temperature image with a 50% transparency. The mean value of in this image is 0.98. Green, and especially dark green, pixels have a more felsic component to their spectra. (A) Ratio map showing the total super-resolved area, using the same scale and orientation as in Figures 1-3 and 1-4. The super-resolved band 9 is shown beneath the color ratio map to provide spatial context. The locations of pixels with a higher ratio than the maximum in (B) are indicated by white arrows. The location shown in (B) is indicated by the large rectangle. (B) Ratio map showing the area surrounding the central crater feature, using the same scale and orientation as in Figures 1-5 and 1-6. The super-resolved band 9 is shown beneath the color ratio map to provide spatial context.

equal emissivity lows in bands 5 and 6 (ratio value = 0.995 – 1.005). Pixels with a mildly stronger quartzofeldspathic absorption account for 20.9%, and pixels having definite quartzofeldspathic signatures make up only 1.5% of the area. The area covered in Figure 1-9B, which is the same area shown in Figure 1-5, has a similar range of values as Figure 1-9A, with a ratio minimum of 0.82, a maximum of 1.29, and a mean of 0.99.

There are seven pixels within the area of Figure 1-9A outside of the area of Figure 1-9B with values between 1.29 and 1.35, greater than the maximum ratio in Figure 1-9B. Four of these pixels form an arc to the north of the crater. This arc trends roughly parallel to the northeastern crater wall. One of the four points is located on the crater wall; the remaining three pixels are located outside the main crater wall, but appear to be along ejecta from the crater. There are also three pixels with a ratio value greater than 1.29 to the south of Figure 1-9B, with one point located on a prominent ridge inside the crater immediately to the south of Figure 1-9B, and the other 2 located to the southeast of the crater, also along crater ejecta. The locations of these pixels are indicated on Figure 1-9A.

1.4 DISCUSSION

The super resolution approach relies on the ISODATA clustering algorithm to create a tree of spectral values, with primary branches formed from the higher-resolution VIS data and secondary branches formed from the lower-resolution TIR data. For the THEMIS data examined here, this tree was the source for roughly 15% of the super-resolved pixels in the final product. There is a large degree of overlap between the two super-resolved data sets presented here

(emissivity and radiance). Nearly 92% of the pixels in both datasets have values from the same original source (either the ISODATA cluster tree or a nearby homogeneous pixel). These are indicated by the light gray pixels in Figure 1-3C. Ideally, the majority of the image would be gray, indicating that results are comparable between runs, even where using a different datasets over the same area. The distance map shows a similar pattern, with just over 83% of the pixels in each dataset having the same distance to their source (Figure 1-4C). This shows that the same matches were made in the two independent runs, even where starting with different datasets (radiance versus emissivity) over the same area.

Source and distance maps both show a generally random spatial distribution, although some patterns are apparent. In particular, the inner crater rim and areas of the outer crater rim can be discerned in the source and distance maps. The pattern is not a complete match, with slightly more super-resolved pixels having an ISODATA tree source along the crater rim. This may reflect a lack of homogeneous pixels in this area, due to the albedo differences between sunlit and shadowed slopes in the visible data. The reduction in homogeneous pixels lowers the chances of finding an appropriate match because the number of homogeneous pixels nearby any given pixel is reduced. A slightly greater than average number of the crater rim pixels are also at a greater than average MD from their source. This may also reflect the limited number of possible matches that could be made for that pixel. The cluster center that forms the best possible match for these pixels is formed from both sunlit and shadowed slopes, and therefore does not match particularly well to either subset. On a larger crater, with greater sunlit or shadowed areas,

a sufficient number of homogeneous pixels would exist. However, with a smaller area shown here, and a smaller number of pixels with which to work, these clusters may not have sufficient members to exceed the minimum cluster size in ISODATA.

The original resolution of IR channel 9 is compared to the super-resolved data of the same band (Figure 1-5). The reduction of shadows and increased spatial resolution allows geomorphic forms to be more clearly discerned in the super-resolved IR data. The clarity is nearly comparable to the original VIS data. In particular, this is evident along the ridges that radiate from the central crater along its southern and western rim, and in the greater detail in the shape of the relatively bright spot between these ridges.

Similar levels of increased detail are also visible in the decorrelation stretch (DCS) images (Figure 1-6). Because the DCS results are dependent on the statistics of the input data (e. g., Gillespie et al., 1986), the super-resolved image does have less color intensity due to the nine-fold increase in the number of pixels. However, each image has the same distribution of color patterns, which lends confidence that the super-resolution approach is not adding spurious data or creating artificial units not seen in the original IR data. The combination of the three DCS images displayed together permits differences to be visualized across a broad spectral region at a single glance (Bandfield et al., 2004a). The units that appear reddish-purple in Figure 1-6E denote emissivity spectra with absorptions at the shorter wavelengths. The spectral minima (e.g., Figure 1-7) have shifted towards band 5, or even band 4, from the band 6 minima seen in the more blue pixels in Figure 1-6E. Similar color variations are present over these regions in Figures 1-6A and 1-6C. Although pixels with these spectral shapes are present in the original

image, the super-resolved image has far more of them showing greater spatial detail. These patterns reveal the intermingling between the two units (i.e., the redder quartzofeldspathic and the bluer basaltic pixels) at a spatial scale not resolved in the original data. Furthermore, the super resolution results show a clear reduction in pixel to pixel noise and image line striping (e.g., Figure 1-6F), both of which are commonly enhanced in the DCS approach. The box in Figure 1-6F is centered on a cluster of pixels with the same coloration as the quartzofeldspathic material in the bottom left of the image. This region is just large enough (~150 m) to be detected in the original resolution IR data; however, noise in the DCS image makes confident detection questionable. This small cluster of quartzofeldspathic pixels is coincident with two small higher albedo mounds that are surrounded by lower albedo material (marked by the square in Figure 1-8A), and seen in greater detail in Figure 1-8C.

The CTX data shown in Figure 1-8A provides an overview of the area surrounding the central crater at over 40 times (5.76 m/pixel) the resolution of the THEMIS VIS/super-resolved THEMIS IR data. It is possible with CTX to compare the location of quartzofeldspathic anomalies originally detected by Bandfield et al. (2004a) and further resolved in the super-resolved data with the corresponding geomorphic features. Two segments of this image have been highlighted to reflect different areas with possible quartzofeldspathic material. The larger of these regions (Figure 1-8B) is a semi-enclosed topographic low, and contains a number of THEMIS IR pixels highlighted by Bandfield et al. (2004a) as containing a significant quartzofeldspathic component in their spectra. The smaller region (Figure 1-8C) is the site of an isolated THEMIS IR pixel in the end-member map by Bandfield et al. (2004a) to the west of the

central crater. This single pixel corresponds to the region of the two isolated mounds. These can be seen more clearly at the meter (Figure 1-8A) and the centimeter (Figure 1-8C) scale, and consist of higher albedo material.

The region denoted by the larger rectangle at the southern end of Figure 1-8A is a topographic low and is shown in Figure 1-8B. This is the site of the highest concentration of quartzofeldspathic material within the extent of Figure 1-8A. The more quartzofeldspathic-rich pixels are located within this topographic low, and not along the ridge separating it from the next depression to the northeast. However, other quartzofeldspathic pixels lie along the rough terrain that marks the southwestern side of the topographic low. In Figure 1-8B, there is indication of eolian reworking (e.g., dunes or dune-like linear features) from this terrain and deposition into the topographic low. These dunes are composed of a higher albedo material similar to that seen in the two small mounds. It appears to be covering the southwestern slope as well as part of the floor of the depression. If these pixels detected to the immediate southwest are associated with the terrain forming the ridge to that side of the depression, these dunes are likely composed of that weathered (i.e., quartz / feldspar minerals or hydrated silica) material (Bandfield et al., 2004a).

Figure 1-8C shows the two high albedo peaks to the west of the central crater, which is clearly visible, even at the scale of THEMIS VIS. The HiRISE image shows these peaks in even more detail (Figure 1-8C). The high albedo material appears to be located along their western sides, facing a region from which much darker albedo material is being weathered and transported to the east. The location of this higher albedo material is consistent across multiple

instruments on multiple satellites, making it unlikely to be the result of sun-facing slopes. It seems likely, given their apparently equal elevations, that these peaks may be composed of the same underlying bedrock as the terrain forming the southwestern ridge of the depression in Figure 1-8B. If this bedrock is part of the putative quartzofeldspathic body proposed by Bandfield et al. (2004a), it would have a minimum north-south extent of 7.5 km in this region. If on the other hand, these peaks are composed of the hydrated silica unit detected by Ehlmann et al. (2009), the results here validate the hypothesis that the material is derived from local bedrock and subsequently weathered.

Some regions highlighted by Bandfield et al. (2004a) as having a higher concentration of the quartzofeldspathic signature are not homogeneous on the subpixel scale. The ratio of band 6 / band 5 in the super-resolved image (Figure 1-9) displays a much greater range of values than in the original IR data. The smaller pixel size would increase the likelihood of detecting compositionally “pure” pixels, composed of only a single end-member. Smaller pixels also permit a more accurate measurement of the surface area covered by spectral end-members with lower percentages. If these end-member outcrops are distributed heterogeneously, with erosion further distributing material, and these are subsequently sampled at a smaller spatial scale, some pixels will have an increase in percentage of that end-member, thus improving detectability. In particular, the concentration towards the lower center of Figure 1-5D is shown to be composed of isolated subpixels (Figure 1-9B). This is the region of the depression shown in Figure 1-8B. Areas with a stronger absorption in band 5 versus band 6, shown as green pixels in Figure 1-9B, tend to overlap regions with the terrain, as well as the dune-like material within the depression highlighted in Figure 1-8B.

The nature of a contact between two unconsolidated sedimentological units (i.e., the quartzofeldspathic material and the basaltic plains material) could provide insight into the process that produced it. Furthermore, the physical mixing of these units within a 100 m pixel would produce a spectrum that showed a linear combination of spectral features in proportion to the areal abundances of those units (Hamilton et al., 2001; Ramsey and Christensen, 1998; Wyatt and McSween, 2002). More traditional linear spectral deconvolution approaches have been used to identify the areal abundance of the spectral end-members using either their image-based or library-based spectra. However, these approaches are limited by the spectral resolution of the IR pixel, the availability and completeness of the spectral library, and the results do not reveal anything about the nature of the actual mixing patterns within that pixel. Using the technique here, the spatial resolution is improved, and the nine-fold increase in the number of IR pixels can be further interrogated using more traditional DCS or linear deconvolution approaches. In order to more accurately assess the results of super resolving THEMIS IR data, emissivity spectra were extracted and examined (Figure 1-7). Homogenous pixels (Figure 1-7A) show very little variation in their super-resolved spectra. The error bars denoting the range of values per band are actually less than the width of the spectral line. Non-homogenous pixels (Figure 1-7B) show shifts in peaks, and data values that vary as much as 0.047 from the original spectrum at any one band. The new spectra show that at least some sub-pixel scale surfaces have been enhanced and are more apparent in the super-resolved data than in the original resolution data. For example, the spectra (Figure 1-7C) of two nearby IR pixels at 36 m/pixel resolution show this effect. The dashed spectrum (quartzofeldspathic-rich) has a shift in the band minimum compared to the dotted spectrum (basaltic plains material). Upon super-resolution, the pixel from which the dotted spectrum was extracted is now subdivided into nine new IR pixels. The spectrum from

one of these has a band minimum, which has a well-defined shift to shorter wavelengths relative to the original pixel. This indicates the presence of more felsic material, and shows a shift towards the quartzofeldspathic solid spectrum in terms of band shape and band minimum (Bandfield et al., 2004a).

1.5 CONCLUSIONS

The super-resolution approach presented here produces enhanced IR images that are both visually appealing and radiometrically accurate, meaning the data can be analyzed with tools such the DCS and linear spectral deconvolution. Unlike pan-resolution techniques, areas where there is a significant difference in the appearance of a surface in the VIS and IR (such as warm dark sand, which would be bright in the IR and dark in the VIS) are handled accurately. Meaningful results are produced over these regions, with a clear reduction in pixel and line image noise, which appear as color variations in the DCS image of the original radiance data. The source of super-resolved emissivity and radiance data show a strong correlation, as does the pattern of Mahalanobis distances between the original and super-resolved pixels. These similarities indicate that it does not matter whether the radiance data is first super-resolved, followed by the separation of temperature and emissivity, or whether the temperature / emissivity separation is performed at the original resolution, and the emissivity data are subsequently super-resolved. However, advantages to super-resolving the radiance data first include the ability to also have a higher resolution temperature image and a decrease in noise in the final product. The ISODATA clustering algorithm is detecting the same patterns within two different data sources.

The similarity of the source and distance maps also indicates that although the ISODATA algorithm is seeded with random data, the algorithm tends to converge upon the same solution, which indicates a high level of reproducibility in the final results.

The results reveal geomorphic and unit contact details within the smaller craters that are not evident in the original resolution. The super-resolved sub-pixels of an original pixel on the border of the area identified as a quartzofeldspathic exposure show a shift in the location of the spectral minima, away from the minimum of the original resolution pixel and towards the location of the minimum of the quartzofeldspathic signature detected in Bandfield et al. (2004a). This indicated a diffuse compositional boundary, likely an eolian contact of two sedimentary surface units rather than a contact from two distinct rock units. The contact and the areal extent of the unit is therefore more accurately defined, which gives strength to the idea that this technique can be used to enhance the detection of small scale deposits previously not identified.

These results are a significant expansion beyond those presented by Tonooka (2005), where three separate spatial / spectral regions collected from Earth-orbit were used. This technique uses a more robust clustering algorithm, has been refined for accuracy, operates on both Earth and Mars orbital data, and can operate with only two spatial / spectral regions. It can be easily applied to any instrument that contains data from two different spatial / spectral regions, provided that both are multi-spectral, where a greater number of bands will produce both more and better fitting cluster centers during step 3. Here it has been used to create THEMIS IR data with a spatial resolution matching that of co-located THEMIS visible data. Future work will expand the model to use data sets collected by other instruments. For example, we expect to

super-resolve THEMIS IR data using higher-resolution, multi- or hyper-spectral data from CRISM or HRSC. The strict requirement for contemporaneous data collection will be assessed to explore utilization of data collected over a longer time span. This will permit a significant increase in the areas that could be super-resolved to resolutions of 20m or better. Further, such relaxation would allow the application of this technique to the same location over multiple time frames. Extended super-resolved coverage would be a powerful tool for detection of recent surface changes, small scale spatial anomalies, and potentially permit the creation of a super-resolved thermal inertia dataset. In examining this region with both super-resolved THEMIS IR data and higher resolution data not available at the time of publication for Bandfield et al. (2004a), it is apparent that the spectra from the quartzofeldspathic material are located in regions composed of sub-pixel scale rocky terrain and material that is likely weathered from it. This material, as speculated in Bandfield et al. (2004a), could be an underlying granitoid body that was fractured during the later impact event. It is at least 7.5 km long in this region on the north-south axis if both regions highlighted in Figure 1-8 are from the same body. Ehlmann et al. (2009) identified hydrated silica in this same area in the form of mobile material derived from an unknown rock unit. The source unit for the mobile material, whether hydrated silica, quartzofeldspathic minerals, or a combination of both, could be the unit seen cropping out.

2.0 MODIFICATION AND ANALYSIS OF A SUPER-RESOLUTION ALGORITHM

2.1 INTRODUCTION

Super-resolution is the process of obtaining a spatial resolution greater than that of the original (or native) resolution of a data source. The super-resolution technique presented here is a modification of an algorithm (Tonooka, 2005) that was tested successfully using multi-resolution data from the Earth orbiting Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument (Yamaguchi et al., 1998). The primary focus of these modifications was to enable the algorithm to process data from the THEMIS instrument (Christensen et al., 2004). Other instruments, with higher spatial resolution in different spectral regions than the THEMIS instrument, have shown a need for higher resolution Thermal Infrared (TIR) data than provided natively by THEMIS. However, it is unlikely that any TIR instrument with a higher spatial resolution than the THEMIS instrument will be sent to Mars in the near future. The modified super-resolution algorithm may help meet the need for this higher resolution TIR data. THEMIS data have different spatial resolutions, different number of bands within each spectral range, lack an intermediate spatial / spectral range, and have data that is organized and processed differently than ASTER. As a result, significant modification of the algorithm was necessary. These modifications include changes to permit the algorithm to use data with only two different spatial resolutions and to expand the range of acceptable input data

for the instrument. In addition, other modifications included performance improvements and the implementation of a different clustering method within the algorithm. After these modifications, testing was done to ensure that no errors were introduced, and that the modified algorithm produced the same results on test data as the original method. The algorithm, as described, has been implemented in the Interactive Data Language (IDL) scripting language and should be easily implementable in other programming languages.

2.2 MODIFICATION OF THE ALGORITHM

The technique initially described in Tonooka (2005) is a ten step process. In the first five steps, the algorithm uses ASTER Visible / Near Infrared (VNIR) data to super-resolve ASTER Shortwave Infrared (SWIR) data. Steps six through ten repeat the same algorithmic process of the first five steps to super-resolve the ASTER TIR data, but make use of ASTER VNIR data and the newly super-resolved ASTER SWIR data as the higher-resolution data set. In adapting this algorithm to apply to the current ASTER and THEMIS instruments' two-resolution data, it was necessary to simplify the algorithm to the first five steps. When intermediate data are available, as in older ASTER data, those data are super-resolved first, and a three-level tree built. After this modification, the super-resolution algorithm can be implemented as a five-stage process. Within each stage, some changes have occurred from the original description provided in Tonooka (2005). The description of each stage and how it varies from the description of the original algorithm is documented below.

2.2.1 Step 1: Convolution with the Point Spread Function

In the first step, higher resolution bands are convolved with the instrument's Point Spread Function (PSF) to produce a dataset with the same pixel size as the lower resolution bands. The PSF describes the amount of blurring induced by the instrument. In an ideal instrument, 100% of the signal sensed within any given pixel would originate only within the area of that pixel. In actual instruments, a large portion of the sensed signal for any given pixel originates within the neighboring pixels. In many instruments, the PSF is either symmetrical or assumed to be symmetrical in both the X and Y axis. The original algorithm defines the PSF of the ASTER instrument as a 2 dimensional Gaussian, with a value determined from pre-flight measurements. In modifying the algorithm, the PSF specification has been converted to use the alpha notation described in Townshend et al. (2000). This provides a more generalized approach for PSF usage, and enables the application of the algorithm to other instruments. In the particular case of the ASTER instrument, the ASTER instrument's PSF's α is calculated to be approximately 6.56 x 10^{-2} based on the Gaussian equation and specified standard deviation given in Tonooka (2005). This translates to 75.5% of a pixel's signal originating from within that pixel, with the rest of the signal originating from the neighboring 8 pixels (Table 2-1). For those pixels that do not have a full complement of 8 neighbors, or edge pixels, a value is still calculated but these pixels are not considered to be fully valid data.

The actual IDL code implements Step 1 as three separate actions. First, the high-resolution image is resized to match the number of pixels in the low-resolution image. This is accomplished using the IDL frebin function, freely available from http://idlastro.gsfc.nasa.gov.

Table 2-1 The ASTER Point Spread Function (PSF) used in this algorithm, based on the Gaussian equation in Tonooka (2005). This PSF is based on an alpha value of 0.065, and shows 75.5% of the signal recorded for a pixel originates within the spatial area of that pixel, and the rest originates within the surrounding pixels.

0.0043	0.0570	0.0043
0.0570	0.7546	0.0570
0.0043	0.0570	0.0043

Next, the resized data are convolved with the PSF using the IDL convolve function, also freely available at the same site. Finally, as the data were in digital number (DN) format originally, the degraded resolution data are rounded to an integer value, and then reconverted to floating point.

2.2.2 Step 2: Identifying Homogeneous Pixels

The second step creates a homogeneous pixel map. The original implementation by Tonooka (2005) does this by examining the data with degraded resolution pixels and their component (original, high resolution) pixels. If the standard deviation, in each band, of the component pixels is less than a specified threshold, that degraded pixel is considered to be homogeneous. In Tonooka (2005), the threshold is defined as the band average of the spatial standard deviation over the whole image of each band. Based on this definition, the code first calculates a standard deviation for each band. Next, these values are averaged together to create the threshold for homogeneity. The component pixel's standard deviation is then calculated for each band. If the greatest standard deviation is still less than the threshold value, that pixel is considered to be homogeneous. All edge pixels are assumed to be non-homogeneous due to their handling in step 1 of the algorithm.

2.2.3 Step 3: The Cluster Tree

The third step generates a tree by initially clustering the homogeneous pixels from the degraded data, and then further clustering co-located low resolution data within each cluster. In both the original (Tonooka, 2005) and this implementation, the distance measurement used for clustering is Mahalanobis distance instead of Euclidian distance. Mahalanobis distance (MD) measures the difference in variance and correlation between bands of data (Mimmack et al., 2001). In effect, MD can be thought of as being equivalent to Euclidian distance for multi-dimensional data, except that it takes into account the differences of scale along each axis and discounts dimensions which are highly correlated. The original implementation uses MD in conjunction with the K-Means clustering algorithm to create a tree of clusters. Because the K-Means algorithm requires an assumption be made *a priori* as to the number of clusters in the data, this implementation made use of a slightly modified ISODATA algorithm (Ball and Hall, 1967). The ISODATA algorithm used within this work was implemented by the author in IDL based on the published algorithm (Ball and Hall, 1967), and uses MD for clustering.

The ISODATA algorithm is similar to the K-Means algorithm, but does not require any assumptions be made about the final number of clusters. In both cases, a number of initial cluster centers are provided to the algorithm as it begins. For ISODATA, choosing too few cluster centers initially will lead to a large number of clusters being split; choosing too many cluster centers initially will lead to a large number of clusters being joined. Either case increases the run-time of the algorithm, and in extreme cases may prevent the algorithm from running to completion in a reasonable amount of time.

In the ISODATA implementation within the modified algorithm, after the user has stated how many initial clusters should be used, this value is increased by 10%. A random pixel is selected from the degraded data. This pixel becomes the cluster center for the first cluster. The MD between this pixel and every other pixel is then calculated. The pixel with the largest MD is selected as the second cluster center. The third cluster center is the pixel with the largest combined MD to the first and second clusters; the fourth cluster center has the largest combined MD to the first three centers, etc. If the MD to any existing cluster center is found to be 0, that pixel is rejected, and not permitted to be a new cluster center. This prevents pixels with the same spectral composition from being selected multiple times, even if that spectral composition is sufficiently different from the rest of the image to give it the largest combined MD. After all potential cluster centers have been calculated, the distance between every cluster center is measured. The 10% additional cluster centers are then removed by finding the cluster center pairs which are closest to one another, and removing the first member of those pairs. This helps de-emphasize data variability near the extreme data values, and leads to a more even distribution of the initial cluster centers across the data. After the 10% additional cluster centers have been removed, all data are clustered within ISODATA using the remaining cluster centers to seed the initial iteration of the algorithm. After the initial iteration, subsequent iterations recalculate the cluster centers based upon the values of the data within that cluster.

2.2.4 Step 4: Assignment of Super-Resolved Values

During step 4, initial values are assigned to each super-resolved pixel. As in the original algorithm, these values are selected from both the data tree created in Step 3 and from nearby homogeneous pixels. The user is required to define a spatial radius, in number of low-resolution

pixels, to check for the best-fit. The super-resolution algorithm then finds all homogeneous pixels within that radial distance of the super-resolved pixel, and calculates the MD to each pixel. As each pixel's MD is calculated, it is compared to the current MD. If the new value is lower, the spectrum of that pixel and the MD value are updated, resulting in the selection of the spectrum that has the lowest MD from within the specified radial distance. After the best fit within the map is found, the data are then compared to the cluster tree. This is done by first finding the minimum MD between the co-located high-resolution pixel and the high-resolution cluster centers. The associated low-resolution sub-clusters are then compared to the original spectrum. If the minimum MD from these comparisons is less than the MD from the map, the spectrum of the sub-cluster center is used instead.

In data with three spatial resolutions available, the MD is calculated by assigning fractional values to the high and middle resolution MD's. Weighting between spectral regions is defined by user input, and required to be a value between 0 and 1. This value, *W*, is multiplied by the higher resolution MD, and 1-W is multiplied by the intermediate resolution MD. The default recommended *W* value is 0.7, reflecting the greater importance given to matching the higher spatial resolution data. The MD of the intermediate data measures differences within only the pixels co-located with the higher resolution cluster; as such, small differences within the limited data are magnified beyond the MD that would be calculated for them when using the entire data.

2.2.5 Step 5: Radiometric Correction

No significant changes from the original algorithm were made in the final step. After allocation of spectra in the previous step, values are only best-fit data. In order to maintain radiometric

accuracy, the newly super-resolved data are degraded back to the original resolution with the PSF for comparison to the original data. A correction factor is then calculated by finding the difference between the original and degraded super-resolved data. Rather than splitting this correction factor equally between the co-located super-resolved pixels, it is instead allocated according to the MD of the super-resolved pixels. Super-resolved pixels with high MD are poorer fits than their neighbors; in this way, the correction is given dominantly to poor fits. The resulting correction is then added to the values created in Step 4.

2.3 TESTING METHODOLOGY

2.3.1 Datasets

Three different sets of data were used for performance and behavior analysis of the algorithm. The first set consisted of ASTER data acquired on May 16, 2000 over Ishioka City, Japan. This scene was the same scene used in Tonooka (2005), and was chosen for comparative purposes. The subset used for super-resolution covers an area of 9.9 x 8.1 kilometers (km). Where pixel locations are specified, pixel 0, 0 in the upper left of the super-resolved area was located at pixel 2496, 2460 of the full scene VNIR data. One significant difference from the Tonooka (2005) paper is that the ASTER SWIR bands were crosstalk corrected prior to the application of the modified super-resolution algorithm. ASTER SWIR bands suffer from crosstalk contamination between detectors due to a stray light error in which incident light to band 4 propagates to other bands via multireflection (Iwasaki and Tonooka, 2005). A correction for this problem was not available at the time of the original super-resolution implementation.

The second data-set consisted of THEMIS data I33902002 (TIR) and V33902003 (VIS), and provided coverage of a putative chloride deposit on Mars. This deposit is located along the southern portion of a crater centered near 180.5° E, -27.0° N, in northwestern Terra Sirenum, with the chloride unit centered near 180.46° E, -27.25° N. The unit covers an area roughly 15 x 10 km. These data were converted to emissivity and atmospherically corrected using previously described methods (Bandfield et al., 2004; Hughes and Ramsey, 2010) prior to application of the super-resolution algorithm.

The third set is an artificially generated terrain, sometimes referred to as TerrGen, using an alpha value of 0.14645 (or 50% of the pixel energy originates from that pixel), two end-members, and two-fold difference in spatial scale between high and low resolution pixels. Both the high and low resolution data have 4 bands. End-member spectra were created and selected separately for each resolution, with values that range between 0 and 1000. These particular values are meaningless. As the super-resolution algorithm looks only for spatially co-located patterns within data, what the data actually measures is irrelevant to the process. The algorithm works equally well on raw DN, radiance, emissivity, or even non-spectral measurement values, with the caveat that the end-user must ensure no other factors can influence patterns in one data set over the other. As an example, the ASTER and THEMIS data are both atmospherically corrected to minimize contributions from the intervening distance.

Pixels were allocated randomly at a low spatial resolution, with a 30% chance of being purely end-member 1, 30% chance of being purely end-member 2, and a 40% chance of being a randomly mixed pixel. In mixed pixels, each sub-pixel had an equal chance of being either end-

member. Mixed sub-pixels could have easily been created in this process as well, but only end-member pixels were used to simplify measurement and interpretation of results. During this process, a high-resolution map and a “super-resolution” map were created. Pixels from the “super-resolution” map were convolved with the PSF for creation of the low-resolution map, leading to a large number of possible final spectra for any given low-resolution pixel. Five possible types of resultant spectra can be created just within the space of the low resolution pixel: purely end-member 1, mostly end-member 1, mixed evenly, mostly end-member 2, and purely end-member 2. Spectra were then modified during convolution with the defined PSF, in which half of the pixel value is derived from the adjacent pixels. As an example of how many ways any individual pixel's spectrum could be modified by the surrounding pixels, ISODATA clustering of this dataset found 245 separate clusters in less than 100 iterations rather than 2 (end-members) or 5 (pixel composition types) that may be initially expected.

The performance of each of the five steps is determined by user input. An initial run was performed for each data-set, using the default values (Table 2-2) for all variables. Default values were based on those provided by Tonooka (2005) for Earth, and used as a guide for THEMIS and TerrGen data. During testing of these variations for each step, one variable was systematically altered while all non-tested user input values were held fixed at their default value. The output data were then compared to each other and to the data from the initial super-resolution results of each data-set. This was done by examining the behavior of the super-resolution algorithm during data processing, the quality of the output in terms of image appearance, and the statistical distribution of the data as compared to the original non-super-resolved data.

Table 2-2 The default values used for variables within the super-resolution algorithm. During testing, one value was allowed to vary in a systematic fashion, whereas all other values were held fixed.

Variable	Earth Data	Mars Data	Artificial Data
Alpha	0.06565	0.1	0.14645
V_T_Distance	10	10	10
V_S_T_Distance	10	n/a	n/a
Weight	0.7	n/a	n/a
ISODATA Variables			
Initial VIS Clusters	50	50	50
Initial SWIR Clusters	10	n/a	n/a
Initial TIR Clusters	5	10	10
ChangeLimit	0.50%	0.50%	0.50%
MaxStdDev	4	4	4
MinDistance	2	2	2
MaxPairs	4	4	4
MinMembers	0.01%	0.01%	0.01%
VIS BandMax	255	0.012	1000
SWIR BandMax	255	n/a	n/a
TIR BandMax	4300	1.05	1000
Limit	100	100	100

2.3.2 Step 1 Testing Methodology

To test the effect of incorrect alpha values on the super-resolution process, data were super-resolved using 5 different alpha values. Each dataset was super-resolved with alpha values equivalent to 0%, 25%, 50%, 75%, and 100% of a pixel's signal originating from within the area of that pixel. In cases where the correct alpha value is close to one of these values, the correct value was used instead. ASTER data were processed with an alpha value equivalent 75.5% instead of the alpha value equivalent to 75%. THEMIS data were processed with an alpha value equivalent to 50.6% instead of the alpha equivalent to 50% used for the ASTER and TerrGen data.

2.3.3 Step 2 Testing Methodology

Homogeneous pixels can be defined in a number of different ways. The super-resolution algorithm defines a single threshold value as the average of the standard deviations of each band for the entire image. To test the impact of other definitions, the code was modified to test the following threshold definitions: a single value defined as the average of the standard deviations of the super-resolved area or area-of-interest, multiple values, with each band being compared to the standard deviation of that band across the entire image, and multiple values, with each band being compared to the standard deviation of the super-resolved area or area-of-interest. For Earth data, the super-resolved area matched that used in Tonooka (2005). For artificial data, the area-of-interest was defined as the middle 250 x 500 low-resolution pixels. For Mars data, the super-resolved area was defined as the area immediately surrounding the putative chloride deposit.

2.3.4 Step 3 Testing Methodology

Step 3 represents the most significant change from the algorithm in Tonooka (2005). There are several user-defined values within the ISODATA process, but the two with the largest impact on the final results are the initial number of clusters used and the maximum number of iterations. These variables define how far the system starts from a final, stable cluster map and whether or not a sufficient number of iterations are permitted to reach this step. Other values, such as the minimum number of members per cluster or maximum number of clusters to join in any iteration, can impact this process, but do not have the same controlling effect. As a result, testing of step 3 was done by varying these values one at a time. The initial number of clusters was varied between 5 and 1000, and the number of iterations was varied between 10 and 10,000. The super-resolved ASTER data results were also compared to Tonooka (2005).

2.3.5 Step 4 Testing Methodology

The main variable impacting the result of step 4 is the radius to use for searching nearby homogeneous pixels for initial super-resolution values. By default, the program uses a radius of 10 low-resolution pixels, which is the same value used in Tonooka (2005). Tests were performed setting this value to 5, 15, and 20 pixels. Data were also super-resolved using a value of 0 (all values from the tree). This permitted the examination of the influence of the cluster-tree on the super-resolution process.

2.3.6 Step 5 Testing Methodology

The correction factor calculated within this step is allocated among the super-resolved sub-pixels based upon their MD within the default code. This can be interpreted in two different ways, with both being tested. In the first method, the difference between the original and degraded super-resolved data was divided by the convolved MD of the super-resolved pixels, and this amount was multiplied by the MD of each super-resolved pixel to create the correction. In the second method, the difference between the original and degraded super-resolved data was divided by the sum of the MD of the associated super-resolved pixels, and this amount was multiplied by the MD of each super-resolved pixel to create the correction. An additional third method was tested, in which just the difference between the original and degraded super-resolved data was used as the correction. A method based upon allocating the correction factor based using the shape of the instrument PSF was also considered, but was discarded when it proved infeasible to apply to data whose size difference was not a multiple of the scale of the PSF.

The clearest way to illustrate the impact of the correction factor and different ways of allocating it can be seen by examining a single low-resolution pixel and its associated super-resolved sub-pixels. ASTER band 10 data from Earth is presented in DN, whereas THEMIS band 4 data are emissivity values. This allows a comparison of the effects of relative scale; ASTER DN are generally several orders of magnitude larger than their associated MD values, THEMIS emissivity are of the same general magnitude. In both cases, the pixel whose upper left super-resolved sub-pixel was located at 240, 240 was examined. By examining a single pixel and a single band, it is feasible to show how the correction factor was developed, and how it was then allocated using different methods.

2.4 RESULTS

2.4.1 Step 1 Results

The DN and calculated calibrated radiance of the super-resolved ASTER data shows the impact of choosing an incorrect PSF (Figure 2-1). If the correct alpha value is used, the mean calibrated radiance of the super-resolved data is closest to that of the original data compared to the other datasets. As the alpha value moves away from the correct value in either direction, there is an increase in the mean calibrated radiance values. Similarly, the correct PSF produced the fewest number of VNIR clusters, with 100 clusters found in ISODATA. As the alpha value used for calculating the PSF moves away from the correct value, a greater number of clusters are found.

The impact of an incorrect PSF on Mars emissivity data is less clear. All five average image spectra plot on the original average image spectrum. The average spectral difference between data super-resolved with any PSF and the original data was 4.0×10^{-5}. The largest difference between original and super-resolved average spectrum in any one band is 1.2×10^{-3}, with total spectral differences across all bands ranging between 9.0×10^{-5} and 0.004. The TerrGen data shows similar results, with all PSF values producing similarly equal results.

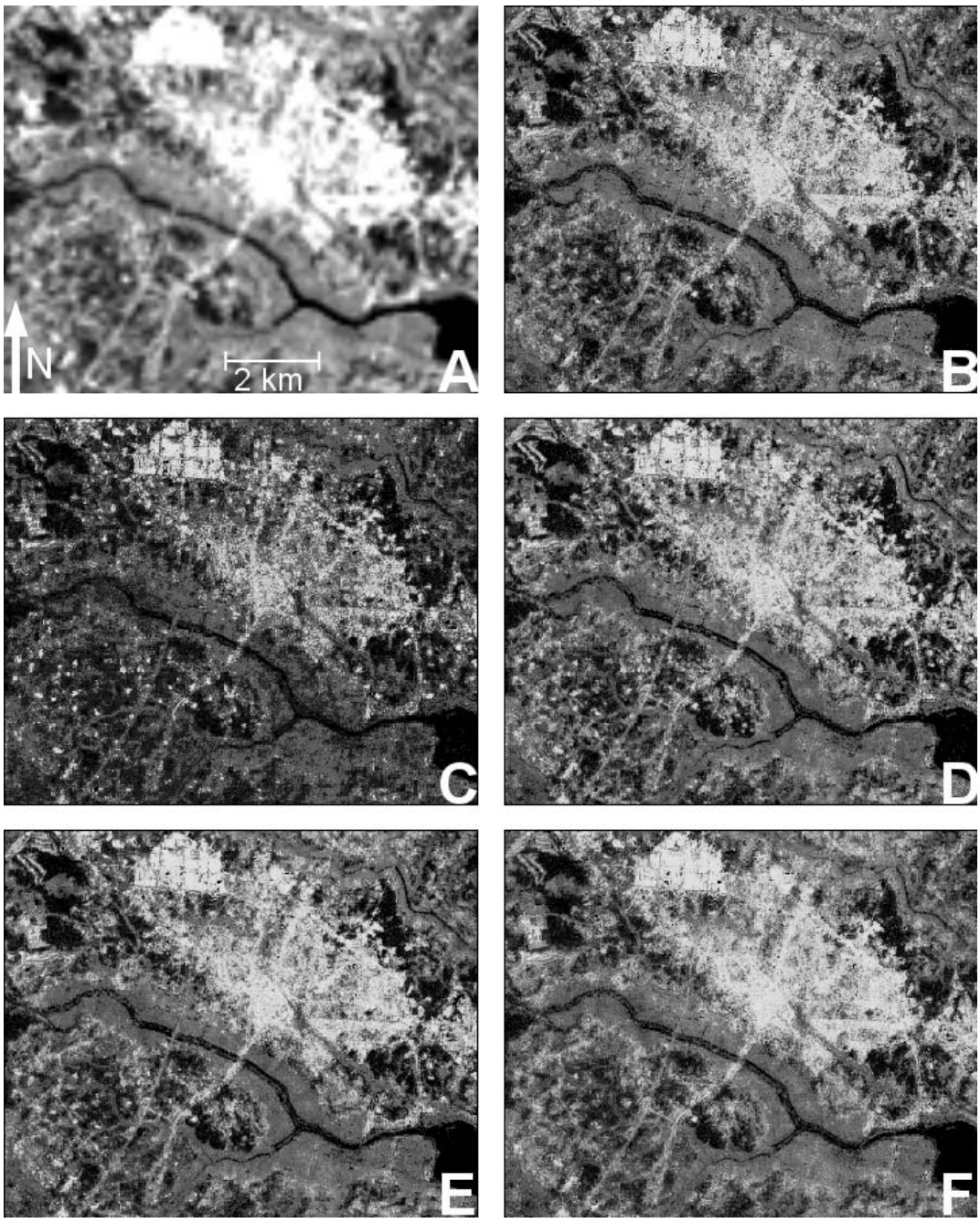

Figure 2-1 The original data compared to super-resolved data using different alpha values for calculation of the Point Spread Function. All images are ASTER Band 14, and are histogram matched to image A for easy comparison. (A) The original Band 14 data, clipped to the area of super-resolved data, with a linear 2% stretch. (B). The super-resolved product with an alpha of 0.06565 (75.5%), which is the correct alpha value to use for ASTER and produces both the most radiometrically accurate and clearest overall result. (C) The super-resolved product with an alpha of 0.5, or 0% of the pixel's energy originates from within its spatial area. (D). The super-resolved product with an alpha of 0.25 (25%). (E). The super-resolved product with an alpha of 0.14645 (50%). (F). The super-resolved product with an alpha of 0.0 (100%)

2.4.2 Step 2 Results

The threshold used for determining whether or not a degraded high-resolution pixel is homogeneous or not is defined in Tonooka (2005) as the band average of the spatial standard deviation over the whole image of each VNIR band. This was implemented into the work examined here as a single value, calculated by averaging the standard deviation of each band together. For the ASTER data, the average standard deviation of the entire scene for bands 1, 2, and 3N is 32 Digital Number (DN, rounded to the nearest integer value). For the sub-scene, it is less than half that amount, (15 DN). As expected from this being an average, the threshold value is greater than the standard deviation of at least one band in both cases. Where the threshold values are determined by band instead, the thresholds become 35, 25, and 37 (for the whole image) or 12, 13, and 20 (for the sub-image). The difference between 12 DN (band 1, area of interest only) and 32 DN (the average of the standard deviations of the entire scene) is significant; 12 DN represents less than 5% of the total possible data range (0 – 255), while 32 DN is 12.5%. The default threshold value (32 DN) identifies 82,442 pixels convolved to the intermediate spatial resolution (93% of the scene) and 6,991 pixels convolved to the low spatial resolution (70% of the scene) as homogeneous.

The Mars data had a default threshold of 1.43×10^{-4} emissivity, or 2.4% of the data range, with the standard deviation of band 4 being one quarter and band 5 being one half of this value. If only the area of interest is considered, this single threshold decreases to 1.09×10^{-4} emissivity, with an equivalent drop in standard deviation across all bands. The default value selects 74,393 pixels convolved to the lower resolution as homogeneous, or 11% of the scene.

The difference between per-band and single value can be more clearly illustrated within the TerrGen artificial data. Where a single threshold value is used from the entire image, the calculated value of 380.8 represents over one third of the data range. As the TerrGen data is randomly distributed, unlike a natural surface, there is no significant difference between the threshold values of the whole image and the image subset. Ignoring the effect of the PSF, this does correctly select only those pixels which are purely one of the two end-members. For those pixels in which there was an even contribution from both end-members, 3 bands exceed this threshold, whereas the pixels in which the contribution is mostly from one end-member have 2 bands exceeding the threshold. As the impact of the PSF is considered though, contributions from surrounding pixels work to homogenize the data. As a result, the super-resolution program found 322, 648 homogeneous pixels instead of the actual 300,000 created homogeneous pixels. This represents 4.5% of the total scene being incorrectly selected.

2.4.3 Step 3 Results

During processing of both Earth and Mars data, a limitation within the IDL environment was encountered. Within some operating systems, IDL array creation of over 800 megabytes (MB) is problematic, with problems commonly occurring even before this size is achieved. Within the version of IDL used for this work, there is also a limit of 2.0 gigabytes (GB) for array size for any operating system. There were tens of thousands of homogeneous pixels in both data sets, and cluster assignment is done by creating a floating point array (4 bytes per array cell) with dimensions equal to the number of clusters on one axis and the number of homogeneous pixels on the other in order to track MD between pixels and cluster centers. This array rapidly exceeded

the memory allocation or memory addressing capability of IDL, which prevented testing of very large values of initial number of clusters (i.e. all homogeneous pixels as their own cluster center), and limited the number of iterations possible for moderately large starting values.

The artificially generated data were not examined within this test due to these memory limitations. With over 322,000 homogeneous pixels, there was a low limit to the number of possible clusters. For an initial 500 clusters, this limitation was reached before 10 iterations. For any larger number of initial clusters, this limitation was reached during array creation and before any initial cluster assignments could be made. This limitation is implementation dependant; the algorithm would not have this issue in a different language, or if steps were taken to allow writing of memory to disk during array creation within the current implementation.

Results from testing Earth and Mars data are presented in Table 2-3. In both data sets, a metastable state exists around 10 clusters, as seen in the results of an initial 5 end-members regardless of the number of iterations. Due to the small size relative to the initial starting value of the other tests, this metastable state was not encountered at other initial means starting values. Within 50 initial end-members and varying iterations, the default case of 100 iterations shows the greatest number of clusters in the Earth data, and is bordered by fewer means on either side in the Mars data. In most cases, the initial number of means and the number of iterations are both positively correlated with the final number of clusters.

Table 2-3 The initial starting number of VNIR clusters and the maximum number of ISODATA iterations are compared to the final number of VNIR clusters. These two variables are the dominant factors in determining the final number of clusters. Both show positive correlation with the final number.

Mars		Iterations			
		10	100	1000	10000
Initial Clusters	5	10	10	10	11
	50	85	118	92	142
	500	534	1192	1190	1275
	1000	673	2117	2000	2059

Earth		Iterations			
		10	100	1000	10000
Initial Clusters	5	10	10	10	10
	50	64	100	95	98
	500	249	472	528	566
	1000	284	470	540	575

2.4.4 Step 4 Results

In the method of Tonooka (2005), a circle with a radius of 10 low-resolution pixels is checked first for the best-fit homogeneous pixel spectrum before comparison to the cluster tree. In the same data used by Tonooka (2005), this resulted in a mean MD of 1.14 and 71.8% of the super-resolved pixels sourced from the map (Figure 2-2). With increasing radii, a decrease is seen in the mean MD and the percentage sourced from the map appears to asymptotically approach 82%. Decreasing the radius sees a significant drop-off in both fit, as measured by average MD, and percentage sourced from the map, with a radius of 0 showing 99.7% from the tree and an average MD of 2.64. The 0.3% not sourced from the tree were pixels whose MD exceeded the default value used to indicate no fit within the map; the same issue is present within the artificially generated terrain but not within the Mars data. The overall appearance of the image, in terms of visual appeal, improves with increasing radius. The artificial terrain and Mars data show the same trend in mean MD, although the artificial terrain sources entirely from the image by a radius of 5, reflecting the relatively few true end-members present within the artificial terrain data.

2.4.5 Step 5 Results

Table 2-4 shows the distance, initial value, and correction factors of each of the three different radiometric correction methods for the ASTER data. The original low-resolution pixel associated with these data had a value of 1205 DN, whereas the uncorrected DN values range from 1129 to

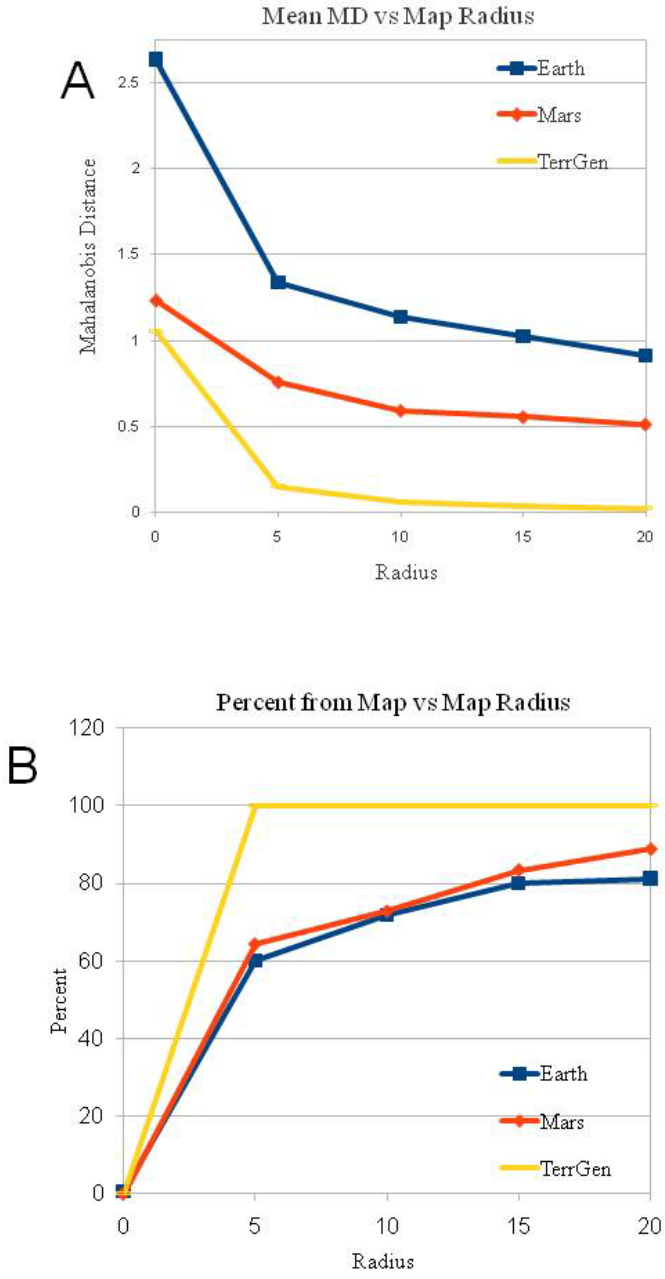

Figure 2-2 This figure shows the effect of altering the radius searched within the image for an adjacent homogeneous pixel. (A) In all three data-sets, there is a decrease of mean Mahalanobis Distance (MD) with increase in radius, asymptotically approaching a lowest value for each data set. In the Earth data, this appears to be 0.9, for the Mars data it appears to be 0.5, and for the artificial terrain it appears to be 0.02. Better data fits have lower MD values. (B) The percentage of super-resolved pixels sourced from the image asymptotically approaches a final value. In the natural data (Earth and Mars), this value is less than 100%. This shows that the cluster-tree is a necessary component of the super-resolution process, and improves the final product. However, as can be seen in a radius of 0, it is not sufficient on its own.

Table 2-4 Super-resolved pixel 240, 240 of ASTER band 10 is traced from the initially assigned super-resolved value through each of the three different radiometric correction methods. The associated original resolution pixel has a value of 1205 DN.

Earth distance map (M.D.)
pixels 240:245, 240:245

1.2657	0.4841	0.3093	0.3374	1.4847	0.6375
0.1921	0.3466	0.3236	0.6581	0.1890	0.6772
0.2679	0.8097	0.3748	0.5881	0.4190	0.2892
0.0314	0.4584	0.5856	0.2464	0.6438	0.7273
0.4084	0.3574	0.7225	1.2173	0.6196	0.5358
0.2605	0.5037	0.1964	0.1469	0.3966	0.5775

Earth equal correction (DN)
pixels 240:245, 240,245

18.2693	18.2693	18.2693	18.2693	18.2693	18.2693
18.2693	18.2693	18.2693	18.2693	18.2693	18.2693
18.2693	18.2693	18.2693	18.2693	18.2693	18.2693
18.2693	18.2693	18.2693	18.2693	18.2693	18.2693
18.2693	18.2693	18.2693	18.2693	18.2693	18.2693
18.2693	18.2693	18.2693	18.2693	18.2693	18.2693

Earth initial values (DN)
pixels 240:245, 240:245

1170	1167	1192	1195	1196	1184
1203	1192	1214	1195	1188	1173
1202	1196	1195	1213	1195	1194
1161	1196	1195	1209	1234	1192
1169	1183	1129	1205	1195	1195
1169	1170	1194	1140	1195	1129

Earth convol corrected (DN)
pixels 240:245, 240:245

1215	1184	1203	1207	1248	1206
1210	1204	1225	1218	1195	1197
1211	1225	1208	1234	1210	1204
1162	1212	1216	1218	1257	1218
1183	1196	1154	1248	1217	1214
1178	1188	1201	1145	1209	1149

Earth convol correction (DN)
pixels 240:245, 240:245

44.5812	17.0522	10.893	11.8831	52.2968	22.4544
6.76696	12.2077	11.3971	23.1811	6.65795	23.8537
9.43707	28.5205	13.2033	20.7135	14.7598	10.187
1.10428	16.1467	20.6272	8.67833	22.6755	25.6179
14.3847	12.5898	25.449	42.8773	21.8264	18.8732
9.17682	17.7412	6.91888	5.17361	13.9714	20.3404

Earth sum corrected (DN)
pixels 240:245, 240:245

1171	1167	1192	1195	1197	1185
1203	1192	1214	1196	1188	1174
1202	1197	1195	1214	1195	1194
1161	1196	1196	1209	1235	1193
1169	1183	1130	1206	1196	1196
1169	1171	1194	1140	1195	1130

Earth sum correction (DN)
pixels 240:245, 240:245

1.2643	0.4836	0.3089	0.3370	1.4831	0.6368
0.1919	0.3462	0.3232	0.6574	0.1888	0.6765
0.2676	0.8088	0.3744	0.5874	0.4186	0.2889
0.0313	0.4579	0.5850	0.2461	0.6431	0.7265
0.4079	0.3570	0.7217	1.2159	0.6190	0.5352
0.2602	0.5031	0.1962	0.1467	0.3962	0.5768

Earth equal corrected (DN)
pixels 240:245, 240:245

1188	1185	1210	1213	1214	1202
1221	1210	1232	1213	1206	1191
1220	1214	1213	1231	1213	1212
1179	1214	1213	1227	1252	1210
1187	1201	1147	1223	1213	1213
1187	1188	1212	1158	1213	1147

1234. After correction, the equal allocation method has the value closest to original data, with a convolved value of 1204.82. As the DN values are integers, this is rounded to 1205 DN. The weighted method using the convolved distance of 0.5187 as the divisor to calculate the correction factor produces the next best results, with a convolved value of 1204.71; this also rounds to 1205 DN. The weighted method using the summed distance of 18.2893 produces non-radiometrically accurate results. The convolved and rounded value produced with this method 1190 DN.

The results of the same tests to THEMIS band 4 are shown in Table 2-5. The original low-resolution pixel associated with these data had an atmospherically corrected emissivity of 0.9766, whereas the non-corrected pixels had values ranging between 0.9689 and 0.9770. The same pattern of correction results was seen as in the Earth data. The equally allocated correction produced a convolved value of 0.9763, the weighted by convolved distance correction produced a convolved value of 0.9761, and the weighted by summed distance method produced a convolved value of 0.9737.

Table 2-5 Super-resolved pixel 240, 240 of THEMIS band 4 is traced from the initially assigned super-resolved value through each of the three different radiometric correction methods. The original resolution pixel associated with these data has an atmospherically corrected emissivity value of 0.9766.

Mars distance map (M.D)
pixels 240:242, 240:242

0.0623	0.0835	0.0890
0.1742	0.2030	0.1805
0.0096	0.5068	0.0704

Mars initial values (emissivity)
pixels 240:242, 240:242

0.9696	0.9769	0.9697
0.9735	0.9770	0.9704
0.9689	0.9708	0.9731

Mars convol correction (emissivity)
pixels 240:242, 240:242

0.0012	0.0016	0.0017
0.0034	0.0039	0.0035
0.0002	0.0098	0.0014

Mars sum correction (emissivity)
pixels 240:242, 240:242

0.0002	0.0002	0.0003
0.0005	0.0006	0.0005
0.0000	0.0015	0.0002

Mars equal correction (emissivity)
pixels 240:242, 240:242

0.0041	0.0041	0.0041
0.0041	0.0041	0.0041
0.0041	0.0041	0.0041

Mars convol corrected emissivity
pixels 240:242, 240:242

0.9708	0.9785	0.9715
0.9769	0.9809	0.9739
0.9691	0.9807	0.9744

Mars sum corrected emissivity
pixels 240:242, 240:242

0.9697	0.9771	0.9700
0.9740	0.9776	0.9709
0.9690	0.9723	0.9733

Mars equal corrected emissivity
pixels 240:242, 240:242

0.9737	0.9810	0.9739
0.9776	0.9811	0.9745
0.9731	0.9750	0.9772

2.5 DISCUSSION

2.5.1 Step 1 Discussion

The choice of an incorrect PSF can lead to the creation of radiometrically inaccurate products during the super-resolution process of Earth data. This effect is due to the contribution from surrounding pixels; as the calculated contribution moves away from the correct value, an increasing DN value is seen. At larger alpha values, this was expected; the larger alpha value indicates a decrease in the contribution from the pixel, with a greater signal coming from the surrounding pixels. However, the radiometric correction is applied only and entirely to the pixel, leading to overcorrection. At smaller than actual alpha values, the opposite process leads to the same result. During the calculation of the amount of correction necessary to make the resultant super-resolved product radiometrically accurate, the image is convolved with the PSF. If too little contribution is calculated as coming from the surrounding region, the correction factor that is calculated is too large. However, this correction factor is still split only within the super-resolved sub-pixels. The correct alpha value is therefore critical, and occupies an optima for the radiometric correction process.

Within the artificial data, with only two end-members, and the Mars data, the impact of an incorrect PSF is negligible. In these datasets, there is a trend towards better results with an increasingly centered PSF, or a higher percentage originating from within that pixel. The Mars data has a limited dynamic and, arguably, represents very few end-members. This commonality between the two datasets may be a result of few end-members. The choice of PSF may become important only with greater diversity, as more can be contributed from the surrounding pixels.

2.5.2 Step 2 Discussion

The Earth data shows an extremely high percentage of homogeneous pixels at the default value. This may be accurate, given that the scene has a significant urban component, however, it may lead to too many pixels, relative to the total number, being used to create the cluster-tree. The Mars data presents a perhaps more accurate case, in selecting significantly fewer pixels. Both scenes select roughly the same absolute number of pixels as homogeneous; this value falls within the range that creates good cluster-trees.

Where the Earth and Mars data are compared as percentages of homogeneity, non-intuitive values are observed. Mars is generally spectrally bland, with a surface heavily dominated by basalt and dust. The Earth, in contrast, appears to have numerous end-members to cause heterogeneity. The difference is the comparison to the average standard deviation. Earth data, due to the numerous end-members, has a significantly higher standard deviation relative to the Mars data. This difference is responsible for the lower percentage of homogeneous pixels on Mars. The very spectral blandness present in most of the Mars data sets a higher threshold for homogeneity. A threshold choice other than standard deviation may result in better selection in such cases.

The artificially generated terrain with two end-members is an extreme example, created by selecting the two most dissimilar end-members from 1000 randomly created ones. As a result, it has a single threshold value over both the image and spatial subset of over one third the data range, and even the smallest threshold on a per-band basis is nearly one quarter of the data range. This led to the incorrect selection of 22,648 non-homogeneous pixels in the default case. The

impact of non-homogeneous pixels, selected by having too low a threshold, reduces the utility of the cluster-tree created in the next step. A balance is necessary between having too low a threshold, and including non-homogeneous pixels within the cluster-tree creation, and having too few pixels for good clustering. It appears to be better to be conservative, and as such, use either a per-band threshold, defined only from the area being super-resolved, or a single threshold defined by the minimum per-band standard deviation threshold.

2.5.3 Step 3 Discussion

In Tonooka (2005), k-means clustering was used to build the cluster-tree, with 50 VNIR clusters, 10 SWIR clusters per VNIR cluster (or 500 clusters total), and 5 TIR clusters per VNIR/SWIR cluster (or 2500 clusters total). As these clusters were derived within k-means, these numbers were invariant, and required significant knowledge of the scene prior to clustering in order to make valid choices. ISODATA clustering, as used in this algorithm, uses these same values as the initial number of clusters, but allows the number of clusters to vary, alternatively splitting if they grow too large or joining as their centers become too close. As seen in Table 2-3, the final number of clusters shows a strong positive correlation to both the initial number of clusters and to the number of iterations used for clustering.

In several cases, a relatively low number of final clusters are found, with values near or less than the initial number of clusters. This fact is driven by two conditions within the data. First, initial means are selected by picking a random starting homogeneous pixel as the first means. After that, initial means are selected by picking those homogeneous pixels which are

furthest in total MD from the previously selected means, and are not equal to any of the previously selected means (so as to prevent extreme end-member over-selection). Because the resulting initial mean pixels will be chosen from the periphery of the dataspace, there is a tendency towards consolidation of clusters within the first several iterations. It is not uncommon for the number of clusters to drop to 1/2 to 1/3 of the initial starting value. It is only after peripheral cluster centers have migrated inward that splitting of clusters becomes dominant, and the number of clusters found starts to grow. Finally, when metastable states exist, such as the case for 10 end-members in both datasets, clustering will halt or return repeatedly to this state. This returning trend can be seen in the case of Mars data with 5 initial end-members and 10,000 iterations, in which the 11^{th} cluster was a small offshoot of one of the other clusters. Given several more iterations, it is likely that this cluster would be eliminated as its members migrated back to the originating cluster. The formation of the offshoot cluster was driven by a few extreme pixels, relative to the originating cluster. There were an insufficient number of these pixels, however, to meet the minimum cluster size, and so the cluster was not stable across multiple iterations.

This section of the algorithm is the one of the most time-intensive in terms of cpu usage, and can cause significant issues for memory management. However, improvement is possible. Future work should focus upon improvements to this code. These can be done by making use of implementations available in the libraries of other languages, such as C.

2.5.4 Step 4 Discussion

The ASTER data shows a definite limit to the percentage of data sourced from the map versus from the cluster-tree. A radius of 10 produces acceptable results; an increasing radius results in a lower mean MD, which can be used as a measure of the fit of the model data to the original. A radius of 10 should take less time to compute than a larger one, and profiling of the algorithm shows this step to be the single most time-intensive step within the program. However, the amount of time added by increasing the radius is not linear. The majority of the time cost for this step is up-front, with some additional time needed for increasing the radius. The time to complete processing with double the radius is less than double the original time required to complete this step.

Super-resolution of some environments may also benefit from a larger radius. In areas in which there are dispersed homogeneous deposits, such as certain putative chloride units on Mars, a radius of 10 may not be sufficient to find those units within the map. These units will be clustered together within the tree, and so an initial spectrum can be selected from the tree. However, the strength and weakness of the cluster-tree is that it averages together a large number of pixels; smaller scale spectral variability is lost. This is helpful when trying to determine what an image end-member might be, but is less useful when trying to provide a specific pixel's spectrum. As a result of the loss of small scale spectral variability when sourcing from the tree, a larger correction factor in step 5 would be necessary.

The artificial data was created with only two end-members. As a result, the use of the tree falls off immediately, with the original super-resolved values deriving entirely from the tree by a radius of 5 low-resolution pixels. This was due to the clustering results within the super-resolution algorithm, resulting in 25 to 40 low-resolution clusters per run. Due to the relatively large threshold value, equal to slightly over 1/3 the data range, a significant percentage of the image was marked homogeneous. This test was meant to examine the effect of altering the radius with which to search for adjacent homogeneous pixels in Step 4, but served as a good illustration of the problems associated with choosing too large a threshold value in Step 2.

When the large numbers of "homogeneous" pixels were averaged together into the clusters, they produced cluster centers much closer to the 5 core pixel values. However, as there were so many pixels with different values, a closer fit to the associated high-resolution pixel was always found within the immediate area. As the area searched increased, the fit of this match improved (i.e., mean MD decreases). The same rate of decreasing mean MD was observed in the artificially generated terrain as in the Earth scene. In particular, the maximum MD for a given radius had a trend of being roughly 2x the average MD of 5 pixel smaller radius.

The suggested radial value for super-resolution has been increased from the 10 proposed by Tonooka (2005) to 20 based on these results. This value balances the cost of computational time and the improvements in spectral fits, particularly for natural data. Testing shows that the calls to the tree are necessary, as the mean MD for a purely image-based sourcing is more than 2x that of the radius-10 case within the Earth data. The artificial terrain data shows a similar decrease in MD with an increase in radius.

2.5.5 Step 5 Discussion

Although the equally weighted correction produced the most radiometrically accurate results when the data were convolved back to the original resolution, this method is not recommended. Ignoring the fitness of the data, as measured by the MD of their original assignment, and treating them as all equally good or bad is too simplistic an approach. Allocating the correction factor strictly according to the summed weight fails due to under-correction. Larger spatial scale differences between data-sets will lead to larger under-corrections, with the only redeeming quality of this method being that it will not over-correct. The method first used in Tonooka (2005), based on the equations 6, 7, 13, and 14 of that work, provides the best results by allocating the correction according to the MD of the super-resolved pixels.

The correction allocations of all three methods examined ignore the impact of convolution with the PSF. In most cases, this leads to pixels that are radiometrically accurate when not convolved with their surroundings, but become slightly inaccurate when convolved. The magnitude of the inaccuracy is inversely proportional to the alpha value used for PSF generation. This is due to the correction necessary for any given low-resolution pixel being allocated solely among its sub-pixels, with the correction based upon uncorrected values of neighboring pixels. Methods of radiometrically correcting data that take this into account might improve the ultimate super-resolution results. Similarly, a method such as the summed distance method that consistently under-corrects may be applied multiple times. This may lead to more accurate results without the loss of the improved spatial clarity or risk of over-correction in other pixels.

2.6 CONCLUSION

A super-resolution technique was developed for the Earth-orbiting ASTER instrument, and presented in Tonooka (2005). Although this technique was originally developed for just the ASTER instrument; it was adapted to work with data from the Mars-orbiting THEMIS instrument. In the process of implementing and adapting the algorithm, significant changes were necessary to process THEMIS data. These changes have also been applied to ASTER data, and found to produce equal or better results.

The super-resolution method examined in this work is a 5 step process and the impact of making alterations to each step was examined. During the first step, the impact of choosing an incorrect PSF was examined. More accurate estimates of the PSF produced more radiometrically accurate final products, though results from an inaccurate PSF still have qualitative value. In step 2, the choice of threshold is seen to influence clustering results. A threshold defined on a per-band basis or as a band-average based only on the area of interest produces better results than the original threshold definition. The third step clusters the data; the ISODATA algorithm was found to be adaptable enough to recover from many poor user choices of initial starting conditions. The most critical choices were the starting number of clusters and the number of iterations before forcing a halt. With too few initial clusters, ISODATA may prematurely halt in or return repeatedly to a metastable state. Due to the method of end-member selection, the ISODATA algorithm will undergo an initial reduction in the number of clusters; a low number of permitted iterations will cause the clustering to end during this stage. Performance issues associated with the algorithm were noted in this step, particularly during the ISODATA assignment of homogeneous pixels to new cluster centers. The fourth step is also cpu-intensive. Increasing the

radius searched for good matches improves the final results at a small cost in time. The majority of the time-cost in this step is up-front instead of scaling linearly or exponentially with radius. Different methods of allocating the radiometric correction in the final step were examined; the method proposed in Tonooka (2005) seems to provide the best results and also accounts for radiometric differences at the super-resolved scale.

During the processing of data for super-resolution, some performance trade-offs are necessary. The algorithm can run more rapidly at the cost of greater memory usage, but will hit limits imposed both by the operating system and the programming environment. The current implementation is also limited to 5 or fewer bands for high resolution data, and 10 or fewer bands for low resolution data due to these same memory restrictions. Workarounds for such arbitrary limits should be examined in future work. Similarly, the problem of determining how the radiometric correction should account for the PSF needs to be addressed. The results produced by the current algorithm are good and interpretable, but improvement is possible.

The modified algorithm is an improvement of the original implementation. The algorithm has been extended to run on multiple data types, including both THEMIS data and atmospherically corrected ASTER data. As part of extending the algorithm to other data, a user-defined PSF also has been implemented, using alpha notation. The requirement for intermediate spatial / spectral resolution data, provided by the ASTER SWIR instrument in the original implementation, has been removed after it was determined that it was not required for correct performance. Multiple data have been used to examine the suggested values for user input, and in some cases better default values have been found. These new default values were found to

perform better with both ASTER and THEMIS data. ISODATA has been used in place of K-Means to build the cluster tree in the modified algorithm examined in this work. This replacement requires less *a priori* knowledge by the user, and provides greater flexibility to the knowledgeable user. The modified algorithm is more transparent to the user, and provides data from intermediate steps as well as runtime output. This permits users to determine how their final product was created and aids in tracing the propagation of incorrect input through the algorithm. In all steps, the cpu cost of the functions have been optimized for performance.

The algorithm relies on a number of choices by the end user; these choices determine the quality of the super-resolved data. Testing of the modifications has led to a new set of default recommendations for these values. This default configuration produces good results for both ASTER and THEMIS data, and does so at a reasonable time-cost. Users with no insight into the algorithm can super-resolve ASTER and THEMIS data with good results by making use of the default values. The modified algorithm will accept other data sources for super-resolution as well, permitting this process to be extended to other sources of Earth and Mars data, as well as to other planets. These other data sets need not be limited to the spectral range examined within this work either. Users should ensure they understand the statistical distribution of their data prior to application of the algorithm, although the new default recommendations will be a good starting point for most data.

3.0 SUPER-RESOLUTION OF A PUTATIVE CHLORIDE SITE ON MARS

3.1 INTRODUCTION

Deposits with spectral signatures similar to those of chlorides have been identified on Mars using THEMIS data (Osterloo et al., 2008). These deposits, located within the southern highlands of Mars, have a spectrally distinct signature in the Thermal Infrared (TIR) portion of the spectrum and are present within topographic low points (Figure 3-1). Spectra of these deposits are generally featureless throughout the Visible / Near Infrared (VNIR) and TIR spectral region. This is consistent with anhydrous chloride, but is also consistent with unoxidized sulfides and homopolar compounds (Jensen and Glotch, 2011). The geologic setting may indicate that the deposits formed as evaporites, and therefore makes it improbable that they are sulfides or homopolar compounds (Jensen and Glotch, 2011). However, these materials cannot be eliminated spectrally, and chlorides alone cannot explain the spectral slope seen in the VNIR (Hunt et al., 1972; Jensen and Glotch, 2011) leading to some uncertainty as to the identity of the deposits. If these deposits are chlorides, it may be possible to detect small-scale associated mineral assemblages which may help elucidate their formation process and confirm their composition. These exposures can be too small for detection at the native scale of the THEMIS TIR instrument (100 m / pixel); however super-resolution of these pixels may increase their

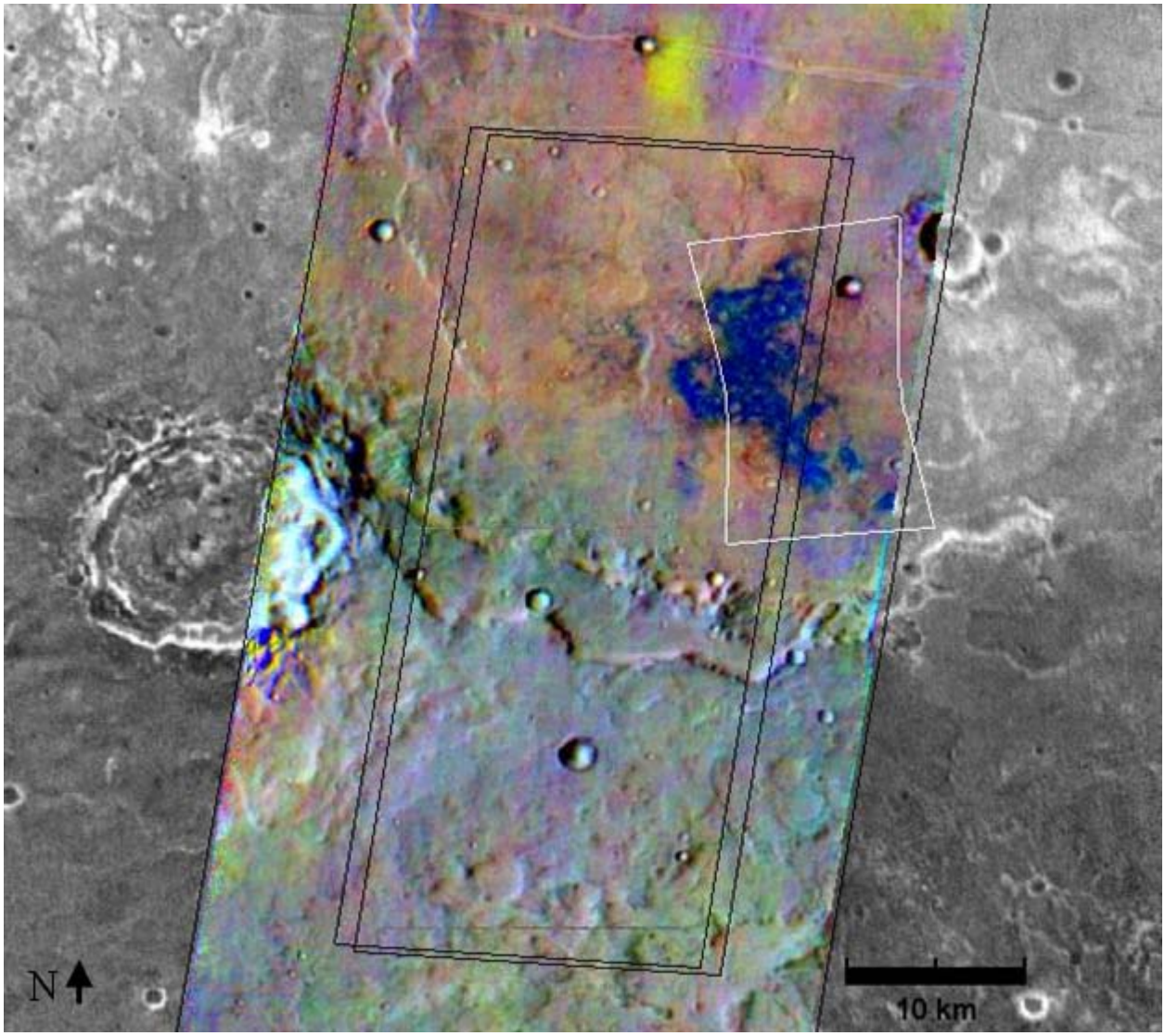

Figure 3-1 The location of the putative chloride units examined in this work. The location of the THEMIS VIS scenes are shown as black rectangles. The location of Figure 3-2 is shown as a white keystone shape. The color strip through the figure center is a THEMIS Decorrelation Image (DCS) with THEMIS TIR bands 8 / 7 / 5. In this DCS combination, the blue unit within the black rectangles and white keystone is the putative chloride unit. The underlying grayscale data is from the THEMIS nighttime 100 m global mosaic. The center of the image is near 180.5° E, -27.25° N.

detectability. This increased detectability originates by increasing the percentage of the pixel occupied by small-scale units, raising them above the detectability threshold. Examination of these regions using other instruments, such as High Resolution Imaging Science Experiment (HiRISE) and the Compact Reconnaissance Imaging Spectrometer for Mars (CRISM) will also provide insight into the morphology and mineralogical makeup of these exposures. Greater understanding of these exposures may help us improve our understanding of the Martian paleoclimate. In addition, the deposits could be considered a high priority science target for future Mars missions. Not only could insights here have planetary-scale ramifications, but fluid inclusions within salts on Earth have been shown to preserve microbial life (Schubert et al., 2009).

3.1.1 Evaporites on Earth

Evaporites are precipitated from a saturated solution or brine, at the surface or nearsurface, through hydrologic processes driven by solar evaporation (Warren, 2006). Evaporite salts can accumulate as capillary efflorescence on arid surfaces, on the floor of briny lakes, or in nearsurface pore space (Warren, 2010). Anhydrite and halite are the two dominant evaporite minerals on Earth (Warren, 2010). Evaporites on Earth have formed at a variety of scales, from ancient marine-fed basins covering more than 1,000,000 km^2 to mineral crystallization onto individual sediment grains (Osterloo et al., 2008; Warren, 2006; Warren, 2010). Evaporites can also form in colder climates, where the dilute water is frozen, leading to a concentration of the ionic content within the remaining liquid (Marion et al., 1999; Warren, 2010) or through heating

of hydrothermal brines in volcanically active regions (Warren, 2010). The most important factor in the resultant mineral assemblages is the composition of the dilute water at the start of evaporation (Osterloo et al., 2008; Tosca and McLennan, 2006).

3.1.2 Putative Chlorides on Mars

Widespread deposits of putative chlorides have been detected throughout the southern highlands of Mars (Glotch et al., 2010; Osterloo et al., 2008; Wray et al., 2009). These deposits have been found at a variety of elevations, but generally within local topographic lows (Glotch et al., 2010; Osterloo et al., 2008). Deposits have been identified within sinuous channels and in small craters (Osterloo et al., 2008), and have formed inverted channel-fill (Glotch et al., 2010). Chlorides occur stratigraphically above phyllosilicate units, and are seen to embay phyllosilicate bearing hills within Terra Sirenum (Glotch et al., 2010; Wray et al., 2009).

The spectra of the putative chloride deposits are generally featureless, exhibiting a negative slope toward longer TIR wavelengths. Such a slope can result during the conversion of radiance data to emissivity from incorrectly derived surface temperatures, due to thermal heterogeneity on the sub-pixel scale (Gillespie et al., 1998; Ramsey and Dehn, 2004; Rose and Ramsey, 2009; Ruff et al., 1997), or due to an incorrect assumption of unit or near-unit emissivity during the separation of temperature and emissivity (Ruff et al., 1997). The most likely explanation for anisothermality on Mars is uneven warming due to sub-pixel scale surface roughness. Where a pixel is anisothermal due to a high degree of sub-pixel scale surface roughness, this negative slope does not continue to longer wavelengths, and the derived emissivity displays variability associated with the solar incidence angle at the time of collection

(Bandfield, 2009). Higher resolution imaging data over the putative chloride sites do not show areas with a high degree of surface roughness (Bandfield, 2009; Osterloo et al., 2008) or variability associated with solar incidence angle. If a pixel's spectrum has a negative slope due to generally featureless spectra and non-unit emissivity (graybody) effects, this negative slope will continue to longer wavelengths, and the derived emissivity is largely independent of the solar incidence angle at the time of collection (Bandfield, 2009). Based on these observations, pixels on Mars with negatively spectra contain a material whose spectrum is a graybody in the TIR (Bandfield, 2009; Glotch et al., 2010; Osterloo et al., 2008). Geologic materials with graybody effects over the spectral range covered by THEMIS are rare; however, chlorides do fit this description.

Within the Martian system, chlorides could form from the evaporation of acidic fluids derived from the weathering of basaltic rocks (Hurowitz et al., 2010; Milliken et al., 2009; Osterloo et al., 2008; Tosca and McLennan, 2006). Evaporite minerals such as halite and gypsum can form in acid environments, and have been identified on Mars from satellite data and in Martian meteorites (Benison and LaClair, 2003; Bridges and Grady, 2000; Fishbaugh et al., 2007; Gooding et al., 1991; Osterloo et al., 2008; Wentworth et al., 2005). Hydrated minerals formed in acidic environments, such as sulfates and kaolinite, have been detected by the CRISM and OMEGA instruments (Bibring et al., 2005; Bishop et al., 2008; Ehlmann et al., 2008; Ehlmann et al., 2009; Gendrin et al., 2005; Langevin et al., 2005; Mustard et al., 2008; Poulet et al., 2005). Precipitates consistent with an acidic environment, such as alunite, hematite, jarosite,

and Fe- or Mg-rich smectites (Milliken and Bish, 2010; Murchie et al., 2009; Weitz et al., 2010; Wray et al., 2009) also have been detected. These factors all support a widespread acidic depositional environment.

One possible acidic depositional environment on Mars is similar to Earth's acid lakes (Benison and LaClair, 2003; Osterloo et al., 2008; Tosca and McLennan, 2006). The acid saline lake systems of Western Australia, with their variety of pH environments, have been used as analogs to the deposits found thus far on Mars (Baldridge et al., 2009). Such a system would strengthen the case for a hydrologic history of Mars. If an analogous acidic environment existed on Mars, it would accommodate both the deposition of both hydrated sulfate minerals and phyllosilicates (Baldridge et al., 2009) and the need for an alkaline environment for the deposition of many of the detected phyllosilicates (Bibring et al., 2005). In such a system, chloride deposition could occur both at the surface from the evaporation of water or as a cement in subsurface sedimentary deposits. Were this to occur, these deposits should have a higher thermal inertia (TI) than the surrounding sedimentary beds. Pixels with significantly higher TI have been previously identified as collocated with chlorides deposits (Osterloo et al., 2008), as have chloride-bearing inverted channel-fill (Glotch et al., 2010).

3.1.3 Approach

Super-resolution is the process of creating a higher spatial resolution than that of the original (or native) data source. Higher-resolution data permit a finer examination of the deposits in the TIR than is possible with existing data. This increase in resolution will bring the spatial resolution of the THEMIS TIR instrument to the same general scale as a number of VNIR instruments now in

orbit around Mars, and improve the comparison between instruments. By making use of a super-resolution algorithm, the area encompassed by any one pixel will be decreased, leading to better detectability of sub-pixel scale outcrops. These smaller pixels can be examined in greater detail for both their sub-pixel composition and for their distribution. Composition may give insight into the formation processes at work for the main deposit, and indicate whether or not it is possible for the deposit to have formed as an evaporite. The combination of a better comparison with other existing instruments and a finer-scale composition map will give an independent way to test the hypothesis that these deposits are chlorides.

3.2 METHODOLOGY

3.2.1 Data Collection

A number of putative chloride sites have been previously identified by Osterloo (2008). The largest of these, in terms of surface area exposure, were identified, and four of these were selected as regions of interest (ROI) for further targeting for this research by the THEMIS instrument (Christensen et al., 2004). The best site was then chosen for further study (Figure 3-1). Data collected over these targeted areas were multi-spectral in both the visible and TIR wavelength regions. THEMIS uses the same telescope for the VIS and IR instruments, with 5 VIS channels between 0.40 and 0.88 μm at 18 m/pixel spatial resolution and 9 IR channels at 100 m/pixel spatial resolution between 6.27 and 12.98 μm (the first two of which are duplicated for better signal to noise). An additional tenth IR channel at the same spatial resolution is centered at 14.88 μm and is used for detection of atmospheric temperatures (Christensen et al., 2004).

The VIS instrument contains a 1024 x 1024 array of silicon elements with a 2.66° x 2.44° field of view (FOV), and a 5 stripe filter, with all elements exposed simultaneously (McConnochie et al., 2006). Groundtrack motion is also used to expand spatial coverage (McConnochie et al., 2006), making it both a framing and pushbroom system. For the visible data, collection of bands 2 through 5 was requested. Pixel summing (aggregation of pixel data) at 36 m / pixel is commonly used to acquire VIS images over larger areas (Christensen et al., 2004). Summing was set to a maximum of two for the visible data, resulting in pixels of either roughly 18 (no summing) or 36 (2x2 summing) square meters. Preference was expressed for the 36 m / pixel data geometry, as it provided a greater areal coverage, and thus better mapping for the super-resolution process.

The IR instrument has a 320 x 240 array of uncooled microbolometer elements with a 4.6° (cross-track) x 3.5° FOV (Christensen et al., 2004). The IR instrument has ten stripe filters, representing nine separate wavelength ranges (bands 1–10) with an image being formed in a pushbroom fashion (Christensen et al., 2004). TIR data used the DAY IR (all band) specification for data collection. Bands 1, 2, and 10 were not used for super-resolution analysis.

3.2.2 Site Location

The site examined in this work (Figure 3-1) is located along the southern portion of a crater centered near 180.5° E, -27.0° N, in northwestern Terra Sirenum. The chloride deposit lies near 180.46° E, -27.25° N and covers an area roughly 15 km x 10 km. This area is well covered by THEMIS data, including two separate contemporaneous image pairs collected in August and

December of 2009. In addition, there are HiRISE and CRISM data acquired during October of 2007. The CRISM data (Figure 3-2) covers the majority of the deposit, missing only a small zone of the deposit extending to the west. The HiRISE data (Figure 3-3) covers the eastern margin of the deposit, with the color bands passing just along the easternmost margin. These data show the interfingering between the light-toned putative chloride deposits and the darker toned surrounding area.

3.2.3 Super-Resolution

Prior to super-resolution, data were pre-processed as described in Hughes and Ramsey (2010). This included degrading the resolution of the IR data to 108 m/pixel and correction of the spatial alignment between the two datasets prior to super-resolution. This resolution degradation works best when the lower resolution pixel can be completely co-located with an integer number of higher resolution pixels (18 or 36 m). Spatial alignment is necessary as there is a significant difference in the registered location of features (i.e., craters) of several lower resolution pixels between the two sets. Previous application of the super-resolution approach (Tonooka, 2005) discovered that an alignment within 0.2 of the larger pixel size was sufficient for valid results. Given the relative resolution of the VIS and IR data, this algorithm is limited to an alignment within 0.33 of the larger pixel. As the resampling of the IR data to 108 m / pixel occurs before manual alignment, and the alignment is performed using the smallest resolvable surface features within this resampled data, it is assumed that this resampling does not introduce further misalignment beyond the one VIS pixel limit. A significant misalignment between the datasets would result in a failure of the super-resolution algorithm. Super-resolution is limited to the spatial region covered by both data.

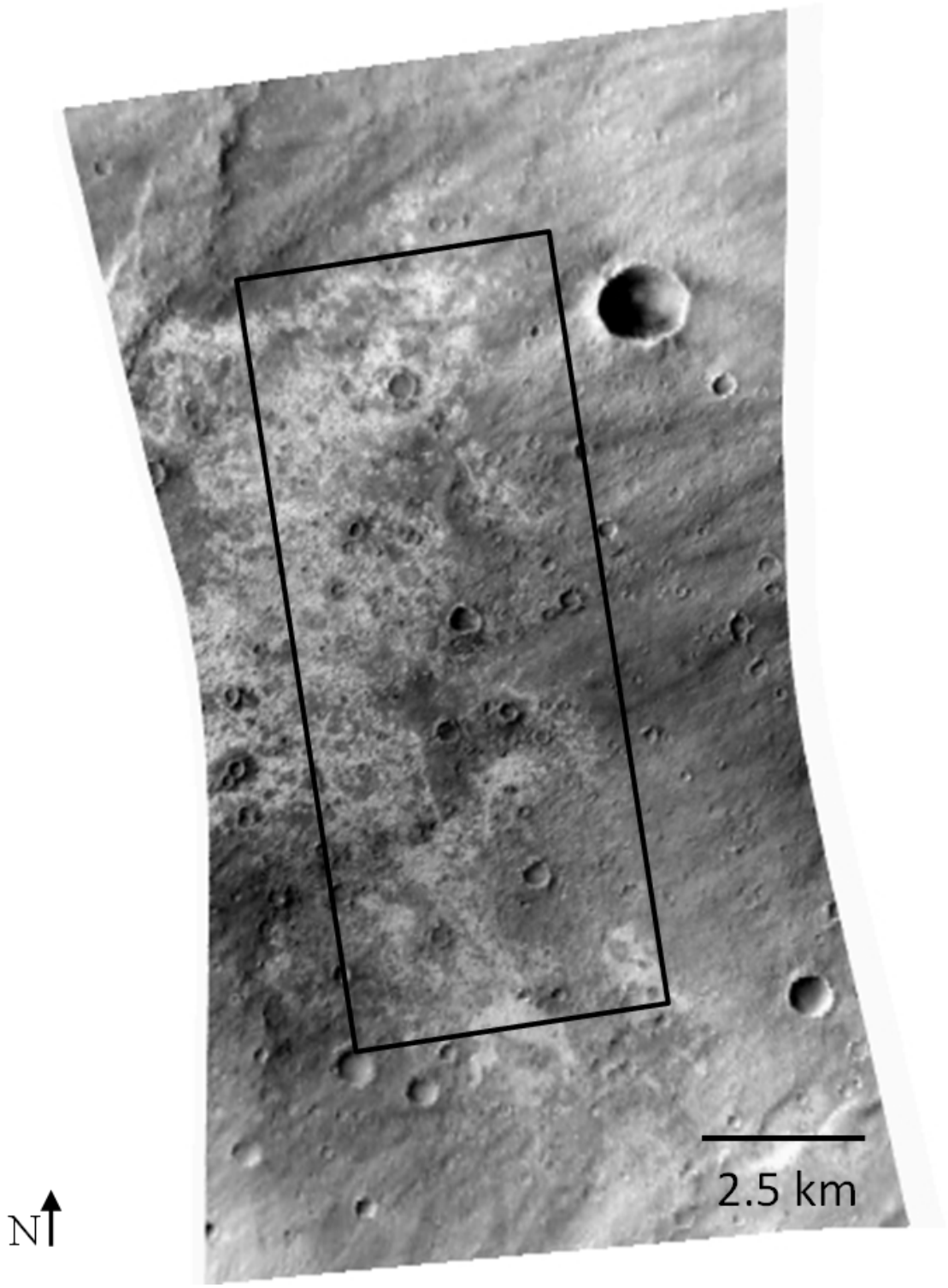

Figure 3-2 CRISM data HRL000082DA_07_IF181L_TRR2, showing IR surface brightness at 1.3 μm. The CRISM data covers the majority of the chloride unit, seen here along the eastern edge to center of the image as the diffuse lighter-toned area. This is collocated with the blue unit seen in Figure 3-1. The black rectangle shows the location of the HiRISE data in Figure 3-3.

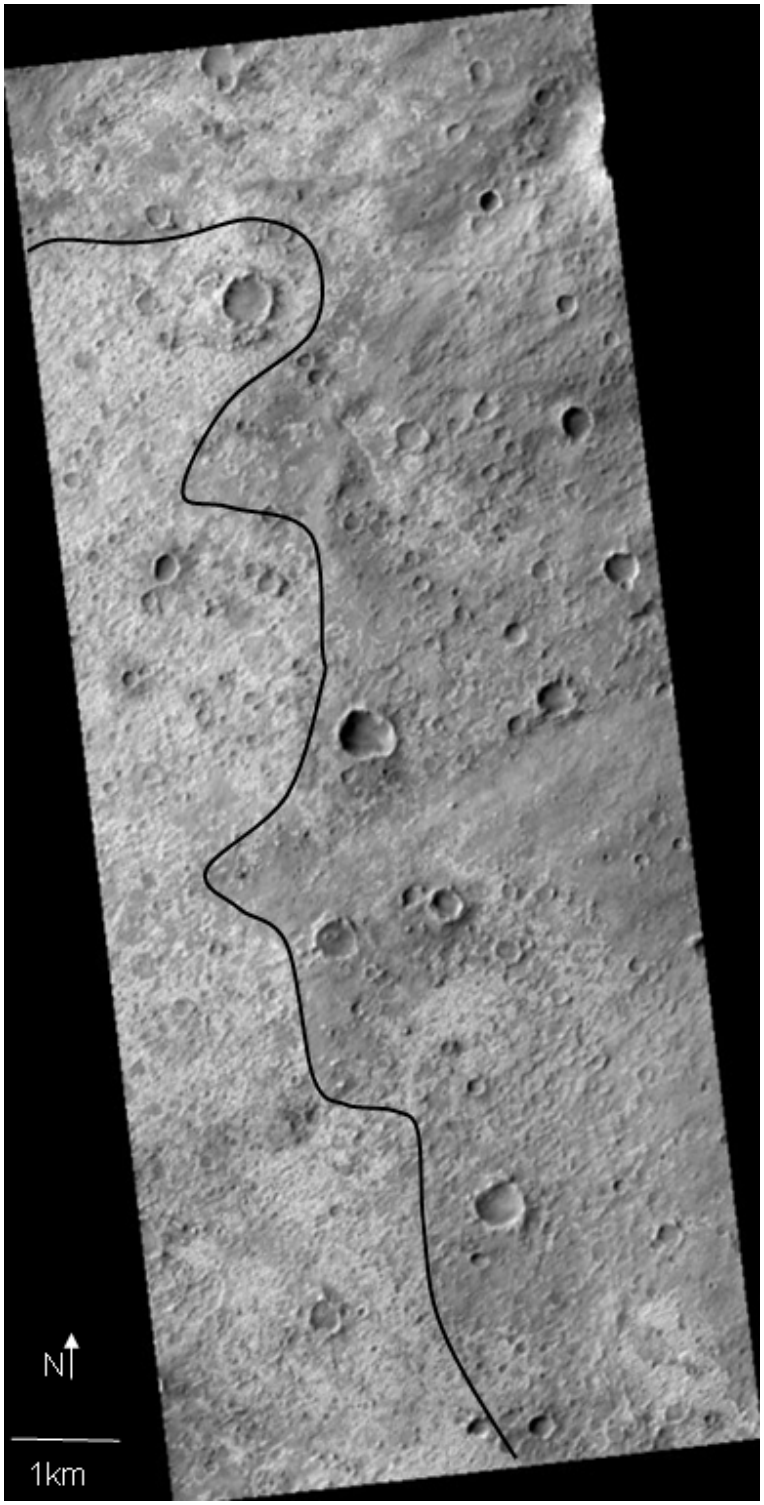

Figure 3-3 HiRISE data PSP_005680_1525_RED. The HiRISE data cover the eastern margin of the chloride deposit. The putative chlorides can be seen as a lighter-toned unit to the west of the added boundary and the surrounding darker-toned material is present to the east.

3.2.4 Linear Deconvolution

TIR spectra can be assembled through the linear addition of spectra of sub-pixel materials relative to their areal abundance (Ramsey and Christensen, 1998; Thomson and Salisbury, 1993). This means that spectra can be forward-modeled by adding sub-pixel component spectra together in the proper proportions. A similar process using a library of end-members can be used to linearly deconvolve a TIR spectrum into its subcomponent spectra and relative abundances. A number of previous studies have made use of this to derive sub-pixel abundance based on TIR data from Mars or Mars analogue sites (e.g., Baldridge et al., 2004; Bandfield, 2008; Bandfield et al., 2004; Christensen et al., 2005; Glotch and Rogers, 2007; Michalski et al., 2004; Osterloo et al., 2008; Stockstill et al., 2007). This process is limited by a number of factors. First, the results will be largely dependent upon the quality and breadth of the spectral library used for deconvolution. Second, in order to remain mathematically valid, the number of possible end-members within a pixel must be equal to or less than the number of bands used to create the spectrum (Adams et al., 1986; Ramsey and Christensen, 1998). This is not a limiting factor in laboratory derived spectra with hundreds of separate bands, but it does limit end-member spectra from THEMIS data to 8 or less end-members.

For the purpose of this work, a maximum of six end-members were permitted after eliminating four bands from use in linear deconvolution. Bands 1 and 2, even averaged together, displayed a significantly greater degree of variability across the data than other bands, reflecting the limited IR signal at these wavelengths over common Mars surface temperatures. Band 3 is typically used for temperature / emissivity separation, and as such shows only minimal

variability across the scene. An emissivity value is typically assumed and assigned to this band for the purpose of pixel temperature determination. Band 10 is generally used for atmospheric analysis, and the amount of surface signal is negligible. A blackbody was also used as an end-member to account for variable spectral contrast between actual and laboratory end-members, further limiting the number of possible informational end-members, but is normalized out where calculating mineral end-member values for each pixel (e.g., Ramsey and Fink, 1999).

The spectral libraries used included those available from the ASU spectral library (Christensen et al., 2000) as well as common martian surface end-members such as Surface Types 1 and 2 (ST 1 and ST 2) (Bandfield et al., 2000). In addition, a spectrum based upon the spectrally distinct unit (the putative chloride) documented in Figure 3A of Osterloo et al (2008) was added. Various combinations of end-members were applied, with the final reported result being optimized for appearance, RMS error values and spatial distribution, and a frugality of end-members. A solution with fewer end-members would be preferred over one with more, even at the expense of slightly greater average RMS error for consistency with the relatively homogeneous nature of the martian surface.

During the deconvolution process, a number of errors can be introduced. These errors can include over-fitting the data, problems with pixel spectra whose emissivity is less than that of any end-member, or whose emissivity already exceeds 1 in the starting data. As a result, masks were defined based on the output of the deconvolution algorithm. Pixels whose blackbody contribution exceeded 100% were masked out before statistics were calculated; if spurious results still existed, pixels with emissivity values that were too high (significantly over 1.0) or

too low (under 0.8) were masked out as well. These pixels were examined separately from the valid results to identify the associated error. In addition, the super-resolved data were created using entire VIS scenes, including null-value pixels used for image rectification. The pre-existing mask from this data was always applied before examining the super-resolved data, as the results from the super-resolution process are invalid over these null-value pixels and are reset to 0 before being output.

3.3 RESULTS

3.3.1 Super-Resolution Results

Two THEMIS TIR datasets were super-resolved. THEMIS TIR scene I35462005 was super-resolved using THEMIS VIS scene V35462006. The resulting super-resolved TIR data were 726 x 1350 pixels. During application of the super-resolution algorithm, data were clustered into 26 groups based on the VIS data, and then further subdivided into 265 further clusters based on TIR data. The super-resolved data (Figure 3-4) shows slightly greater data diversity than the original TIR data (Table 3-1). The band averages are roughly the same; however, the standard deviation, per band, is greater in the super-resolved data. Similarly, THEMIS TIR scene I33902002 was super-resolved with data from THEMIS VIS scene V33902003. The resulting super-resolved scene (Figure 3-5) also showed slightly greater data diversity than the original data (Table 3-1). During the super-resolution process, data from this scene-pair were clustered into 10 groups based on the VIS data, and then further subdivided into 34 clusters based on the TIR data.

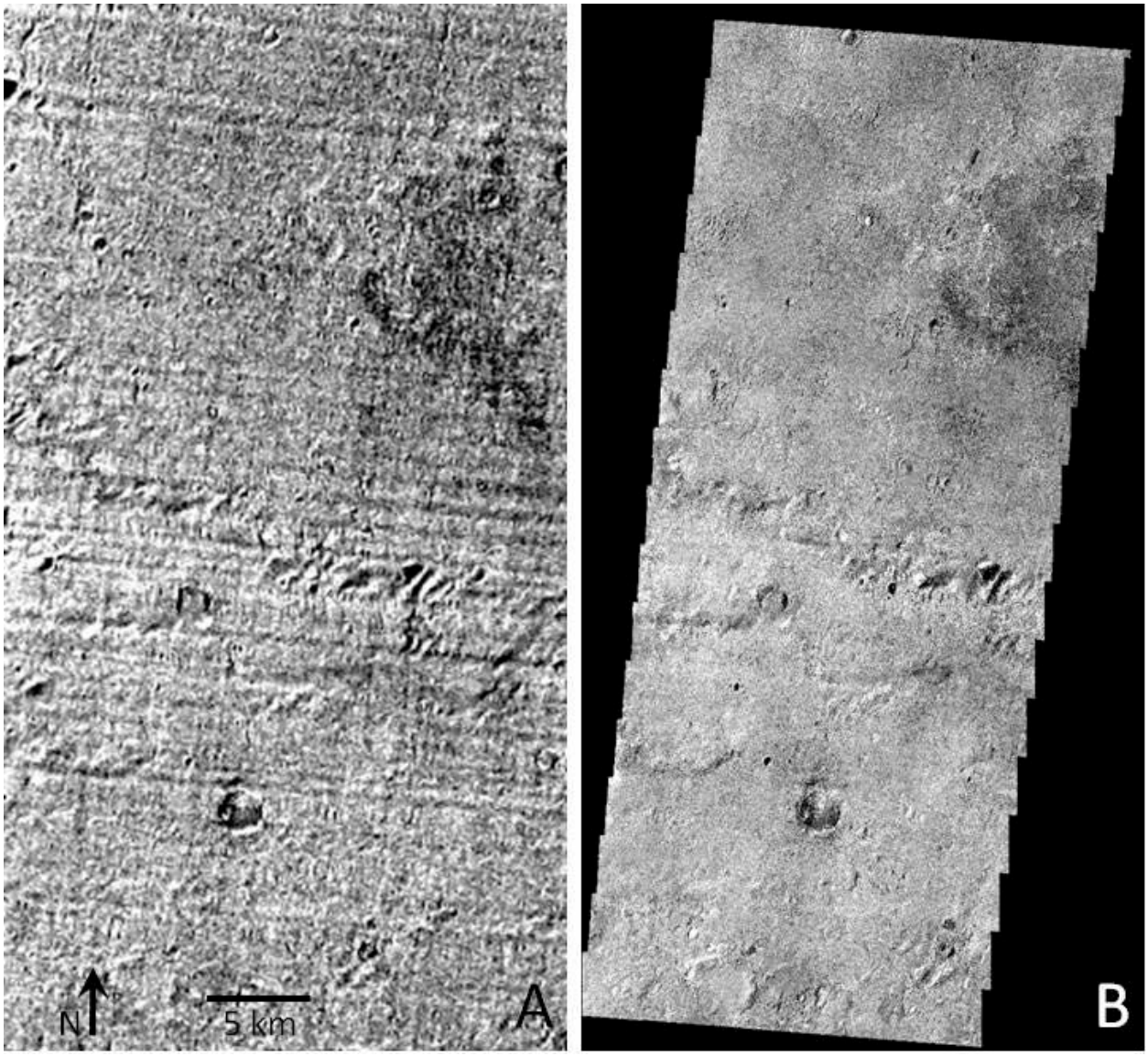

Figure 3-4 THEMIS TIR I35462005 emissivity band 9. The chloride unit is shown as a darker (lower emissivity) irregularly shaped polygon in the northeast portion of the image. The darker color of the unit at longer wavelengths reflects the negative slope of these units in the TIR. The rough terrain crossing the figures in a roughly east-west orientation is the crater rim. (A) Original resolution data. (B) Super-resolved data.

Table 3-1 Statistical comparison of the original and super-resolved data for the two THEMIS scenes. The super-resolved data shows the same mean (within 0.0001 emissivity), but slightly greater data diversity.

Site 1: I33902002								
	Original				Super-Resolved			
	Min	Max	Mean	Stdev	Min	Max	Mean	Stdev
Band 3	0.9764	0.9764	0.9764	0.0000	0.9764	0.9765	0.9764	0.0000
Band 4	0.9304	1.0157	0.9748	0.0047	0.9300	1.0138	0.9748	0.0047
Band 5	0.9323	1.0104	0.9742	0.0056	0.9319	1.0095	0.9742	0.0056
Band 6	0.9439	1.0141	0.9824	0.0054	0.9436	1.0135	0.9824	0.0055
Band 7	0.9273	1.0094	0.9788	0.0047	0.9295	1.0043	0.9789	0.0047
Band 8	0.9387	1.0001	0.9773	0.0042	0.9383	1.0015	0.9773	0.0042
Band 9	0.9331	0.9977	0.9744	0.0052	0.9366	0.9980	0.9744	0.0052

Site 1:35462002								
	Original				Super-Resolved			
	Min	Max	Mean	Stdev	Min	Max	Mean	Stdev
Band 3	0.9174	0.9765	0.9765	0.0005	0.9551	0.9776	0.9765	0.0001
Band 4	0.9064	1.0379	0.9749	0.0091	0.9034	1.0714	0.9749	0.0092
Band 5	0.9077	1.0820	0.9740	0.0095	0.9164	1.0592	0.9740	0.0095
Band 6	0.9089	1.0612	0.9832	0.0095	0.9162	1.0834	0.9832	0.0096
Band 7	0.8806	1.0672	0.9800	0.0092	0.9069	1.0552	0.9799	0.0094
Band 8	0.8855	1.0275	0.9786	0.0089	0.8793	1.0410	0.9785	0.0090
Band 9	0.8740	1.0333	0.9766	0.0091	0.9087	1.0416	0.9765	0.0092

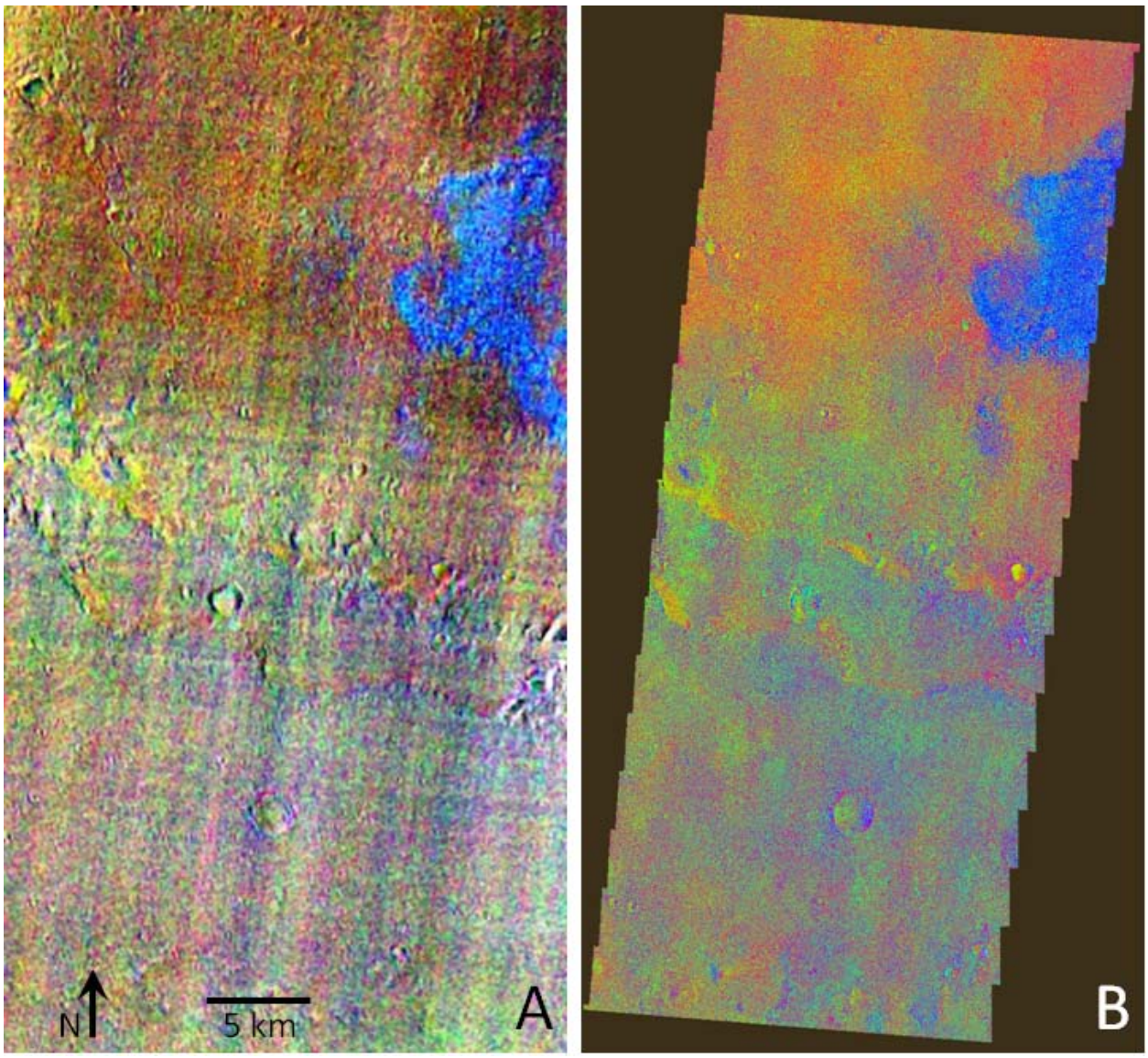

Figure 3-5 DCS of THEMIS TIR data I33902002 bands 8 / 7 / 5 in RGB over Site 1. The chloride unit is shown as the blue irregular polygon in the northeast. (A) The original resolution data show some striping still present within the data. (B) The super-resolved data define features with greater clarity and mute the striping.

3.3.2 Linear Deconvolution Results

After the exclusion of pixels specified below, statistics from the model spectra of both the base and super-resolved data can be found in Table 3-2. The differences between the mean values of the original data and the model data are consistent between the two different resolutions. The model data shows a slightly lower standard deviation than the starting data; this reflects that the spectra are being built from a handful of end-members and a blackbody, whereas the natural surface is composed of significantly more end-members. Table 3-3 shows the contribution of each end-member to building the model spectra.

Four end-members plus a blackbody were found to produce the optimal deconvolution from the library derived end-members for the original resolution data. Spectra used were ST2 (Bandfield et al., 2000), two different labradorite spectra from the ASU spectral library (Christensen et al., 2000), and the chloride unit from Osterloo et al (2008). Labradorite was included in the possible end-member list as Glotch et al (2010) reported that the CRISM spectra of the putative chloride deposits in Terra Sirenum were consistent with a ratio of a mixture halite and labradorite to pure labradorite between 1.0 and 2.6 μm. However, it was not possible to replicate these results using either labradorite end-member alone or an average of the two. Using either the average spectrum or either labradorite on its own as an end-member resulted in a mean contribution of under 2%, versus the nearly 9% between the two separate labradorite samples. This may be a good example of data being over-fitted. Deconvolution was re-performed as a result with no labradorite, as it was a minor contributor if only a single spectrum of it was used.

Table 3-2 Statistics of the original and model spectra.

Starting Data								
	Original Res				Super Res			
	Min	Max	Mean	Stdev	Min	Max	Mean	Stdev
Band 4	0.9304	1.0157	0.9751	0.0048	0.9300	1.0138	0.9748	0.0047
Band 5	0.9306	1.0195	0.9746	0.0059	0.9319	1.0095	0.9742	0.0056
Band 6	0.9439	1.0184	0.9827	0.0058	0.9436	1.0135	0.9824	0.0055
Band 7	0.9273	1.0147	0.9792	0.0050	0.9295	1.0043	0.9789	0.0047
Band 8	0.9336	1.0195	0.9776	0.0046	0.9383	1.0015	0.9773	0.0042
Band 9	0.9331	1.0136	0.9746	0.0054	0.9366	0.9980	0.9744	0.0052

Model From Library Spectra								
	Original Res				Super Res			
	Min	Max	Mean	Stdev	Min	Max	Mean	Stdev
Band 4	0.9476	1.0000	0.9842	0.0037	0.9473	1.0000	0.9840	0.0036
Band 5	0.9305	0.9995	0.9747	0.0057	0.9302	0.9973	0.9744	0.0054
Band 6	0.9340	0.9994	0.9758	0.0051	0.9337	0.9968	0.9755	0.0048
Band 7	0.9398	0.9993	0.9773	0.0042	0.9396	0.9961	0.9770	0.0040
Band 8	0.9450	0.9991	0.9786	0.0040	0.9448	0.9957	0.9784	0.0037
Band 9	0.9407	0.9999	0.9768	0.0049	0.9450	0.9963	0.9765	0.0046

Table 3-3 End-member contributions to the model spectra.

Model From Library Spectra								
	Original Res				Super Res			
	Min	Max	Mean	Stdev	Min	Max	Mean	Stdev
ST 2	0.00	100.00	47.27	16.88	0.00	100.00	47.70	16.39
Chloride	0.00	100.00	52.73	16.88	0.00	100.00	52.30	16.39
Blackbody	0.00	98.57	39.48	13.36	0.00	92.85	38.72	12.41
RMS Error	0.0012	0.0198	0.0052	0.0010	0.0011	0.0157	0.0052	0.0010

Some pixels were found with incorrect model emissivity values. A total of 207 original resolution pixels (0.19%) were excluded due to model emissivity values exceeding 1.05. A total of 61 pixels (0.06%) were excluded in which modeling failed, resulting in an emissivity of 0.0 in all bands. The majority of the excluded pixels were found along crater rims, particularly in the area south of the chloride deposit along the main crater wall and with the putative chloride deposit. Within the super-resolved data, a total of 1,026 (0.15%) pixels were excluded. Ninety-one (0.01%) were found in which modeling failed, resulting in an emissivity of 0.0 in all bands. The remaining 935 (0.14%) pixels had emissivity values that exceed 1.05 in at least one band. These excluded pixels tended (1010 of the 1026) to coincide with the putative chloride deposit. All had their model spectrum derive exclusively from chloride, to which a small component of blackbody contribution was added, leading to excessive emissivity values.

3.4 DISCUSSION

3.4.1 Super-Resolved TIR Emissivity

The super-resolved emissivity data (Figure 3-5) derived from I33902002 show a greater spatial clarity than the original data. The super-resolved data, when subjected to a decorrelation stretch, shows the same vivid blue in the 8 / 7 / 5 decorrelation stretch that is seen in the original resolution data. The deposit itself appears to be less homogenously covered with the putative chlorides. Individual pixels show a greater degree of variability, perhaps reflecting greater and lesser degrees of erosion or the irregular distribution of sub-pixel scaled craters and their ejecta, as either would tend to expose topographically lower layers to view.

Super-resolved data does a better job of showing the composition of the interface between the putative chloride deposit and the surrounding material. The irregular nature of the edge of the deposit is more apparent in the super-resolved data, with regions of non-chlorides visibly indenting the feature along both the southern and northern margin. These regions also can be seen in other data sets, such as CRISM data HRL000082DA (Figure 3-2) and, to a smaller spatial extent, HiRISE data PSP005680_1525_RED (Figure 3-3). The DCS of the super-resolved data (Figure 3-5) shows that the main chloride unit, in blue, is surrounded first by a rim of purple pixels and then a larger extent of red pixels. Smaller scatterings of each are present within the main deposit as well. The red units in Figure 3-5 are associated with more weathered ST-2 surrounding material. The purple rim seems to be closely associated with the interface between the weathered surrounding material and the putative chlorides. If this is so, this region may represent the buried margin of the chloride unit, and show the extent of excavation. The smaller exposure of putative chloride to the southwest shows bands of this purple, less than 100 m wide, cutting across the unit. These may represent regions of heavily weathered ST-2 that were slightly more resistant than the completely eroded surrounding material. Within these regions, there are small scale (sub-pixel in the super-resolved data) exposures of the chloride. This can be seen both in the purple color of the DCS data as well as the presence of blue pixels contained within the purple bands. These blue pixels are larger windows through the ST-2, showing the underlying deposit. No patterns are found within the THEMIS data, at the original or super-resolved spatial resolution, which correspond with the presence of phyllosilicates in this region. Glotch et al (2010) report that the chloride unit is stratigraphically above the phyllosilicates, and do not report any exposures in this region. The weathering here may not be sufficient to expose any underlying phyllosilicates in detectable amounts.

3.4.2 Linear Deconvolution

The blackbody contribution (Figure 3-6) shows a strong correlation with the striping in the original data, as well as with crater walls and rims in both the original and super-resolved data. The super-resolved data also shows this correlation between striping in the data and the blackbody end-member, though the striping is greatly muted in the super-resolved data. The large crater wall and the smaller crater rims are rough topography. These areas should contain numerous small-scale surface irregularities that act as shadowed regions (effectively blackbodies) to surface emissivity. Energy emitted from these surfaces is reflected within this terrain and never reaches the satellite. The location of the putative chloride unit shows an anti-correlation with the blackbody end-member in both resolutions; these spectra only required minimal contributions from the blackbody for modeling. A bright region in Figure 3-6 can be seen immediately north of the putative chloride, extending approximately 8,000 square meters. This region is shown as shadowed in other THEMIS data, and may represent a sharp slope and shadowed region not visible to the satellite as a result of the topography.

The RMS error maps (Figure 3-7) show a mild positive correlation with the main putative chloride unit, and an apparently weaker correlation for the super-resolved data. Super-resolved data have both a lower maximum RMS error value and fewer pixels with higher values. This skewing of the super-resolved data to lower RMS error values causes the deposit to not stand out as clearly from the surroundings. Both resolutions show noisy data, with the original resolution data having better correlation to crater rims and ejecta. Neither resolution has consistently higher RMS error values associated with the smaller exposure to the southwest of the main deposit.

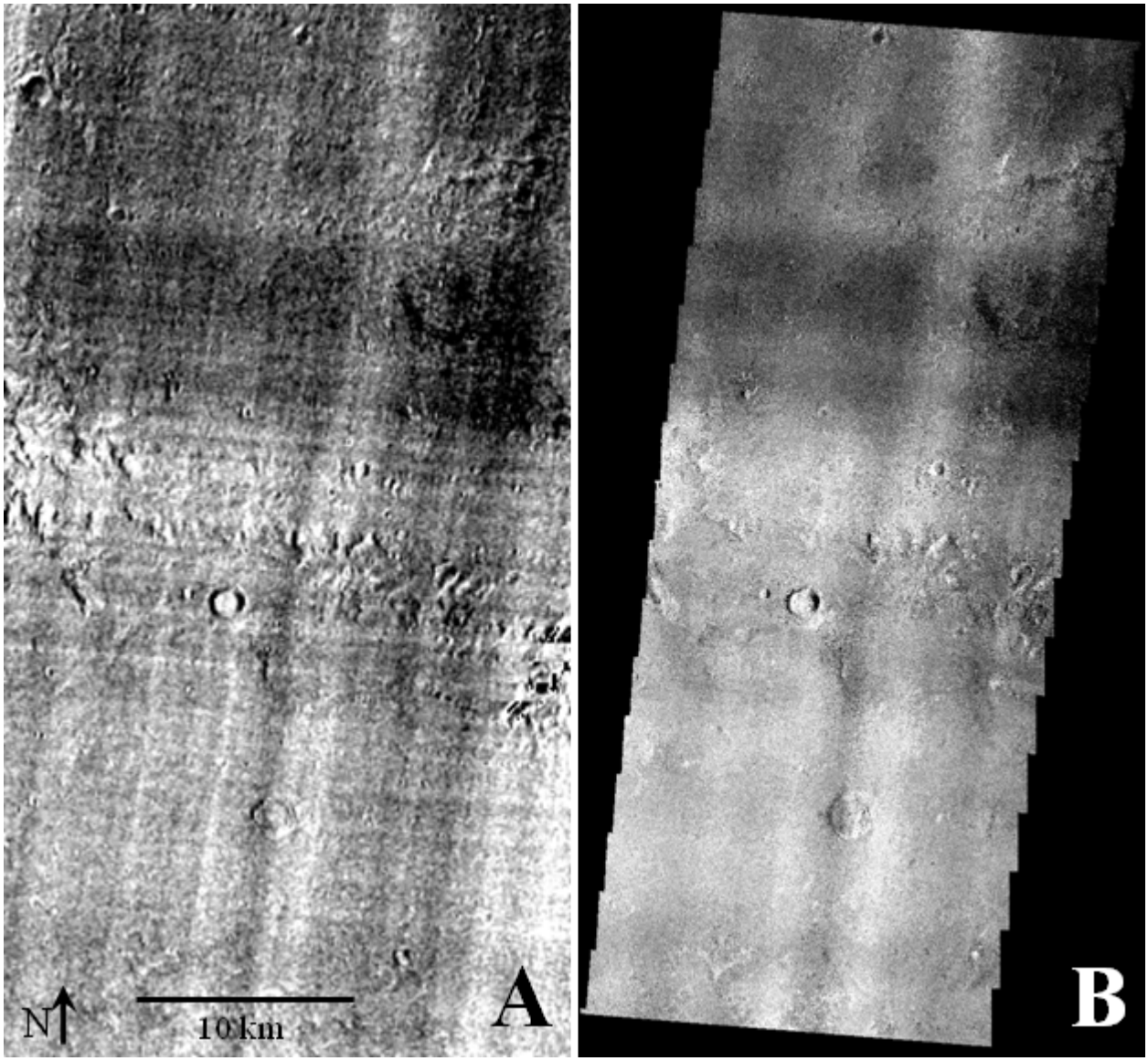

Figure 3-6 Blackbody end-member contributions to linear unmixing. (A) Original resolution data. (B) Super-resolved data.

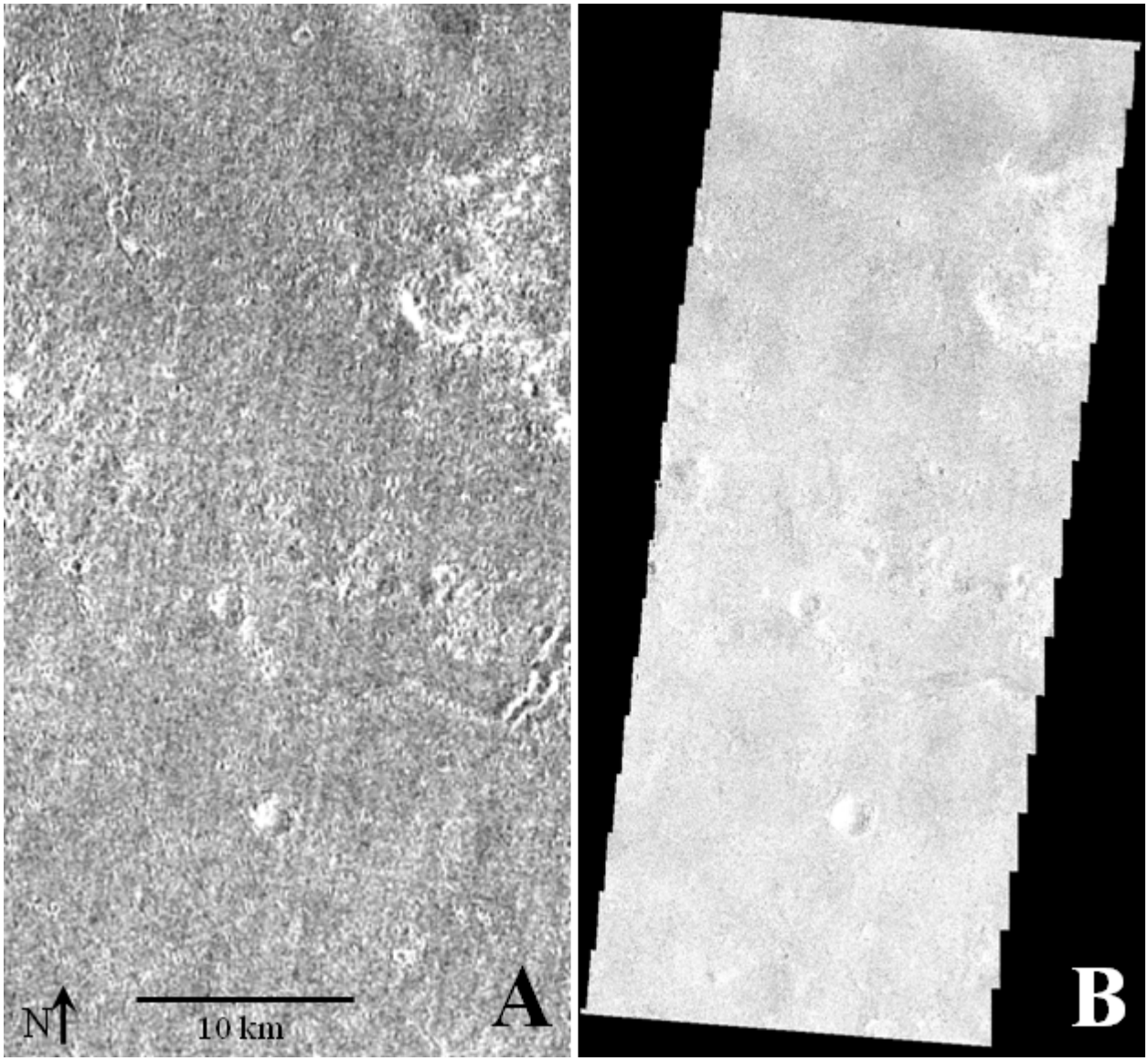

Figure 3-7 RMS error map for deconvolution. Brighter pixels denote higher error values, indicating a poorer fit of the model spectrum to the original spectrum for that pixel. (A) Original resolution data. (B) Super-resolved data.

End-member distributions (Figure 3-8) do a good job of isolating and highlighting the chloride units. The smaller exposure to the southwest is connected to the main deposit with a dispersed chain of blue pixels, representing exposures of chloride greater than 1200 m^2 in extent. The area between the exposures, and continuing to the south of the secondary exposure, is purple in color. This area shows contribution of both chloride and ST-2, and is larger in extent than the similar area seen in the DCS image (Figure 3-5). Within this region, ST-2 and chloride are mixed at the sub-pixel scale, with rare exposures of larger chlorides (blue) or ST-2 (red). In both the original and super-resolved data, the putative chloride unit is modeled as being purely chloride, with no contribution from the ST-2 end-member. The super-resolved data shows the western margin of the main unit to have a more dispersed chloride signature. In the original resolution data, a region of ST-2 is seen covering parts of this margin; in the super-resolved data, this area contains a mixture of red (ST-2), blue (chloride) and purple (both) pixels instead.

The smaller exposure to the southwest and the northwestern portion of the main deposit are shown in a lighter blue tone. These regions required less blackbody contributions than the main chloride unit during spectral deconvolution. This may indicate that these regions are less weathered. This would be consistent with the interpretation of the compositional striping over the southwestern exposure in the DCS image. If these materials are relatively fresh, they would have a smoother upper surface, and require less blackbody. However, super-resolved model data shows only a single complete stripe of mixed ST-2 and chloride material bisecting the southwestern exposure. The modeling data shows a number of partial stripes; these partial stripes are collocated with the other stripes seen in the DCS data. The original resolution data do not show these stripes, but does have isolated pixels of ST-2 composition within the deposit.

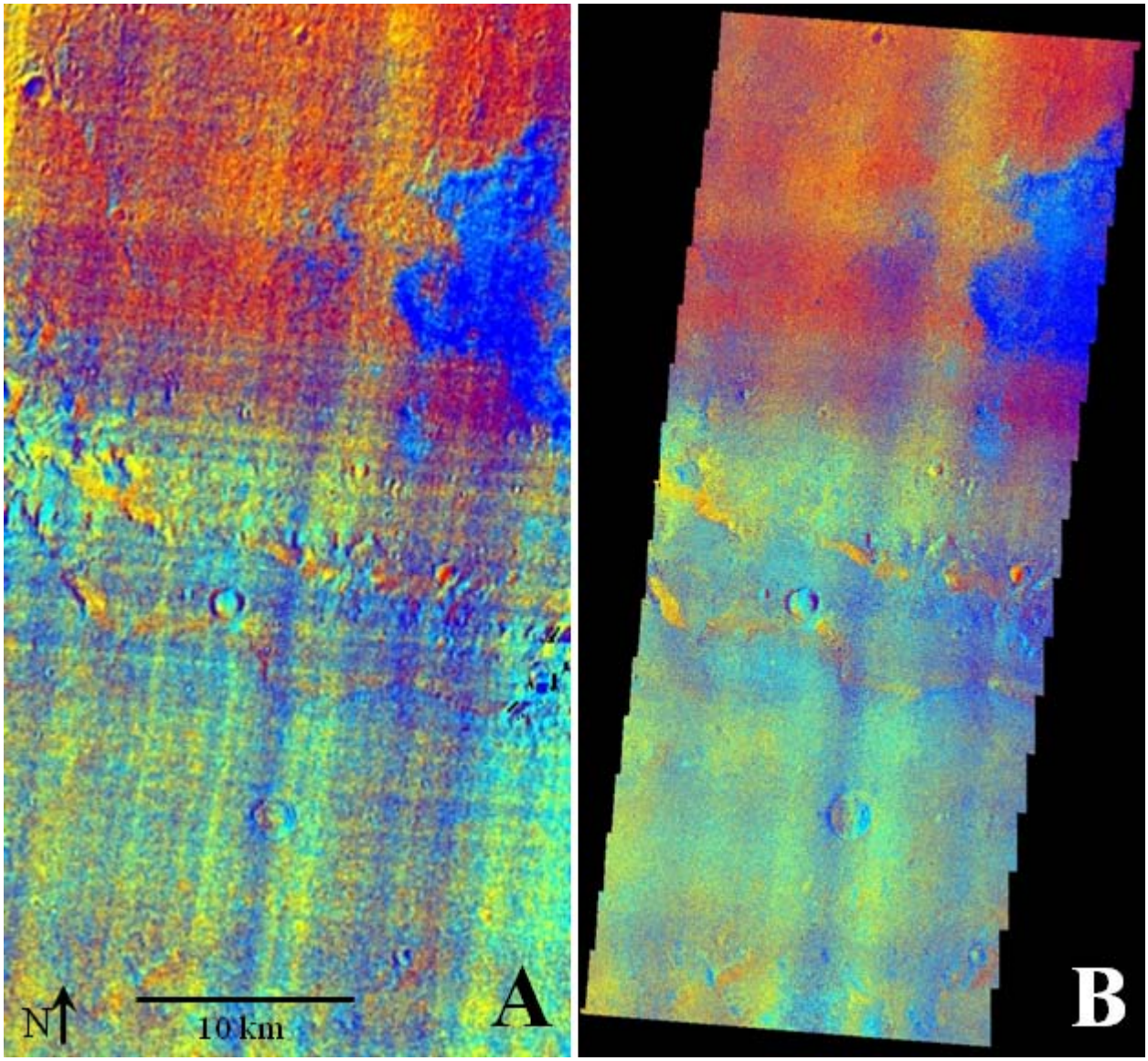

Figure 3-8 End-member contribution to the model spectra. Surface Type 2 is shown as red, the blackbody end-member is shown as green, and chloride is shown as blue. (A) Original resolution data. (B) Super-resolved data.

3.4.3 Comparison Between THEMIS TIR and Other Data

The super-resolved data has an improved resolution for comparison to other Mars data. Alignment between the blue unit in the super-resolved data and the lighter-toned unit in the CRISM IR brightness data (Figure 3-2) is excellent. The HiRISE data (Figure 3-3) also aligns well, with the irregular margin between the lighter and darker toned units collocated with the purple pixels in the DCS data. These pixels are associated with a mixed composition between more weathered ST-2, based on the appearance in CRISM and HiRISE data, and the chloride units. The small-scale nature of this mixture can be illustrated at the bend in the margin within the HiRISE data (Figure 3-3). The surrounding surface in this area can be seen to overlay the chloride unit, and both were later impacted by a crater.

Compositional information from the super-resolved data show that the deposit examined in this work is thin, with a thickness less than 100 m. This is also seen with limited exposures in CRISM data. The super-resolved data shows additional ST-2 material below this point. This material may represent the surface on which the putative chlorides were emplaced, or it may be from later crater ejecta that filled in topographic lows that would expose a different underlying material. Neither CRISM nor THEMIS data indicate the presence of phyllosilicates in this immediate region. It is likely there are underlying phyllosilicates based on Glotch et al. (2010), but this material may not be exposed in detectable quantities here. The underlying ST-2 unit may be protecting any phyllosilicates from weathering. Alternatively, the phyllosilicate-rich material seen in Glotch et al. (2010) may be part of a continuous lithostratigraphic unit with the ST-2 material below the chlorides, but has been subjected to a different weathering environment.

3.5 CONCLUSIONS

The super-resolution process produced good results at the site examined in this work. The super-resolved data showed an increase in data diversity over limited spatial areas, with indication that this diversity is associated with sub-pixel scale exposures of surfaces. Data from higher resolution instruments, such as CRISM and HiRISE, show small-scale variations collocated with these differences, but the THEMIS data can provide mineralogic composition of these difference units. Features, such as small craters, are more apparent in the super-resolved data. The relationship between the surrounding terrain, the putative chloride unit, and crater ejecta can be more readily determined within the super-resolved data. By using the super-resolved data in conjunction with CRISM data, it is possible to infer the thickness of the putative chloride unit to be less than 100 meters. Some features do not have the sharp margins apparent in the original resolution data when examined with super-resolved data. The large putative chloride exposure shows a more mixed region to the west and south.

The putative chloride deposit shows a haloed effect, with a blend of super-resolved chloride and non-chloride pixels surrounding the edges of the unit. This halo shows a definite gradation. Within the main exposure, there are few ST-2 pixels, and only slightly more mixed pixels. These mixed pixels, seen in both DCS and linear deconvolution of super-resolved data, are composed of sub-pixel scale exposures of both ST-2 and chloride. After a relatively sharp margin of pure chloride pixels, there is a rim composed almost exclusively of mixed pixels for several hundred meters. This region is the margin of the exposure through the overlying ST-2 material. Isolated sub-pixel scale windows through the ST-2 expose the underlying chloride to the surface. Surrounding all of this is a large region modeled as nearly pure ST-2.

Moving southwest from the main exposure, there is a corridor of these mixed pixels, intermingled with purer pixels of either end-member in linear deconvolution that stretches to a smaller exposure of the putative chloride. Unlike the main exposure, this unit is collocated with bands of mixed pixels. In the linear deconvolution results, these bands are partial, with only one band stretching the complete length of the deposit. In the DCS data, several smaller bands cover the length of the deposit. In either case, these bands represent heavily eroded ST-2, with windows of the putative chloride exposed at the sub-pixel scale over a large areal portion of the pixel. The centers of these bands are likely composed of heavily eroded ST-2, with pixels whose areal exposure is mostly chloride. This southwestern exposure appears similar to the northwestern segment of the larger exposure; both areas are less weathered exposures of the underlying unit. In the linear deconvolution results, these areas require less blackbody contribution.

No exposures of phyllosilicates are seen at this location. There are few exposures through the putative chloride. In those areas where erosion has continued through the deposit, additional ST-2 material is exposed. This ST-2 may have one of two origins. They may represent areas in which ST-2 was remobilized and deposited, following the deposition of the putative chloride unit. Alternatively, these ST-2 exposures could be part of the same lithostratigraphic unit as the phyllosilicate-rich unit in Glotch et al. (2010), but subjected to different weathering conditions.

The putative chloride site here is composed of a larger homogeneous unit with a smaller associated exposure to the southwest. These deposits are located within a regional topographic low, have a spectrally bland appearance, and show a negative spectral slope within the TIR. A

rough, cracked surface can be seen in higher resolution data, such as HiRISE. Taken together, this would seem to indicate that these units are evaporites, that their depositional environment was as wet one, and that these conditions were not isolated to small areas of Mars. The possible presence of a pre-existing ST-2 surface beneath these exposures may represent a basaltic aquifer, leading to acidification of surface waters. This and other identified putative chloride sites are being examined with other data sets from orbit, but, presents a worthwhile target for future lander or rover missions.

4.0 THE USE OF LUNAR LAKE PLAYA AS A PLANETARY ANALOG AND CALIBRATION TEST SITE

4.1 INTRODUCTION

Lunar Lake Playa (Figure 4-1) is located at 38.4°N, 116°W, approximately 100 kilometers (km) northeast of Tonopah, Nevada (NV) and 125 km southwest of Ely, NV. The playa lies within the Lunar Lake caldera, along its southeastern margin. This is the youngest caldera in the multiple-caldron complex of central Nevada (Ekren et al., 1974). The playa is roughly kidney shaped, and covers an area of approximately 7 km^2, with a perimeter of approximately 11 km. The axes of the playa are 4.3 km and 1.9 km, with the strike of the major axis extending 053º. It exists within a closed basin, with an elevation of 1751 meters (m) above sea level (ASL). There is almost no topography within the playa; relief is on the centimeter-scale. Surrounding peaks generally have elevations between 1700 and 1800 m ASL, with an arcuate ridge ("The Wall") at 1800 m ASL for an extended distance.

To the north and east of the playa, there are Quaternary basaltic lava flows and tuff cones. These are associated with Lunar Crater (4.25 km west of the playa) and related volcanics. Volcanics are associated with northeast trending faults (Ekren et al., 1974). The south and west are bounded by uplifted Tertiary rhyolitic tuff, which are also associated with northeast trending

Figure 4-1 ASTER data acquired on July 30, 2006 at 11:38 am local time of Lunar Lake Playa and its immediate surroundings. ASTER bands 3N (0.807 µm), 2 (0.661µm), and 1 (0.556µm) are shown in RGB. Area covered in clasts can be seen along the western end of the playa, adjacent to the perimeter, as a medium gray unit extending into the lighter-toned playa. Squares are areas in which pixel surveys were conducted; the large square on the on the eastern side is the Survey Area 2, and the square on the western side is Survey Area 3. The small diamond south of Survey Area 2 is Survey Area 1, the site of the 30 m x 30 m pixel survey conducted in March, 2010. The yellow cross marks the location of the FLIR camera during ASTER overpasses on July 27 (day) and 28 (night), 2010. The black line extending across the playa marks the location of the partial transect. The circle on the south of the playa shows the location from which samples March-2A through March-2E were collected. The inset shows the approximate location of the site within Nye County, Nevada.

faults (Ekren et al., 1974; Petroy and Arvidson, 1990).The northern and western shores of the playa contact basaltic flows and tuff cones, and the eastern margin is marked by a sand berm. Small areas along the perimeter of the playa are covered to various extents with centimeter scale cobbles derived primarily from the surrounding basalts. Meter to decameter scale areas within the playa, away from the perimeter, are similarly covered with centimeter scale cobbles. Along the northern and western margin, these cobbles can be decimeter scale. The playa surface is smooth, hardpacked, and fine-grained.

4.1.1 Uses of Lunar Lake Playa

Lunar Lake playa and the surrounding region have a history of use as a planetary analog extending back 50 or more years. In the 1960s, the surrounding region was used along with the Nevada Test Site as a training site for the Apollo astronauts. During the late 1980s and early 1990s, the immediate region was used as an analog for radar remote sensing of Venus, particularly the younger basaltic flows (Arvidson et al., 1992). The region has also been used as an analog for Mars, both for remote sensing and for rover exploration (Baumgartner; Petroy and Arvidson, 1990). Part of the reason for the multiple uses for the site is its choice as the primary location for the Geologic Remote Sensing Field Experiment (GRSFE), in which the Lunar Lake region was examined in great detail by a number of remote sensing systems during a field campaign.

Lunar Lake also has a history of use as a calibration target for Earth-orbiting sensors. It is among one of several sites used for the vicarious calibration of remote sensing instruments because it is homogeneous on a sufficiently large scale, shows little topographic relief, has high

reflectance across a broad spectral range, experiences low atmospheric aerosol loading and has a relatively cloud-free environment (Bruegge et al., 2002; Thome et al., 1998). The designation of the site for the GRSFE led to it being used for other calibration measurements. Lunar Lake has been used for vicarious calibration of the Earth Orbiting Systems (EOS) instruments (Bruegge et al., 2002; Thome et al., 1998), IKONOS (Pagnutti et al., 2003), and Landsat-5 TM & Landsat-7 ETM+ (Thome et al., 2004). Due to its smaller size, the neighboring Railroad Valley playa is used at times for the calibration of moderate to low spatial resolution instruments. Given the proximity of the two playas, both are commonly imaged in the same scene. As a result of the spectral, spatial, and temporal range of data collected for Lunar Lake, the site is very well characterized, which makes it an excellent target for the testing of new instruments and new data processing methods.

4.2 DATA AND METHODS

4.2.1 ASTER

The ASTER instrument is composed of three wavelength sub-systems, all using separate telescopes. ASTER has 3 channels between 0.52 and 0.86 microns at 15 m/pixel spatial resolution, 6 channels in the 1.6 to 2.43 micron short wave infrared (SWIR) region with at a spatial resolution of 30 m/pixel, and 5 channels between 8.13 and 11.65 microns at 90 m/pixel spatial resolution (Fujisada et al., 1998). The Signal to Noise Ratio (SNR) values range between 44 – 368 for low gain radiance values, and 156 – 466 for high gain radiance values (Fujisada et al., 1998). The VIS instrument uses a 5000 x 4 element array of silicon detectors; the SWIR

instrument uses a 2048 x 6 element array of PtSi detectors. Both form images with a pushbroom configuration. The TIR instrument is a 10 x 5 array of HgCdTe elements used in a whiskbroom configuration (Yamaguchi et al., 1998). The FOV for all three instruments is 6.09° square. A fourth telescope pointed backward collects data in the near-infrared, and uses 4100 elements of the silicon detector to create along track digital elevation models (DEMs). Since April, 2008, ASTER has been acquiring useable multispectral / multispatial data in only two wavelength regions. Data from the SWIR instrument are no longer usable due to a failed cryo-cooler.

Although 282 ASTER scenes are available with coverage of Lunar Lake playa, the data used in this work were collected on July 30, 2006 at 11:38 am local time. This scene was chosen for being cloud-free, having data in all three spectral regions, and for being at the same time of year as fieldwork conducted at Lunar Lake playa. Data in this scene were subset to lines 2278 through 2677 and samples 420 through 755 of the 15 m / pixel resolution VIS bands (Figure 4-1). Co-located pixels were used for the SWIR (30 m / pixel) and TIR (90 m / pixel) bands.

4.2.2 AVIRIS

The AVIRIS instrument is a 224 band airborne whisk-broom imaging spectrometer, with a spectral range between 0.4 and 2.5 μm, 10 nm band spacing, and a 1 μrad Instantaneous Field of View (IFOV) (Green et al., 1998). There are four spectrometer subsystems within the instrument: The A subsystem covers 400 – 700 nm, B covers 700 – 1300 nm, C covers 1300 – 1900 nm, and D covers 1900 – 2500 nm. Each subsystem is optimized for its particular wavelength region. Data are collected from these spectrometers and digitized to 12 bit DN; the SNR exceeds 100:1 in all bands, and frequently exceeds 500:1 in wavelengths through 1800 nm.

NASA and the AVIRIS team have made four sets of data to the public through the AVIRIS website at http://aviris.jpl.nasa.gov; one of these sets covers the Lunar Lake region. Over 2 GB of data were collected by AVIRIS over Lunar Lake playa on August 19, 2009. These data replace previously available AVIRIS data collected in June of 1997, and require both different and less calibration processing. The 2009 data are orthorectified and geocorrected, and have a spatial resolution of 15 m / pixel. These data were subset to the area surrounding the playa by producing a 600 x 600 x 224 data cube. The cube was then rotated by 78.0° to align North to the top of the image using a 1st degree polynomial rotation, scale and translation (RST) and cubic convolution, which best preserves the data statistics and spectra. The centermost 400 x 400 pixels were then used in all further analysis (Figure 4-2).

4.2.3 Aerial Photography

Geolocated and orthorectified aerial data of Lunar Lake playa were obtained from Microsoft Research Maps at http://www.msrmaps.com, with images sourced from the United States Geologic Survey (USGS). In addition, topography, hydrography, and geolocated / orthorectified aerial photography were obtained from The National Map Project at http://viewer.nationalmap.gov/viewer. The MSRMaps aerial image is a single band with a resolution of 4 m / pixel. These data also clearly show where separate images were mosaiced together to create the MSRMaps data, due to differences in distribution of DN and the stretches applied to the data. The National Map aerial data were collected by the National Agriculture Imagery Program (NAIP), and are provided at a 1 m / pixel resolution with a horizontal accuracy of at least 5 m. The topography data were collected as part of the National Elevation DATA (NED) at a resolution of roughly 10800 pixels per degree (ppd). This translates to 1/3 of an

Figure 4-2 Lunar Lake Playa and its immediate surroundings are shown using data collected by the airborne AVIRIS instrument on August 19, 2009. The figure shown is in approximate-true color, with bands 30 (0.648 µm), 16 (0.511 µm), and 10 (0.453 µm) shown in RGB. The spatial resolution is 15 m, the same as Figure 4-1, with the data clipped to show only the playa and its immediate surroundings. Areas covered in clasts are significantly more visible in the true color image than in the ASTER data. The stretch applied to this data also reveals some of the variation in the reflection of the playa.

arc-second or 10 m / pixel horizontal resolution, and 0.1 m vertical accuracy. Due to the size and resolution differences, areas were first examined in the MSRMaps data, and then only points of interest were examined in the NAIP and NED data.

4.2.4 Fieldwork

Fieldwork was conducted at Lunar Lake playa by a team from the University of Pittsburgh in March, 2010. A second field campaign was conducted by the author in late July, 2010. During the March campaign, harsh winter conditions were present at the site making field data collection nearly impossible. The playa surface had a centimeter-scale layer of ice, covering the entire surface except for the larger cobbles. During July, expected summer conditions prevailed with the exception of one day with significant rainfall.

Fieldwork was used to better characterize Lunar Lake playa than was possible from remote sensing data alone. During the fieldwork, areas around the playa were characterized for areal abundances of surface units through pixel surveys. Samples were collected during these surveys, as well as during a transect across the playa. Sampling was done to permit mineralogic analysis, and the results used to better characterize areas of the playa from remote sensing data not examined in detail in fieldwork. Mapping was also done of the margin between the clay playa surface and the surrounding clast-rich shoreline and of strewn fields on the playa. These mapped areas can be compared to both original and super-resolved ASTER data, with the fieldwork mapping acting as a ground-truth for the remote sensing data.

Pixel surveys are designed to produce a reasonably-accurate measure of ground cover and are used as a data/algorithm validation technique. These surveys were conducted in both March and July of 2010. Due to the ice, the March pixel survey encompassed only the shoreline area along the eastern side of the playa. A 30 area (Survey Area 1) was marked out, designed to encompass four ASTER VNIR pixels. Two individuals each walked the diagonal and noted the ground cover within one meter of their stopping points every 3 meters. During the July fieldwork, two 90 m pixel survey were conducted. The first (Survey Area 2) was near Survey Area 1, and only partially on the shoreline. A second 90 m pixel survey (Survey Area 3) was conducted on the northwestern side of the playa. Both July pixel surveys were done by marking nine contiguous 30 m pixels in a 3 x 3 array, and walking the diagonals of each 30 m pixel. As in the March survey, the average composition of each pixel was determined by noting the ground cover at each stop along each diagonal. The nine sub-pixel values were then averaged together, producing one overall set of ground cover value per survey site.

During the March campaign, poor weather conditions prevented the collection of a significant number of samples. However, two types of samples were collected from three locations. Soil samples were collected by pressing a 25 mm diameter plastic vial into the surface. The vials were 52 mm high; however, the vial did not penetrate more than 15 – 20 mm before encountering frozen soil. Soil samples March-1A through March-1F were collected from on the playa, inward of Survey Area 1 (Figure 4-1). These samples were collected after first removing ice from the surface, and pressing the vials through the water into the soil below. These samples were collected at 5 different distances, up to 100 m from the edge of the ice, and from both clear areas (no clasts) and strewn fields. Soil samples March-2A through March-2E were collected

from a site near the southern tip of the playa (Figure 4-1). These samples were collected from within a meter of the ice edge, on either side. Finally, clast samples were collected during the pixel survey. Clasts were sorted into two separate types: basaltic clasts and metamorphic / other clasts. Clast samples were also sorted by size into small and medium size. Small clasts were defined as centimeter-scale and medium clasts were 5 – 10 cm in diameter.

Both soil and clast samples were collected during the July fieldwork. Soil samples were collected after a day of significant rainfall from a portion of the playa surface that appeared dry and from an adjacent area that appeared wet one hour after the end of the July 26th rainfall using the same method as March. Both locations are within the Survey Area 2 (Figure 4-1). Clast samples were collected from both pixel survey areas, and sorted by size into small, medium, and large clasts. Small and medium clasts from the July fieldwork have the same scale as the March fieldwork and large clasts had at least 2 dimensions greater than 10 cm

During the July fieldwork, a transect was conducted across the playa (Figure 4-1). The transect started in the vicinity of the Survey Area 1, at the shoreline, and moved in a straight line across the lake. A total of 93 soil samples were collected, with an average spacing between samples of approximately 15 m. Rare small clasts were sometimes within 1-2 meters of a sample location; however, none were present at any sampling site. As a result, only soil samples were collected. During the transect, the playa surface was too hard to penetrate with the vials beyond an approximate 5 mm depth. As a result, vials were used instead to dig very shallow (under 5 mm) trenches in the surface, and this material was collected.

4.2.5 Laboratory Analysis

Collected samples were analyzed using a Nicolet Nexus 670 Fourier Transform InfraRed (FTIR) Spectrometer at the University of Pittsburgh Image Visualization and Infrared Spectroscopy (IVIS) facility. Sampling was done at 2 cm^{-1} spectral resolution between 400 and 2000 wavenumber (cm^{-1}), or ~5 – 25 μm, using the calibration method described in Ruff et al. (1997). Before analysis of any samples, and at least once every four hours, blackbodies were scanned at 70° C and 100° C. The blackbody used for this work is a custom-designed cylinder in which a conical section has been hollowed out and a spectrally flat paint has been applied. The spectrometer and the analysis chamber are purged with air that has most carbon dioxide and water vapor removed, so as to minimize absorption features by these gasses.

All samples were heated to 80° C for at least 24 hours prior to analysis. Soil and rock samples were analyzed using two different methods. Soils and small clasts were placed in copper cups painted with the same blackbody paint, and then onto a heating stage. This stage kept the samples at a stable temperature of 80° C during analysis. Larger rock samples were removed from the oven and allowed to cool during spectral acquisition. The thermal inertia was assumed to keep them at an adequate temperature for analysis. These samples were placed within a rock holder that provided no active heating.

Rock samples were processed in two different states; half of the analyzed samples were left as-collected, while the other half were briefly rinsed under water, patted dry, rinsed with acetone, and patted dry a final time. Although there was very minor risk of altering the chemistry, and hence the spectra, of the basaltic clasts through wetting, the risk was considered

minimal due to: their collection from a natural surface on which it had rained immediately before collection; the rapid drying immediately following rinsing, followed by a rinse in acetone; and, their immediate placement within an oven after rinsing. The decision to rinse half of the samples was made to increase the probability of detecting a stronger signal from the underlying weathered rock surface; with a depth of penetration measured in microns, a coating of soil on the clast would likely conceal some or all of the rock spectrum.

The blackbody, rock, or heating stage (dependent on what was being analyzed) was raised into a controlled environment within the analysis chamber; this area was kept at a constant temperature of 25° C. By keeping the environmental temperature constant, downwelling energy originating from multiple temperature sources in the environment could be minimized and modeled using the known temperature of the chamber. The spot-size is approximately 2 cm in diameter. During analysis, each sample was scanned 512 times, and those data averaged together to minimize the noise.

Following collection of the spectra, the data were analyzed in two different ways. First, spectra were examined at a qualitative level, to ensure good data acquisition and to assess the materials that may be present. Following this, spectra were processed with linear deconvolution (Ramsey and Christensen, 1998; Thomson and Salisbury, 1993). Mixed spectra can be modeled by deconvolving the spectrum against a spectral library to determine the sub-pixel component. This process is limited by a number of factors though. First, the results will be largely dependent upon the quality and breadth of the spectral library used for deconvolution. Second, in order to remain mathematically valid, the number of possible end-members within a pixel must be equal

to or less than the number of bands used to create the spectrum (Adams et al., 1986; Ramsey and Christensen, 1998). For spectra acquired in the laboratory, this factor does not pose a constraint because these spectra have hundreds of separate bands. Linear deconvolution of the ASTER data is limited to only 5 end-members (the number of TIR bands). The Arizona State University (ASU) spectral library was used for the analysis (Christensen et al., 2000).

4.3 RESULTS

4.3.1 Fieldwork Results

During the March fieldwork, Survey Area 1 was found to have significant spatial heterogeneity (Figure 4-3). Clasts covered between 0 and 95% of the surface across the survey points. Despite its location on the side of the playa closer to the rhyolitic tuffs, clasts tagged as “basaltic” outnumbered “metamorphic / other” clasts nearly 4:3. Across the space of the entire survey area, the three end-members were split nearly evenly, with clay/soil covering 33.5%, basalt covering 38%, and other clasts covering 28.5%. The predicted spectrum based on these results is shown compared to the actual ASTER spectrum and a super-resolved spectrum in Figure 4-4. This super-resolved spectrum was the one located physically closest to the center of the survey area. The modeled spectrum featured a good fit to both the original and super-resolved spectra.

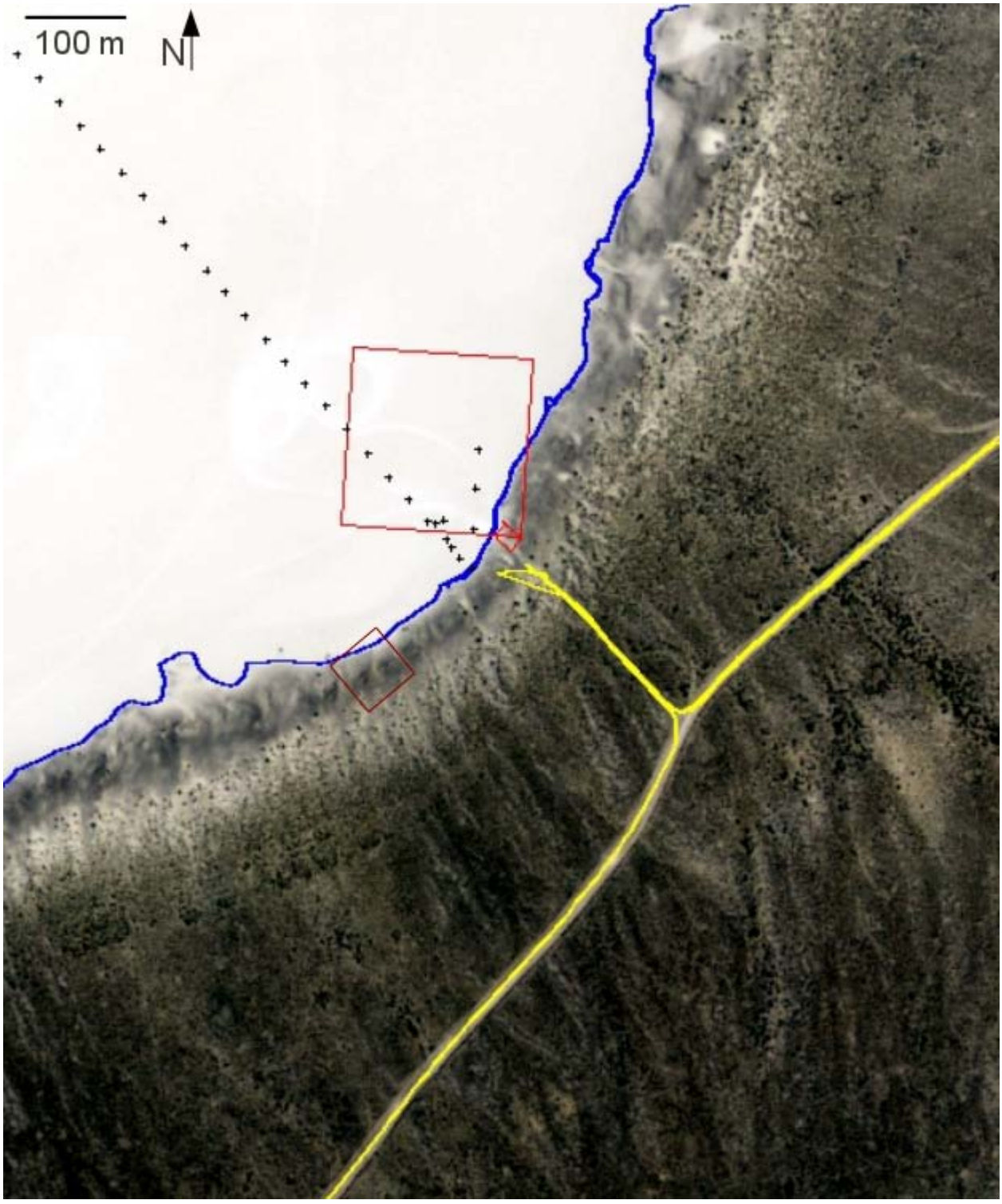

Figure 4-3 Survey Areas 1 and 2 shown on the NAID map. The medium red square is the location of Survey Area 1, and the large red square is the location of Survey Area 2. The mapped boundary of the playa is shown as a thick blue line, and roads are shown in yellow for reference. Black crosses show the locations of soil samples collection.

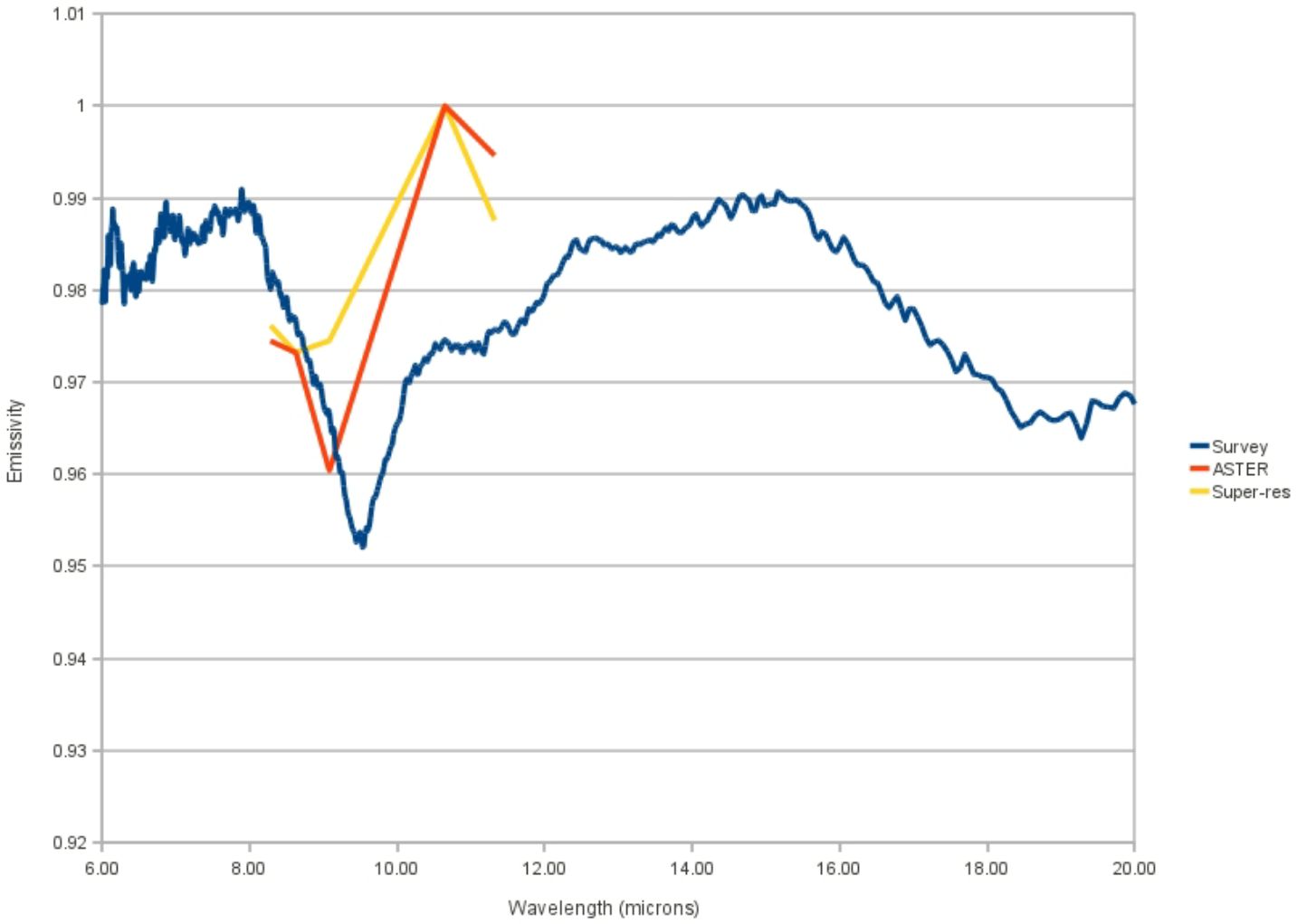

Figure 4-4 Laboratory spectra are combined in a weighted average based on the Survey Area 1 pixel survey results. Also shown are the original resolution ASTER TIR and the super-resolved TIR spectra.

During the July fieldwork, Survey Area 2 was generally homogeneous (Figure 4-3). After survey, the pixel had 3.8% of its surface covered with clasts. Within the nine 30m x 30m pixels, there was a significant variety in the amount of clast coverage, with values ranging between 0.72% and 20.57%. It should be noted that the southeastern 30m pixel was almost entirely on the "beach" area (20.57% clast), and the central-eastern pixel was mostly on the beach (5.65% clast); the rest of the 90m pixel covered the playa (0.72% - 1.4% clast). As there were no apparent differences within the clast composition, all clasts were considered basaltic. The predicted spectrum based on these results is shown compared to the actual ASTER spectrum of the pixel and a super-resolved spectrum in Figure 4-5. The super-resolved spectrum shown was chosen from the 36 total super-resolved spectra by finding the one which best fit both the original and super-resolved data.

Survey Area 3 (Figure 4-6) featured significant heterogeneity in surface composition, and was the site used for surveying with a FLIR camera during ASTER overpasses. Within the survey region, there were "light" (playa) and "dark" (basaltic clast) areas. Light areas had playa as greater than 90% of their surface. Dark areas had less than 60% playa exposure. During the pixel survey, the only survey points with playa exposure between these amounts were all on the border between light and dark areas. Areas of the same type (light or dark) were continuous on a decameter scale. As a whole, the 90m pixel had an average composition of 60.0% playa material, 32.8% small basaltic clast, 6.8% medium basaltic clast, and 0.4% large basaltic clast. Within each of the nine 30m areas, the minimum average playa exposure was 43.4% and the largest was 85.6%. The spectrum based on these results is shown compared to the actual ASTER spectrum the super-resolved spectrum in Figure 4-7.

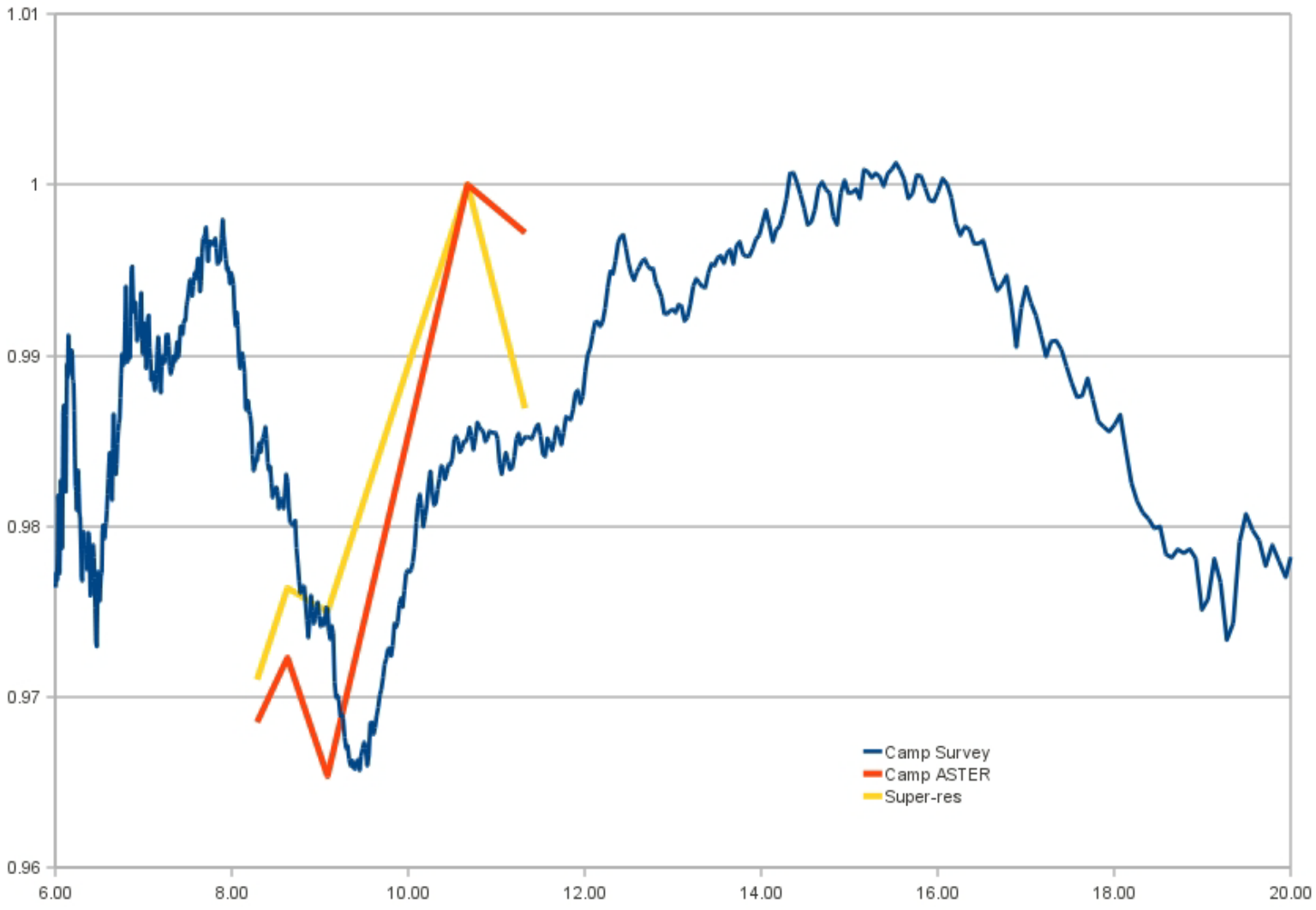

Figure 4-5 Laboratory spectra are combined in a weighted average based on the Survey Area 2 pixel survey results. Also shown are the original resolution ASTER TIR and the super-resolved TIR spectra.

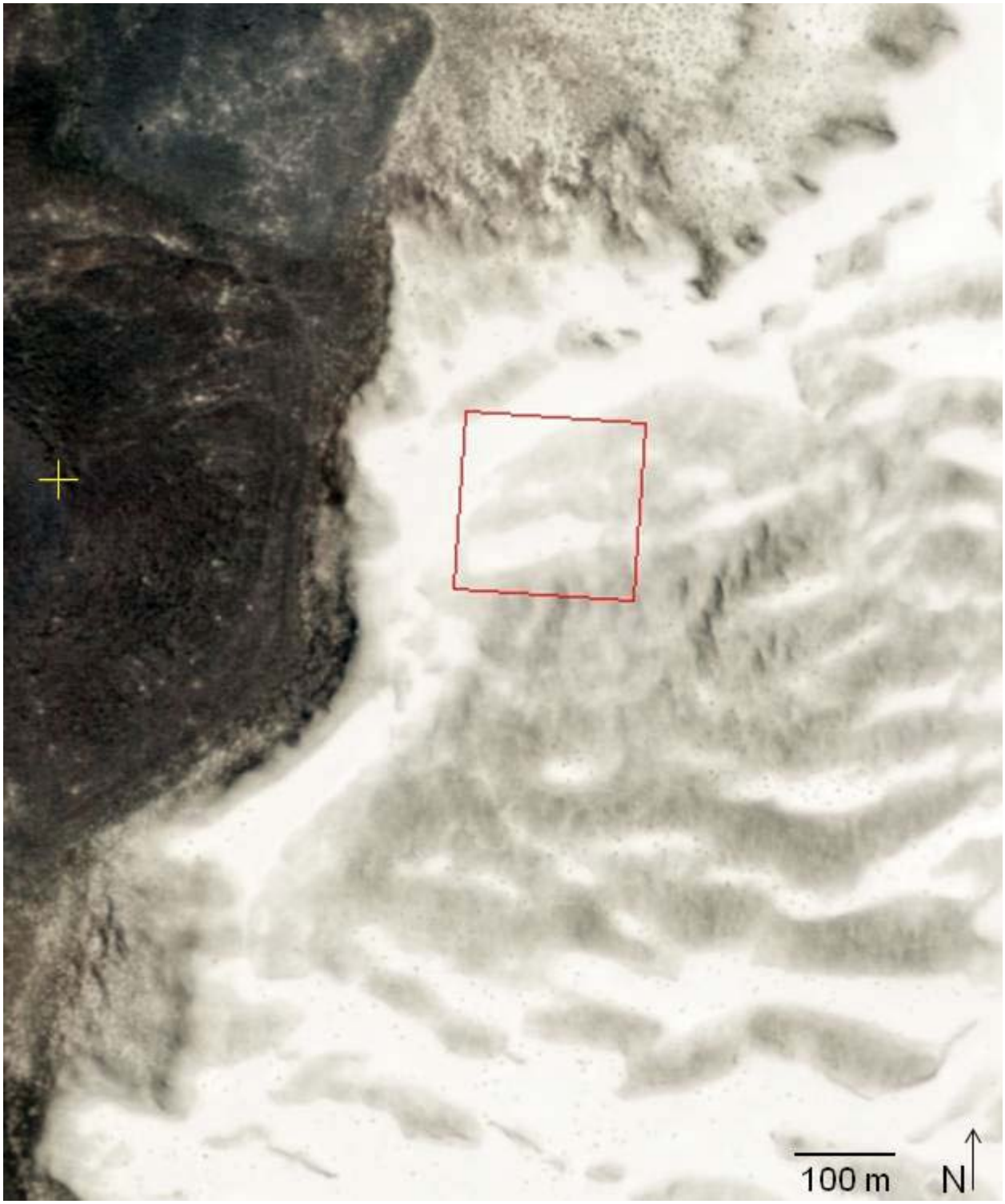

Figure 4-6 Survey Area 3 shown on the NAID map. The red square marks the location of Survey Area 3 and the yellow cross marks the location of the FLIR camera during ASTER overpasses.

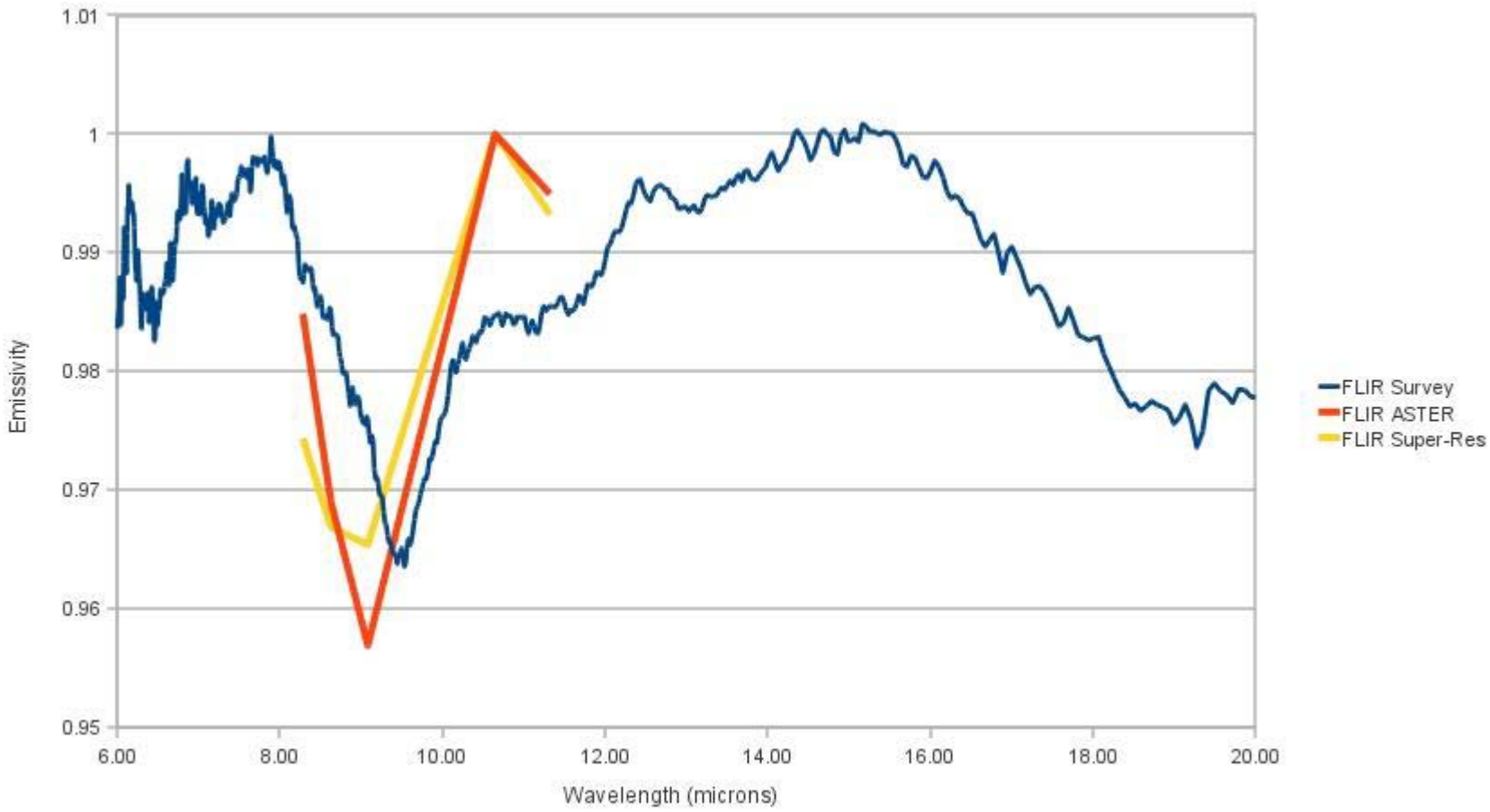

Figure 4-7 Laboratory spectra are combined in a weighted average based on the Survey Area 3 pixel survey results. Also shown are the original resolution ASTER TIR and the super-resolved TIR spectra.

Spectra from the samples collected during the transect (Figure 4-3) can be seen in Figure 4-8. There is a trend for the maximum absorption feature, between 9.0 and 10.0 μm, to shallow and shift to shorter wavelengths with increasing distance from shoreline. Similarly, other spectral features become muted with increasing distance from the shore as well. However, features are generally easily discernible at the same spectral location across all spectra.

4.3.2 Spectral Analysis Results

A subset of the laboratory spectra of playa surface samples and of clasts are shown in Figure 4-9 and Figure 4-10. Figure 4-9 samples are representative of several different types of spectra seen during data collection. The spectra in Figure 4-10 show several different basalt clast spectra as well as the spectrum of the rhyolitic clasts from Survey Area 1. Spectra from both figures were linear deconvolved to derive end-member areal abundances. These spectra also were used in the creation of the model spectra shown in Figures 4-4, 4-5, and 4-7.

Unmixing of the spectra shown in Figures 4-9 and 4-10 using the ASU spectral library (Christensen et al., 2000) produced good results. Model fits had average RMS errors between 0.0037 (Playa II) and 0.0145 (Playa IV). Eight of the 14 minerals found through linear deconvolution are clays; the remaining minerals are pyroxenes, plagioclase, iron oxide, and potassium feldspar. Six of the nine samples contained Ca-montmorillonite and Fe-smectite, and palygorskite was found in five. Several minerals were found in only one sample; the “small clast other” spectrum was modeled as several of these. These clasts were classified during transects as representing either rhyolitic or metamorphic material.

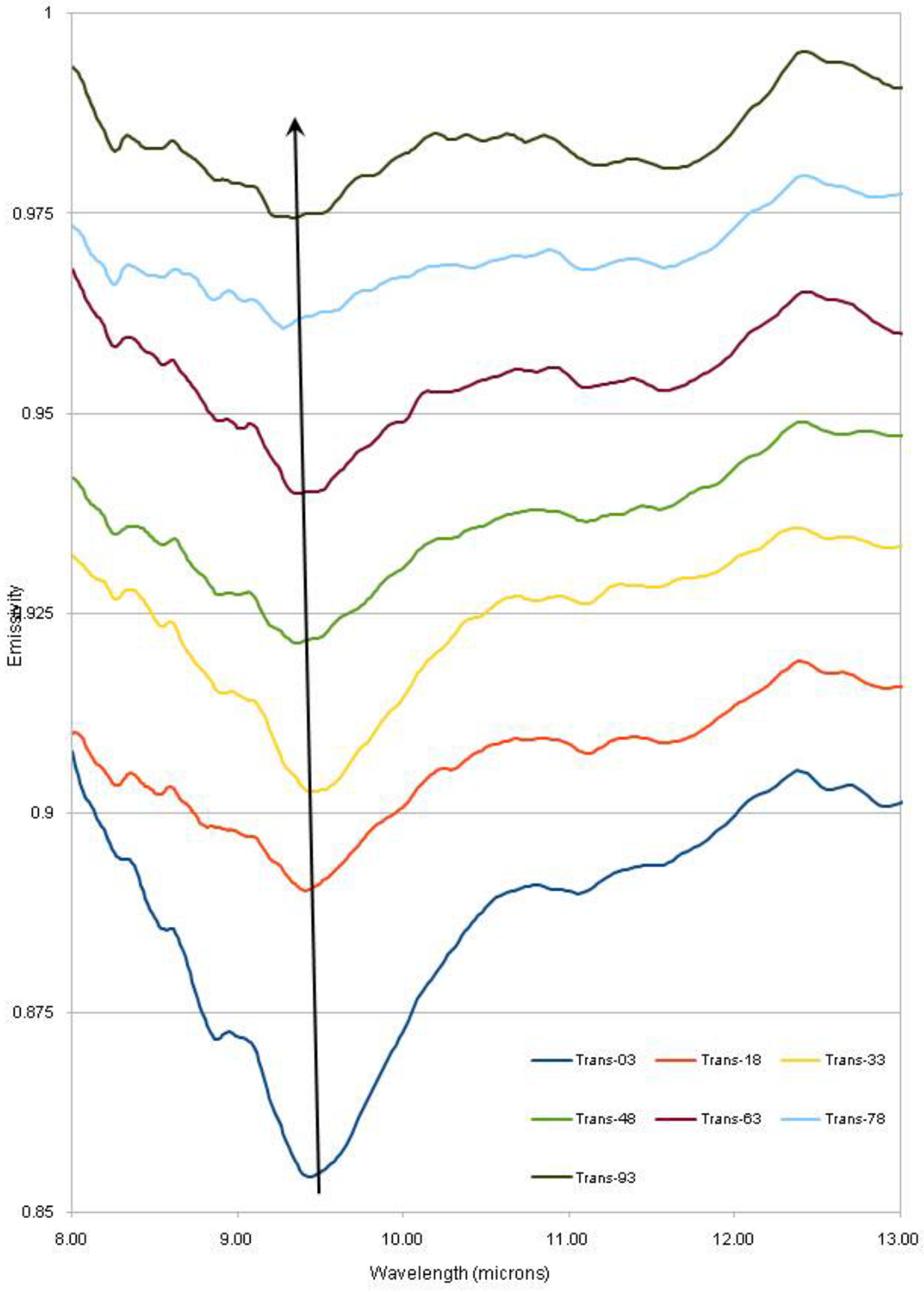

Figure 4-8 Transect sample spectra, offset for clarity, between 8.0 and 13.0 µm. There is a readily apparent trend (shown with the black arrow) for the main absorption feature between 9.0 and 10.0 µm to shift to shorter wavelengths and become shallower with increasing distance from the shoreline.

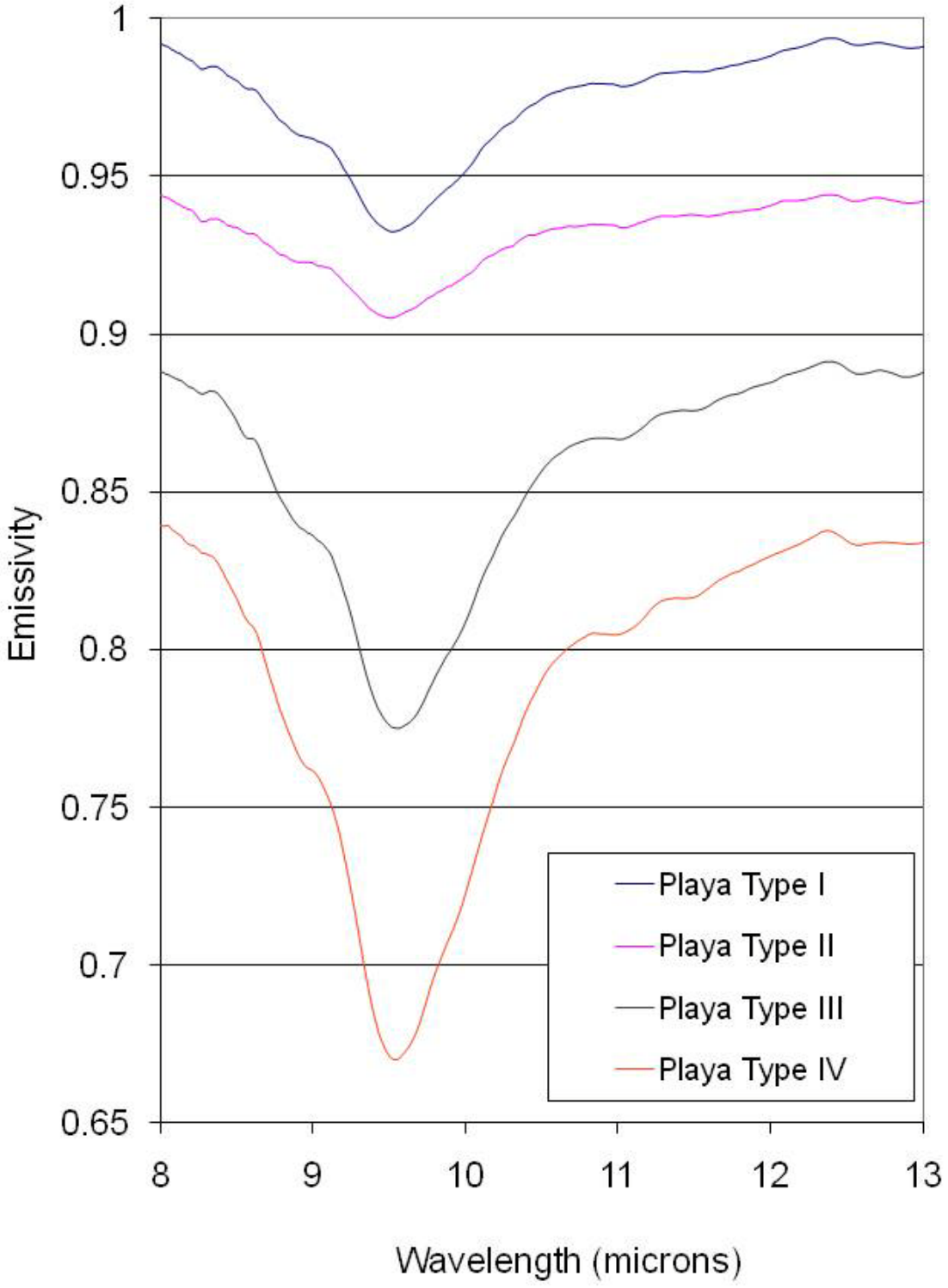

Figure 4-9 Spectra of representative playa surface material, offset for clarity, between 8 and 13 μm.

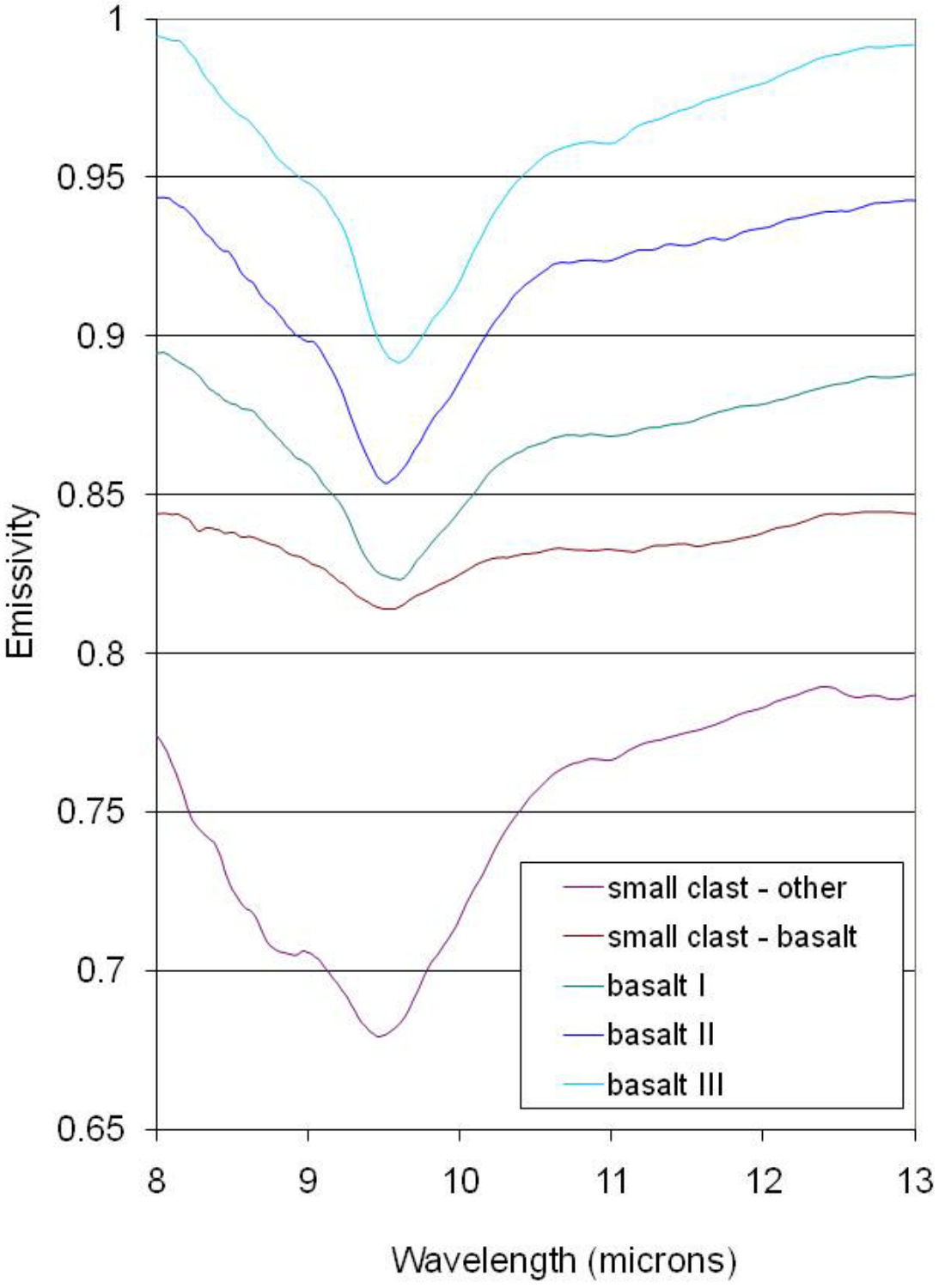

Figure 4-10 Spectra of representative clast material, offset for clarity, between 8 and 13 µm.

End-members derived from linear deconvolution were then used for deconvolution of satellite data. Linear deconvolution is valid within the TIR spectral range, but not in the VIR or SWIR regions. As a result, only ASTER data were deconvolved. During the unmixing of the laboratory spectra, 14 end-members were found. To reduce this number to five or less in order to be used with the ASTER emissivity data, end-members found only in 1 sample of the laboratory data were eliminated. The nine remaining end-members did not produce results which appeared valid. During unmixing, RMS errors were significantly high either within the playa or in the immediate surroundings, depending on which set of end-members were used. A decorrelation stretch of ASTER bands 13, 12, and 10 was done instead (Figure 4-11A), in order to pick out interesting areas and to see if there is correlation between the decorrelation stretch and mineral locations based on transect sampling. The same comparison was used with super-resolved ASTER TIR data (Figure 4-11B) for mapping of light and dark areas in Survey Area 3. As it was not possible to perform linear deconvolution on the AVIRIS data, a cluster analysis (Figure 4-12) was done for comparison purposes.

4.4 DISCUSSION

Data were collected at a variety of spatial and spectral scales during the fieldwork and laboratory spectral analysis. Integration of these differing scales revealed information that was not previously apparent within the data. An analysis of remote sensing data alone would have shown no systematic change in the mineral content away from the shoreline. Collection of sufficient samples to see these patterns across the entire playa would be impractical. Data integration reveals Lunar Lake playa to be more varied than previously documented.

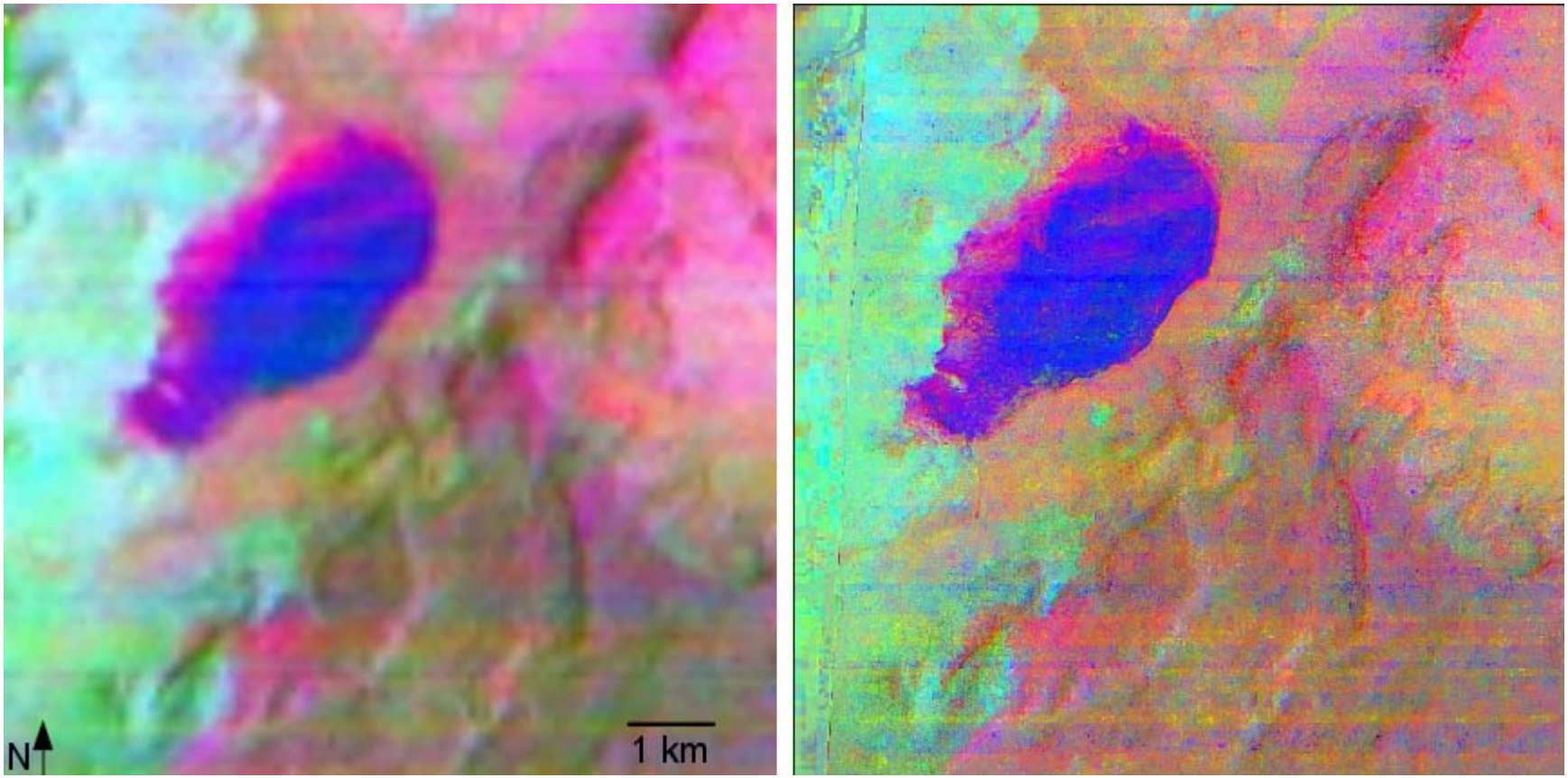

Figure 4-11 ASTER data of the area surrounding Lunar Lake playa are shown using a decorrelation stretch of bands 13 (10.657 µm), 12 (9.075 µm), and 10 (8.291 µm) in RGB. Within this image, the basaltic flows are cyan, the rhyolitic material is orange, playa material is blue, and the clasts on the playa show up as magenta. (A) Original resolution. (B) Super-resolved.

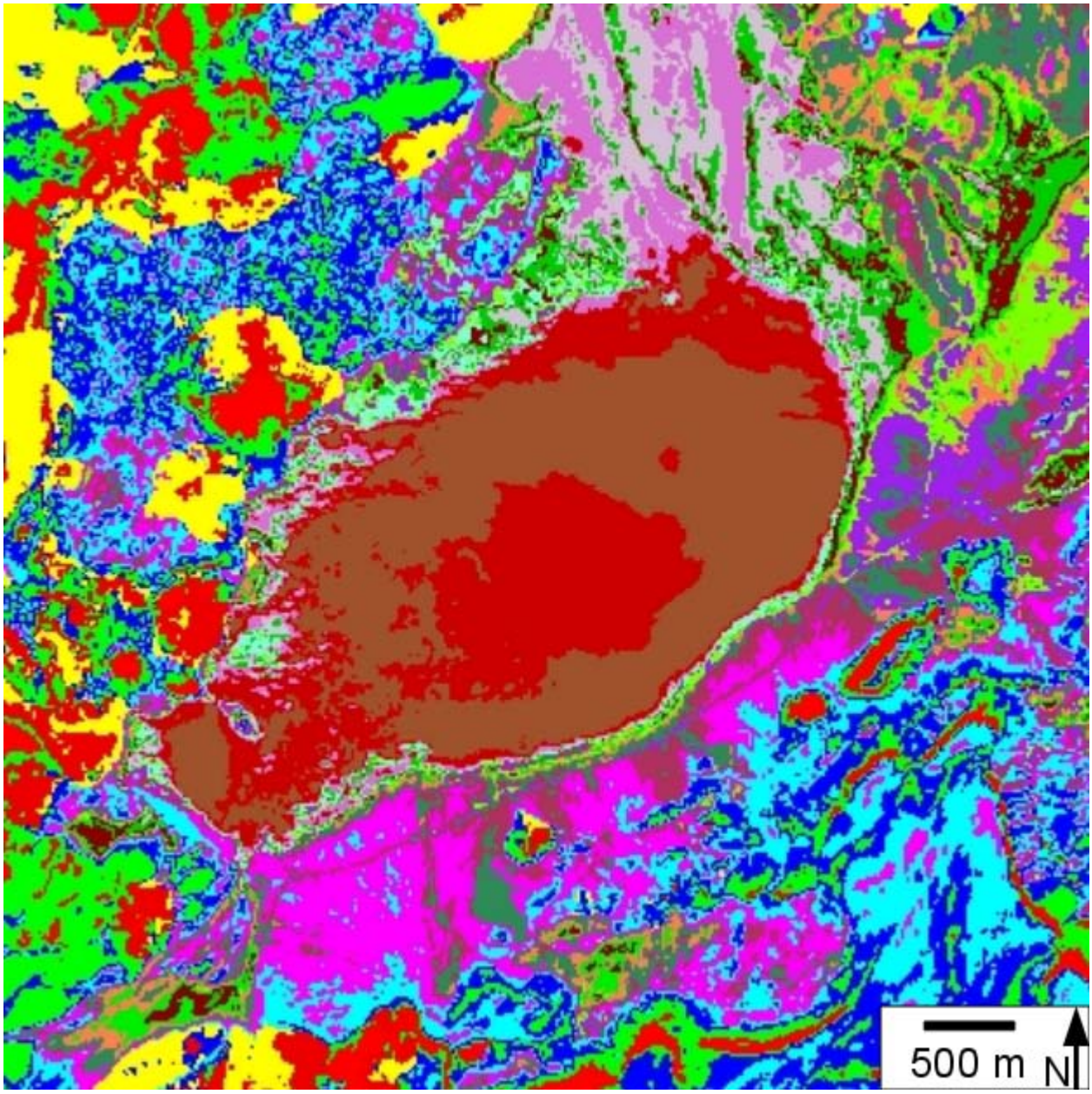

Figure 4-12 The AVIRIS data was clustered using the ISODATA algorithm, with a limit of 10 iterations and between 5 and 20 end-members.

4.4.1 Fieldwork Discussion

In Survey Area 1, the apparent location of the emissivity minima in both the original and super-resolved data is shifted to longer wavelengths relative to the laboratory spectrum built by the weighted average of pixel contents. This shift could be due to the non-perfect alignment between the ASTER and the surveyed pixel. The surveyed pixel is also 11% of the area of an ASTER TIR pixel. As a significant portion of the associated ASTER pixel is over the playa surface, any absorption minima will be shifted to shorter wavelengths as less proportionate signal is derived from mafic material and more from the clays of the playa surface. The super-resolved spectrum overlying this survey area has a different spectral shape than the original resolution ASTER pixel. This is consistent with the laboratory spectra, in which clasts have shallower features than the playa surface clays. As this pixel survey was almost entirely on the beach area, relatively little playa surface would be exposed within the super-resolved area.

Survey Area 2 was homogeneous and almost entirely clay, with minimal clast material. In the course of laying out the survey, only one of the 30 m subpixels was not entirely on the playa. Roughly 30% of the southeastern subpixel extended beyond the playa edge. As a result, clay exposed on the surface of this subpixel dropped to just below 80%, whereas the minimum clay exposure for the other eight 30 m subpixels was 94%. Clasts were evenly distributed between and within the eight playa subpixels. There were differences in how quickly the playa absorbed water after rainfall, but these differences were not associated with clast distribution or surface coloration when the surface dried. Differences between the wet sample and dry sample collected post-rain are minimal, and can be attributed to system noise.

Figure 4-5 shows a good association between the original resolution ASTER pixel and the model spectrum of Survey Area 2. The slight drop between bands 10 and 11 of ASTER is attributed to imperfect atmospheric correction. The magnitude of the spectral absorption feature near 9 μm is well captured within the ASTER data, and the apparent shift between the minima is an artifact of the location of the band centers and the number of bands on ASTER. The same is true for the super-resolved spectra. The majority of the 36 super-resolved spectra were nearly identical. Given the compositional homogeneity of this pixel, most super-resolved pixels will contain only playa material, with insignificant (fewer than 5%) contributions from clasts.

Survey Area 3 was within a region with significant clast coverage (Figure 4-6). This was reflected in the survey, when individual samples would show a jump from 40% playa exposure at one survey point to 98% exposure at the next. A number of large clasts were also found on this side of the playa, unlike Survey Areas 1 or 2. This is most likely attributable to the relative proximity to the volcanic area to the immediate north and west of Survey Area 3. This area of the playa featured steep slopes to within a few meters of the playa, and then a surface composed of nearly-interlocking large basaltic clasts, with little exposure of any clay.

The model spectrum based on Survey Area 3 (Figure 4-7) shows a better correlation to the original resolution ASTER spectrum than seen in the other two pixel surveys. A slight shoulder on the shorter wavelength side of the constructed spectrum is visible as an inflection point at band 11 of the original resolution ASTER data. The super-resolved spectrum shows this same feature. Unlike the Survey Area 2, there were noticeable differences among the 36 super-resolved spectra associated with this pixel. These differences correlate to the differences in

distribution between clast and playa material on the surface. The super-resolved data does a good job of discriminating between the two surface types in this Survey Area despite the limited spectral resolution of the ASTER instrument.

Spectral features in the TIR became more muted in transect samples with increasing distance from the playa edge, although no systematic differences were apparent in the VIS. A shift to shorter wavelengths can be seen in the main silicate absorption feature between 9.0 and 10.0 μm in the TIR data as well. These changes are due to two separate differences with the increased distance from the playa edge. First, as water flows into the playa, it will slow down as it can spread over a greater surface area. This will result in larger particles coming out of suspension and settling, leading to a decrease in particle size with increased distance from the playa edge. Decreasing particle size generally has the effect of muting spectral features. Then as the water evaporates into the dry environment, any molecules in solution will become increasingly concentrated. The remaining brine will be located at increasing distances from the playa edge. This will alter the chemistry of the evaporitic minerals that are deposited during evaporation, and lead to a shift in the spectra based on the mineral composition.

4.4.2 Spectral Analysis Discussion

Playa surface samples collected in March generally have a deeper absorption feature than the July samples. The March playa samples were collected under wet conditions. When these samples were analyzed at the University of Pittsburgh, they were still wet. This allowed them to be spread into the copper cups for spectral analysis, and to dry as a single unit. In comparison, spectra from the transect samples show much shallower features, despite some of them being

collected from the same general vicinity. These samples were already dry before being placed into the copper cups for spectral analysis, and so were granular instead of spread like a paste. This led to particle size effects during analysis of these samples.

Clast spectra also show a difference in the magnitude of spectral features. The cleaned clast samples showed similar features to their natural counterparts. The surface layer of playa material generally muted features. During collection, clasts had been sorted by size. Spectral analysis of the three sizes of basaltic clasts showed no significant differences. Clasts that were categorized as being either rhyolitic or metamorphic in the field were differentiated from the basaltic clasts. These spectra possessed a slight shoulder in their main absorption feature, on the shorter wavelength side of the absorption. This shoulder is due to the presence of non-clay minerals, particularly microcline and augite, within the samples.

Linear deconvolution model data were a good fit to the original spectra (Figures 4-9 and 4-10). The dominant minerals from deconvolution are Ca-montmorillonite and Fe-smectite. Given the playa environment and a surrounding composed of rhyolitic tuff and basalt, these clay minerals are expected. The next two most common minerals are also clays. Halloysite is frequently formed from the weathering of rhyolitic units, and hectorite from the weathering of glass. No evaporites were found through linear deconvolution. Playa IV, the worst model fit based on RMS error, had better fits with minerals such as jade and chlorite; however, these minerals were deemed unlikely and removed from the end-member list before repeating the deconvolution. Prior to the removal of these problem end-members, gypsum was found in several of the samples. Given the environment, it is possible that this was a valid result.

A decorrelation stretch can emphasize spectral differences within multivariable data at the possible expense of magnifying noise. Figure 4-11 shows DCS images generated using decorrelated ASTER bands 13, 12, and 10 in RGB. The original resolution image in Fig. 4-11 was generated after first clipping the data to the super-resolved area. Basaltic flows and explosive deposits are shown to the west of the playa, in cyan. The rhyolitic tuff units dominate to the west. Within the playa, a ring of magenta can be seen; this area represents surfaces with greater than 10% clast coverage. The color of this unit is slightly darker on the eastern side. This may be a result of mixing between basalt-derived and tuff-derived clasts on that side, but data can be over-interpreted using results from such a limited area.

AVIRIS data provides significantly more bands of data than ASTER and the spectral region is far more sensitive to phyllosilicate mineralogy. ISODATA clustering was performed using between 5 and 20 end-members, with the algorithm selecting 20 end-members after the 10^{th} iteration. (Figure 4-12). It is immediately apparent that there are two separate clusters over the area of the playa surface, shown as brown and red in Figure 4-12. This result is reproducible across multiple runs. Fieldwork, laboratory spectra, and ASTER data provide no additional insight into the difference between these two units given their spatial distribution. The spectral resolution of the ASTER SWIR instrument may not be sufficient to detect this difference, and no differences were apparent in the VIS or TIR. The eastern side of the playa shows at least five different clusters along its edge. These clusters form concentric rings within the intermediate region between the clast-free playa and the clast-covered shoreline. Based upon fieldwork and laboratory spectra, these rings are primarily defined by the ratio of playa surface to clast coverage. In addition, they are modified by the type of clasts present, with one of the classes

associated with areas more rich in rhyolitic tuff clasts than basalt clasts. The unit shown in light pink in the northern corner of the playa is co-located with inflow channels, visible in both the aerial and satellite data, and is associated with the plants growing in those channels.

4.5 CONCLUSION

Lunar Lake playa is significantly less spectrally homogeneous than previously thought. There are multiple clay compositions present. These clays appear to have been dominantly developed in situ and influenced by the immediate surroundings. Changes in surface composition were observed moving away from the playa edge, indicating that surface flow is a dominant process in determining this composition. The surface of the playa also appears to be free of evaporitic deposits, though this may be biased by the timing of the fieldwork after a particularly wet winter. Finally, there are differences in composition within the surrounding material, with rhyolitic tuff dominating the eastern side and basalt the western. As these areas weather, clasts are transported onto the playa surface. These clasts can accumulate in large strewn fields on the surface, altering the spectra of the surface for that area and reducing the available surface area for a calibration target. These same strewn field areas have strong value for planetary analog studies. The mixture of basalt clasts onto a clay playa, as in Survey Area 3, produces an area that possibly replicates the boundaries of martian clay deposits. The high spatial resolution of the AVIRIS data, combined with its spectral range and resolution, provide an additional dataset with which to compare this area to other planets.

Super-resolved ASTER data did a good job of discriminating surface differences not apparent in the original resolution ASTER data. The super-resolved ASTER data differentiated between clast-covered and clast-free areas within Survey Area 3, but failed to differentiate the shoreline subpixel area in Survey Area 2. The relatively clast-rich subpixel in Survey Area 2 would be considered a clast-free area in Survey Area 3. This provides some constraints to the amount of change necessary to differentiate these surfaces, with an apparent requirement for a difference of 20% abundance to be noticeable in the super-resolved ASTER data.

Neither the original nor super-resolved ASTER data detected the clay compositional differences seen in the AVIRIS data and laboratory spectra of transect samples. In both the AVIRIS and laboratory data, there is a change of composition moving inward from the playa edge. Both of these data are hyperspectral, and detecting the surface composition of a small area. Super-resolved ASTER data, with the same spatial resolution as the AVIRIS instrument, did not detect these differences in any of the three spectral regions, indicating that the requirement for this detection may be the number of bands. With an increase in the number of bands and a decrease in the width of these bands, more subtle spectral features can be detected. The ASTER instrument may not possess sufficient spectral resolution to detect these differences.

The super-resolution process was tested with data from this site, and confirmed with fieldwork. No surprises were found in this confirmation process. As this process has been used for two different planets, processing signals from two different satellites, and run on multiple types of environments, the super-resolution process used within this work can be considered robust and a useful tool for future studies.

APPENDIX A

SUPER-RESOLUTION CODE

A.1 SUPERRES_S_T.PRO

```
;Topher Hughes, spring semester 2006
;implement algorithm from Tonooka super-resolution paper
;first cut - do subset of ASTER imagery
;===================================
;===================================
;STEP 0:
;   open crosstalk corrected file
;   define region of interest
;   address / select region of interest in all 14 bands
;   calculate per-band StdDev in region of interest
;   calculate size of image @ 15, 30, and 90m
;band - total band image
;band1, band2, band3N, band4,....,band14 - subsetted area of band - orig resln
;band1_S, band1_T,...,band4_V,band4_T,....,band14_V,band14_S - alt resolution
;band1_stddev, band2_stddev,...,band14_stddev - std dev of WHOLE band


ReadData, band1, band2, band3N, band4, band5, band6, band7, band8,$
        band9, band10, band11, band12, band13, band14, V_stddev,$
        S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
        vis_image, swir_image, tir_image, Orig_Variable_Value, Orig_Variable_Name
```

```
Orig_band1=band1
Orig_band2=band2
Orig_band3N=band3N
Orig_band4=band4
Orig_band5=band5
Orig_band6=band6
Orig_band7=band7
Orig_band8=band8
Orig_band9=band9
Orig_band10=band10
Orig_band11=band11
Orig_band12=band12
Orig_band13=band13
Orig_band14=band14

Orig_V_stddev=V_stddev
Orig_S_stddev=S_stddev
Orig_T_stddev=T_stddev

Orig_vis_image=vis_image
Orig_swir_image=swir_image
Orig_tir_image=tir_image

read, alpha, prompt="Alpha value to use for PSF? (ASTER: 0.06565, THEMIS: 0.1) "
Orig_Variable_Value[8]=alpha
Orig_Variable_Name[8]='alpha'

;;;==================VNIR ISODATA Variables================================;;;
read, MeansSize, prompt="Initial # of VIS clusters? "
Orig_Variable_Value[9]=MeansSize
Orig_Variable_Name[9]='MeansSize'

;need to use temp variables - IDL passes array members by value and not reference

Reason="VNIR Clusters"
```

```
;doing
Temp_BandMax=max(Orig_vis_image)

ISODAT_Step_One, Temp_ChangeLimit, Temp_MinMembers, Temp_MaxStdDev,$
        Temp_MinDistance, Temp_MaxPair, Temp_BandMax, Temp_Limit,          Reason

V_ChangeLimit=Temp_ChangeLimit
Orig_Variable_Value[11]=Temp_ChangeLimit
Orig_Variable_Name[11]="V_ChangeLimit"

V_MinMembers=Temp_MinMembers
Orig_Variable_Value[12]=Temp_MinMembers
Orig_Variable_Name[12]="V_MinMembers"

V_MaxStdDev=Temp_MaxStdDev
Orig_Variable_Value[13]=Temp_MaxStdDev
Orig_Variable_Name[13]="V_MaxStdDev"

V_MinDistance=Temp_MinDistance
Orig_Variable_Value[14]=Temp_MinDistance
Orig_Variable_Name[14]="V_MinDistance"

V_MaxPair=Temp_MaxPair
Orig_Variable_Value[15]=Temp_MaxPair
Orig_Variable_Name[15]="V_MaxPair"

BandMax_V=Temp_BandMax
Orig_Variable_Value[16]=Temp_BandMax
Orig_Variable_Name[16]="BandMax_V"

V_Limit=Temp_Limit
Orig_Variable_Value[17]=Temp_Limit
Orig_Variable_Name[17]="V_Limit"

;;;==================SWIR ISODATA Variables================================;;;
read, SWIR_MeansSize, prompt='Initial # of SWIR clusters per VNIR cluster?:'
Orig_Variable_Value[35]=SWIR_MeansSize
```

```
Orig_Variable_Name[35]="SWIR_MeansSize"

Reason="SWIR Clusters"
Temp_BandMax=max(Orig_swir_image)

ISODAT_Step_One, Temp_ChangeLimit, Temp_MinMembers, Temp_MaxStdDev,$
        Temp_MinDistance, Temp_MaxPair, Temp_BandMax, Temp_Limit, Reason

S_ChangeLimit=Temp_ChangeLimit
Orig_Variable_Value[18]=Temp_ChangeLimit
Orig_Variable_Name[18]="S_ChangeLimit"

S_MinMembers=Temp_MinMembers
Orig_Variable_Value[19]=Temp_MinMembers
Orig_Variable_Name[19]="S_MinMembers"

S_MaxStdDev=Temp_MaxStdDev
Orig_Variable_Value[20]=Temp_MaxStdDev
Orig_Variable_Name[20]="S_MaxStdDev"

S_MinDistance=Temp_MinDistance
Orig_Variable_Value[21]=Temp_MinDistance
Orig_Variable_Name[21]="S_MinDistance"

S_MaxPair=Temp_MaxPair
Orig_Variable_Value[22]=Temp_MaxPair
Orig_Variable_Name[22]="S_MaxPair"

BandMax_S=Temp_BandMax
Orig_Variable_Value[23]=Temp_BandMax
Orig_Variable_Name[23]="BandMax_S"

S_Limit=Temp_Limit
Orig_Variable_Value[24]=Temp_Limit
Orig_Variable_Name[24]="S_Limit"
```

```
;;;==================TIR ISODATA Variables================================;;;
read, TIR_MeansSize, prompt='Initial # of TIR clusters per VNIR/SWIR cluster?:'
Orig_Variable_Value[10]=TIR_MeansSize
Orig_Variable_Name[10]="TIR_MeansSize"

Reason="TIR Clusters"
Temp_BandMax=max(Orig_tir_image)

ISODAT_Step_One, Temp_ChangeLimit, Temp_MinMembers, Temp_MaxStdDev,$
        Temp_MinDistance, Temp_MaxPair, Temp_BandMax, Temp_Limit, Reason

T_ChangeLimit=Temp_ChangeLimit
Orig_Variable_Value[25]=Temp_ChangeLimit
Orig_Variable_Name[25]="T_ChangeLimit"

T_MinMembers=Temp_MinMembers
Orig_Variable_Value[26]=Temp_MinMembers
Orig_Variable_Name[26]="T_MinMembers"

T_MaxStdDev=Temp_MaxStdDev
Orig_Variable_Value[27]=Temp_MaxStdDev
Orig_Variable_Name[27]="T_MaxStdDev"

T_MinDistance=Temp_MinDistance
Orig_Variable_Value[28]=Temp_MinDistance
Orig_Variable_Name[28]="T_MinDistance"

T_MaxPair=Temp_MaxPair
Orig_Variable_Value[29]=Temp_MaxPair
Orig_Variable_Name[29]="T_MaxPair"

BandMax_T=Temp_BandMax
Orig_Variable_Value[30]=Temp_BandMax
Orig_Variable_Name[30]="BandMax_T"

T_Limit=Temp_Limit
Orig_Variable_Value[31]=Temp_Limit
```

```
Orig_Variable_Name[31]="T_Limit"

;;;==================Steps 4/5 and 9/10 variables================================;;;
read, V_T_Distance, prompt="How many pixels away at native resolution to search for matches (V/S or V/T)
[default is 10]: "
Orig_Variable_Value[32]=V_T_Distance
Orig_Variable_Name[32]="V_T_Distance"

read, V_S_T_Distance, prompt="How many pixels away at native resolution to search for matches (V/S/T) [default
is 10]: "
Orig_Variable_Value[33]=V_S_T_Distance
Orig_Variable_Name[33]="V_S_T_Distance"

read, Weight, prompt="Amount to weight VNIR vs SWIR? (0-1, 0.7 default) "
Orig_Variable_Value[34]=Weight
Orig_Variable_Name[34]="Weight"

basestring=' '
read, basestring, prompt='Enter filename base to which details will be appended: '
Orig_basestring=basestring

print, " 1. alpha                11. SWIR ChangeLimit   21. TIR MinDistance"
print, " 2. initial VIS clusters   12. SWIR MinMembers    22. TIR MaxPair"
print, " 3. inital TIR clusters    13. SWIR MaxStdDev     23. TIR BandMax"
print, " 4. VIS ChangeLimit        14. SWIR MinDistance   24. TIR Limit"
print, " 5. VIS MinMembers         15. SWIR MaxPair       25. V_T_Distance"
print, " 6. VIS MaxStdDev          16. SWIR BandMax       26. V_S_T Distance"
print, " 7. VIS MinDistance        17. SWIR Limit         27. Weight"
print, " 8. VIS MaxPair            18. TIR ChangeLimit    28. SWIR MeanSize"
print, " 9.  VIS BandMax           19. TIR MinMembers"
print, " 10. VIS Limit             20. TIR MaxStdDev"
print, "If not varying anything, choose any of the above, and enter final value equal to initial value"
read, Variable_To_Vary_Index, prompt="Index in Orig_Variable_xxx of variable to vary? "

;display name here - re-read if not correct
NotRightVariable=1
While NotRightVariable Do Begin
```

```
  ;offset for initial variables
  Variable_To_Vary_Index+=7
  print, "Variable to be varied is "+Orig_Variable_Name[Variable_To_Vary_Index]
  IsRight=' '
  read, IsRight, prompt="Is this correct? (y/n) "
  if IsRight eq 'y' or IsRight eq 'Y' then NotRightVariable=0 $
    else read, Variable_To_Vary_Index, prompt="Index in Orig_Variable_xxx of variable to vary? "
EndWhile; NotRightVariable

Orig_Variable_Value[99]=Variable_To_Vary_Index
Orig_Variable_Name[99]="Variable_To_Vary_Index"

read, Variable_To_Vary_FinalValue, prompt="Final value of variable? "

Orig_Variable_Value[98]=Variable_To_Vary_FinalValue
Orig_Variable_Name[98]="Variable_To_Vary_FinalValue"

read, Variable_To_Vary_StepSize, prompt="Vary by how much in each step (NEVER enter 0 - even if not
varying)? "
if not(Variable_To_Vary_StepSize) then Variable_To_Vary_StepSize=1

Orig_Variable_Value[97]=Variable_To_Vary_StepSize
Orig_Variable_Name[97]="Variable_To_Vary_StepSize"

read, Variable_To_Vary_StepCount, prompt="How many loops per step? "

Orig_Variable_Value[96]=Variable_To_Vary_StepCount
Orig_Variable_Name[96]="Variable_To_Vary_StepCount"

num_loops=Variable_To_Vary_FinalValue-Orig_Variable_Value[Variable_To_Vary_Index]
num_loops/=Variable_To_Vary_StepSize
num_loops*=Variable_To_Vary_StepCount
num_loops+=Variable_To_Vary_StepCount ;to get full count on final variable value

Orig_Variable_Value[95]=num_loops
Orig_Variable_Name[95]="num_loops"
```

```
;we always increase (or decrease if neg) by stepsize on loop 1. adjust
;for that
Orig_Variable_Value[Variable_To_Vary_Index]-=Variable_To_Vary_StepSize

;all user input variables must be initialized by this point - ready to begin looping
for loop_count=1, num_loops, 1 do begin
  ;before doing anything, set values to what they ought to be
  ;use this to increment / decrement any variable variables ( ;-)  ) too

  ;first, save our loop_count for later output
  Orig_Variable_Value[94]=loop_count
  Orig_Variable_Name[94]="loop_count"

  if (not((loop_count-1) mod Variable_To_Vary_StepCount)) then $ ;its time to iterate
      Orig_Variable_Value[Variable_To_Vary_Index]+=Variable_To_Vary_StepSize
   ReInitVariables, Orig_Variable_Value, alpha, MeansSize, TIR_MeansSize, V_ChangeLimit, V_MinMembers,$
            V_MaxStdDev, V_MinDistance, V_MaxPair, BandMax_V, V_Limit, S_ChangeLimit,$
            S_MinMembers, S_MaxStdDev, S_MinDistance, S_MaxPair, BandMax_S, S_Limit,$
            T_ChangeLimit, T_MinMembers, T_MaxStdDev, T_MinDistance, T_MaxPair, BandMax_T,$
            T_Limit, V_T_Distance, V_S_T_Distance, Weight, SWIR_MeansSize, Orig_basestring,$
            basestring, loop_count, Variable_To_Vary_Index, Orig_Variable_Name


  ;dont forget pass by value / pass by reference issues.


  ;====================================
  ;====================================
  ;STEP 1:
  ;
  ;
  ;
  ;
  ;degrade resolution of VNIR to 30M
  ;
  ; local vars through here - result, max_x, max_y
  ; result - stores size() result [1] = x, [2] = y
```

```
;first thing - reduce the size of my image by 2.0
mag_ratio=2.0

;and define my Point Spread Function to convolve with my image
PSF_S=CalcPSF(alpha)

;create my 30m resolution VNIR images now

band1_S=Degrade(band1, PSF_S, mag_ratio)
band2_S=Degrade(band2, PSF_S, mag_ratio)
band3N_S=Degrade(band3N, PSF_S, mag_ratio)

;====================================
;====================================
;STEP 2:
;
;
;
;

;generate homogeneous pixel map
;
; local vars through here -
; result - stores size() result [1] = x, [2] = y
; band1_stddev, band2_stddev, etc calc. in step 0 above

result = size(band1_S)

VNIR_30m_homog = fltarr(result[1],result[2]) ;zeroed out initially
mag_ratio=2.0

Homog_Pix_VNIR, band1, band2, band3N, V_stddev[0], mag_ratio, VNIR_30m_homog

Homog_30m=VNIR_30m_homog
Homog_30m*=255
```

```
;====================================
;====================================
;STEP 3:
;
;
;
;
;build the V/S tree, using ISODATA and mahalanobis distance
;
; local vars through here -
; result - stores result of size() function
; VNIR_30m - the total vnir image
; swir_image - the total swir image
; seed - array involved in PRNG - DO NOT MESS WITH THIS!!!!
; homog - array of indices of homogeneous pixels
; Means - the 1D list of cluster means for ISODATA

;create 2D array - each image is a column in final result
result=size(band1_S)
VNIR_30m=[[reform(band1_S,result[4],1)],$
        [reform(band2_S,result[4],1)],$
        [reform(band3N_S,result[4],1)]]
VNIR_30m=rotate(VNIR_30m,4)

swir_image=Orig_swir_image


;swir_image is of type BYTE. Convert to type Int (-32K to +32K) so
;that when we go to find diffs (for MahaDist) we don't overflow
;Not sure why program has worked without this one. 20 Sep 07
swir_image=fix(swir_image)

;get indices of homogeneous pixels
;in theory - VNIR_30m[*,homog] (or swir_image) is pixel values
;of a homogeneous pixel where homogeneous is based on VNIR pixels
homog=where(VNIR_30m_homog)
```

```
  ;Tonooka says 50 initial starters for VNIR - I'm going to end up
  ;with a list of roughly 50 indices in homog as a result.
  ;using -1 for blank values in next two lines
  result=size(homog, /n_elements)
  print, 'You have',result,' homogeneous VNIR pixels'
;   read, MeansSize, prompt='Initial number of VNIR clusters?:'
  ;increase MeansSize by 10%, so I can eliminate the closest
  MeansSize=round(Orig_Variable_Value[9] * 1.1)
  VNIRMeans=make_array(MeansSize,1,/long,value=-1L);sizex1 array, all equal -1

  ;calculate the Inverted Covariance Matrix for VNIR
  subimage=VNIR_30m[*,homog]
  ;make sure we have more than 1 pixel for covariance
  sizecheck=size(subimage)
  sizecheck=sizecheck[0]-1 ;[0] is 1 if 1, so sizecheck now 0 if 1

  if (sizecheck) then begin
    VNIR30mInvCovMat=GetInvCovMatrix(subimage) ;calc only for Homog Pix
  endif else begin
    VNIR30mInvCovMat=make_array(3,3,/float,value=1.0)
    print, "WARNING - FOUND ONLY A SINGLE HOMOGENEOUS VNIR PIXEL!!!!!!"
  endelse

  Distance=make_array(size(homog, /n_elements), /float, value=0.0)
  VNIRMeans=GetInitialMeans(homog,VNIR_30m,VNIRMeans,VNIR30mInvCovMat,Distance)

  ;now get rid of those extra 10% - get rid of ones closest to another
  result=size(VNIRMeans, /n_elements)
  numremove=round(result*10/float(11))
  numremove=result-numremove
  MeansVals=VNIR_30m[*,homog[VNIRMeans]]
  VNIRMeans=RemoveLowestMeans(VNIRMeans,numremove,MeansVals,VNIR30mInvCovMat)

  ;have my initial Means to work with - time to k-means sort all of
  ;these. I need to get back the modified means for each cluster and
```

```
;the cluster to which each pixel belongs. I think this means I need
;a PROcedure to get back more than one thing
;given the image array, all homogeneous pixels, the initial means,
;and Cluster - another 1-D array, same size as homog that will list
;the cluster to which the pixel at the common index of homog belongs
;ie - cluster[x] is cluster to which VNIR_30m[*,homog[x]] belongs


;VNIR_30m and homog shouldn't change
;
;Means will flip from a 1-D list of homog indices to a 3 column average
;of all pixels assigned to that cluster. Since we're dealing with DN here
;the means will also remain byte values
;
;Cluster will be defined as a 1-D list of same length as homog that can
;have a value between 0 and 49 - value is index of Means to which it
;is closest.
VNIRClusters=homog  ;ok, its cheap, but it does get me the same size
VNIRClusters[*]=-1  ;cheap too - set them all to -1

VNIRMeans=VNIR_30m[*,homog[VNIRMeans]]

ClusterChecksum=make_array(V_Limit*4, /ulong, value=0)
;why *4? we exceed Limit sometimes, and *4 seems a good safe limit

ISODATA, VNIRMeans, V_Limit, VNIR_30m, homog, VNIRClusters, VNIR30mInvCovMat,$
        V_ChangeLimit, V_MinMembers, V_NumMembers, V_MaxStdDev, V_MinDistance,$
        V_MaxPair, BandMax_V, ClusterChecksum

test=histogram(VNIRClusters, omin=ohmi, omax=ohma)
print, "post ISODATA omin omax and histogram of VNIRClusters is ",$
      ohmi, ohma, " and ", test


;====================================
;====================================
;STEP 3B:
```

```
;
;For each VNIR cluster, cluster associated SWIR pixels
;
;

;at this point, I have my 30m VNIR pixels clusters - for each VNIR cluster
;select all of the SWIR pixels, and cluster them.

;increase MeansSize by 10%, so I can eliminate the closest
SWIRMeansSize=round(Orig_Variable_Value[35] * 1.1)

;save them for posterity (and changes in the for loop)
TrueChangeLimit=S_ChangeLimit
TrueMinMembers=S_MinMembers
TrueMaxPair=S_MaxPair

;how many VNIR clusters are there?
NumVNIRClusters=max(VNIRClusters)
NumVNIRClusters+=1

;make an array to store start and stop array values for VNIR arrays
VNIRtoSWIR=make_array(NumVNIRClusters*2)

SWIRClusters=VNIRClusters
SWIRClusters[*]=-1 ;same cheap trick used to create VNIRClusters

;ok - giving up on doing this mathematically properly for now, and seeing
;if I can get through this with just a single inverse covariance matrix for
;each band. It should really be on a per cluster basis, but I am
;running into too many problems with trying to invert the covariance
;matrix - I'm getting a singular matrix (determinant before invert
;is 0) too often. As a result, the invert isn't reliable. This really
;stinks. 27 Feb 07.

;this is bad if we only have a few homogenous pixels. Of couse, if we
;only have a few homogeneous pixels, we have bigger issues.
```

```
SWIR30mInvCovMat=GetInvCovMatrix(swir_image[*,homog])

ohma=-1.0

;for each cluster, calculate the initial means, and then ISODATA it
for VNIRclus=0, NumVNIRClusters-1, 1 do begin

  ;need to reset these after each iteration
  S_ChangeLimit=TrueChangeLimit
  S_MinMembers=TrueMinMembers
  S_MaxPair=TrueMaxPair

  ThisSWIRMeans=make_array(SWIRMeansSize,1,/long,value=-1L)
  SWIRhomog=where(VNIRClusters eq VNIRclus)

  ;want to keep the indices so I can reassign to them later
  ThisSWIRIndex=SWIRhomog

  ;actual pixel values from homog for this VNIR cluster
  SWIRhomog=homog[SWIRhomog]

  ;cheap way to get a cluster list of the right size
  ThisSWIRCluster=SWIRhomog

  ;define MinMembers, ChangeLimit, and NumMembers per cluster
  NumMembers=size(SWIRhomog, /n_elements)

  if ( size(ThisSWIRMeans,/n_elements) ge NumMembers ) then begin
    ThisSWIRMeans=indgen(NumMembers)
  endif else begin
    Distance=make_array(size(homog, /n_elements), /float, value=0.0)
    ThisSWIRMeans=GetInitialMeans(SWIRhomog,swir_image,ThisSWIRMeans,$
          SWIR30mInvCovMat,Distance)
    ;now get rid of those extra 10% - get rid of ones closest to another
    result=size(ThisSWIRMeans, /n_elements)
    numremove=round(result*10/float(11))
    numremove=result-numremove
```

```
  SWIRMeansVals=swir_image[*,SWIRhomog[ThisSWIRMeans]]
  ThisSWIRMeans=RemoveLowestMeans(ThisSWIRMeans,numremove,SWIRMeansVals,$
                SWIR30mInvCovMat)
endelse; ThisSWIRMeans elements vs NumMembers

ThisSWIRMeans=swir_image[*,SWIRhomog[ThisSWIRMeans]]

;ok, now I can call ISODATA for these
ClusterChecksum=make_array(S_Limit*4, /ulong, value=0)

ISODATA, ThisSWIRMeans, S_Limit, swir_image, SWIRhomog, ThisSWIRCluster,$
       SWIR30mInvCovMat, S_ChangeLimit, S_MinMembers, NumMembers,$
       S_MaxStdDev, S_MinDistance, S_MaxPair, BandMax_S, ClusterChecksum

;ok, now add ThisSWIRCluster list to SWIRClusters and mark the sizes
;in VNIRtoSWIR so I can index it. To do this:
;first, modify the cluster values with the previous ohma so that
;they don't conflict with results from other VNIR clusters
ThisSWIRCluster+=(ohma+1)

;take a histogram of ThisSWIRCluster and set VNIRtoSWIR[VNIRclus*2]
;to ohmi and VNIRtoSWIR[(VNIRClus*2)+1] to ohma
ThisSWIRhisto=histogram(ThisSWIRCluster,omin=ohmi,omax=ohma,$
              reverse_indices=ri)

VNIRtoSWIR[(VNIRclus*2)]=ohmi
VNIRtoSWIR[(VNIRclus*2)+1]=ohma

;finally, copy cluster values from ThisSWIRCluster to SWIRClusters
SWIRClusters[ThisSWIRIndex]=ThisSWIRCluster

;end by recording our Means permanant-like
if (~ohmi) then begin
  SWIRMEANS=ThisSWIRMeans
endif else begin
  SWIRMeans=[[SWIRMeans],[ThisSWIRMeans]]
endelse
```

```
endfor; VNIRclus=0

print, "Final VNIRtoSWIR is ", VNIRtoSWIR

;====================================
;====================================
;STEP 4:
;
;The super-resolved images of the six SWIR bands are generated by
;allocating the most likely SWIR spectrum to each 15m-resolution pixel
;based on spectral similarity in VNIR.

;I hadn't bothered building a vis_image image yet - do it now, as I'll
;need it for this step

mag_ratio=2.0

vis_image=Orig_vis_image

SWIR_15m=SuperRes_OneBand(VNIRMeans,SWIRMeans,$
                VNIRtoSWIR,homog,vis_image,VNIR_30m,swir_image,$
                VNIR30mInvCovMat,SWIR30mInvCovMat,$
                X_15m,Y_15m,mag_ratio, V_T_Distance)


;====================================
;====================================
;STEP 5:
;
;The super-resolved image (SWIR_15m) may not be radiometrically accurate,
;so it needs to be modified. This modification is weighted by the distance
;between the VNIR spectra of the 15m pixel and VNIR spectra of the SWIR
;spectra assigned to the 15m pixel (whether from tree or image)

SWIR_15m=ModifySuperRes(SWIR_15m, swir_image, PSF_S, X_15m, Y_15m, mag_ratio)
```

```
;at this point SWIR 0-5 are the images, 6 is the distance map, and 7 is
;the source map (I think). I probably need to add bands 8-13 for the
;correction maps, so that I can save those as well.  26 Oct 2006

;zero out my border pixels - top row, bottom row, first column, last column
;top row

SWIR_15m=ZeroLines(SWIR_15m, X_15m, Y_15m)

;====================================
;====================================
;STEP 6:
;
;having super-resolved the SWIR bands, it is now time to super-resolve
;the TIR bands. "This step is almost the same with step 1. The degraded
;VNIR and SWIR images with a 90m resolution are derived...."

;define my Point Spread Function to convolve with my image
PSF_T=CalcPSF(alpha)

;VNIR 15 to 90 m resolution switch first

;reduce the size of my image by 6.0 for VNIR
mag_ratio=6.0

;now actually degrade my resolution
band1_T=Degrade(band1, PSF_T, mag_ratio)
band2_T=Degrade(band2, PSF_T, mag_ratio)
band3N_T=Degrade(band3N, PSF_T, mag_ratio)

result=size(band1_T)

VNIR_90m=[[reform(band1_T, result[4], 1)],$
        [reform(band2_T, result[4], 1)],$
        [reform(band3N_T, result[4], 1)]]
VNIR_90m=rotate(VNIR_90m,4)
```

```
;SWIR 30 to 90 m resolution switch first

;reduce the size of my image by 3.0 for VNIR
mag_ratio=3.0

;now actually degrade my resolution
band4_T=Degrade(band4, PSF_T, mag_ratio)
band5_T=Degrade(band5, PSF_T, mag_ratio)
band6_T=Degrade(band6, PSF_T, mag_ratio)
band7_T=Degrade(band7, PSF_T, mag_ratio)
band8_T=Degrade(band8, PSF_T, mag_ratio)
band9_T=Degrade(band9, PSF_T, mag_ratio)

result=size(band4_T)

SWIR_90m=[[reform(band4_T, result[4], 1)],$
        [reform(band5_T, result[4], 1)],$
        [reform(band6_T, result[4], 1)],$
        [reform(band7_T, result[4], 1)],$
        [reform(band8_T, result[4], 1)],$
        [reform(band9_T, result[4], 1)]]
SWIR_90m=rotate(SWIR_90m,4)

;====================================
;====================================
;STEP 7:
;
;"This step is almost the same with the step 2.  A homogeneous pixel is
; here defined as a 90m-resolution pixel that each of the standard
; deviations of 36 original VNIR pixels and 9 original SWIR pixels in a
; 90m-resolution pixel is smaller than each threshold of VNIR and SWIR.
; Using the original VNIR and SWIR images, the 90m-resolution homogeneous
; pixel map is generated."

;
```

```
;generate homogeneous pixel map
;

;do VNIR first
result = size(band1_T)

VNIR_90m_homog = fltarr(result[1],result[2]) ;zeroed out initially
mag_ratio=6.0

Homog_Pix_VNIR, band1, band2, band3N, V_stddev[0], mag_ratio, VNIR_90m_homog

;now do SWIR
;result is size of 90m array - same for VNIR, SWIR, or TIR

SWIR_90m_homog = fltarr(result[1],result[2]) ;zeroed out initially
mag_ratio=3.0

Homog_Pix_SWIR, band4, band5, band6, band7, band8, band9, S_stddev[0],$
        mag_ratio, X_30m, Y_30m, SWIR_90m_homog



Homog_90m=floor((VNIR_90m_homog + SWIR_90m_homog) / 2.0)
Homog_90m*=255

;====================================
;====================================
;STEP 8:
;
;
;
;
;build the V/S/T tree, using ISODATA and mahalanobis distance
;
; local vars through here -
; result - stores result of size() function
; VNIR_90m - the total vnir image
; SWIR_90m - the total swir image
```

```
; seed - array involved in PRNG - DO NOT MESS WITH THIS!!!!
; homog - array of indices of homogeneous pixels
; Means - the 1D list of cluster means for ISODATA

;get indices of homogeneous pixels
;in theory - Homog_90m is pixel values of a homogeneous pixel
;where homogeneous is based on VNIR and SWIR pixels
homog=where(Homog_90m)



;Tonooka says 50 initial starters for VNIR - I'm going to end up
;with a list of roughly 50 indices in homog as a result.
;using -1 for blank values in next two lines
result=size(homog, /n_elements)
print, 'You have',result,' homogeneous VNIR/SWIR 90m pixels'
;   read, MeansSize, prompt='Initial number of VNIR clusters?:'
;increase MeansSize by 10%, so I can eliminate the closest

MeansSize=round(Orig_Variable_Value[9] * 1.1)
VNIRMeans=make_array(MeansSize,1,/long,value=-1L);sizex1 array, all equal -1

;calculate the Inverted Covariance Matrix for VNIR
subimage=VNIR_90m[*,homog]
;make sure we have more than 1 pixel for covariance
sizecheck=size(subimage)
sizecheck=sizecheck[0]-1 ;[0] is 1 if 1, so sizecheck now 0 if 1

if (sizecheck) then begin
  VNIR90mInvCovMat=GetInvCovMatrix(subimage) ;calc only for Homog Pix
endif else begin
  VNIR90mInvCovMat=make_array(3,3,/float,value=1.0)
  print, "ONLY ONE 90m HOMOGENEOUS VNIR PIXEL?!?!?!?"
endelse

Distance=make_array(size(homog, /n_elements), /float, value=0.0)
VNIRMeans=GetInitialMeans(homog,VNIR_90m,VNIRMeans,VNIR90mInvCovMat,Distance)
```

```
;now get rid of those extra 10% - get rid of ones closest to another
result=size(VNIRMeans, /n_elements)
numremove=round(result*10/float(11))
numremove=result-numremove
MeansVals=VNIR_90m[*,homog[VNIRMeans]]
VNIRMeans=RemoveLowestMeans(VNIRMeans,numremove,MeansVals,VNIR90mInvCovMat)

;have my initial Means to work with - time to ISODATA sort all of
;these. I need to get back the modified means for each cluster and
;the cluster to which each pixel belongs. I think this means I need
;a PROcedure to get back more than one thing
;given the image array, all homogeneous pixels, the initial means,
;and Cluster - another 1-D array, same size as homog that will list
;the cluster to which the pixel at the common index of homog belongs
;ie - cluster[x] is cluster to which VNIR_90m[*,homog[x]] belongs


;VNIR_90m and homog shouldn't change
;
;Means will flip from a 1-D list of homog indices to a 3 column average
;of all pixels assigned to that cluster. Since we're dealing with DN here
;the means will also remain byte values
;
;Cluster will be defined as a 1-D list of same length as homog that can
;have a value between 0 and max cluster less 1 - value is index of Means
;to which it is closest.
VNIRClusters=homog  ;ok, its cheap, but it does get me the same size
VNIRClusters[*]=-1  ;cheap too - set them all to -1

VNIRMeans=VNIR_90m[*,homog[VNIRMeans]]

NumMembers=size(homog, /n_elements)

ClusterChecksum=make_array(V_Limit*4, /ulong, value=0)


ISODATA, VNIRMeans, V_Limit, VNIR_90m, homog, VNIRClusters, VNIR90mInvCovMat,$
```

```
    V_ChangeLimit, V_MinMembers, V_NumMembers, V_MaxStdDev, V_MinDistance,$
    V_MaxPair, BandMax_V, ClusterChecksum

test=histogram(VNIRClusters, omin=ohmi, omax=ohma)
print, "post ISODATA omin omax and histogram of VNIRClusters is ", ohmi, ohma, " and ", test

;====================================
;====================================
;STEP 8B:
;
;For each VNIR cluster, cluster associated SWIR pixels
;
;
;at this point, I have my 90m VNIR pixels clusters - for each VNIR cluster
;select all of the SWIR pixels, and cluster them.

;increase MeansSize by 10%, so I can eliminate the closest
SWIRMeansSize=round(Orig_Variable_Value[35] * 1.1)

;save them for posterity (and changes in the for loop)
TrueChangeLimit=S_ChangeLimit
TrueMinMembers=S_MinMembers
TrueMaxPair=S_MaxPair

;how many VNIR clusters are there?
NumVNIRClusters=max(VNIRClusters)
NumVNIRClusters+=1

;make an array to store start and stop array values for VNIR arrays
VNIRtoSWIR=make_array(NumVNIRClusters*2)

SWIRClusters=VNIRClusters
SWIRClusters[*]=-1 ;same cheap trick used to create VNIRClusters

;ok - giving up on doing this properly for now, and seeing if I can
;get through this with just a single inverse covariance matrix for
;each band. It should really be on a per cluster basis, but I am
```

```
;running into too many problems with trying to invert the covariance
;matrix - I'm getting a singular matrix (determinant before invert
;is 0) too often. As a result, the invert isn't reliable. This really
;stinks. 27 Feb 07.

;this is bad if we only have a few homogenous pixels. Of course, if we
;only have a few homogeneous pixels, we have bigger issues.

SWIR90mInvCovMat=GetInvCovMatrix(SWIR_90m[*,homog])

ohma=-1.0

;for each cluster, calculate the initial means, and then ISODATA it
for VNIRclus=0, NumVNIRClusters-1, 1 do begin

  ;need to reset these after each iteration
  S_ChangeLimit=TrueChangeLimit
  S_MinMembers=TrueMinMembers
  S_MaxPair=TrueMaxPair

  ThisSWIRMeans=make_array(SWIRMeansSize,1,/long,value=-1L)
  SWIRhomog=where(VNIRClusters eq VNIRclus)

  ;want to keep the indices so I can reassign to them later
  ThisSWIRIndex=SWIRhomog

  ;actual pixel values from homog for this VNIR cluster
  SWIRhomog=homog[SWIRhomog]

  ;cheap way to get a cluster list of the right size
  ThisSWIRCluster=SWIRhomog

  ;define MinMembers, ChangeLimit, and NumMembers per cluster
  NumMembers=size(SWIRhomog, /n_elements)

  if ( size(ThisSWIRMeans,/n_elements) ge NumMembers ) then begin
    ThisSWIRMeans=indgen(NumMembers)
```

```
    endif else begin
      Distance=make_array(size(SWIRhomog, /n_elements), /float, value=0.0)
      ThisSWIRMeans=GetInitialMeans(SWIRhomog,SWIR_90m,ThisSWIRMeans,$
              SWIR30mInvCovMat,Distance)
;      ThisSWIRMeans=CalcMahaDist(SWIRhomog,SWIR_90m,ThisSWIRMeans,$
;                          SWIR90mInvCovMat)

      ;now get rid of those extra 10% - get rid of ones closest to another
      result=size(ThisSWIRMeans, /n_elements)
      numremove=round(result*10/float(11))
      numremove=result-numremove
      SWIRMeansVals=SWIR_90m[*,SWIRhomog[ThisSWIRMeans]]
      ThisSWIRMeans=RemoveLowestMeans(ThisSWIRMeans,numremove,$
                          SWIRMeansVals,SWIR90mInvCovMat)
    endelse; ThisSWIRMeans elements vs NumMembers

    ThisSWIRMeans=SWIR_90m[*,SWIRhomog[ThisSWIRMeans]]

    ;ok, now I can call ISODATA for these

    ClusterChecksum=make_array(S_Limit*4, /ulong, value=0)

    ISODATA, ThisSWIRMeans, S_Limit, SWIR_90m, SWIRhomog, ThisSWIRCluster,$
        SWIR90mInvCovMat, S_ChangeLimit, S_MinMembers, S_NumMembers,$
        S_MaxStdDev, S_MinDistance, S_MaxPair, BandMax_S, ClusterChecksum

    ;ok, now add ThisSWIRCluster list to SWIRClusters and mark the sizes
    ;in VNIRtoSWIR so I can index it. To do this:

    ;first, modify the cluster values with the previous ohma so that
    ;they don't conflict with results from other VNIR clusters
    ThisSWIRCluster+=(ohma+1)

    ;take a histogram of ThisSWIRCluster and set VNIRtoSWIR[VNIRclus*2]
    ;to ohmi and VNIRtoSWIR[(VNIRClus*2)+1] to ohma
    ThisSWIRhisto=histogram(ThisSWIRCluster,omin=ohmi,omax=ohma,$
                reverse_indices=ri)
```

```
  VNIRtoSWIR[(VNIRclus*2)]=ohmi
  VNIRtoSWIR[(VNIRclus*2)+1]=ohma

  ;finally, copy cluster values from ThisSWIRCluster to SWIRClusters
  SWIRClusters[ThisSWIRIndex]=ThisSWIRCluster

  ;end by recording our Means permanant-like
  if (~ohmi) then begin
    SWIRMEANS=ThisSWIRMeans
  endif else begin
    SWIRMeans=[[SWIRMeans],[ThisSWIRMeans]]
  endelse

endfor; VNIRclus=0

;====================================
;====================================
;STEP 8C:
;
;For each SWIR cluster, cluster associated TIR pixels
;
;
;create my TIR array

tir_image=Orig_tir_image

;at this point, I have my 90m SWIR pixels clusters - for each SWIR cluster
;select all of the TIR pixels, and cluster them.

;first, find out how many Means we want per cluster


;increase MeansSize by 10%, so I can eliminate the closest
TIRMeansSize=round(Orig_Variable_Value[10] * 1.1)

;save them for posterity (and changes in the for loop)
```

```
TrueChangeLimit=T_ChangeLimit
TrueMinMembers=T_MinMembers
TrueMaxPair=T_MaxPair

;how many SWIR clusters are there?
NumSWIRClusters=max(SWIRClusters)
NumSWIRClusters+=1

;make an array to store start and stop array values for SWIR arrays
SWIRtoTIR=make_array(NumSWIRClusters*2)

TIRClusters=SWIRClusters
TIRClusters[*]=-1 ;same cheap trick used to create SWIRClusters

;ok - giving up on doing this properly for now, and seeing if I can
;get through this with just a single inverse covariance matrix for
;each band. It should really be on a per cluster basis, but I am
;running into too many problems with trying to invert the covariance
;matrix - I'm getting a singular matrix (determinant before invert
;is 0) too often. As a result, the invert isn't reliable. This really
;stinks. 27 Feb 07.

;this is bad if we only have a few homogenous pixels. Of couse, if we
;only have a few homogeneous pixels, we have bigger issues.

TIR90mInvCovMat=GetInvCovMatrix(tir_image[*,homog])

ohma=-1.0

;for each cluster, calculate the initial means, and then ISODATA it
for SWIRclus=0, NumSWIRClusters-1, 1 do begin

  ;need to reset these after each iteration
  T_ChangeLimit=TrueChangeLimit
  T_MinMembers=TrueMinMembers
  T_MaxPair=TrueMaxPair
```

```
ThisTIRMeans=make_array(TIRMeansSize,1,/long,value=-1L)
TIRhomog=where(SWIRClusters eq SWIRclus)

;want to keep the indices so I can reassign to them later
ThisTIRIndex=TIRhomog

;actual pixel values from homog for this SWIR cluster
TIRhomog=homog[TIRhomog]

;cheap way to get a cluster list of the right size
ThisTIRCluster=TIRhomog

;define MinMembers, ChangeLimit, and NumMembers per cluster
NumMembers=size(TIRhomog, /n_elements)


if ( size(ThisTIRMeans,/n_elements) ge NumMembers ) then begin
  ThisTIRMeans=indgen(NumMembers)
endif else begin
  Distance=make_array(size(TIRhomog, /n_elements), /float, value=0.0)
  ThisTIRMeans=GetInitialMeans(TIRhomog,tir_image,ThisTIRMeans,$
            TIR90mInvCovMat,Distance)

  ;now get rid of those extra 10% - get rid of ones closest to another
  result=size(ThisTIRMeans, /n_elements)
  numremove=round(result*10/float(11))
  numremove=result-numremove
  TIRMeansVals=tir_image[*,TIRhomog[ThisTIRMeans]]
  ThisTIRMeans=RemoveLowestMeans(ThisTIRMeans,numremove,TIRMeansVals,$
                        TIR90mInvCovMat)
endelse; ThisTIRMeans elements vs NumMembers

ThisTIRMeans=tir_image[*,TIRhomog[ThisTIRMeans]]

;ok, now I can call ISODATA for these

ClusterChecksum=make_array(T_Limit*4, /ulong, value=0)
```

```
  ISODATA, ThisTIRMeans, T_Limit, tir_image, TIRhomog, ThisTIRCluster,$
        TIR90mInvCovMat, T_ChangeLimit, T_MinMembers, T_NumMembers,$
        T_MaxStdDev, T_MinDistance, T_MaxPair, BandMax_T, ClusterChecksum

  ;ok, now add ThisTIRCluster list to TIRClusters and mark the sizes
  ;in SWIRtoTIR so I can index it. To do this:

  ;first, modify the cluster values with the previous ohma so that
  ;they don't conflict with results from other SWIR clusters
  ThisTIRCluster+=(ohma+1)

  ;take a histogram of ThisTIRCluster and set SWIRtoTIR[SWIRclus*2]
  ;to ohmi and SWIRtoTIR[(SWIRClus*2)+1] to ohma
  ThisTIRhisto=histogram(ThisTIRCluster,omin=ohmi,omax=ohma,$
                reverse_indices=ri)

  SWIRtoTIR[(SWIRclus*2)]=ohmi
  SWIRtoTIR[(SWIRclus*2)+1]=ohma

  ;finally, copy cluster values from ThisTIRCluster to TIRClusters
  TIRClusters[ThisTIRIndex]=ThisTIRCluster

  ;end by recording our Means permanent-like
  if (~ohmi) then begin
    TIRMEANS=ThisTIRMeans
  endif else begin
    TIRMeans=[[TIRMeans],[ThisTIRMeans]]
  endelse

endfor; SWIRclus=0

print, "Final SWIRtoTIR is ", SWIRtoTIR

;====================================
;====================================
;STEP 9:
```

```
;
;The super-resolved images of the five TIR bands are generated by
;allocating the most likely TIR spectrum to each 15m-resolution pixel
;based on spectral similarity in VNIR and SWIR.

mag_ratio=6.0

TIR_15m=SuperRes_TwoBands(VNIRMeans,SWIRMeans,TIRMeans,$
                VNIRClusters,SWIRClusters,TIRClusters,$
                VNIRtoSWIR,SWIRtoTIR,homog,$
                vis_image,VNIR_90m,SWIR_15m,SWIR_90m,tir_image,$
                VNIR90mInvCovMat,SWIR90mInvCovMat,TIR90mInvCovMat,$
                X_15m,Y_15m, mag_ratio, V_S_T_Distance, Weight)



;====================================
;====================================
;STEP 10:
;
;The super-resolved image (TIR_15m) may not be radiometrically accurate,
;so it needs to be modified. This modification is weighted by the distance
;between the VN/SWIR spectra of the 15m pixel and VN/SWIR spectra of the TIR
;spectra assigned to the 15m pixel (whether from tree or image)

;just in case its not set right
mag_ratio=6.0

TIR_15m=ModifySuperRes_T(TIR_15m, tir_image, PSF_T, X_15m, Y_15m, mag_ratio)

;at this point TIR 0-4 are the images, 5 is the distance map, 6 is
;the source map, and 7 - 11 are the correction maps. 8/15/07

;zero out my border pixels - top row, bottom row, first column, last column
;top row

TIR_15m=ZeroLines(TIR_15m, X_15m, Y_15m)
```

```
  ;create our calibrated radiance file
  ;
  ;          HARD CODING BELOW
  ;          HARD CODING BELOW
  ;          HARD CODING BELOW
  ;
  TIR_calib_rad=CalibrateTIRRadiance(TIR_15m[0:4,*])
  SWIR_calib_rad=CalibrateSWIRRadiance(SWIR_15m[0:5,*])

  ;add our calibrated radiances to our _15m files
  TIR_15m=[TIR_15m,TIR_calib_rad]
  SWIR_15m=[SWIR_15m,SWIR_calib_rad]

  ;Save out our data to files with a common basename
  WriteOutASTERData, tir_15m, homog_90m, swir_15m, homog_30m, basestring, Orig_Variable_Value,
Orig_Variable_Name

endfor; major loop for repeated runs

  print, "Program Finished."

end
```

A.2 SUPERRES_THEMIS.PRO

```
;====================================
;====================================
;STEP 0:
;   open crosstalk corrected file
;   define region of interest
;   address / select region of interest in all 14 bands
;   initially - print value of upper left and lower right corners

;ok - start with initialization
;these variables are first defined in step 0
```

```
;V=VNIR S=SWIR T=TIR UL=Upper Left LR=Lower Right

;other variables used
;band - total band image
;band1, band2, band3N, band4,....,band14 - subsetted area of band - orig resln
;band1_S, band1_T,...,band4_V,band4_T,....,band14_V,band14_S - alt resolution
;band1_v, band2_stddev,...,band14_stddev - std dev of WHOLE band

;====================================
;====================================
;STEP 0:
;
;
;   THEMIS step 0
;
;open file, calculate standard deviations, etc

print, "Welcome to the super-res program for THEMIS"

ReadThemisData, Vband1, Vband2, Vband3, Vband4, Vband5, V_pix_size,$
            V_numbands, X_vis, Y_vis, vis_image, Tband1, Tband2,$
            Tband3, Tband4, Tband5, Tband6, Tband7, Tband8, Tband9,$
            Tband10, T_pix_size, T_numbands, X_tir, Y_tir, tir_image,$
            V_nonzeroes, T_nonzeroes, V_stddev, T_stddev,$
            Orig_Variable_Value, Orig_Variable_Name

;save all those values out to variables that I know won't change in the loop
;these are arrays, etc - things that don't fit the Orig_Variable_Value / Name model
;these are also all things that won't be iterated during looping (ie, non-variables)

Orig_Vband1=Vband1
Orig_Vband2=Vband2
Orig_Vband3=Vband3
Orig_Vband4=Vband4
Orig_Vband5=Vband5
Orig_vis_image=vis_image
Orig_Tband1=Tband1
```

```
Orig_Tband2=Tband2
Orig_Tband3=Tband3
Orig_Tband4=Tband4
Orig_Tband5=Tband5
Orig_Tband6=Tband6
Orig_Tband7=Tband7
Orig_Tband8=Tband8
Orig_Tband9=Tband9
Orig_Tband10=Tband10
Orig_tir_image=tir_image
Orig_V_nonzeroes=V_nonzeroes
Orig_T_nonzeroes=T_nonzeroes
Orig_V_stddev=V_stddev
Orig_T_stddev=T_stddev

read, alpha, prompt="Alpha value to use for PSF? (ASTER: 0.06565, THEMIS: 0.1) "
Orig_Variable_Value[8]=alpha
Orig_Variable_Name[8]='alpha'

read, MeansSize, prompt="Initial # of VIS clusters? "
Orig_Variable_Value[9]=MeansSize
Orig_Variable_Name[9]='MeansSize'

;need to use temp variables - IDL passes array members by value and not reference
Reason="VNIR Clusters"
;doing
Temp_BandMax=max(Orig_vis_image)

ISODAT_Step_One, Temp_ChangeLimit, Temp_MinMembers, Temp_MaxStdDev,$
        Temp_MinDistance, Temp_MaxPair, Temp_BandMax, Temp_Limit, Reason

V_ChangeLimit=Temp_ChangeLimit
Orig_Variable_Value[11]=Temp_ChangeLimit
Orig_Variable_Name[11]="V_ChangeLimit"

V_MinMembers=Temp_MinMembers
Orig_Variable_Value[12]=Temp_MinMembers
```

```
Orig_Variable_Name[12]="V_MinMembers"

V_MaxStdDev=Temp_MaxStdDev
Orig_Variable_Value[13]=Temp_MaxStdDev
Orig_Variable_Name[13]="V_MaxStdDev"

V_MinDistance=Temp_MinDistance
Orig_Variable_Value[14]=Temp_MinDistance
Orig_Variable_Name[14]="V_MinDistance"


V_MaxPair=Temp_MaxPair
Orig_Variable_Value[15]=Temp_MaxPair
Orig_Variable_Name[15]="V_MaxPair"

BandMax_V=Temp_BandMax
Orig_Variable_Value[16]=Temp_BandMax
Orig_Variable_Name[16]="BandMax_V"

V_Limit=Temp_Limit
Orig_Variable_Value[17]=Temp_Limit

Orig_Variable_Name[17]="V_Limit"

read, TIR_MeansSize, prompt='Initial # of TIR clusters per VNIR cluster?:'
Orig_Variable_Value[10]=TIR_MeansSize
Orig_Variable_Name[10]="TIR_MeansSize"

Reason="TIR Clusters"
Temp_BandMax=max(Orig_tir_image)

ISODAT_Step_One, Temp_ChangeLimit, Temp_MinMembers, Temp_MaxStdDev,$
        Temp_MinDistance, Temp_MaxPair, Temp_BandMax, Temp_Limit, Reason

T_ChangeLimit=Temp_ChangeLimit
Orig_Variable_Value[25]=Temp_ChangeLimit
Orig_Variable_Name[25]="T_ChangeLimit"
```

```
T_MinMembers=Temp_MinMembers
Orig_Variable_Value[26]=Temp_MinMembers
Orig_Variable_Name[26]="T_MinMembers"

T_MaxStdDev=Temp_MaxStdDev
Orig_Variable_Value[27]=Temp_MaxStdDev
Orig_Variable_Name[27]="T_MaxStdDev"

T_MinDistance=Temp_MinDistance
Orig_Variable_Value[28]=Temp_MinDistance
Orig_Variable_Name[28]="T_MinDistance"

T_MaxPair=Temp_MaxPair
Orig_Variable_Value[29]=Temp_MaxPair
Orig_Variable_Name[29]="T_MaxPair"

BandMax_T=Temp_BandMax
Orig_Variable_Value[30]=Temp_BandMax
Orig_Variable_Name[30]="BandMax_T"

T_Limit=Temp_Limit
Orig_Variable_Value[31]=Temp_Limit
Orig_Variable_Name[31]="T_Limit"

read, V_T_Distance, prompt="How many pixels away at native resolution to search for matches [default is 10]: "
Orig_Variable_Value[32]=V_T_Distance
Orig_Variable_Name[32]="V_T_Distance"


basestring=' '
read, basestring, prompt='Enter filename base to which details will be appended: '
Orig_basestring=basestring

print, " 1. alpha                7. VIS MinDistance    13. TIR MaxStdDev"
print, " 2. initial VIS clusters   8. VIS MaxPair        14. TIR MinDistance"
print, " 3. inital TIR clusters    9. Vis BandMax        15. TIR MaxPair"
```

```
print, " 4. VIS ChangeLimit        10. VIS Limit         16. TIR BandMax"
print, " 5. VIS MinMembers         11. TIR ChangeLimit   17. TIR Limit"
print, " 6. VIS MaxStdDev          12. TIR MinMembers    18. Distance"
print, "If not varying anything, choose any of the above, and enter final value equal to initial value"
read, Variable_To_Vary_Index, prompt="Index in Orig_Variable_xxx of variable to vary? "

;display name here - re-read if not correct
NotRightVariable=1
While NotRightVariable Do Begin
  ;offset for initial variables and SWIR variables (real indices btwn 18 and 24)
  if (Variable_To_Vary_Index le 10) then Variable_To_Vary_Index+=7 else Variable_To_Vary_Index+=14
  print, "Variable to be varied is "+Orig_Variable_Name[Variable_To_Vary_Index]
  IsRight=' '
  read, IsRight, prompt="Is this correct? (y/n) "

  if IsRight eq 'y' or IsRight eq 'Y' then NotRightVariable=0 $
  else read, Variable_To_Vary_Index, prompt="Index in Orig_Variable_xxx of variable to vary? "
EndWhile; NotRightVariable

Orig_Variable_Value[99]=Variable_To_Vary_Index
Orig_Variable_Name[99]="Variable_To_Vary_Index"

read, Variable_To_Vary_FinalValue, prompt="Final value of variable? "

Orig_Variable_Value[98]=Variable_To_Vary_FinalValue
Orig_Variable_Name[98]="Variable_To_Vary_FinalValue"

read, Variable_To_Vary_StepSize, prompt="Vary by how much in each step (NEVER enter 0 - even if not
varying)? "
if not(Variable_To_Vary_StepSize) then Variable_To_Vary_StepSize=1

Orig_Variable_Value[97]=Variable_To_Vary_StepSize
Orig_Variable_Name[97]="Variable_To_Vary_StepSize"

read, Variable_To_Vary_StepCount, prompt="How many loops per step? "

Orig_Variable_Value[96]=Variable_To_Vary_StepCount
```

```
Orig_Variable_Name[96]="Variable_To_Vary_StepCount"

num_loops=Variable_To_Vary_FinalValue-Orig_Variable_Value[Variable_To_Vary_Index]
num_loops/=Variable_To_Vary_StepSize
num_loops*=Variable_To_Vary_StepCount
num_loops+=Variable_To_Vary_StepCount ;to get full count on final variable value

Orig_Variable_Value[95]=num_loops
Orig_Variable_Name[95]="num_loops"

;SWIR variables so stuff doesn't crash
S_ChangeLimit=1
S_MinMembers=1
S_MaxStdDev=1
S_MinDistance=1
S_MaxPair=1
BandMax_S=1
S_Limit=1
SWIR_MeansSize=1
Weight=1
V_S_T_Distance=1

;we always increase (or decrease if neg) by stepsize on loop 1. adjust
;for that

Orig_Variable_Value[Variable_To_Vary_Index]-=Variable_To_Vary_StepSize

;all user input variables must be initialized by this point - ready to begin looping
for loop_count=1, num_loops, 1 do begin
  ;before doing anything, set values to what they ought to be
  ;use this to increment / decrement any variable variables ( ;-)  ) too

  ;first, save our loop_count for later output
  Orig_Variable_Value[94]=loop_count
  Orig_Variable_Name[94]="loop_count"

  if (not((loop_count-1) mod Variable_To_Vary_StepCount)) then $ ;its time to iterate
```

```
    Orig_Variable_Value[Variable_To_Vary_Index]+=Variable_To_Vary_StepSize


;dont forget pass by value / pass by reference issues.
ReInitVariables, Orig_Variable_Value, alpha, MeansSize, TIR_MeansSize, V_ChangeLimit, V_MinMembers,$
            V_MaxStdDev, V_MinDistance, V_MaxPair, BandMax_V, V_Limit, S_ChangeLimit,$
            S_MinMembers, S_MaxStdDev, S_MinDistance, S_MaxPair, BandMax_S, S_Limit,$
            T_ChangeLimit, T_MinMembers, T_MaxStdDev, T_MinDistance, T_MaxPair, BandMax_T,$
            T_Limit, V_T_Distance, V_S_T_Distance, Weight, SWIR_MeansSize, Orig_basestring,$
            basestring, loop_count, Variable_To_Vary_Index, Orig_Variable_Name


;====================================
;====================================
;STEP 1:
;
;
;
;
;degrade resolution of VNIR to match TIR
;
; local vars through here - result, max_x, max_y
; result - stores size() result [1] = x, [2] = y

  ;first thing - reduce the size of my image by 3.0
  mag_ratio=T_pix_size / V_pix_size

  ;and define my Point Spread Function to convolve with my image

  ;this PSF is equivalent to an alpha of 0.1 (where center of the PSF
  ;has a value of (1-2alpha)^2. As a comparison, the PSF for ASTER I've
  ;been using is tighter, with an alpha value of 0.06565.
  ;

  PSF_T=CalcPSF(alpha)

  ;create my TIR resolution VNIR images now
```

```
    Vband1_T=Degrade(Vband1, PSF_T, mag_ratio)
    Vband2_T=Degrade(Vband2, PSF_T, mag_ratio)
    Vband3_T=Degrade(Vband3, PSF_T, mag_ratio)
    Vband4_T=Degrade(Vband4, PSF_T, mag_ratio)
    Vband5_T=Degrade(Vband5, PSF_T, mag_ratio)



  ;====================================
  ;====================================
  ;STEP 2:
  ;
  ;
  ;
  ;
  ;generate homogeneous pixel map
  ;
  ; local vars through here -
  ; result - stores size() result [1] = x, [2] = y
  ; band1_stddev, band2_stddev, etc calc. in step 0 above

    ;create an array of same size as TIR to indicate homogeneous VNIR pix
    V_T_homog = fltarr(X_tir,Y_tir) ;zeroed out initially
;    New_V_T_homog=fltarr(X_tir,Y_tir) ;zeroed out initially

    Homog_THEMIS_Pix, Vband1, Vband2, Vband3, Vband4, Vband5,$
                V_stddev[0], V_nonzeroes, V_T_homog, mag_ratio

 ;    New_Homog_THEMIS_Pix, Vband1, Vband2, Vband3, Vband4, Vband5,$
 ;                V_stddev[0], New_V_T_homog, mag_ratio



  ;====================================
  ;====================================
  ;STEP 3:
  ;
```

```
;
;
;
;build the V/S tree, using ISODATA and mahalanobis distance
;
; local vars through here -
; result - stores result of size() function
; seed - array involved in PRNG - DO NOT MESS WITH THIS!!!!
; homog - array of indices of homogeneous pixels
; Means - the 1D list of cluster means for ISODATA

  ;create 2D array - each image is a column in final result
  VNIR_T=make_THEMIS_vis_array(Vband1_T,Vband2_T,Vband3_T,$
                        Vband4_T,Vband5_T,V_numbands)
;should do this only on my non-zero'd pixels
NonZeroPix=where(VNIR_T ne 0.0)
test=histogram(VNIR_T[NonZeroPix], omin=ohmi, omax=BandMax_V)
print, "omin omax of VNIR_T are ", ohmi, BandMax_V

  TIR_T=make_THEMIS_tir_array(Tband1,Tband2,Tband3,Tband4,Tband5,$
                        Tband6,Tband7,Tband8,Tband9,Tband10,$
                        T_numbands)

 ;changed from unindented diagnostic code to indented code 17 Sep 09
 ;should do this only on my non-zero'd pixels
  NonZeroPix=where(TIR_T ne 0.0)
  test=histogram(TIR_T[NonZeroPix], omin=ohmi, omax=BandMax_T)
  print, "omin omax of TIR_T is ", ohmi, BandMax_T

  ;get indices of homogeneous pixels
  ;in theory - VNIR_T[*,homog] is pixel values of a homogeneous
  ;pixel where homogeneous is based on VNIR pixels
  homog=where(V_T_Homog)

  ;find out how many clusters to work with
  result=size(homog, /n_elements)
  print, 'You have',result,' homogeneous VNIR pixels'
```

```
;      read, MeansSize, prompt='Initial number of VNIR clusters?:'
       ;increase MeansSize by 10%, so I can eliminate the closest
       MeansSize=round(Orig_Variable_Value[9] * 1.1)
       VNIRMeans=make_array(MeansSize,1,/long,value=-1L);sizex1 array, all equal -1

       ;calculate the Inverted Covariance Matrix for VNIR
       subimage=VNIR_T[*,homog]
       ;make sure we have more than 1 pixel for covariance
       sizecheck=size(subimage)
       sizecheck=sizecheck[0]-1 ;[0] is 1 if 1, so sizecheck now 0 if 1

       if (sizecheck) then begin
         VNIR90mInvCovMat=GetInvCovMatrix(VNIR_T[*,homog]);calc only for Homog Pix
         TIR90mInvCovMat=GetInvCovMatrix(TIR_T[*,homog]) ;only current cluster
       endif else begin
         VNIR90mInvCovMat=make_array(V_numbands,V_numbands,/float,value=1.0)
         TIR90mInvCovMat=make_array(T_numbands,T_numbands,/float,value=1.0)
         print, "WARNING - FOUND ONLY ONE HOMOGENEOUS PIXEL!!!!!!!!"
       endelse

       Distance=make_array(size(homog, /n_elements), /float, value=0.0)
       VNIRMeans=GetInitialMeans(homog,VNIR_T,VNIRMeans,VNIR90mInvCovMat,Distance)

       ;now get rid of those extra 10% - get rid of ones closest to another
       result=size(VNIRMeans, /n_elements)
       numremove=round(result*10/float(11))
       numremove=result-numremove
       MeansVals=VNIR_T[*,homog[VNIRMeans]]
       VNIRMeans=RemoveLowestMeans(VNIRMeans,numremove,MeansVals,VNIR90mInvCovMat)

       ;have my initial Means to work with - time to ISODATA cluster all of
       ;these. I need to get back the modified means for each cluster and
       ;the cluster to which each pixel belongs. I think this means I need
       ;a PROcedure to get back more than one thing
       ;given the image array, all homogeneous pixels, the initial means,
       ;and Cluster - another 1-D array, same size as homog that will list
       ;the cluster to which the pixel at the common index of homog belongs
```

```
;ie - cluster[x] is cluster to which VNIR_T[*,homog[x]] belongs


;VNIR_T and homog shouldn't change
;
;Means will flip from a 1-D list of homog indices to an X column average
;of all pixels assigned to that cluster.
;
;Cluster will be defined as a 1-D list of same length as homog that can
;have a value between 0 and 49 - value is index of Means to which it
;is closest.
VNIRClusters=homog  ;ok, its cheap, but it does get me the same size
VNIRClusters[*]=-1  ;cheap too - set them all to -1

VNIRMeans=VNIR_T[*,homog[VNIRMeans]]

ClusterChecksum=make_array(V_Limit*4, /ulong, value=0)
 ;why *4? we exceed Limit sometimes, and *4 seems a good safe limit

NumMembers=size(homog, /n_elements)

ISODATA, VNIRMeans, V_Limit, VNIR_T, homog, VNIRClusters, VNIR90mInvCovMat,$
      V_ChangeLimit,V_MinMembers, NumMembers, V_MaxStdDev, V_MinDistance,$
      V_MaxPair, BandMax_V, ClusterChecksum

;changed from unindented diagnostic code to indented real code 17 Sep 09
test=histogram(VNIRClusters, omin=ohmi, omax=ohma)
print, "post ISODATA omin omax and histogram of VNIRClusters is ", ohmi, ohma, " and ", test


;====================================
;====================================
;STEP 3B:
;
;For each VNIR cluster, cluster associated TIR pixels
;
;
```

```
;at this point, I have my TIR sized VNIR pix clusters - for each cluster
;select all of the TIR pixels, and cluster them.

;increase MeansSize by 10%, so I can eliminate the closest
TIRMeansSize=round(Orig_Variable_Value[10] * 1.1)



;save them for posterity (and changes in the for loop)
;oy - so I'll have it doubly protected now for loops within loops. *headache* 4 Oct 09
;these don't change in main loop - in ISODATA.pro file
TrueChangeLimit=T_ChangeLimit
TrueMinMembers=T_MinMembers
TrueMaxPair=T_MaxPair

;how many VNIR clusters are there?
NumVNIRClusters=max(VNIRClusters)
NumVNIRClusters+=1

;make an array to store start and stop array values for VNIR arrays
VNIRtoTIR=make_array(NumVNIRClusters*2, /float)

TIRClusters=float(VNIRClusters)
TIRClusters[*]=-1.0 ;same cheap trick used to create VNIRClusters

ohma=-1.0

;for each cluster, calculate the initial means, and then ISODATA it
for VNIRclus=0, NumVNIRClusters-1, 1 do begin

  ;need to reset these after each iteration
  T_ChangeLimit=TrueChangeLimit
  T_MinMembers=TrueMinMembers
  T_MaxPair=TrueMaxPair

  ThisTIRMeans=make_array(TIRMeansSize,1,/float,value=-1.0)
```

```
TIRhomog=where(VNIRClusters eq VNIRclus)

;want to keep the indices so I can reassign to them later
ThisTIRIndex=TIRhomog

;actual pixel values from homog for this VNIR cluster
TIRhomog=homog[TIRhomog]

;cheap way to get a cluster list of the right size
ThisTIRCluster=TIRhomog


;define MinMembers, ChangeLimit, and NumMembers per cluster
NumMembers=size(TIRhomog, /n_elements)


;ok, now I can call ISODATA for these


if ( size(ThisTIRMeans,/n_elements) ge NumMembers ) then begin
  ThisTIRMeans=indgen(NumMembers)
endif else begin
  Distance=make_array(size(TIRhomog, /n_elements), /float, value=0.0)
  ThisTIRMeans=GetInitialMeans(TIRhomog,TIR_T,ThisTIRMeans,$
          TIR90mInvCovMat,Distance)
  ;now get rid of those extra 10% - get rid of ones closest to another
  result=size(ThisTIRMeans, /n_elements)
  numremove=round(result*10/float(11))
  numremove=result-numremove
  TIRMeansVals=TIR_T[*,TIRhomog[ThisTIRMeans]]
  ThisTIRMeans=RemoveLowestMeans(ThisTIRMeans,numremove,TIRMeansVals,$
                  TIR90mInvCovMat)
endelse; ThisTIRMeans elements vs NumMembers


ThisTIRMeans=TIR_T[*,TIRhomog[ThisTIRMeans]]
```

```
  ClusterChecksum=make_array(T_Limit*4, /ulong, value=0)

  ISODATA, ThisTIRMeans, T_Limit, TIR_T, TIRhomog, ThisTIRCluster,$
        TIR90mInvCovMat, T_ChangeLimit, T_MinMembers, NumMembers,$
        T_MaxStdDev, T_MinDistance, T_MaxPair, BandMax_T, ClusterChecksum

  ;ok, now add ThisTIRCluster list to TIRClusters and mark the sizes
  ;in VNIRtoTIR so I can index it. To do this:

  ;first, modify the cluster values with the previous ohma so that
  ;they don't conflict with results from other VNIR clusters
  ThisTIRCluster+=(ohma+1)

  ;take a histogram of ThisTIRCluster and set VNIRtoTIR[VNIRclus*2]
  ;to ohmi and VNIRtoTIR[(VNIRClus*2)+1] to ohma
  ThisTIRhisto=histogram(ThisTIRCluster,omin=ohmi,omax=ohma,$
                  reverse_indices=ri)

  VNIRtoTIR[(VNIRclus*2)]=ohmi
  VNIRtoTIR[(VNIRclus*2)+1]=ohma

  ;finally, copy cluster values from ThisTIRCluster to TIRClusters
  TIRClusters[ThisTIRIndex]=ThisTIRCluster

  ;end by recording our Means permanant-like
  if (~ohmi) then begin
    TIRMEANS=ThisTIRMeans
  endif else begin
    TIRMeans=[[TIRMeans],[ThisTIRMeans]]
  endelse

endfor; VNIRclus=0

;originally non-indented test code, but I've come to like having it reported. indented on 17 Sep 09
print, "VNIRtoTIR now ", VNIRtoTIR
test=histogram(TIRClusters, omin=ohmi, omax=ohma)
print, "post ISODATA omin omax and histogram of TIRClusters is ", ohmi, ohma, " and ", test
```

```
;====================================
;====================================
;STEP 4:
;
;The super-resolved images of the six SWIR bands are generated by
;allocating the most likely SWIR spectrum to each vis-resolution pixel
;based on spectral similarity in VNIR.

;I hadn't bothered building a VNIR_V image yet - do it now, as I'll
;need it for this step

openw, speclun, basestring+'--vnir-means--post-step3.txt', /get_lun
printf, speclun, VNIRMeans
free_lun, speclun

openw, speclun, basestring+'--tir-means-post-step3.txt', /get_lun, width=200
printf, speclun, TIRMeans/10000000.0
free_lun, speclun

VNIR_V=make_THEMIS_vis_array(Vband1,Vband2,Vband3,Vband4,Vband5,V_numbands)

TIR_V=SuperRes_OneBand(VNIRMeans,TIRMeans,$
                VNIRtoTIR,homog,VNIR_V,VNIR_T,TIR_T,$
                VNIR90mInvCovMat,TIR90mInvCovMat,$
                X_vis,Y_vis,mag_ratio, V_T_Distance)

openw, speclun, basestring+'--vnir-means--post-step4.txt', /get_lun
printf, speclun, VNIRMeans
free_lun, speclun

openw, speclun, basestring+'--tir-means-post-step4.txt', /get_lun, width=200
```

```
printf, speclun, TIRMeans/10000000.0
free_lun, speclun




;====================================
;====================================
;STEP 5:
;
;The super-resolved image (TIR_V) may not be radiometrically accurate,
;so it needs to be modified. This modification is weighted by the distance
;between the VNIR spectra of the vis pixel and VNIR spectra of the TIR
;spectra assigned to the vis_res pixel (whether from tree or image)

Orig_TIR_V=TIR_V

TIR_V=ModifySuperRes(TIR_V, TIR_T, PSF_T, X_vis,Y_vis, mag_ratio)



;at this point TIR 0-T_numbands-1 are the images, T_numbands is the
;source map, and T_numbands+1 is the distance map. Bands
;T_numbands+2 to (2*T_numbands)+1 are correction maps



;annoyed with seeing Emissivity values > 1.0 as they can't be
;real. Let's fix those.

;this whole thing needs a check to see if this is emissivity data we're reading

;first - check if JENVI (8 band) or THMPROC (9 band) emissivity

;    if (T_numbands eq 8) then $
;       qq=TIR_V[1:T_numbands-1,*] $
;    else $
;       qq=TIR_V[2:T_numbands-1,*]
```

```
;     num_toohigh=where(qq gt 10200000)
;     num_toolow=where(qq lt 8500000)

;     rad_fix_counter=1

;     while (num_toohigh[0] ne -1) do begin
;       qq[num_toohigh]*=0.95
;       if (num_toolow[0] ne -1) then qq[num_toolow]*=1.05
;       if (T_numbands eq 8) then $
;         TIR_V[1:T_numbands-1,*]=qq $
;       else $
;         TIR_V[2:T_numbands-1,*]=qq
;
;
;       TIR_V=ModifySuperRes(TIR_V, TIR_T, PSF_T, X_vis,Y_vis, mag_ratio)
;
;       if (T_numbands eq 8) then $
;         qq=TIR_V[1:T_numbands-1,*] $
;       else $
;         qq=TIR_V[2:T_numbands-1,*]
;
;       rad_fix_counter+=1
;
;       if (rad_fix_counter lt 1000) then begin
;         num_toohigh=where(qq gt 10200000)
;         num_toolow=where(qq lt 8500000)
;       endif else begin
;         print, "Too many loops through Rad Fix"
;         num_toohigh=-1
;       endelse
;     endwhile


    ;zero out my border pixels - top row, bottom row, first column, last column
    ;top row
    TIR_V=ZeroLines(TIR_V, X_vis, Y_vis)
```

```
    ;also zero out the offset areas surrounding the images
    TIR_V=ZeroThemisOffset(TIR_V, V_nonzeroes)

    ;finally, write out our data
    WriteOutTHEMISData, TIR_V, V_T_Homog, VNIRClusters, TIRClusters, $
                basestring, Orig_Variable_Value, Orig_Variable_Name

  ;end looping here

    openw, speclun, basestring+'--vnir-means--final.txt', /get_lun
    printf, speclun, VNIRMeans
    free_lun, speclun

    openw, speclun, basestring+'--tir-means-final.txt', /get_lun, width=200
    printf, speclun, TIRMeans/10000000.0
    free_lun, speclun

endfor ;loop_count

print, "sitting at end of program"

end
```

A.3 THEMIS-FUNCS.PRO

```
PRO ReadThemisData, Vband1, Vband2, Vband3, Vband4, Vband5, V_pix_size,$
            V_numbands, X_vis, Y_vis, vis_image, Tband1, Tband2,$
            Tband3, Tband4, Tband5, Tband6, Tband7, Tband8, Tband9,$
            Tband10, T_pix_size, T_numbands, X_tir, Y_tir, tir_image,$
            V_nonzeroes, T_nonzeroes, V_stddev, T_stddev,$
            Orig_Variable_Value, Orig_Variable_Name

;create an array to start storing original values, so we can re-init them during looping
Orig_Variable_Value=fltarr(100)
Orig_Variable_Name=strarr(100)
```

```
;
;Read in vis image and get its size
;first, ask pixel size, number of bands, size of image
read, V_pix_size, prompt="THEMIS vis pixel size (in m)? "
Orig_Variable_Value[0]=V_pix_size
Orig_Variable_Name[0]='V_pix_size'

read, V_numbands, prompt="How many bands are there in the vis image? "
Orig_Variable_Value[1]=V_numbands
Orig_Variable_Name[1]='V_numbands'

read, X_vis, Y_vis, prompt="Size of vis image? (format: column,row) "
Orig_Variable_Value[2]=X_vis
Orig_Variable_Name[2]='X_vis'
Orig_Variable_Value[3]=Y_vis
Orig_Variable_Name[3]='Y_vis'

;create array to hold visible image
vis_image=fltarr(X_vis, Y_vis, V_numbands)

dirname=' '
filename=Dialog_Pickfile(filter='*.img', title="Select THEMIS VIS File ",get_path=dirname)
openr, vislun, filename, /get_lun
readu, vislun, vis_image
free_lun, vislun

;while here, set a default directory too, so the TIR search is easier

;size it upwards to make dealing with it much easier
vis_image*=1000000.0  ;1 million

;Read in tir image
;first, ask pixel size, number of bands, size of image
read, T_pix_size, prompt="THEMIS tir pixel size (in m)? "
Orig_Variable_Value[4]=T_pix_size
Orig_Variable_Name[4]='T_pix_size'
```

```
read, T_numbands, prompt="How many bands are there in the tir image? "
Orig_Variable_Value[5]=T_numbands
Orig_Variable_Name[5]='T_numbands'

read, X_tir, Y_tir, prompt="Size of tir image? (format: column,row) "
Orig_Variable_Value[6]=X_tir
Orig_Variable_Name[6]='X_tir'
Orig_Variable_Value[7]=Y_tir
Orig_Variable_Name[7]='Y_tir'



;create array to hold visible image
tir_image=fltarr(X_tir, Y_tir, T_numbands)

filename=Dialog_Pickfile(filter='*.img', title="Select THEMIS TIR File ", path=dirname)
openr, tirlun, filename, /get_lun
readu, tirlun, tir_image
free_lun, tirlun

;somewhere we convert from float to integer. As a result, size TIR
;files upward to make dealing with them much easier.
;works regardless of radiance or emissivity. probably should make sure
;we convert it back (divide by 10 mil) before saving out the file
tir_image*=10000000.0  ;10 million
;tir_image*=1.0

;now have two different image files. Need to convert these over to bands
;and calculate the standard deviations for individual bands as well as for
;spectral ranges

;vis band first
;assume that we have at least one band, with a potential maximum of 5
Vband1=vis_image[*,*,0]

;check V_numbands and then assign bands 2,3, and 4 as needed
if (V_numbands ge 2) then Vband2=vis_image[*,*,1] else Vband2=-1.0
if (V_numbands ge 3) then Vband3=vis_image[*,*,2] else Vband3=-1.0
```

```
if (V_numbands ge 4) then Vband4=vis_image[*,*,3] else Vband4=-1.0

;band5 may not be optimal quality -ask if the user wants it included
;if there is one available
if (V_numbands eq 5) then begin

  print, "There are 5 vis bands present. However, the fifth band is not always reliable. "
  IncludeFifth=' '
  read, IncludeFifth, prompt="Do you want to include the 5th vis band? (y/n)"

  if ((IncludeFifth eq 'y') || (IncludeFifth eq 'Y') || (IncludeFifth eq 'yes') || (IncludeFifth eq 'Yes')) then begin
    print, "Including Vis Band 5 in analysis"
    Vband5=vis_image[*,*,4]
  endif else begin
    print, "Not including Vis Band 5 in analysis"
    Vband5=-1.0
    V_numbands=4
  endelse; user indicated to include vis band 5
endif else begin
  Vband5=-1.0
endelse; V_numbands eq 5



;tir bands next
;assume that we have at least one band, with a potential maximum of 10
;(this max is unlikely given the current pre-processing necessary to open
; the file under IDL. The jenvi process_themis procedure currently
; averages bands 1 and 2 together, and gets rid of band 10)

Tband1=tir_image[*,*,0]

;check T_numbands and then assign bands 2 through 9 as needed
;switch may be more elegant - just not sure how to assign -1s to it
if (T_numbands ge 2) then Tband2=tir_image[*,*,1] else Tband2=-1.0
if (T_numbands ge 3) then Tband3=tir_image[*,*,2] else Tband3=-1.0
if (T_numbands ge 4) then Tband4=tir_image[*,*,3] else Tband4=-1.0
if (T_numbands ge 5) then Tband5=tir_image[*,*,4] else Tband5=-1.0
```

```
if (T_numbands ge 6) then Tband6=tir_image[*,*,5] else Tband6=-1.0
if (T_numbands ge 7) then Tband7=tir_image[*,*,6] else Tband7=-1.0
if (T_numbands ge 8) then Tband8=tir_image[*,*,7] else Tband8=-1.0

;this one is probably unlikely with our pre-processing
if (T_numbands ge 9) then Tband9=tir_image[*,*,8] else Tband9=-1.0

;thermal band 10 has "near infinite opacity" - probably don't want it.
if (T_numbands eq 10) then begin
  print, "There are 10 TIR bands present. Band 10 has near infinite opacity."
  IncludeTenth=' '
  read, IncludeTenth, prompt="Do you want to include the 10th TIR band? (y/n)"

  if ((IncludeTenth eq 'y') || (IncludeTenth eq 'Y') || (IncludeTenth eq 'yes') || (IncludeTenth eq 'Yes')) then begin
    print, "Including TIR band 10 in super-resolution"
    Tband10=tir_image[*,*,9]
  endif else begin
    print, "Not including TIR band 10 in super-resolution"
    Tband10=-1.0
    T_numbands=9
  endelse; user indicated to include tir band 10
endif else begin
  Tband10=-1.0
endelse; T_numbands=10

;calculate standard deviations for the bands we have - do not include the
;0 value pixels that surround the image and make it rectangular though

;once again, assume we have at least one band and also assume that we
;have pixels that overlap sufficiently that we only need to find the
;non-zeroes for one band in each spectral range.
V_nonzeroes=where(Vband1 gt 0.0)
V_stddev=[-1.0,-1.0,-1.0,-1.0,-1.0,-1.0]

V_stddev[1]=stddev(Vband1[V_nonzeroes])

if (V_numbands ge 2) then V_stddev[2]=stddev(Vband2[V_nonzeroes])
```

```
if (V_numbands ge 3) then V_stddev[3]=stddev(Vband3[V_nonzeroes])
if (V_numbands ge 4) then V_stddev[4]=stddev(Vband4[V_nonzeroes])
if (V_numbands ge 5) then V_stddev[5]=stddev(Vband5[V_nonzeroes])

V_stddev[0]=( (total(V_stddev[1:V_numbands])) / V_numbands)


;ok, now do the same for TIR and make the same assumptions
T_nonzeroes=where(Tband1 gt 0.0)
T_stddev=[-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0,-1.0]


T_stddev[1]=stddev(Tband1[T_nonzeroes])

if (T_numbands ge 2) then T_stddev[2]=stddev(Tband2[T_nonzeroes])
if (T_numbands ge 3) then T_stddev[3]=stddev(Tband3[T_nonzeroes])
if (T_numbands ge 4) then T_stddev[4]=stddev(Tband4[T_nonzeroes])
if (T_numbands ge 5) then T_stddev[5]=stddev(Tband5[T_nonzeroes])
if (T_numbands ge 6) then T_stddev[6]=stddev(Tband6[T_nonzeroes])
if (T_numbands ge 7) then T_stddev[7]=stddev(Tband7[T_nonzeroes])
if (T_numbands ge 8) then T_stddev[8]=stddev(Tband8[T_nonzeroes])
if (T_numbands ge 9) then T_stddev[9]=stddev(Tband9[T_nonzeroes])
if (T_numbands ge 10) then T_stddev[10]=stddev(Tband10[T_nonzeroes])

T_stddev[0]=( (total(T_stddev[1:T_numbands])) / T_numbands)

print, "VIS standard deviations (avg and by band) are: "
print, V_stddev

print, "TIR standard deviations (avg and by band) are: "
print, T_stddev

END;PRO ReadThemisData

FUNCTION Calc_Variance, band, half_size, x_centers, y_centers
;so as to not type this 5 times for THEMIS vis
```

```
    var_band=image_variance(band,half_size,/pop)
    var_band=var_band[x_centers,*]
    var_band=var_band[*,y_centers]

    return, var_band

END; FUNCTION Calc_Variance

PRO New_Homog_THEMIS_Pix, Vband1, Vband2, Vband3, Vband4, Vband5,$
                    thresh, V_T_homog, mag_ratio

; re-writing to speed up - attempt to avoid for loops,
; esp. nested for loops, by using image_variance() to
; calculate the variance. Because stddev is the sqrt()
; of variance, I can compare the correct pixels (center
; of a mag_ratio x mag_ratio neighborhood) to thresh*thresh
; to figure out homogeneity.
;
; Hopefully this will be faster than the previous method

   ;set up indices for the center pixel locations that we want to hold on to
   result = size(Vband1)

   x_centers=findgen(result[1])
   y_centers=findgen(result[2])

   ignore=where(x_centers mod mag_ratio,complement=x_centers)
   x_centers+=1
   ;if mag_ratio is 3, x_centers is now [1,4,7,....]


   ignore=where(y_centers mod mag_ratio,complement=y_centers)
   y_centers+=1
   ;if mag_ratio is 3, y_centers is now [1,4,7....]

   ;the image_variance function expects the 1/2width of our neighborhood
   knlsize=((mag_ratio-1.0)/2.0)
```

```
  print, "Threshold used for VNIR homogeneity is ", thresh

  ;we could take the sqrt() of the variance images to get their stddev() - or we could square thresh
  newthresh=thresh*thresh

  ;surely, there's a prettier way, but.....
  ;an undefined band will return -1 in its [0] position - undefinedband[0]+1 is equal to 0
  if(Vband1[0]+1) then var_band1=Calc_Variance(vband1,knlsize,x_centers,y_centers) $
          else var_band1=make_array([result[1]/mag_ratio,result[2]/mag_ratio],/float,value=thresh);we know
thresh < thresh*thresh
  if(Vband2[0]+1) then var_band2=Calc_Variance(vband2,knlsize,x_centers,y_centers) $
          else var_band2=make_array([result[1]/mag_ratio,result[2]/mag_ratio],/float,value=thresh);we know
thresh < thresh*thresh
  if(Vband3[0]+1) then var_band3=Calc_Variance(vband3,knlsize,x_centers,y_centers) $
          else var_band3=make_array([result[1]/mag_ratio,result[2]/mag_ratio],/float,value=thresh);we know
thresh < thresh*thresh
  if(Vband4[0]+1) then var_band4=Calc_Variance(vband4,knlsize,x_centers,y_centers) $
          else var_band4=make_array([result[1]/mag_ratio,result[2]/mag_ratio],/float,value=thresh);we know
thresh < thresh*thresh
  if(Vband5[0]+1) then var_band5=Calc_Variance(vband5,knlsize,x_centers,y_centers) $
          else var_band5=make_array([result[1]/mag_ratio,result[2]/mag_ratio],/float,value=thresh);we know
thresh < thresh*thresh



  ;want to get max value at each location to compare to that newthresh value
  max_var=max([[[var_band1]],[[var_band2]],[[var_band3]],[[var_band4]],[[var_band5]]],dimension=3)

  ;in theory, max_var is now a 2D array with the maximum band variance at each location
  max_var=where(max_var lt newthresh)
  V_T_Homog[max_var]=1

  ;V_T_Homog is same as old one - but still has where data is all 0s marked as homog
  ;need to find good way to remove that before using this method

  ;assume that if all 0s in band1, same as other bands
  checkforzero=smooth(vband1,mag_ratio)
```

```
   checkforzero=checkforzero[x_centers,*]
   checkforzero=checkforzero[*,y_centers]
   checkforzero=where(checkforzero lt 0.1)

read,pauseval,prompt='check here? '
   V_T_Homog[checkforzero]=0 ;remove any that were over all 0s

END ;PRO New_Homog_THEMIS_Pix

PRO Homog_THEMIS_Pix, Vband1, Vband2, Vband3, Vband4, Vband5,$
                  thresh, V_nonzeroes, V_T_homog, mag

; variables used in this procedure
; result - stores result of size() function
; max_x, max_y - size of X and Y in image
; x, y - loop variables to iterate of 2D image
;
;n.b. - mag is normally mag_ratio; shortened variable name in this one

   result = size(Vband1)

   ;and set our maxes so that we don't exceed beyond our border pixels
   max_x = result[1] - (mag +1)
   max_y = result[2] - (mag +1)

print, "Threshold used for VNIR homogeneity is ", thresh

;as of Oct 27 2006 I no longer remember why I do not check edge pixels
;however, since I did not in my original Homog_Pix_15m I am leaving it
;that way for now.

;as of 25 Feb 07 - the reason we start at mag_ratio in and down, and end
;before we're mag_ratio from the opposite sides is b/c of the convol() that
;we do. We cannot rely on the convol of edge pixels to be a good answer.

   for y = mag, max_y, mag do begin
      for x = mag, max_x, mag do begin
```

```
      ;first thing first - if we're over something thats all 0s, bail.
      ;it speeds it up, AND, we don't want these marked as being homog
      if (total(Vband1[x:x+(mag-1),y:y+(mag-1)]) eq 0.00) then continue

      ;going in band order, since we dont know how many bands we have

      if( (Vband1[0] ge 0) &&$
         (stddev(Vband1[x:x+(mag-1),y:y+(mag-1)]) ge thresh) )$
      then continue

      if( (Vband2[0] ge 0) &&$
         (stddev(Vband2[x:x+(mag-1),y:y+(mag-1)]) ge thresh) )$
      then continue

      if( (Vband3[0] ge 0) &&$
         (stddev(Vband3[x:x+(mag-1),y:y+(mag-1)]) ge thresh) )$
      then continue

      if( (Vband4[0] ge 0) &&$
         (stddev(Vband4[x:x+(mag-1),y:y+(mag-1)]) ge thresh) )$
      then continue

      if( (Vband5[0] ge 0) &&$
         (stddev(Vband5[x:x+(mag-1),y:y+(mag-1)]) ge thresh) )$
      then continue

      ;if we made it here, pix were under thresh in all bands that
      ;were found to exist
      V_T_homog[x / mag, y / mag] = 1

    endfor; x_loop
  endfor ;y_loop

END; PRO Homog_Pix_VNIR

FUNCTION make_THEMIS_vis_array, Vband1, Vband2, Vband3, Vband4,$
```

```
                    Vband5, V_numbands

  ;based on V_numbands, create a 2D array of my image

  res=size(Vband1)

  CASE V_numbands OF
    1:arr=reform(Vband1,res[4],1)
    2:arr=[[reform(Vband1, res(4), 1)],$
          [reform(Vband2, res(4), 1)]]
    3:arr=[[reform(Vband1, res(4), 1)],$
          [reform(Vband2, res(4), 1)],$
          [reform(Vband3, res(4), 1)]]
    4:arr=[[reform(Vband1, res(4), 1)],$
          [reform(Vband2, res(4), 1)],$
          [reform(Vband3, res(4), 1)],$
          [reform(Vband4, res(4), 1)]]
    5:arr=[[reform(Vband1, res(4), 1)],$
          [reform(Vband2, res(4), 1)],$
          [reform(Vband3, res(4), 1)],$
          [reform(Vband4, res(4), 1)],$
          [reform(Vband5, res(4), 1)]]
  ENDCASE ;V_numbands

  arr=rotate(arr,4)

  return, arr

END; FUNCTION create_THEMIS_vis_array

FUNCTION make_THEMIS_tir_array, Tband1, Tband2, Tband3, Tband4, Tband5,$
                    Tband6, Tband7, Tband8, Tband9, Tband10,$
                    T_numbands

  ;based on T_numbands, create a 2D array of my image

  res=size(Tband1)
```

```
CASE T_numbands OF
  1:arr=  reform(Tband1, res[4], 1)
  2:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)]]
  3:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)]]
  4:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)],$
       [reform(Tband4, res(4), 1)]]
  5:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)],$
       [reform(Tband4, res(4), 1)],$
       [reform(Tband5, res(4), 1)]]
  6:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)],$
       [reform(Tband4, res(4), 1)],$
       [reform(Tband5, res(4), 1)],$
       [reform(Tband6, res(4), 1)]]
  7:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)],$
       [reform(Tband4, res(4), 1)],$
       [reform(Tband5, res(4), 1)],$
       [reform(Tband6, res(4), 1)],$
       [reform(Tband7, res(4), 1)]]
  8:arr=[[reform(Tband1, res(4), 1)],$
       [reform(Tband2, res(4), 1)],$
       [reform(Tband3, res(4), 1)],$
       [reform(Tband4, res(4), 1)],$
       [reform(Tband5, res(4), 1)],$
       [reform(Tband6, res(4), 1)],$
       [reform(Tband7, res(4), 1)],$
```

```
        [reform(Tband8, res(4), 1)]]
    9:arr=[[reform(Tband1, res(4), 1)],$
        [reform(Tband2, res(4), 1)],$
        [reform(Tband3, res(4), 1)],$
        [reform(Tband4, res(4), 1)],$
        [reform(Tband5, res(4), 1)],$
        [reform(Tband6, res(4), 1)],$
        [reform(Tband7, res(4), 1)],$
        [reform(Tband8, res(4), 1)],$
        [reform(Tband9, res(4), 1)]]
    10:arr=[[reform(Tband1, res(4), 1)],$
        [reform(Tband2, res(4), 1)],$
        [reform(Tband3, res(4), 1)],$
        [reform(Tband4, res(4), 1)],$
        [reform(Tband5, res(4), 1)],$
        [reform(Tband6, res(4), 1)],$
        [reform(Tband7, res(4), 1)],$
        [reform(Tband8, res(4), 1)],$
        [reform(Tband9, res(4), 1)],$
        [reform(Tband10, res(4), 1)]]
  ENDCASE ;T_numbands

  arr=rotate(arr,4)

  return, arr

END; FUNCTION make_THEMIS_tir_array

FUNCTION ZeroTHEMISOffset, TIR_V, V_nonzeroes

  mask=TIR_V
  mask[*]=0.0
  mask[*,V_nonzeroes]=1.0

  return, TIR_V*mask

END; FUNCTION ZeroTHEMISOffsets
```

A.4 REINITVARIABLES.PRO

```
PRO ReInitVariables, Orig_Variable_Value, alpha, MeansSize, TIR_MeansSize, V_ChangeLimit,
V_MinMembers,$
        V_MaxStdDev, V_MinDistance, V_MaxPair, BandMax_V, V_Limit, S_ChangeLimit,$
        S_MinMembers, S_MaxStdDev, S_MinDistance, S_MaxPair, BandMax_S, S_Limit,$
        T_ChangeLimit, T_MinMembers, T_MaxStdDev, T_MinDistance, T_MaxPair, BandMax_T,$
        T_Limit, V_T_Distance, V_S_T_Distance, Weight, SWIR_MeansSize, Orig_basestring,$
        basestring, loop_count, Variable_To_Vary_Index, Orig_Variable_Name

;first thing first - diagnostics
print, "------------------------Start of ReInit--------------------------------"
print, "alpha", alpha
print, "MeanSize", MeansSize
print, "TIRMeanSize", TIR_MeansSize
print, "V_ChangeLimit", V_ChangeLimit
print, "V_MinMembers", V_MinMembers
print, "V_MaxStdDev", V_MaxStdDev
print, "V_MinDistance", V_MinDistance
print, "V_MaxPair", V_MaxPair
print, "BandMax_V", BandMax_V
print, "V_Limit", V_Limit
print, "S_ChangeLimit", S_ChangeLimit
print, "S_MinMembers", S_MinMembers
print, "S_MaxStdDev", S_MaxStdDev
print, "S_MinDistance", S_MinDistance
print, "S_MaxPair", S_MaxPair
print, "BandMax_S", BandMax_S
print, "S_Limit", S_Limit
print, "T_ChangeLimit", T_ChangeLimit
print, "T_MinMembers", T_MinMembers
print, "T_MaxStdDev", T_MaxStdDev
print, "T_MinDistance", T_MinDistance
```

```
print, "T_MaxPair", T_MaxPair
print, "BandMax_T", BandMax_T
print, "T_Limit", T_Limit
print, "V_T_Distance", V_T_Distance
print, "V_S_T_Distance", V_S_T_Distance
print, "Weight", Weight
print, "SWIR_MeansSize", SWIR_MeansSize
print, "Orig_basestring: ", Orig_basestring
print, "Current basestring: ", basestring

;ok, now reset all of them.
alpha=Orig_Variable_Value[8]
MeanSize=Orig_Variable_Value[9]
TIRMeanSize=Orig_Variable_Value[10]
V_ChangeLimit=Orig_Variable_Value[11]
V_MinMembers=Orig_Variable_Value[12]
V_MaxStdDev=Orig_Variable_Value[13]
V_MinDistance=Orig_Variable_Value[14]
V_MaxPair=Orig_Variable_Value[15]
BandMax_V=Orig_Variable_Value[16]
V_Limit=Orig_Variable_Value[17]
S_ChangeLimit=Orig_Variable_Value[18]
S_MinMembers=Orig_Variable_Value[19]
S_MaxStdDev=Orig_Variable_Value[20]
S_MinDistance=Orig_Variable_Value[21]
S_MaxPair=Orig_Variable_Value[22]
BandMax_S=Orig_Variable_Value[23]
S_Limit=Orig_Variable_Value[24]
T_ChangeLimit=Orig_Variable_Value[25]
T_MinMembers=Orig_Variable_Value[26]
T_MaxStdDev=Orig_Variable_Value[27]
T_MinDistance=Orig_Variable_Value[28]
T_MaxPair=Orig_Variable_Value[29]
BandMax_T=Orig_Variable_Value[30]
T_Limit=Orig_Variable_Value[31]
V_T_Distance=Orig_Variable_Value[32]
V_S_T_Distance=Orig_Variable_Value[33]
```

```
Weight=Orig_Variable_Value[34]
SWIRMeanSize=Orig_Variable_Value[35]
basestring=Orig_basestring+"--loop--"+StrTrim(loop_count,2)+"--of--"+StrTrim(Orig_Variable_Value[95],2)+$
        "--varying-variable-is--"+Orig_Variable_Name[Orig_Variable_Value[99]]

;and, as diagnostics - show where I am at the end
print, "------------------------End of ReInit--------------------------------"
print, "alpha", alpha
print, "MeanSize", MeanSize
print, "TIRMeanSize", TIRMeanSize
print, "V_ChangeLimit", V_ChangeLimit
print, "V_MinMembers", V_MinMembers
print, "V_MaxStdDev", V_MaxStdDev
print, "V_MinDistance", V_MinDistance
print, "V_MaxPair", V_MaxPair
print, "BandMax_V", BandMax_V
print, "V_Limit", V_Limit
print, "S_ChangeLimit", S_ChangeLimit
print, "S_MinMembers", S_MinMembers
print, "S_MaxStdDev", S_MaxStdDev
print, "S_MinDistance", S_MinDistance
print, "S_MaxPair", S_MaxPair
print, "BandMax_S", BandMax_S
print, "S_Limit", S_Limit
print, "T_ChangeLimit", T_ChangeLimit
print, "T_MinMembers", T_MinMembers
print, "T_MaxStdDev", T_MaxStdDev
print, "T_MinDistance", T_MinDistance
print, "T_MaxPair", T_MaxPair
print, "BandMax_T", BandMax_T
print, "T_Limit", T_Limit
print, "V_T_Distance", V_T_Distance
print, "V_S_T_Distance", V_S_T_Distance
print, "Weight", Weight
print, "SWIRMeanSize", SWIR_MeansSize
print, "Orig_basestring: ", Orig_basestring
print, "Current basestring: ", basestring
```

```
END; PRO ReInitVariables
```

A.5 STEP0.PRO

```
;====================================
;====================================
;STEP 0:
;   open crosstalk corrected file
;   define region of interest
;   address / select region of interest in all 14 bands
;   initially - print value of upper left and lower right corners

PRO ReadData, band1, band2, band3N, band4, band5, band6, band7, band8,$
        band9, band10, band11, band12, band13, band14, V_stddev,$
        S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
        vis_image, swir_image, tir_image, Orig_Variable_Value, Orig_Variable_Name


;05 Sep 07 - this function is strictly a wrapper function for
;           determining if we're reading an ASTER L1B file or
;           a set of 3 AST_09 files. Need to decide how to handle
;           the VNIR->TIR case for _09Ts (inclined to ignore it for
;           now).
;
;           I am going to use this function to ask if we're looking
;           at atm corrected data or not, and then call either the
;           ReadL1BData or Read09TData function based on the answer
;

;create an array to start storing original values, so we can re-init them during looping
Orig_Variable_Value=fltarr(100)
Orig_Variable_Name=strarr(100)
```

```
Orig_Variable_Value[0]=15
Orig_Variable_Name[0]='V_pix_size'

Orig_Variable_Value[1]=3
Orig_Variable_Name[1]='V_numbands'

Orig_Variable_Value[4]=90
Orig_Variable_Name[4]='T_pix_size'

Orig_Variable_Value[5]=5
Orig_Variable_Name[5]='T_numbands'



print, ' '
print, ' '
print, ' '
print, '1. super-resolve ASTER L1B file (no atm correction)'
print, ' '
print, '2. super-resolve ASTER 09T files (atm correction; 3 files needed)
print, ' '
print, ' '
read, whichtype,  prompt='Which super-resolve? (1 or 2): '

;ask for coords in VNIR for UL and LR corners - want format x,y
read, VUL_X, VUL_Y,$
     prompt='Array Location of Upper Left VNIR Corner(form x,y):'
read, VLR_X, VLR_Y,$
     prompt='Array Location of Lower Right VNIR Corner(form x,y):'



;now that we have corners, verify valid UL and LR
;VNIR is 1/2 of SWIR and 1/6 of TIR - X,Y must /6 with no modulus

if (not VUL_X mod 6) then begin
  SUL_X = VUL_X / 2
  TUL_X = VUL_X / 6
endif else begin
```

```
  print, 'UL X coord not divisible by 6 - invalid'
endelse

if (not VUL_Y mod 6) then begin
  SUL_Y = VUL_Y / 2
  TUL_Y = VUL_Y / 6
endif else begin
  print, 'UL Y coord not divisible by 6 - invalid'
endelse

if (not (VLR_X+1) mod 6) then begin
  SLR_X = (VLR_X-1) / 2
  TLR_X = (VLR_X-5) / 6
endif else begin
  print, 'LR X coord not divisible by 6 - invalid'
endelse

if (not (VLR_Y+1) mod 6) then begin
  SLR_Y = (VLR_Y-1) / 2
  TLR_Y = (VLR_Y-5) / 6
endif else begin
  print, 'LR Y coord not divisible by 6 - invalid'
endelse


;now that we've defined our corners, lets define our lengths
X_15m=(VLR_X-VUL_X)+1
Y_15m=(VLR_Y-VUL_Y)+1

Orig_Variable_Value[2]=X_15m
Orig_Variable_Name[2]='X_vis'
Orig_Variable_Value[3]=Y_15m
Orig_Variable_Name[3]='Y_vis'


X_30m=(SLR_X-SUL_X)+1
Y_30m=(SLR_Y-SUL_Y)+1
```

```
X_90m=(TLR_X-TUL_X)+1
Y_90m=(TLR_Y-TUL_Y)+1

Orig_Variable_Value[6]=X_90m
Orig_Variable_Name[6]='X_tir'
Orig_Variable_Value[7]=Y_90m
Orig_Variable_Name[7]='Y_tir'

case whichtype of

1: begin
  print, 'Processing L1B file for super-resolution'
  ReadL1BData, band1, band2, band3N, band4, band5, band6, band7, band8,$
     band9, band10, band11, band12, band13, band14, V_stddev,$
     S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
     VLR_X, SLR_X, TLR_X, VLR_Y, SLR_Y, TLR_Y, VUL_X, SUL_X, TUL_X,$
     VUL_Y, SUL_Y, TUL_Y, vis_image, swir_image, tir_image,$
     Orig_Variable_Value, Orig_Variable_Name
  end


2:begin
  print, 'Processing 09T files for super-resolution'
  Read09TData, band1, band2, band3N, band4, band5, band6, band7, band8,$
     band9, band10, band11, band12, band13, band14, V_stddev,$
     S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
     VLR_X, SLR_X, TLR_X, VLR_Y, SLR_Y, TLR_Y, VUL_X, SUL_X, TUL_X,$
     VUL_Y, SUL_Y, TUL_Y, vis_image, swir_image, tir_image,$
     Orig_Variable_Value, Orig_Variable_Name
  end

else:begin
  print, 'Valid choices are 1 or 2'
  ReadData, band1, band2, band3N, band4, band5, band6, band7, band8,$
       band9, band10, band11, band12, band13, band14, V_stddev,$
       S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
```

```
        vis_image, swir_image, tir_image, Orig_Variable_Value, Orig_Variable_Name

  end
endcase; whichtype

END;  PRO ReadData    - ASTER S and T version

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PRO Read09TData, band1, band2, band3N, band4, band5, band6, band7, band8,$
      band9, band10, band11, band12, band13, band14, V_stddev,$
      S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
      VLR_X, SLR_X, TLR_X, VLR_Y, SLR_Y, TLR_Y, VUL_X, SUL_X, TUL_X,$
      VUL_Y, SUL_Y, TUL_Y, vis_image, swir_image, tir_image,$
      Orig_Variable_Value, Orig_Variable_Name

  ;this procedure makes an assumption that the radiance data is in the
  ;swath before the sky irradiance data. if that order changes, I'll need
  ;to come up with some way to differentiate between the BandX in radiance
  ;and the BandX in sky_irradiance. 05 Sep 07


  foundvis = 0

  title='Select the VIS 09XT file'

  repeat begin
    filename=Dialog_Pickfile(FILTER='*.hdf', TITLE=title)

    ;if its an hdf file, open it
    if HDF_IsHDF(filename) then begin
      sd_id = HDF_SD_START(filename)

      ;we know we have an HDF file - but is it the right HDF file?
      ;since I'm not certain how names are parsed look for 3N first
```

```
    ;if we find it, and can read it, we're looking at the right file
    foundvis = HDF_SD_NAMETOINDEX(sd_id,'Band3N')
  endif

  ;set our title in case we didn't actually open the VIS 09XT file
  title = 'That was not a VIS 09XT file. Please select the VIS 09XT file.'

endrep until (foundvis ne -1) ; open up VIS 09T file

;we now have the AST_09XT file for the VIS bands. Get our corner values,
;read in the data, and calculate our standard deviations for our image area

index = HDF_SD_NAMETOINDEX(sd_id,'Band1')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band1= band[VUL_X:VLR_X,VUL_Y:VLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band2')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band2= band[VUL_X:VLR_X,VUL_Y:VLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band3N')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band3N= band[VUL_X:VLR_X,VUL_Y:VLR_Y]

V_stddev=[-1,-1,-1,-1]

V_stddev[0]=(stddev(band1)+stddev(band2)+stddev(band3N))/3.0
V_stddev[1]=stddev(band1)
V_stddev[2]=stddev(band2)
V_stddev[3]=stddev(band3N)
```

```
result=size(band1)
vis_image=[[reform(band1,result[4],1)],$
        [reform(band2,result[4],1)],$
        [reform(band3N,result[4],1)]]
vis_image=rotate(vis_image,4)

HDF_SD_END, sd_id

foundswir = 0

title='Select the SWIR 09XT file'

repeat begin
  filename=Dialog_Pickfile(FILTER='*.hdf', TITLE=title)

  ;if its an hdf file, open it
  if HDF_IsHDF(filename) then begin
    sd_id = HDF_SD_START(filename)

    ;we know we have an HDF file - but is it the right HDF file?
    foundswir = HDF_SD_NAMETOINDEX(sd_id,'Band4')
  endif

  ;set our title in case we didn't actually open the SWIR 09XT file
  title = 'That was not a SWIR 09XT file. Please select the SWIR 09XT file.'

endrep until (foundswir ne -1) ; open up VIS 09T file

index = HDF_SD_NAMETOINDEX(sd_id,'Band4')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band4 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band5')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
```

```
HDF_SD_ENDACCESS, sds_id
band5 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band6')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band6 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band7')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band7 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band8')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band8 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band9')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band9 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

S_stddev=[-1,-1,-1,-1,-1,-1,-1]
S_stddev[0]=(stddev(band4)+stddev(band5)+stddev(band6)+$
        stddev(band7)+stddev(band8)+stddev(band9))/6.0
S_stddev[1]=stddev(band4)
S_stddev[2]=stddev(band5)
S_stddev[3]=stddev(band6)
S_stddev[4]=stddev(band7)
S_stddev[5]=stddev(band8)
S_stddev[6]=stddev(band9)
```

```
result=size(band4)
swir_image=[[reform(band4,result[4],1)],$
        [reform(band5,result[4],1)],$
        [reform(band6,result[4],1)],$
        [reform(band7,result[4],1)],$
        [reform(band8,result[4],1)],$
        [reform(band9,result[4],1)]]
swir_image=rotate(swir_image,4)

HDF_SD_END, sd_id

foundtir = 0

title='Select the TIR 09XT file'

repeat begin
  filename=Dialog_Pickfile(FILTER='*.hdf', TITLE=title)

  ;if its an hdf file, open it
  if HDF_IsHDF(filename) then begin
    sd_id = HDF_SD_START(filename)

    ;we know we have an HDF file - but is it the right HDF file?
    foundtir = HDF_SD_NAMETOINDEX(sd_id,'Band14')
  endif

  ;set our title in case we didn't actually open the TIR 09T file
  title = 'That was not a TIR 09T file. Please select the TIR 09T file.'

endrep until (foundvis ne -1)

index = HDF_SD_NAMETOINDEX(sd_id,'Band10')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band10 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]
```

```
index = HDF_SD_NAMETOINDEX(sd_id,'Band11')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band11 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band12')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band12 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band13')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band13 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'Band14')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band14 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

T_stddev=[-1,-1,-1,-1,-1,-1]
T_stddev[0]=(stddev(band10)+stddev(band11)+stddev(band12)+$
      stddev(band13)+stddev(band14))/5.0
T_stddev[1]=stddev(band10)
T_stddev[2]=stddev(band11)
T_stddev[3]=stddev(band12)
T_stddev[4]=stddev(band13)
T_stddev[5]=stddev(band14)

result=size(band10)
tir_image=[[reform(band10,result[4],1)],$
       [reform(band11,result[4],1)],$
       [reform(band12,result[4],1)],$
```

```
            [reform(band13,result[4],1)],$
            [reform(band14,result[4],1)]]
   tir_image=rotate(tir_image,4)
   tir_image=long(tir_image)

   HDF_SD_END, sd_id

END; PRO Read09TData

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PRO ReadL1BData, band1, band2, band3N, band4, band5, band6, band7, band8,$
      band9, band10, band11, band12, band13, band14, V_stddev,$
      S_stddev, T_stddev, X_15m, X_30m, X_90m, Y_15m, Y_30m, Y_90m,$
      VLR_X, SLR_X, TLR_X, VLR_Y, SLR_Y, TLR_Y, VUL_X, SUL_X, TUL_X,$
      VUL_Y, SUL_Y, TUL_Y, vis_image, swir_image, tir_image,$
      Orig_Variable_Value, Orig_Variable_Name

;ok - start with initialization
;these variables are first defined in step 0
;V=VNIR S=SWIR T=TIR UL=Upper Left LR=Lower Right
;VUL_X,VUL_Y,VLR_X,VLR_Y,SUL_X,SUL_Y,SLR_X,SLR_Y,TUL_X,TUL_Y,TLR_X,TLR_Y=0

;other variables used
;filename - name of HDF (ASTER L1B) file
;sd_id - SDS format file handle, I think?
;sds_id - SDS object handle, I think?
;index - index number for band variable name
;band - total band image
;band1, band2, band3N, band4,....,band14 - subsetted area of band - orig resln
;band1_S, band1_T,...,band4_V,band4_T,....,band14_V,band14_S - alt resolution
;band1_stddev, band2_stddev,...,band14_stddev - std dev of WHOLE band

;initially - select file, verify hdf format, open file
filename = Dialog_Pickfile (filter='*.hdf')
```

```
;if its an hdf file, open it
if HDF_IsHDF(filename) then begin
  sd_id = HDF_SD_START(filename)

  index = HDF_SD_NAMETOINDEX(sd_id,'ImageData1')
  sds_id = HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETDATA, sds_id, band
  HDF_SD_ENDACCESS, sds_id
  band1 = band[VUL_X:VLR_X,VUL_Y:VLR_Y]

  index = HDF_SD_NAMETOINDEX(sd_id,'ImageData2')
  sds_id = HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETDATA, sds_id, band
  HDF_SD_ENDACCESS, sds_id
  band2 = band[VUL_X:VLR_X,VUL_Y:VLR_Y]

  index = HDF_SD_NAMETOINDEX(sd_id,'ImageData3N')
  sds_id = HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETDATA, sds_id, band
  HDF_SD_ENDACCESS, sds_id
  band3N = band[VUL_X:VLR_X,VUL_Y:VLR_Y]

  V_stddev=[-1,-1,-1,-1]
  V_stddev[0]=(stddev(band1)+stddev(band2)+stddev(band3N))/3.0
  V_stddev[1]=stddev(band1)
  V_stddev[2]=stddev(band2)
  V_stddev[3]=stddev(band3N)

  result=size(band1)
  vis_image=[[reform(band1,result[4],1)],$
        [reform(band2,result[4],1)],$
        [reform(band3N,result[4],1)]]
  vis_image=rotate(vis_image,4)

  index = HDF_SD_NAMETOINDEX(sd_id,'ImageData4')
  sds_id = HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETDATA, sds_id, band
```

```
HDF_SD_ENDACCESS, sds_id
band4 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData5')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band5 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData6')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band6 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData7')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band7 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData8')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band8 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData9')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band9 = band[SUL_X:SLR_X,SUL_Y:SLR_Y]

S_stddev=[-1,-1,-1,-1,-1,-1,-1]
S_stddev[0]=(stddev(band4)+stddev(band5)+stddev(band6)+$
      stddev(band7)+stddev(band8)+stddev(band9))/6.0
S_stddev[1]=stddev(band4)
```

```
S_stddev[2]=stddev(band5)
S_stddev[3]=stddev(band6)
S_stddev[4]=stddev(band7)
S_stddev[5]=stddev(band8)
S_stddev[6]=stddev(band9)

result=size(band4)
swir_image=[[reform(band4,result[4],1)],$
        [reform(band5,result[4],1)],$
        [reform(band6,result[4],1)],$
        [reform(band7,result[4],1)],$
        [reform(band8,result[4],1)],$
        [reform(band9,result[4],1)]]
swir_image=rotate(swir_image,4)

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData10')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band10 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData11')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band11 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData12')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
band12 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

index = HDF_SD_NAMETOINDEX(sd_id,'ImageData13')
sds_id = HDF_SD_SELECT(sd_id,index)
HDF_SD_GETDATA, sds_id, band
HDF_SD_ENDACCESS, sds_id
```

```
  band13 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

  index = HDF_SD_NAMETOINDEX(sd_id,'ImageData14')
  sds_id = HDF_SD_SELECT(sd_id,index)
  HDF_SD_GETDATA, sds_id, band
  HDF_SD_ENDACCESS, sds_id
  band14 = band[TUL_X:TLR_X,TUL_Y:TLR_Y]

  T_stddev=[-1,-1,-1,-1,-1,-1]
  T_stddev[0]=(stddev(band10)+stddev(band11)+stddev(band12)+$
        stddev(band13)+stddev(band14))/5.0
  T_stddev[1]=stddev(band10)
  T_stddev[2]=stddev(band11)
  T_stddev[3]=stddev(band12)
  T_stddev[4]=stddev(band13)
  T_stddev[5]=stddev(band14)


  result=size(band10)
  tir_image=[[reform(band10,result[4],1)],$
          [reform(band11,result[4],1)],$
          [reform(band12,result[4],1)],$
          [reform(band13,result[4],1)],$
          [reform(band14,result[4],1)]]
  tir_image=rotate(tir_image,4)
  tir_image=long(tir_image)

  HDF_SD_END, sd_id

;if it wasn't an hdf file, output some error
endif else begin
  print, 'there was a problem - ending early'
endelse

END;  PRO ReadL1BData    - ASTER S and T version
```

A.6 STEP1.PRO

```
;STEP 1 - "The degraded VNIR images with a 30m resolution are generated
;         by pixel aggregation with the PSF of SWIR" - Tonooka
;
; Tonooka assumes sigma = 0.44 based on MTF results in Fujisada et al 1998
;
;
; Topher Hughes, Spring Semester 2006
;
; This should follow the quote above  - we'll see what our results look like

FUNCTION CalcPSF, alpha
;variables used:
;
;
  PSF=fltarr(3,3)
  PSF[0,0]=alpha*alpha
  PSF[0,2]=PSF[0,0]
  PSF[2,0]=PSF[0,0]
  PSF[2,2]=PSF[0,0]

  PSF[0,1]=alpha*(1-(2*alpha))
  PSF[1,0]=PSF[0,1]
  PSF[2,1]=PSF[0,1]
  PSF[1,2]=PSF[0,1]

  PSF[1,1]=(1-(2*alpha))*(1-(2*alpha))

  return, PSF

END; Function CalcPSF

FUNCTION Degrade, OrigBand, PSF, mag_ratio
```

```
;variables used:
;
;result - stores result of size() call
;DegradedBand - the new, degraded resolution image for the band

  result=size(OrigBand)

  if (~result[0]) then begin
    print, "Passed Empty Band - returning -1 value"
    DegradedBand=-1
  endif else begin
    DegradedBand=frebin(OrigBand, result[1]/mag_ratio, result[2]/mag_ratio)
    DegradedBand=convolve(DegradedBand, PSF)
    DegradedBand=float(round(DegradedBand))
  endelse; OrigBand contains data
  return, DegradedBand

END; FUNC Degrade
```

A.7 STEP2.PRO

```
;STEP 2 - "A homogenous pixel is here defined as a 30m-resolution pixel that
;   the standard deviation of four original VNIR pixels in a 30m-resolution
;   pixcel is smaller than a threshold. Using the original VNIR images, the
;   30m-resolution homogeneous pixel map is generated" - Tonooka
;
;   "In the step 2, the threshold determining homogeneous pixels was defined
;   as the band average of the spatial standard deviation over the whole
;   image of each VNIR band: if all VNIR bands had the spatial standard
;   deviation smaller than this threshold in a 30m-resolution pixel, this
;   pixel was assumed to be homogeneous" - Tonooka
;
;
; Tonooka assumes sigma = 0.44 based on MTF results in Fujisada et al 1998
```

```
;
;
; Topher Hughes, Spring Semester 2006
;
; I am not certain about the 'band average of the spatial standard deviation
; over the whole image of each VNIR band' part. In the main program, I define
; my threshold as the average of the stddev() result for each band's whole
; image.


PRO Homog_Pix_VNIR, band1, band2, band3, thresh, mag_ratio, homog_map

; variables used in this procedure
; result - stores result of size() function
; max_x, max_y - size of X and Y in image
; x, y - loop variables to iterate of 2D image

  result = size(band1)

  ;make homog_map 2D
  homog_map=reform(homog_map, result[1] / mag_ratio, result[2] / mag_ratio)

  ;and set our maxes so that we don't exceed check our border pixels
  max_x = result[1] - (mag_ratio +1)
  max_y = result[2] - (mag_ratio +1)

print, "Threshold used for VNIR homogeneity is ", thresh

;as of Oct 27 2006 I no longer remember why I do not check edge pixels
;however, since I did not in my original Homog_Pix_15m I am leaving it
;that way for now.

  for y = mag_ratio, max_y, mag_ratio do begin
    for x = mag_ratio, max_x, mag_ratio do begin

      ;going 3-2-1, since that is order of likeliest fails in test images
```

```
        if(stddev(band3[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
         then continue
        if(stddev(band2[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
         then continue
        if(stddev(band1[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
         then continue

        homog_map[x / mag_ratio, y / mag_ratio] = 1

     endfor; x_loop
  endfor ;y_loop

END

PRO Write_StdDevVals, band1,band2,band3N
; variables used in this procedure
; result - stores result of size() function
; max_x, max_y - size of X and Y in image
; x_loop, y_loop - loop variables to iterate of 2D image
; band_map - 2D array of std dev of pixels
;
; ASSUMES VNIR TO SWIR (4:1)
;
;

   result = size(band1)
   band_map1 = fltarr(result[1]/2 ,result[2]/2)
   band_map2 = fltarr(result[1]/2,result[2]/2)
   band_map3N = fltarr(result[1]/2,result[2]/2)

   max_x = result[1] - 2
   max_y = result[2] - 2

   for y_loop = 0, max_y, 2 do begin
     for x_loop = 0, max_x, 2 do begin

        band_map1[(x_loop/2),(y_loop/2)] = $
```

```
            float(stddev(band1[x_loop:x_loop+1,y_loop:y_loop+1]))

        band_map2[(x_loop/2),(y_loop/2)] = $
            float(stddev(band2[x_loop:x_loop+1,y_loop:y_loop+1]))

        band_map3N[(x_loop/2),(y_loop/2)] = $
            float(stddev(band3N[x_loop:x_loop+1,y_loop:y_loop+1]))

      endfor; x_loop
    endfor ;y_loop

    ;write band std dev to disk

    openw, lun, 'band1_stddev.dat', /get_lun
    writeu, lun, band_map1
    free_lun, lun

    openw, lun, 'band2_stddev.dat', /get_lun
    writeu, lun, band_map2
    free_lun, lun

    openw, lun, 'band3N_stddev.dat', /get_lun
    writeu, lun, band_map3N
    free_lun, lun

END
```

A.8 STEP3.PRO

```
;STEP 3 - Generation of the multi-way V/S tree
;"The multi-way V/S tree is generated from the 30m-resolution VNIR and SWIR
; images, and the 30m-resolution homogeneous pixel map. First all homogeneous
; VNIR pixels (30m) are classified based on the Mahalanobis' generalized
; distance, so that VNIR spectra similar to each other are merged and averaged.
```

```
; Then, all homogeneous SWIR pixels (30m) are classifed for each VNIR merged
; spectrum (class) in the same way, so that one or more SWIR merged spectra
; are given to each VNIR merged spectrum." - Tonooka
;
;"3)generate a multi-way tree for VNIR and SWIR spectra (referred to as the
; V/S tree) by stepwise clustering for the 30m-resolution VNIR and SWIR
; images" - Tonooka
;
;"In steps 3 and 8, the k-means clustering based on Mahalanobis' generalized
; distance was stepwise applied to each of VNIR, SWIR, and TIR. The number
; of initial clusters given is 50 for VNIR and 10 for SWIR in step 3"
; - Tonooka
;
;

FUNCTION GetInvCovMatrix, total_image
;calculate the variance-covariance matrix and then the mahabanolis distance
;for images
;
;variables used:
;result - store result of size() function
;image_temp - image redone as a 1D array
;total_image - array of all images, 1 image per column
;cov_matrix - the covariance matrix of total_image
;inv_cov_matrix - the inverted covariance matrix
;status - the status of the inversion

  ;since we have them 1 band per column, calculate the covariance matrix
  cov_matrix=correlate(total_image,/covariance)

  ;result here should be a #band x #band matrix.
  ;we now invert it. why? because! (b/c mahalanobis uses inverted matrix)
  inv_cov_matrix=invert(cov_matrix, status)



;
;   this block was here during original programming
```

```
;   I don't think I've ever seen it since then
;   commenting out in preparation for conversion to looping 16 Sep 09
;
;   if status then begin
;      print, "STATUS VALUE NOT 0 - SET TO ", status
;      read, PauseVal, prompt="Would you like to see how you screwed up? "
;   endif

   return, inv_cov_matrix

END ;GetInvCovMatrix



FUNCTION GetMahaDist, loc1,loc2,InvCovMat
;given - two separate pixels and the Inverted Covariance Matrix
;calculate - the Mahalanobis distance between the two points
;variables used -
;result - used to store result of size() operation
;diff - difference between loc1 and loc2
;MahaDist - first, an array, then a scalar

   ;first - calculate X sub i minus mean - let loc2 be the mean

   ;may not need the abs() call below; leaving for now to play
   ;it safe. 17 Sep 07

   ;comfortable that I don't need to abs the below line. deleting
   ;the enclosing call to abs() from below. 30 Sep 07

   return, ( (loc1-loc2) # InvCovMat)  #  (loc1-loc2)

END ;GetMahaDist

FUNCTION GetAvgMahaDist, HomogIndex, total_image, InvCovMat

   result=size(HomogIndex, /n_elements)
   avgdist=0.0
```

```
  for firsthomog=0.0, result-2.0, 1.0 do begin
    for secondhomog=firsthomog+1.0, result-1.0, 1.0 do begin
      avgdist=avgdist+(GetMahaDist(total_image[*,HomogIndex[firsthomog]],$
                  total_image[*,HomogIndex[secondhomog]],$
                  InvCovMat)/(result))
    endfor ;secondhomog
  endfor; firsthomog

print, "Average Mahalanobis Distance between all homogeneous pixels is"
print, avgdist

  return, avgdist

END ;GetAvgMahaDist

FUNCTION RemoveLowestMeans, MeansList, NumToRemove, MeansVals, InvCovMat
;given - a MahaList with 10% more Means than needed
;remove the NumToRemove members that are closest to other members
;variables used -
;whichremove - which of NumToRemove Means I am currently calculating
;minval - the lowest calculated distance between two Means
;minval - the current calculated distance to another Means from CurrentMeans
;CurrentMeans - the member of MeansList being examined
;CompMeans - the member of MeansList being measured with CurrentMeans
;MeansToRemove - the member of MeansList with lowest MahaDist to another
;result - used to store result of size() operation
;index - index values for MeansList and MeansVal (same order)
;keeplist - all indices that are not MeansToRemove


  ;I have numremove additional elements. I want to remove those
  ;Means that are close to other means - ie, try to account for
  ;outliers increasing the average distance so Means near one another
  ;are both chosen. As a pair (at minimum) will be selected - remove
  ;the first one found, and then recalcuate the minimums.
```

```
  for whichremove=1, NumToRemove, 1 do begin
    result=size(MeansList, /n_elements)
    minval=GetMahaDist(MeansVals[*,1],MeansVals[*,0],InvCovMat)
    MeansToRemove=0
    for CurrentMeans=0, result-2, 1 do begin
      for CompMeans=CurrentMeans+1, result-1, 1 do begin
        curval=GetMahaDist(MeansVals[*,CompMeans],$
            MeansVals[*,CurrentMeans], InvCovMat)
        if (curval lt minval) then begin
          minval=curval
          MeansToRemove=CurrentMeans
        endif; curval lt minval
      endfor; CompMeans
    endfor; CurrentMeans

    ;minval should be closest distance btwn 2 points, and
    ;MeansToRemove should be its index

    index=replicate(1L,result)
    index[MeansToRemove]=0L
    keeplist=where(index eq 1)
    MeansList=MeansList(keeplist)
    MeansVals=MeansVals(*,keeplist)
   endfor; whichremove
  return, MeansList

END; RemoveLowestMeans


;FUNCTION CalcMahaDist, HomogIndex, total_image, MahaList, InvCovMat
;;given - the index of all homogeneous pixels, the full array
;;(X x Y x 3 or X x Y x 6 or X x Y x 5), the number of means to calc,
;;the array of mean pixels to calculate against, and the inverted covariance
;;matrix - find the Mahalanobis Distance for each pixel to all mean pixels
;;
;;variables used:
;;subimage - only the members of total_image in HomogIndex
```

```
;;result - used to store result of size() operation
;;MahaList - list of starting points. Need to have NumToCalc values in it
;;current - index of the MahaList we're trying to fill
;;thispix - index of already existing MahaList values
;;existing - index of already existing MahaList values
;;total - the total Mahalanobis distance for thispix to all current means
;;find_thispix - check to see if thispix is already a member of MahaList
;;maxval - the highest total Mahalanobis distance to all current means
;;newmeans - the index of HomogIndex that had maxval
;;numremove - the # of Means that are additional, and can be removed
;
;   ;if first time called
;   if (MahaList[0] eq -1) then begin
;      ;pick a random index of a homogeneous pixel to start calculations
;      ;first, figure out our range - its the total size of the homog array
;      result=(size(HomogIndex, /n_elements))-1
;      MahaList[0]=long(round(RANDOMU(seed)*result))
;
;      ;ok, so total_image[HomogIndex[MahaList[0]]] is my random choice
;      ;and I have an Inverted Covariance Matrix,so, call myself
;      MahaList=CalcMahaDist(HomogIndex,total_image,MahaList,InvCovMat)
;
;   endif else begin
;      ;calculate Mahalanobis distance between each existing mean
;      ;and all other points. The new point has the greatest total
;      ;distance from all existing points.
;      ;Might need to toss out any that have a distance=0 since
;      ;those would already be in my list
;
;      ;have MahaList, with some values and some -1
;      ;if we use where to get all -1, the first of these is the index
;      ;of the slot we're looking to fill. This value - 1 is the last
;      ;existing means.
;
;      ;current = index of all non-set means
;      current = where(MahaList eq -1)
;
```

```
;     ;current = first one - this is the one we're trying to fill
;     ;current is now an index value in MahaList
;     current = current[0]
;
;     ;set our maxval=0 to init it, will compare total to it later
;     maxval=0.0
;     newmeans=-1
;
;     ;get size of homog so we know how many pixels to iterate over
;     result=size(HomogIndex, /n_elements) - 1.0
;
;     for thispix = 0.0, result, 1 do begin
;       ;initialize our total
;       total=0.0
;       ;start adding distances - calculate the total MahaDist for thispix
;       for existing = 0, current-1, 1 do begin
;         ;added amount previously ended with "/(current+1)" but I no
;         ;longer see the reason for the divide. We don't need the real
;         ;value - we just want to get which one is furthest.
;
;         dist_to_this_one=GetMahaDist(total_image[*,HomogIndex[thispix]],$
;                         total_image[*,HomogIndex[MahaList[existing]]],$
;                         InvCovMat)
;
;         ;if we're here, we have a Maha Dist of 0.0 - that would
;         ;be the same point as an existing center I think so break
;         ;out of the for loop
;
;         if ~(dist_to_this_one) then total+=dist_to_this_one else break
;
;       endfor ;existing
;       ;if my total is greater than maxval,
;       ;and if my pixel isn't already in the list,
;       ;replace my value with maxval
;
;       if (total gt maxval) and (dist_to_this_one) then begin
;           maxval = total
```

```
;              newmeans = thispix
;        endif ;total gt maxval
;
;     endfor ;thispix
;
;     ;at this point, newmeans should be index of farthest pixel
;     ;with value maxval - go ahead and add it to the list
;
;     MahaList[current]=newmeans
;
;     ;if current was not the last one in list, call me again
;
;     result=size(MahaList, /n_elements)
;     if (current ne result-1) then begin
;        MahaList=CalcMahaDist(HomogIndex,total_image,MahaList,InvCovMat)
;     endif
;   endelse
;   return,MahaList
;
;END ;CalcMahaDist

FUNCTION GetInitialMeans,HomogIndex,total_image,MahaList,InvCovMat,Distance
;given - the index of all homogeneous pixels, the full array
;(X x Y x 3 or X x Y x 6 or X x Y x 5), the number of means to calc,
;the array of mean pixels to calculate against, and the inverted covariance
;matrix - find the Mahalanobis Distance for each pixel to all mean pixels
;

   ;if first time called
if (MahaList[0] eq -1) then begin
   ;pick a random index of a homogeneous pixel to start calculations
   ;first, figure out our range - its the total size of the homog array
   result=(size(HomogIndex, /n_elements))-1
   MahaList[0]=long(round(RANDOMU(seed)*result))

   ;ok, so total_image[HomogIndex[firstpix]] is my random choice
   ;and I have an Inverted Covariance Matrix,so, call myself
```

```
  MahaList=GetInitialMeans(HomogIndex,total_image,MahaList,InvCovMat,Distance)

endif else begin
  ;this is not our first means
  ;we need to add the distance to the last selected means to our total
  ;we need to to select pixels to exclude:
    ;these include any which are already in the list
    ;it also incudes any that had 0 distance to the last added means
    ;we select out the pixel with the max distance from the non-excluded
  ;we call GetInitialMeans again, with the new distances and mean

  ;first - find our most recently added means
  current=where(MahaList eq -1)
  current=current[0]
  current-=1

  ;how many pixels are we looking at:
  num_homog=size(HomogIndex, /n_elements)

  ;record old distances
  old_dist=distance

  for cp=0.0, num_homog-1, 1 do begin
    distance[cp]+=GetMahaDist(total_image[*,HomogIndex[cp]],$
                  total_image[*,HomogIndex[MahaList[current]]],$
                  InvCovMat)
  endfor

  ;distance is now a measure of a pixels distance to all previously chosen
  ;means. Find our exclusions, eliminate them, and pick our max from the
  ;remain values

  exclude=setunion(MahaList[0:current],where(distance eq old_dist))
  include=indgen(num_homog, /float)
  include=SetDifference(include, exclude)
```

```
  if (include[0] eq -1) then begin
    ;if we're here - there were no new means
    ;shorten our MahaList, and return to end our search
    MahaList=MahaList[0:current]
    print, "Found less than the specified number of means - shortened list."
    return, MahaList
  endif; include list is empty


  ;measurement of the maximum value
  maxval=max(distance[include])
  ;indices where maxval is located
  maxval=where(distance eq maxval)
  ;get rid of indices that were already excluded
  maxval=SetDifference(maxval, exclude)
  ;in case there is more than 1, pick the first
  maxval=maxval[0]

  ;maxval is now HomogIndex index

  MahaList[current+1]=maxval

  if (current+1 ne size(MahaList, /n_elements)-1 ) then begin
    MahaList=GetInitialMeans(HomogIndex,total_image,MahaList,InvCovMat,Distance)
  endif; current+1 ne size of MahaList
endelse; MahaList not empty
return, MahaList

END ;GetInitialMeans

PRO ShowMeansDistances, total_image, homog, Means, InvCovMat
;function will be used to measure distance between all means
;and write the values to a formatted text file.
;It seems like maybe my means choices aren't great and I want
;to know why
;
;variables used in this procedure
```

```
;result - used to store the result of size

result=size(Means, /n_elements)

Output=make_array(result,result,/float)
temp=total_image[*,homog[Means]]


for x=0,result-2,1 do begin
  for y=x,result-1,1 do begin
print, 'Means A is ', temp[*,x], 'and Means B is ', temp[*,y], 'and their distance is ',
GetMahaDist(temp[*,y],temp[*,x],InvCovMat)
    Output[x,y]=GetMahaDist(temp[*,y],temp[*,x],InvCovMat)
  endfor
endfor

print, Output
END
```

A.9 ISODATA.PRO

```
PRO ISODAT_Step_One, ChangeLimit, MinMembers, MaxStdDev, MinDistance, MaxPair, BandMax, Limit,
Reason
  ;this procedure will be used to set initial parameters for ISODATA

  print, "The following questions control ", Reason

  read, Limit, prompt='Maximum iterations for ISODATA?: '

  read, ChangeLimit, prompt='Iterations end when % changed < (0.5 default): '

  read, MinMembers, prompt='% of total in cluster to not delete (0.01 default): '
  read, MaxStdDev, prompt='Std Dev threshold before splitting? (5-10 default): '
```

```
  read, MinDistance, prompt='Minimum distance between centroids before joining? (2 default): '

  read, MaxPair, prompt='Maximum clusters to join together per iteration? (5 default): '

  print, "The current Max DN value in these bands is "+StrTrim(float(BandMax),2)
  read, prompt='Max DN value to use?', BandMax

END; ISODAT_Step_One

FUNCTION ISODAT_Step_Two_C, Image, HomogIndex, Means, InvCovMat
;distribute all homogeneous pixels among the current cluster centers

NumHomog=size(HomogIndex, /n_elements)
meansize=size(Means)
bands=meansize[1]

if (size(meansize, /n_elements) eq 4) then begin
  meansize=1
endif else begin
  meansize=meansize[2] ; should be number of rows - which is # of clusters
endelse

Distance=make_array(NumHomog, meansize, /float)

HomogImage=Image[*,HomogIndex]

ex_ICM=reform(rebin(InvCovMat,bands,bands,NumHomog),bands,bands*NumHomog)

for cm=0, meansize-1, 1 do begin


  ;calc out our row
  Row=HomogImage-rebin(Means[*,cm],bands, NumHomog, /sample)
  ;expand it to match ex_ICM
  Row=rebin(Row,bands,bands*NumHomog, /sample)
  ;uniq_entries - so I can pull out my original stuff later
  uniq_entries=bands*indgen(NumHomog, /float)
```

```
  stage1=total(ex_ICM*Row,1)

  ;set Row back to unexpanded, and make it a 1D list
  Row=reform(Row[*,uniq_entries],bands*NumHomog)

  Distance[*,cm]=total(reform(stage1*Row,bands,NumHomog),1)

endfor
;Distance is now a NumHomog wide, meansize deep array - pick out the mins

if (meansize eq 1) then begin
  ClusterList=make_array(NumHomog,value=0L)
endif else begin
  Distance=min(Distance,ClusterList,dim=2,/absolute)
  ClusterList= floor(temporary(ClusterList) / NumHomog)
endelse

return, ClusterList

END; ISODAT_Step_Two_C

PRO ISODAT_Step_Three, Means, ClusterList, MinMembers, WereDeleted
;remove any cluster with less than MinMembers in it.
;variables used:
;current_cluster - the cluster being examined
;meansize - used to store result of size(Means)
;members - the pixels from ClusterList in cluster curr_cluster
;nummembers - # of pixels from ClusterList in cluster curr_cluster
;NewMeans - the Means list without the cluster being deleted

meansize=size(Means)
bands=meansize[1]
if (size(meansize, /n_elements) eq 4) then begin
  meansize=1
endif else begin
  meansize=meansize[2] ; should be number of rows - which is # of clusters
```

```
endelse

histo_array=histogram(ClusterList, omin=ohmi, omax=ohma, REVERSE_INDICES=ri)
histo_array=where(histo_array lt MinMembers)
histo_array=reverse( histo_array[ sort(histo_array) ] )

last=ohma

if (histo_array[0] ne -1) then begin

  WereDeleted=1 ;ok - so we set this multiple times

  for clus_ind=0, size(histo_array, /n_elements)-1, 1 do begin
    clus_to_del=histo_array[clus_ind]

    ;set the members of clus_to_del to -1 if there are any
    if ri[clus_to_del] ne ri[clus_to_del+1] then begin
      ClusterList[ri[ri[clus_to_del]:ri[clus_to_del+1]-1]]=-1
    endif

    ;unless last cluster, set all members gt clus_to_del to one less
    if ((clus_to_del ne meansize)&&(ri[clus_to_del+1] ne ri[meansize])) then begin
      ClusterList[ri[ri[clus_to_del+1]:*]]-=1
    endif

    CASE clus_to_del OF
        0: Means=Means[*,1:*]
      last: Means=Means[*,0:last-1]
      else: Means=reform([Means[0:(clus_to_del*bands)-1],$
              Means[(bands*(clus_to_del+1)):*]],bands,last)
    ENDCASE

    last-=1
  endfor; clus_ind

  ;keep my histograms sane - set all my -whatevers to -1
  deleted_members=where(ClusterList le -1)
```

```
  if (deleted_members[0] ne -1) then begin
    ClusterList[deleted_members]=-1
  endif

endif; histo_array has clusters to delete

END; ISODAT_Step_Three

FUNCTION ISODAT_Step_Four, Image, HomogIndex, Means, ClusterList
;this step recomputes the means of the clusters based on their
;assigned pixels. In theory, if we deleted clusters in step 3, we
;have unassigned pixels - they are ignored during this step and only
;the pixels assigned to a cluster during step 2 are used to compute
;the new means
;variables used:
;current_cluster - the cluster we're working on
;current_band - the band we're working on
;meansize - used to store result of size()
;          result[1] on Means should be #bands
;          result[2] on Means should be #clusters
;cluster_size - used to store result of size() on ClusterList
;members - all members (using where) from ClusterList in a cluster
;NewMeans - store the newly calculated Means
;
;
;changed from procdedure to function, returning NewMeans 20 Sep 07
;
;
meansize=size(Means)
NewMeans=Means

bands=meansize[1]

if (size(meansize, /n_elements) eq 4) then begin
  meansize=1
endif else begin
  meansize=meansize[2] ; should be number of rows - which is # of clusters
```

```
endelse

;for each cluster, for each band compute the new mean

for current_cluster=0, meansize-1, 1 do begin

  ;select all member of cluster
  members=where(ClusterList eq current_cluster)

  for current_band=0, bands-1, 1 do begin

    NewMeans[current_band, current_cluster]=$
      round(mean(Image[current_band,HomogIndex[members]]))

  endfor; current_band

endfor; current_cluster

return, NewMeans

END ; ISODAT_Step_Four


FUNCTION ISODAT_Step_Five_A, Image, Homog, Means, ClusterList,$
                    CurrClusters, InvCovMat

;compute the average distance of the samples from their cluster centers
;variables used in this function
;current_cluster - the cluster whose average is being computed
;members - members of current_cluster
;nummembers - size(members, /n_elements)
;current_pix - the pixel whose distance is being measured

NumHomog=size(Homog, /n_elements)
bands=size(Means)
bands=bands[1]
```

```
;calc out our row, and replicate it to match ex_ICM
Row=rebin((Image[*,Homog]-Means[*,ClusterList]),bands, bands*NumHomog, /sample)

;expand our Inverted Covariance Matrix
ex_ICM=reform(rebin(InvCovMat, bands, bands, NumHomog), bands, bands*NumHomog)

;set uniq_entries to recover unique entries from Row
uniq_entries=bands*indgen(NumHomog, /float)

stage1=total(ex_ICM*Row,1)

;set Row back to unexploded version and stretch it out into 1-D
Row=reform(Row[*,uniq_entries],bands*NumHomog)

Row=total(reform(stage1*Row,bands,NumHomog),1)

;row is now ClusterList long, and contains the distance from that pixel to
;its cluster's mean

AverageDistance=histogram(ClusterList,omax=maxclus, REVERSE_INDICES=ri)
AverageDistance=float(AverageDistance)

for cc=0, maxclus, 1 do AverageDistance[cc]=mean(Row[ri[ri[cc]:ri[cc+1]-1]])

return, AverageDistance

END; ISODAT_Step_Five_A

FUNCTION ISODAT_Step_Six_A, AverageDistance, ClusterList
;compute the weighted overall average distance
;variables used in this function-

clus_weight=histogram(ClusterList)

return, total(clus_weight*AverageDistance)/total(clus_weight)

END; ISODAT_Step_Six_A
```

```
PRO ISODAT_Step_Seven, MaxPair, SkipToEleven, RunLimit, Iter, CurrClusters, MaxClusters
;this function determines whether we go to step 8 or skip ahead to 11

if (Iter eq RunLimit) then begin
  MinDistance=0
  SkipToEleven=1
endif

if ((Iter mod 2) eq 0 ) then begin
  SkipToEleven=1
endif

if (CurrClusters ge (2*MaxClusters) ) then begin
  ;How is it even possible to reach this point?
  SkipToEleven=1
endif

END; ISODAT_Step_Seven

FUNCTION ISODAT_Step_Eight_B,Image,HomogIndex, Means,ClusterList,CurrClusters
;calculate the per cluster per band standard deviation
;variables used in this function-
;current_cluster - cluster being examined
;current_band - band being examined
;members - the members of the cluster
;nummembers - the size of members
;numbands - the number of bands in this spectral range
;pixbandval - value of pixel current_pix at band current_band
;meanbandval - value of cluster current_cluster mean at band current_band
;
;
; this version does not require the avgstd.pro file
;
;
  numbands=size(Means)
  numbands=numbands[1]
```

```
  StdDevVector=make_array(numbands, CurrClusters, /float, value=0.0)

  for current_cluster=0, CurrClusters-1, 1 do begin
    members=where(ClusterList eq current_cluster)

    if (size(members, /n_elements) lt 2) then begin
      StdDevVector[0:numbands-1, current_cluster]=0.0
      continue
    endif

    for current_band=0, numbands-1, 1 do begin
      StdDevVector[current_band,current_cluster]=$
          STDDev(Image[current_band,HomogIndex[members]])
    endfor; current_band
  endfor; current_cluster

  return, StdDevVector

END; ISODAT_Step_Eight_B

FUNCTION ISODAT_Step_Nine, StdDevVector

  MaxStdDevVector=max(StdDevVector, dimension=1)

return, MaxStdDevVector

END; ISODAT_Step_Nine


PRO ISODAT_Step_Ten, Means, MaxStdDev, StdDevVector,MaxStdDevVector,$
                AverageDistance, OverallAvgDist, ClusterList,$
                CurrClusters, MaxClusters, MinDistance, BandMax, Split
;this function is the splitting step. If we have one of two conditions, we
;will split a cluster into two new means.
;
;The two conditions are:
```

```
;
; if MaxStdDevVector > MaxStdDev   AND EITHER

; A - if the cluster avg dist is > overall avg distance AND
;           nummembers > ((2*MinMembers)+1)
;
; OR
;
; B -  CurrClusters <= MaxClusters/2
;
;
;variables used in this function:
;current-cluster - the cluster being examined
;SplitThis - whether or not to split the current cluster
;result - stores the result of size()
;NewMeans - holder for our expanded Means
;band - the band of the Max StdDevVector
;k - the fraction to multiply times MaxStdDevVector for my +/-
;NewClusters - the new number of clusters
;current_band - the band which I am examining

NewClusters=CurrClusters

for current_cluster=0, CurrClusters-1, 1 do begin

  members=where(ClusterList eq current_cluster)
  nummembers=size(members, /n_elements)
  SplitThis=0
  if (MaxStdDevVector[current_cluster] gt MaxStdDev) then begin

    if (AverageDistance[current_cluster] gt OverallAvgDist) then begin
      if (nummembers gt (2*(MinDistance+1))) then begin
        Split=1
        SplitThis=1
      endif; nummembers > 2(MinDistance+1)
    endif; AverageDistance > Overall Average Distance
```

```
  if (NewClusters le (MaxClusters / 2.0)) then begin
    Split=1
    SplitThis=1
  endif; CurrClusters > MaxClusters/2


endif; MaxStdDevVector > MaxStdDev

if (SplitThis eq 1) then begin
  meansize=size(Means)

  if (size(meansize, /n_elements) eq 4) then begin
    numbands=meansize[1]
    meansize=1
  endif else begin
    numbands=meansize[1]
    meansize=meansize[2]
  endelse

  NewMeans=make_array(numbands,meansize+1)
  NewMeans[*,0:meansize-1]=Means
  NewClusters=NewClusters+1

  k=0.75

  ;find the max component. avg of that band +/- (k*maxstd) are new centers
  band=where(StdDevVector[*,current_cluster] eq MaxStdDevVector[current_cluster])

  ;sometimes there is a tie - go for the first one.
  band=band[0]

  ;set new means equal to current means for now
  NewMeans[*,meansize]=NewMeans[*,current_cluster]

  ;modify MaxStdDev band - current is lower, new is higher
```

```
    if ((NewMeans[band,current_cluster] +$
     round(k * MaxStdDevVector[current_cluster])) gt BandMax) then begin
       NewMeans[band,meansize]=BandMax
    endif else begin
       NewMeans[band,meansize]=NewMeans[band,current_cluster] +$
                    round(k * MaxStdDevVector[current_cluster])
    endelse


    if((NewMeans[band, current_cluster]-$
     round(k * MaxStdDevVector[current_cluster])) lt 0) then begin
       NewMeans[band,current_cluster]=0
    endif else begin
       NewMeans[band,current_cluster]=NewMeans[band, current_cluster]-$
                    round(k * MaxStdDevVector[current_cluster])
    endelse

    Means=NewMeans

  endif; SplitThis=1

endfor; current_cluster


CurrClusters=NewClusters

END; ISODAT_Step_Ten


FUNCTION ISODAT_Step_Eleven, Means, ClusterList, InvCovMat
;this function computes the distances between centroids
;variables used in this function-
;result - store the result of size()
;current_cluster - cluster being compared
;current_means - the cluster being measured to
;MeansDistances - the 2-D array of distances between means
```

```
;calculate the number of pairings we have
;THE NEXT TWO LINES ASSUMED THAT ALL MEANS HAD DATA - NOT TRUE AFTER STEP 10
;pairs=size(Means)
;pairs=total(indgen(pairs[2]-1))

meansize=size(Means)
meansize=meansize[2]

histo_array=histogram(ClusterList, omin=ohmi, omax=ohma)

;do some sanity checks - no fixes but do put out some errors
if (ohmi ne 0) then print, 'IN STEP 11 - OHMI NE 0 - SET TO: ', ohmi
if (ohma+1 gt meansize) then print, 'OHMA+1 GT MEANSIZE - OHMA SET TO: ', ohma

MeansDistances=make_array(3,total(indgen(ohma+1)),/float, value=-1.0)

current_pair=0.0

for current_means=0.0, ohma-1.0, 1.0 do begin
  for current_cluster=current_means+1.0, ohma, 1.0 do begin
    MeansDistances[0,current_pair]=current_means
    MeansDistances[1,current_pair]=current_cluster
    MeansDistances[2, current_pair]=$
        Means[*,current_cluster]#InvCovMat#Means[*,current_means]
    current_pair+=1
  endfor; current_cluster
endfor; current_means

return, MeansDistances
END; ISODAT_Step_Eleven


FUNCTION ISODAT_Step_Eleven_A, Means, ClusterList, InvCovMat
;this function computes the distances between centroids
;variables used in this function-
;result - store the result of size()
```

```
;current_cluster - cluster being compared
;current_means - the cluster being measured to
;MeansDistances - the 2-D array of distances between means

;calculate the number of pairings we have
;THE NEXT TWO LINES ASSUMED THAT ALL MEANS HAD DATA - NOT TRUE AFTER STEP 10
;pairs=size(Means)
;pairs=total(indgen(pairs[2]-1))

meansize=size(Means)
meansize=meansize[2]

histo_array=histogram(ClusterList, omin=ohmi, omax=ohma)

;do some sanity checks - no fixes but do put out some errors
if (ohmi ne 0) then print, 'IN STEP 11 - OHMI NE 0 - SET TO: ', ohmi
if (ohma+1 gt meansize) then print, 'OHMA+1 GT MEANSIZE - OHMA SET TO: ', ohma

MeansDistances=make_array(3,total(indgen(ohma+1)),/float, value=-1.0)

current_pair=0.0

for current_means=0.0, ohma-1.0, 1.0 do begin
  for current_cluster=current_means+1.0, ohma, 1.0 do begin
    MeansDistances[0,current_pair]=current_means
    MeansDistances[1,current_pair]=current_cluster
    Diff=(Means[*,current_cluster]-Means[*,current_means])
    MeansDistances[2, current_pair]=$
     Diff#InvCovMat#Diff
    current_pair+=1
  endfor; current_cluster
endfor; current_means

return, MeansDistances
END; ISODAT_Step_Eleven_A

FUNCTION ISODAT_Step_Twelve,MeansDistances, MaxPair, MinDistance
```

```
;select the smallest up to MaxPair distances between 0 and MinDistance
;variables used-
;sorted_indices - indices from sorting values
;tmp_indices - indices from sorting values

sorted_indices=sort(MeansDistances[2,*])

;want at most MaxPair of them
;but first make sure I actually have at least MaxPair first

CurrPair=size(MeansDistances)

if (size(CurrPair, /n_elements) eq 4) then begin
  CurrPair=1
endif else begin
  CurrPair=CurrPair[2]
endelse

if (CurrPair lt MaxPair) then begin
  MaxPair=CurrPair
endif

sorted_indices=sorted_indices[0:MaxPair-1]

;and I don't want any whose value is greater than MinDistance

tmp_indices=where(MeansDistances[2,sorted_indices] le MinDistance)

if (tmp_indices[0] ne -1) then begin
  sorted_indices=sorted_indices[tmp_indices]
  MeansDistances=MeansDistances[0:1,sorted_indices]
endif else begin
  MeansDistances=-1
endelse

return, MeansDistances
```

```
END; ISODAT_Step_Twelve

PRO ISODAT_Step_Thirteen, ClusterList, Means, Lumps, CurrClusters
;this procedure will lump pairs of clusters together and create a new
;Means that is based on the weighted average of the old Means
;variables used:
;lumped: ensuring that no cluster is double-lumped
;numpairs -number of pairs in Lumps
;pair - which pair in Lumps we're working on
;bands - the number of bands we're dealing with
;memone, memtwo - members of the clusters being lumped
;nummemone, nummemtwo - number of members of the clusters being lumped
;cleanup_array - tmp holder for removing lumped Means

histo_array=histogram(ClusterList, omin=ohmi, omax=ohma, REVERSE_INDICES=ri)

flush, 0

if (ohmi ne 0) then print, "Ohmi not 0, instead it is ", ohmi," in Step 13."
if (ohma+1 ne CurrClusters) then print, "Ohma and CurrClusters out of sync - CurrClusters is ", CurrClusters," and
ohma is ", ohma



lumped=make_array(ohma+1, value=0)
removed_clusters=make_array(ohma+1,value=0)
numpairs=size(Lumps)
if (size(numpairs, /n_elements) eq 4) then numpairs=1 else numpairs=numpairs[2]

;make sure we're dealing with floats in here
Means=float(Means)

;assuming that if we're lumping pairs, there is more then one Means
meansize=size(Means)
meansize=meansize[2]

if (ohma+1 ne meansize) then print, "ohma and meansize not in sync."
```

```
for pair=0, numpairs-1, 1 do begin
  if (lumped[Lumps[0,pair]] OR lumped[Lumps[1,pair]]) then continue

  ;set both to indicate that they've been lumped already
  lumped[Lumps[0,pair]]=1
  lumped[Lumps[1,pair]]=1

  ;make sure Clusters to Lump aren't empty
  if (ri[Lumps[0,pair]] eq ri[Lumps[0,pair]+1]) then continue
  if (ri[Lumps[1,pair]] eq ri[Lumps[1,pair]+1]) then continue

  ;reverse indices confuses me - use where for now
  memone=where(ClusterList eq Lumps[0,pair])
  memtwo=where(ClusterList eq Lumps[1,pair])
  nummemone=size(memone, /n_elements)
  nummemtwo=size(memtwo, /n_elements)

  ClusterList[memtwo]=Lumps[0,pair]
  ;calculate the weighted averages of the two clusters
  Means[*,Lumps[0,pair]]=$
  round(((Means[*,Lumps[0,pair]]*nummemone)+$
       (Means[*,Lumps[1,pair]]*nummemtwo))/(nummemone+nummemtwo))
  ;set the 2nd Means to -1s
  Means[0,Lumps[1,pair]]=-1
endfor; pair


;remove the deleted (set to -1) means
kept=where(Means[0,*] ne -1)
removed=where(Means[0,*] eq -1)
Means=Means[*,kept]

;now need to lower cluster values
;start by reversing out removed, so we can work our way down-list
removed=reverse(removed)

for rem_idx=0, size(removed, /n_elements)-1, 1 do begin
```

```
  error_check=where(ClusterList eq removed[rem_idx])
  if (error_check[0] ne -1) then print, "FOUND REMOVED CLUSTER IN STEP 13!"

  lower_list=where(ClusterList gt removed[rem_idx])
  if (lower_list[0] ne -1) then ClusterList[lower_list]-=1

endfor; rem_idx

CurrClusters=size(Means)

if (size(CurrClusters, /n_elements) eq 4) then begin
  CurrClusters=1
endif else begin
  CurrClusters=CurrClusters[2]
endelse

test_histo=histogram(ClusterList, omin=ohmi, omax=ohma)

if (CurrClusters gt (ohma+1)) then begin
  print, "CurrClusters larger than ohma+1 - resizing Means and CurrClusters"
  print, "Is this a removal of means created in step 10?"
  Means=Means[*,0:ohma]
  CurrClusters = ohma+1
endif

if (ohmi ne 0) then begin
  print, "OHMI NE 0 - halting program"
  STOP
endif

END; ISODAT_Step_Thirteen
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
PRO ISODATA, Means, RunLimit, Image, HomogIndex, ClusterList, InvCovMat,$
        ChangeLimit,MinMembers, NumMembers, MaxStdDev, MinDistance,$
        MaxPair, BandMax, ClusterChecksum
;this procedure will take our initial means, the maximum number of loops,
;the total image, the homogeneous pixel list, a blank array for assigning
;homogeneous pixels to a cluster, and the Inverted Covariance Matrix and
;perform an ISODATA clustering on the homogeneous pixels in the total image.
;other variables which will need to be defined and may need user input are:
;
;ChangeLimit- the maximum fraction unchanged between runs before stopping
;MinMembers - delete clusters with less than this percent of the members in it
;MaxStdDev - part of cluster splitting limit. If a cluster's standard deviation
;           exceeds this value, and there are at least twice the MinMembers in
;           it, split the cluster unless there are already too many clusters.
;           I will probably use Mahalanobis distance to check this, maybe.
;MinDistance - the minimum distance between clusters before joining them. It
;           will probably also be checked with Mahalanobis.
;
;
;Other variables used in this procedure:
;
;result,meansize - used to store the result of size()
;current_pix - the current pixel being assigned to a cluster
;current_cluster - the cluster center currently being measured
;closest_center_distance - the distance from current_pix to the closest center
;distance - distance to current_cluster from current_pix
;OldCluster - a copy of ClusterList pre-changes (to check for change %s)
;remlist - list of clusters to delete
;Iter - which iteration we're on. Some things happen on evens, etc.

;initialize stuff - ISODATA step 1
;Tou and Gonzalez (1974) describe step one as:
;Specify the following parameters:
;K=number of cluster centers desired    (initial size of ClusterList)
;0n = parameter against which number of samples is compared (MinMembers)
;0s = standard deviation parameter    (MaxStdDev)
```

```
;0c = lumping parameter (MinDistance)
;L = maximum number of pairs of cluster centers which can be lumped (MaxPair)
;I= number of iterations to allow (RunLimit)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;ISODAT_Step_One;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

OldCluster = ClusterList
MaxClusters=size(Means)
MaxClusters=MaxClusters[2]
CurrClusters=MaxClusters
NumMembers=size(HomogIndex, /n_elements)
Iter=0L
Split=0

ChangeLimit=ChangeLimit/float(100)
ChangeLimit=round(ChangeLimit*NumMembers)

MinMembers=round(MinMembers*float(NumMembers)*0.01)

NumHomog=size(HomogIndex, /n_elements)

;always keep MinMembers greater than 0
if ~MinMembers then MinMembers=1

repeat begin ; do 2-14 until terminate=1
  repeat begin ;do 2-10 until none split in 10
    repeat begin ;do 2-4 until none deleted in 3
      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      ;;;;;;;;;;;;ISODAT_Step_Two;;;;;;;;;;;;;;;
      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;assign pixels to clusters
ClusterList=ISODAT_Step_Two_C(Image, HomogIndex, Means, InvCovMat)
;an iteration is every time that the program goes to step 1 or 2
Iter=Iter+1

;reset my Split and WereDeleted
Split=0
WereDeleted=0



;check for clusters that got silently dropped during the
;assignment period.
clus_histo=histogram(ClusterList, omin=ohmi, omax=ohma)

if ((ohma+1) lt CurrClusters) then begin
  CurrClusters=ohma+1
  Means=Means[*,0:CurrClusters-1]
endif

if (ohmi ne 0) then begin
  Means=Means[*,ohmi:CurrClusters-1]
  CurrClusters=CurrClusters-ohmi
  ClusterList-=ohmi
endif



;now that all homogeneous pixels are assigned to a cluster
;use the checksum function to see if we have previously had
;this pattern of clusters before. If we have - we are caught
;in a loop. We can use the checksums to determine this, and
;can then break out of the clustering sequence.

;get the Checksum for the current ClusterList and save it
;in the array of Checksums we created. I tried passing it
;ClusterChecksum[Iter] as the 2nd argument but it would not
;store the value that way (never modded from 0).
;spawn, 'compute-sha '+ClusterList, spawnresults
```

```
  ;ClusterChecksum[Iter]=spawnresults

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;ISODAT_Step_Three;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  ;delete clusters with less than MinMembers in it
  ISODAT_Step_Three, Means, ClusterList, MinMembers, WereDeleted

  CurrClusters=size(Means)
  if ( size(CurrClusters, /n_elements) eq 4) then begin
    CurrClusters=1
  endif else begin
    CurrClusters=CurrClusters[2]
  endelse

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;ISODAT_Step_Four;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  Means=ISODAT_Step_Four(Image, HomogIndex, Means, ClusterList)

endrep until (WereDeleted eq 0) ;Step 2-4 loop until none deleted

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;ISODAT_Step_Five;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
AverageDistance=ISODAT_Step_Five_A(Image, HomogIndex, Means,$
                        ClusterList,CurrClusters,InvCovMat)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;ISODAT_Step_Six;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;overall average distance is weighted by the #members in a cluster
OverallAvgDist=ISODAT_Step_Six_A(AverageDistance,ClusterList)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;ISODAT_Step_Seven;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

SkipToEleven=0

ISODAT_Step_Seven, MaxPair, SkipToEleven, RunLimit, Iter,$
          CurrClusters, MaxClusters

Split=0

if (SkipToEleven eq 0) then begin
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;ISODAT_Step_Eight;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

  StdDevVector=ISODAT_Step_Eight_B(Image, HomogIndex, Means,$
                      ClusterList, CurrClusters)


  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;ISODAT_Step_Nine;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
    MaxStdDevVector=ISODAT_Step_Nine(StdDevVector)



    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;ISODAT_Step_Ten;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

    if (Iter lt RunLimit) then begin

    ISODAT_Step_Ten, Means, MaxStdDev, StdDevVector, MaxStdDevVector,$
            AverageDistance, OverallAvgDist, ClusterList,$
            CurrClusters, MaxClusters, MinDistance, BandMax, Split
    endif

  endif; SkipToEleven

endrep until (Split eq 0)   ;repeat step 2 - 10 until none split



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;ISODAT_Step_Eleven;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

if (CurrClusters gt 1) then begin
  MeansDistances=ISODAT_Step_Eleven_A(Means, ClusterList, InvCovMat)
endif else begin
  MeansDistances=-1
endelse



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;ISODAT_Step_Twelve;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


if ( (MaxPair) && (MeansDistances[0] ne -1) ) then begin
   ;pukes out when MaxPair is 0
   MeansToLump=ISODAT_Step_Twelve(MeansDistances, MaxPair, MinDistance)
endif else begin
   MeansToLump=-1
endelse



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;ISODAT_Step_Thirteen;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

if (MeansToLump[0] ne -1) then begin
   ISODAT_Step_Thirteen, ClusterList, Means, MeansToLump, CurrClusters
endif

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;ISODAT_Step_Fourteen;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Terminate=0

;if we've looped our limit, terminate
if (Iter ge RunLimit) then begin
   Terminate=1
endif

;if we've changed less then the percent limit, terminate
```

```
  PercentCheck=total(ClusterList ne OldCluster)
  if (PercentCheck le ChangeLimit) then begin
    Terminate=1
  endif else begin
    OldCluster=ClusterList
  endelse

result=size(HomogIndex, /n_elements)

endrep until (Terminate eq 1)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;ok, a little diagnostics....

clus_histo=histogram(ClusterList, omin=ohmi, omax=ohma)

if ((ohmi ne 0) || (ohma ne (CurrClusters-1)) ) then begin
  print, "After final iteration, there were unassigned pixels - redoing step 2-4 loop"

  repeat begin ;do 2-4 until none deleted in 3
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;ISODAT_Step_Two;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

    ;assign pixels to clusters
    ISODAT_Step_Two_C, Image, HomogIndex, Means, InvCovMat, ClusterList

    ;an iteration is every time that the program goes to step 1 or 2
    Iter=Iter+1

    clus_histo=histogram(ClusterList, omin=ohmi, omax=ohma)

    if ((ohma+1) lt CurrClusters) then begin
      CurrClusters=ohma+1
```

```
  Means=Means[*,0:CurrClusters-1]
endif

if (ohmi ne 0) then begin
  Means=Means[*,ohmi:CurrClusters-1]
  CurrClusters=CurrClusters-ohmi
  ClusterList-=ohmi
endif


;now that all homogeneous pixels are assigned to a cluster
;use the checksum function to see if we have previously had
;this pattern of clusters before. If we have - we are caught
;in a loop. We can use the checksums to determine this, and
;can then break out of the clustering sequence.

;get the Checksum for the current ClusterList and save it
;in the array of Checksums we created.
;checksum32, ClusterList, ClusterChecksum[Iter]


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;ISODAT_Step_Three;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;delete clusters with less than MinMembers in it
WereDeleted=0B
ISODAT_Step_Three, Means, ClusterList, MinMembers, WereDeleted

CurrClusters=size(Means)
if ( size(CurrClusters, /n_elements) eq 4) then begin
  CurrClusters=1
endif else begin
  CurrClusters=CurrClusters[2]
endelse
```

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;ISODAT_Step_Four;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

    ISODAT_Step_Four, Image, HomogIndex, Means, ClusterList

  endrep until (WereDeleted eq 0) ;Step 2-4 loop until none deleted
endif

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

result=size(HomogIndex, /n_elements)

END; PRO ISODATA
```

A.10 STEP4.PRO

```
FUNCTION GetHomogWithinDist, HomogIndex, current_X, current_Y, Distance,$
                    LoResX, LoResY
;Use this function to compute all pixels located within a radius of
;Distance to the specificed pixel. Find which of those are Homogeneous.
;Return that list.

  ;specify Number of angles to measure for our circle - more is better, but
  ;less will use significantly less memory for current_X  and current_Y
  Nang=20


  within_distance=circ(current_X, current_Y, Distance, LoResX, LoResY, Nang)
```

```
  within_distance=setintersection(within_distance,HomogIndex)

  return, within_distance

END; GetHomogWithinDist

FUNCTION GetImageDistance, HomogIndex, spectrum, LoResImg, InvCovMat,$
                within_distance, bands

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;Step4 Part 1 - Pixs w/in Distance;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  num_within_dist=size(within_distance, /n_elements)

  if(within_distance[0] ne -1) then begin
    ;now that we know all homog means within distance, create an array
    ;of Mahalanobis distances between pix and means

    ;first create an array of hires spectrum repeated for each lores pix
    diff_array=rebin(spectrum,bands,num_within_dist)

    ;subtract the lores within distance from this array
    diff_array=temporary(diff_array)-LoResImg[*,within_distance]

    ;replicate it so we can create our full # matrix
    diff_array=rebin(diff_array,bands,num_within_dist*bands,/sample)

    ;pull together uniq_homog for our full # matrix
    uniq_homog=bands*indgen(num_within_dist)

    ;expands the InvCovMat for our full # matrix
    ex_InvCovMat=$
     reform(rebin(InvCovMat,bands,bands,num_within_dist),$
              bands,bands*num_within_dist)
```

```
    ;now # it with the InvCovMat and the rotate of itself
    dist_res=total((diff_array*ex_InvCovMat),1)
    dist_res=$
    rotate(reform(dist_res,bands,num_within_dist),4)#$
    diff_array[*,uniq_homog]

    dist_res=diag_matrix(dist_res)

    dist_res=min(dist_res, mindex, /absolute)
    dist_res=[within_distance[mindex],abs(dist_res)]

    ;at this point, dist_res[0] is the index of the homog pixel
    ;with min val and dist_res[1] is the actual value

  endif else begin
    dist_res=[-1,-1]
  endelse ; are there homog LoRes pix within distance?

  return, dist_res

END; GetImageDistance

FUNCTION GetTreeDistance, Means, bands, InvCovMat, num_means, spectrum

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;Step4 Part 2 - Pix vs Tree;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;first create an array of 15m pix spectra repeated for each 30m means
  diff_array=rebin(spectrum,bands,num_means)

  ;next subtract out the Means
  diff_array=temporary(diff_array)-Means
```

```
  ;replicate it so we can create our full # matrix
  diff_array=rebin(diff_array,bands,num_means*bands,/sample)

  ;pull together uniq_homog for our full # matrix
  uniq_homog=bands*indgen(num_means)

  ;expands the InvCovMat for our full # matrix
  ex_InvCovMat=$
   reform(rebin(InvCovMat,bands,bands,num_means),bands,bands*num_means)

  ;now # it with the InvCovMat and the rotate of itself
  means_res=total((diff_array*ex_InvCovMat),1)
  means_res=rotate(reform(means_res,bands,num_means),4)#$
            diff_array[*,uniq_homog]

  means_res=diag_matrix(means_res)

  means_res=min(means_res,mindex, /absolute)
  means_res=[mindex,abs(means_res)]

  ;at this point, means_res[0] is the # of the Means
  ;with min val and means_res[1] is the actual value
  return, means_res


END; GetTreeDistance

FUNCTION SuperRes_OneBand, VNIRMeans,SWIRMeans,$
                VNIRtoSWIR,HomogIndex,VNIR_15m, VNIR_30m, SWIR_30m,$
                        VNIRInvCovMat, SWIRInvCovMat, X_15m, Y_15m, mag_ratio, Distance
;this function calculates the initial spectra for 15m pixels in the SWIR
;bands. It checks first for matches within an initial distance (#pixels)
;and then checks for closer matches within the VNIR-SWIR tree
;
;takes:
;VNIRMeans,SWIRMeans - Means calculated by ISODATA (cluster means)
;VNIRtoSWIR - the SWIR clusters for each VNIR cluster - there are 2*#VNIR
```

```
;           clusters. The first entry is the first SWIR cluster for that
;           VNIR cluster, and the second entry is the last SWIR cluster -
;           VNIR clusters with 1 SWIR cluster will have value recorded in
;           both boxes as a result
;HomogIndex - the 30m resolution index of homogeneous pixels (not sure I need)
;VNIR_15m, VNIR_30m, SWIR_30m  - the 15m and 30m resolution images
;X_15m, Y_15m - the size of the 2D image at 15m (VNIR) resolution
;
;key variables:
;
;SWIR_15m - the 15m SWIR image
;xxbands, xxmeansize - used to store results from size()
;mag_ratio - how super is our resolution (2x to go from 30m to 15m)

  ;create our initial variables
  VNIRbands=size(VNIR_15m)
  VNIRbands=VNIRbands[1]

  SWIRbands=size(SWIR_30m)
  SWIRbands=SWIRbands[1]

  X_15m=float(X_15m)
  Y_15m=float(Y_15m)

  appended_bands=2

  SWIR_15m=make_array((2*SWIRbands)+appended_bands,(X_15m)*(Y_15m),value=-1.0)

  ;commented out for looping - 4 Oct 09
  ;;find out what distance they want to use for the first stage
  ;read, Distance,prompt='How many pixels away at native resolution to search for matches [default is 10]: '

  ;the 15m VNIR spectra is now compared to the 30m homogeneous pixels within
  ;Distance (#30m pix) of it. The SWIR spectrum colocated with the 30m VNIR
  ;pix with the minimum Mahalanobis distnace is is initially assigned to the
  ;15m SWIR pix, although it may change if there is a closer match in the tree.
```

```
;create arrays of size X_15m*Y_15m that has the current X and Y locations
cur_X=floor((reform(rebin(indgen(X_15m),X_15m,1,Y_15m),$
        X_15m*Y_15m,1))/mag_ratio)
cur_Y=floor((reform(rebin(rotate(indgen(Y_15m),4),X_15m,Y_15m,1),$
        X_15m*Y_15m,1)) / mag_ratio)


;calculate out the size of our VNIRMeans (how many do we have?) here
;so we don't have to do it thousands of times in the for loop
VNIR_num=size(VNIRMeans)
if ( size(VNIR_num, /n_elements) eq 4) then begin
  VNIR_num=1
endif else begin
  VNIR_num=VNIR_num[2]
endelse


;for each 15_m pix, check within Distance of it and then within the tree
for cp=0.0, ((X_15m*Y_15m)-1), 1 do begin

  if (~(cp mod (250*X_15m))) then begin
    flush,0
    print, "Started row ", (cp /(X_15m))+1, " of ", Y_15m, "at ", systime()
  endif

  ;find closest MahaDist homog pixel within range:

  ;first, define all homog pixels that are within the range
  within_distance=GetHomogWithinDist(HomogIndex, cur_X[cp], cur_Y[cp],$
                    Distance, X_15m/mag_ratio, Y_15m/mag_ratio)

  ;image_dist[0] will be the index of the homog pixel within range
  ;with the lowest Mahalanobis Distance, and image_dist[1] will be
  ;the actual distance
  image_dist=GetImageDistance(HomogIndex, VNIR_15m[*,cp], VNIR_30m,$
                    VNIRInvCovMat, within_distance, VNIRbands)
```

```
;find closest (Maha Dist) cluster center:

;tree_dist[0] will be the index of the cluster with the lowest
;Mahalanobis Distance, and clust_dist[1] will be the actual distance
tree_dist=GetTreeDistance(VNIRMeans, VNIRbands, VNIRInvCovMat,VNIR_num,$
                          VNIR_15m[*,cp])


;If image_dist exists and has a lower value, assign it and go to the
;next pixel. Otherwise, find the SWIR spectra associated with the
;VNIR means from tree_dist that is closest to the current spectra
;and assign it.

;explicitly using a le rather than the lt that may be specified by
;Tonooka - I think in cases of a tie, preferences should go to "real"
;spectra rather than cluster centers

if ((image_dist[0] ne -1) && (image_dist[1] le tree_dist[1])) then begin
  SWIR_15m[0,cp]=SWIR_30m[*,image_dist[0]]

  ;set the fake 7th band to indicate source - 0 is from circle
  SWIR_15m[SWIRbands,cp]=0

  ;set the fake 8th band to indicate Mahalanobis distance
  SWIR_15m[SWIRbands+1,cp]=image_dist[1]

endif else begin

  ;ok, if we're here we need to figure out which of the associated
  ;SWIR cluster means is closest (Mahalanobis-wise) to the current
  ;pixel, cp.

  min_clus=VNIRtoSWIR[(2*tree_dist[0])]
  max_clus=VNIRtoSWIR[(2*tree_dist[0])+1]
```

```
        ;branch_dist[0] will be the index of the cluster with the lowest
        ;Mahalanobis Distance, and clust_dist[1] will be the actual distance
        branch_dist=GetTreeDistance(SWIRMeans[*,min_clus:max_clus],$
              SWIRbands, SWIRInvCovMat,(max_clus-min_clus)+1,$
              SWIR_30m[*,((cur_Y[cp]*(X_15m/mag_ratio))+cur_X[cp])])


        ;branch_dist[0]+min_clus should be the right means to assign
        SWIR_15m[0,cp]=SWIRMeans[*,min_clus+branch_dist[0]]

        ;set the fake 7th band to indicate source - 255 is from tree
        SWIR_15m[SWIRbands,cp]=255

        ;set the fake 8th band to indicate Mahalanobis distance
        SWIR_15m[SWIRbands+1,cp]=tree_dist[1]

      endelse; Are we dist_res or means_res?

   endfor; cp

return, SWIR_15m

END; SuperRes_SWIR
```

A.11 STEP5.PRO

```
FUNCTION ZeroLines, SuperRes_Image, X_size, Y_size
;this function expects to receive an image in BIP format - 1 column of
;data per band, as well as the X and Y dimensions for the image when in
;2D format.

zeroline=findgen(X_size)
SuperRes_Image[*,zeroline]=0
```

```
;bottom row
zeroline+=(X_size*(Y_size-1.0))
SuperRes_Image[*,zeroline]=0

;first column
zeroline=findgen(Y_size)
zeroline*=X_size
SuperRes_Image[*,zeroline]=0

;last column
zeroline+=(X_size-1.0)
SuperRes_Image[*,zeroline]=0

return, SuperRes_Image

END; FUNCTION ZeroLines

FUNCTION ModifySuperRes, SWIR_15m, SWIR_30m, PSF_S, X_15m, Y_15m, mag_ratio
;since the original (30m) SWIR image is a "measurement" the 15m image
;needs to be compatible with it. This modificiation  is weighted by the
;distance
;
;SWIR_15m 0-5 are the bands
;SWIR_15m 6 is the source map (spectra is from either the image or the tree)
;SWIR_15m 7 is the distance map
;SWIR_15m 8-13 are the correction maps
;
;new SWIR_15m sub k = SWIR_15m sub k + alpha sub k  X distance map

backup_orig=SWIR_15m

swirbands=size(SWIR_15m)
swirbands=(swirbands[1] / 2.0) - 1   ;swirbands=6

distance_map=SWIR_15m[swirbands+1,*]

;back to code pre-dating the HI trip in 11/06 now
```

```
distance_map=reform(distance_map, X_15m, Y_15m)

convol_distance_map=frebin(distance_map, X_15m/mag_ratio, Y_15m/mag_ratio)
convol_distance_map=convolve(convol_distance_map, PSF_S)

correction=make_array(X_15m, Y_15m, swirbands)

;swirbands-1 b/c count from 0
for which_band=0, swirbands-1, 1 do begin
  ;pull together our images first
  orig_15m_image=SWIR_15m[which_band,*]
  orig_15m_image=reform(orig_15m_image, X_15m, Y_15m)

  orig_30m_image=SWIR_30m[which_band,*]
  orig_30m_image=reform(orig_30m_image, X_15m/mag_ratio, Y_15m/mag_ratio)

  ;do our convols
  convol_15m=frebin(orig_15m_image, X_15m / mag_ratio, Y_15m / mag_ratio)
  convol_15m=convolve(convol_15m, PSF_S)

  ;calculate alpha. think of alpha as offset per maha distance
  ;we convolve both the distance map and the hires data to lores
  ;find the difference from the original image, and divide that by
  ;the convolved distance amount.

  ;ok, trying to clean up code 22 Sep 07. We only multiply and divide by
  ;the distance map. If our distance is 0, things become upset. But, if
  ;distance is 0, we really don't need any sort of correction.
  bs=where(convol_distance_map eq 0.0)

  if (bs[0]+1) then convol_distance_map[bs]=1.0

  alpha=((orig_30m_image - convol_15m)/convol_distance_map)

  ;any place that had a convol_distance_map value of 0 shouldn't
  ;be corrected. Set all my bs values to 0. Do same for distance
  if (bs[0]+1) then alpha[bs]=0.0
```

```
   if (bs[0]+1) then convol_distance_map[bs]=0.0

  ; alpha=((orig_30m_image - convol_15m)/convol_distance_map)

;   changing to use the finite() function as I'm generating
;   -NaNs and cant find a way to detect them otherwise 19 Aug 2010
   ;at this point some of alpha is set to 'Inf' - reset those to 0
;   infinites=where(alpha eq !VALUES.F_INFINITY)
;   nans=where(alpha eq !VALUES.F_NAN)

;   if (infinites[0] ne -1) then alpha[infinites]=0.0
;   if (nans[0] ne -1) then alpha[nans]=0.0

   bad_values=finite(alpha)
   bad_values=where(alpha eq 0)
   if (bad_values[0] ne -1) then alpha[bad_values]=0.0



   ;alpha is 30m - double it to get to 15m sizing for multiplication

   alpha=frebin(alpha,X_15m, Y_15m)

   ;correction is alpha X d_15m in paper - assume mult

   correction[0,0,which_band]=alpha*distance_map

endfor; which_band

;now that we've calculated our correction - add it to our SWIR_15m

;ok - convert the correction to columnar form
correction=rotate(reform(correction,X_15m*Y_15m,swirbands),4)

;now add the correction to the columns that are SWIR bands
SWIR_15m[0:swirbands-1,*]+=correction

;round it - DNs are only integers
```

```
SWIR_15m[0:swirbands-1,*]=round(SWIR_15m[0:swirbands-1,*])

;record our correction as well
;SWIR_15m 8 - 13 are our correction (for ASTER)
SWIR_15m[swirbands+2:((2*swirbands)+1),*]=correction

return, SWIR_15m

END
```

A.12 STEP7.PRO

```
;STEP 7 -
;
; I am not certain about the 'band average of the spatial standard deviation
; over the whole image of each VNIR band' part. In the main program, I define
; my threshold as the average of the stddev() result for each band's whole
; image.

PRO Homog_Pix_SWIR, band4, band5, band6, band7, band8, band9, thresh,$
                    mag_ratio, X_30m, Y_30m, homog_map

; variables used in this procedure
; result - stores result of size() function
; max_x, max_y - size of X and Y in image
; x, y - loop variables to iterate of 2D image

  max_x = X_30m - (mag_ratio +1)
  max_y = Y_30m - (mag_ratio +1)

  ;specifying value for ASTER SWIR
  num_bands=6.0

print, "Threshold used for SWIR homogeneity is ", thresh
```

```
;as of Oct 27 2006 I no longer remember why I do not check edge pixels
;however, since I did not in my original Homog_Pix_15m I am leaving it
;that way for now.

  for y = mag_ratio, max_y, mag_ratio do begin
    for x = mag_ratio, max_x, mag_ratio do begin

      if(stddev(band4[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue
      if(stddev(band5[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue
      if(stddev(band6[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue
      if(stddev(band7[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue
      if(stddev(band8[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue
      if(stddev(band9[x:x+(mag_ratio-1),y:y+(mag_ratio-1)]) ge thresh)$
       then continue

      homog_map[x / mag_ratio, y / mag_ratio] = 1

    endfor; x_loop
  endfor ;y_loop

END
```

A.13 STEP9.PRO

```
FUNCTION GetImageTwoBandsDistance, HomogIndex, VNIRspectrum, VNIRLoRes,$
        VNIRInvCovMat, within_distance, VNIRbands, SWIRspectrum,$
        SWIRLoRes, SWIRInvCovMat, SWIRbands, Weight

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;Step9 Part 1 - Pixs w/in Distance;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  num_within_dist=size(within_distance, /n_elements)

  if(within_distance[0] ne -1) then begin
    ;now that we know all homog means within distance, create an array
    ;of Mahalanobis distances between pix and means

    ;VNIR first
    spectrum=VNIRspectrum
    InvCovMat=VNIRInvCovMat
    bands=VNIRbands
    LoResImg=VNIRLoRes

    ;first create an array of hires spectrum repeated for each lores pix
    diff_array=rebin(spectrum,bands,num_within_dist)

    ;subtract the lores within distance from this array
    diff_array-=LoResImg[*,within_distance]

    ;replicate it so we can create our full # matrix
    diff_array=rebin(diff_array,bands,num_within_dist*bands,/sample)

    ;pull together uniq_homog for our full # matrix
```

```
uniq_homog=bands*indgen(num_within_dist)

;expands the InvCovMat for our full # matrix
ex_InvCovMat=$
  reform(rebin(InvCovMat,bands,bands,num_within_dist),$
            bands,bands*num_within_dist)

;now # it with the InvCovMat and the rotate of itself
dist_res=total((diff_array*ex_InvCovMat),1)
dist_res=$
rotate(reform(dist_res,bands,num_within_dist),4)#$
diff_array[*,uniq_homog]

dist_res=diag_matrix(dist_res)

;SWIR second
spectrum=SWIRspectrum
InvCovMat=SWIRInvCovMat
bands=SWIRbands
LoResImg=SWIRLoRes



;first create an array of hires spectrum repeated for each lores pix
sdiff_array=rebin(spectrum,bands,num_within_dist)

;subtract the lores within distance from this array
sdiff_array=temporary(sdiff_array)-LoResImg[*,within_distance]

;replicate it so we can create our full # matrix
sdiff_array=rebin(sdiff_array,bands,num_within_dist*bands,/sample)

;pull together uniq_homog for our full # matrix
uniq_homog=bands*indgen(num_within_dist)

;expands the InvCovMat for our full # matrix
ex_InvCovMat=$
  reform(rebin(InvCovMat,bands,bands,num_within_dist),$
```

```
            bands,bands*num_within_dist)

  ;now # it with the InvCovMat and the rotate of itself
  sdist_res=total((sdiff_array*ex_InvCovMat),1)
  sdist_res=$
  rotate(reform(sdist_res,bands,num_within_dist),4)#$
  sdiff_array[*,uniq_homog]

  sdist_res=diag_matrix(sdist_res)



  ;re-examining code on 15 Sep 07 - I am not sure why
  ;I am dividing the weight factor by the number of bands
  ;as that seems to make _no_ sense. As an example, I think
  ;my default ASTER weighting of 0.7VNIR/0.3SWIR actually
  ;ends up as weighting 0.23VNIR/0.05SWIR (or 82%/18%)
  ;
  ;After thinking about it, I looked at Tonooka's paper - I
  ;have Weight/VNIRbands and (1-W)/SWIRbands because thats what
  ;he says. I don't understand. This is not the first time that
  ;I've felt that way. *sigh*

  dist_res*=(Weight/VNIRbands)
  sdist_res*=((1-Weight)/SWIRbands)

  dist_res=abs(dist_res)+abs(sdist_res)



  ;no abs() or /abs below - already set abs() above this
  dist_res=min(dist_res,mindex)
  dist_res=[within_distance[mindex],dist_res]

  ;at this point, dist_res[0] is the index of the homog pixel
  ;with min val and dist_res[1] is the actual value

endif else begin
  dist_res=[-1,10000000.0]
```

```
  endelse ; are there homog LoRes pix within distance?

  return, dist_res

END; GetImageTwoBandsDistance


FUNCTION SuperRes_TwoBands, VNIRMeans,SWIRMeans, TIRMeans,$
                VNIRClusters,SWIRClusters,TIRClusters,$
                VNIRtoSWIR,SWIRtoTIR,HomogIndex,$
                VNIR_15m,VNIR_90m,Orig_SWIR_15m,SWIR_90m,TIR_90m,$
                VNIRInvCovMat,SWIRInvCovMat,TIRInvCovMat,$
                X_15m,Y_15m, mag_ratio, Distance, Weight

;this function calculates the initial spectra for 15m pixels in the SWIR
;bands. It checks first for matches within an initial distance (#pixels)
;and then checks for closer matches within the VNIR-SWIR tree
;
;takes:
;VNIRMeans,SWIRMeans - Means calculated by ISODATA (cluster means)
;VNIRClusters, SWIRClusters - the cluster assignments in VNIR and SWIR
;VNIRtoSWIR - the SWIR clusters for each VNIR cluster - there are 2*#VNIR
;       clusters. The first entry is the first SWIR cluster for that
;       VNIR cluster, and the second entry is the last SWIR cluster -
;       VNIR clusters with 1 SWIR cluster will have value recorded in
;       both boxes as a result
;HomogIndex - the 90m resolution index of homogeneous pixels
;VNIR_15m, VNIR_30m, SWIR_30m  - the 15m and 30m resolution images
;X_15m, Y_15m - the size of the 2D image at 15m (VNIR) resolution
;
;key variables:
;
;TIR_15m - the 15m SWIR image
;xxbands, xxmeansize - used to store results from size()
;mag_ratio - how super is our resolution (6x to go from 90m to 15m)

  ;create our initial variables
```

```
appended_bands=2.0; we add a source map and a distance map to 15m data

VNIRbands=size(VNIR_15m)
VNIRbands=VNIRbands[1]

;passed in SWIR_15m as Orig_SWIR_15m so I could really only deal with
;with my data bands and not my correction maps, source, or distance map
SWIRbands=size(Orig_SWIR_15m)
SWIRbands=(SWIRbands[1] / 2.0)-1.0
SWIR_15m=Orig_SWIR_15m[0:SWIRbands-1,*]  ;-1 since still counting from 0

TIRbands=size(TIR_90m)
TIRbands=TIRbands[1]

X_15m=float(X_15m)
Y_15m=float(Y_15m)

TIR_15m=make_array((2*TIRbands)+appended_bands,(X_15m)*(Y_15m),value=-1.0)

;the 15m VNIR spectra is now compared to the 90m homogeneous pixels within
;Distance <# 90m pix> of it. The TIR spectrum colocated with the 90m VN/SWIR
;pix with the minimum Mahalanobis distance is initially assigned to the
;15m TIR pix, although it may change if there is a closer match in the tree.

;;;;;;;;;;;;;;
;;;;;;;;;;;;;;
;code from Newsgroup (NOT MINE!) to find members within a circle - first snip
;is over the entire image, then you tell it later from which point and distance
;specify Number of angles to measure for our circle - more is better, but
;less will use significantly less memory for current_X  and current_Y
Nang=20

;create arrays of size X_15m*Y_15m that has the current X and Y locations
cur_X=floor((reform(rebin(indgen(X_15m),X_15m,1,Y_15m),$
        X_15m*Y_15m,1))/mag_ratio)
cur_Y=floor((reform(rebin(rotate(indgen(Y_15m),4),X_15m,Y_15m,1),$
        X_15m*Y_15m,1)) / mag_ratio)
```

```
;;;;;;;;;;;;;;
;;;;;;;;;;;;;;



;for each 15_m pix, check within Distance of it and then within the tree
for cp=0.0, ((X_15m*Y_15m)-1), 1 do begin

  flush,0
  if (~(cp mod (250*X_15m))) then begin
    print, "Started row ", (cp /(X_15m))+1, " of ", Y_15m
  endif



  ;find closest MahaDist homog pixel within range:

  ;first, define all homog pixels that are within the range
  within_distance=GetHomogWithinDist(HomogIndex, cur_X[cp], cur_Y[cp],$
                    Distance, X_15m/mag_ratio, Y_15m/mag_ratio)

  ;image_dist[0] will be the index of xxx_90m within range
  ;with the lowest Mahalanobis Distance that is homogeneous, and
  ;image_dist[1] will be the actual distance
  image_dist=GetImageTwoBandsDistance(HomogIndex, VNIR_15m[*,cp],$
                    VNIR_90m, VNIRInvCovMat, within_distance,$
                    VNIRbands, SWIR_15m[*,cp], SWIR_90m,$
                    SWIRInvCovMat, SWIRbands, Weight)

  ;find closest (Maha Dist) cluster center:

  ;I don't think there's a pretty way to do this. I need to calculate
  ;out my distances to all vnir clusters and to all swir clusters
  ;thats a lot of calls to GetTreeDistance and at least 1 for loop.
  ;I'm going to re-use a lot of my code from the original step9.pro
  ;(unlike step4.pro) to do that.

  ;How many VNIRmeans do I have?
```

```
num_means=size(VNIRMeans)
if ( size(num_means, /n_elements) eq 4) then begin
  num_means=1
endif else begin
  num_means=num_means[2]
endelse

;set up a var to store my Maha Dist
tree_dist=image_dist

for current_VNIRMeans=0, num_means-1, 1 do begin

  ;calculate out my Maha Dist to the current VNIR cluster
  VNIR_dist=VNIR_15m[*,cp] - VNIRMeans[*,current_VNIRMeans]
  VNIR_dist=(VNIR_dist#VNIRInvCovMat)#VNIR_dist


  ;multiply by my weighting factor here
  VNIR_dist*=(Weight/VNIRbands)

  ;set which SWIR clusters to check
  min_clus=VNIRtoSWIR[(2*current_VNIRMeans)]
  max_clus=VNIRtoSWIR[(2*current_VNIRMeans)+1]

  ;branch_dist[0] will be the index of the cluster with the lowest
  ;Mahalanobis Distance, and clust_dist[1] will be the actual distance
  ;
  ;I've got to be honest here. I dont really know what the heck I'm
  ;doing in the SWIR_90m reference below. Its cool though! When did
  ;I write that code? Did I ever double-check it to make sure it was
  ;accurate? I suspect I wrote it within 24 hrs of this comment =(
  ;
  branch_dist=GetTreeDistance(SWIRMeans[*,min_clus:max_clus],$
        SWIRbands, SWIRInvCovMat,(max_clus-min_clus)+1,$
        SWIR_90m[*,((cur_Y[cp]*(X_15m/mag_ratio))+cur_X[cp])])
```

```
  ;branch_dist[0]+min_clus is swir cluster of minumum Maha
  ;branch_dist[1] is an unweighted Maha Dist though; fix that
  branch_dist[0]+=min_clus
  branch_dist[1]*=( (1-Weight) /SWIRbands )
  branch_dist[1]+=VNIR_dist

  ;ok, check my distances. If I'm closer, replace tree_dist
  if (branch_dist[1] lt tree_dist[1]) then tree_dist=branch_dist
endfor; current_VNIRMeans



;at this point, if there is a better spot in the tree, we have it
;if tree_dist lt image_dist, we need to find the TIR
;spectra associated with it that is closest, else if it doesn't exist
;assign dist_res to the current pixel.

if (tree_dist[1] ge image_dist[1]) then begin
  ;image is closer; record spectra from image match
  TIR_15m[0,cp]=TIR_90m[*,image_dist[0]]

  ;set the fake 6th band to indicate source - 0 is from circle
  TIR_15m[TIRbands,cp]=0

  ;set the fake 7th band to indicate Mahalanobis distance
  TIR_15m[TIRbands+1,cp]=image_dist[1]
endif else begin
  ;ok, if we're here we need to figure out which of the associated
  ;TIR cluster means is closest (Mahalanobis-wise) to the current
  ;pixel, cp.

  min_clus=SWIRtoTIR[(2*branch_dist[0])]
  max_clus=SWIRtoTIR[(2*branch_dist[0])+1]

  ;if tree is VNIR, and branch is SWIR.....twig for TIR?
  twig_dist=GetTreeDistance(TIRMeans[*,min_clus:max_clus],$
        TIRbands, TIRInvCovMat,(max_clus-min_clus)+1,$
        TIR_90m[*,((cur_Y[cp]*(X_15m/mag_ratio))+cur_X[cp])])
```

```
        ;mindex+min_clus should be the right means to assign
        TIR_15m[0,cp]=TIRMeans[*,min_clus+twig_dist[0]]

        ;set the fake 6th band to indicate source - 255 is from tree
        TIR_15m[TIRbands,cp]=255

        ;set the fake 7th band to indicate Mahalanobis distance
        TIR_15m[TIRbands+1,cp]=branch_dist[1]
      endelse; tree_dist vs image_dist

   endfor; cp

return, TIR_15m

END; SuperRes_TwoBands
```

A.14 STEP10.PRO

```
FUNCTION ModifySuperRes_T, TIR_15m, TIR_90m, PSF_T, X_15m, Y_15m, mag_ratio
;since the original (30m) SWIR image is a "measurement" the 15m image
;needs to be compatible with it. This modificiation  is weighted by the
;distance
;
;TIR_15m 0-4 are the bands
;TIR_15m 5 is the source map (spectra is from either the image or the tree)
;TIR_15m 6 is the distance map
;TIR_15m 7-11
;
;new TIR_15m sub k = TIR_15m sub k + alpha sub k  X distance map

backup_orig=TIR_15m
```

```
tirbands=size(TIR_15m)
tirbands=(tirbands[1] / 2.0)-1.0

distance_map=TIR_15m[tirbands+1,*]

;back to code pre-dating the HI trip in 11/06 now
distance_map=reform(distance_map, X_15m, Y_15m)
convol_distance_map=frebin(distance_map, X_15m/mag_ratio, Y_15m/mag_ratio)
convol_distance_map=convolve(convol_distance_map, PSF_T)



;create array to store correction for each band
correction=make_array(X_15m, Y_15m, tirbands)

for which_band=0, tirbands-1, 1 do begin
  ;pull together our images first
  orig_15m_image=TIR_15m[which_band,*]
  orig_15m_image=reform(orig_15m_image, X_15m, Y_15m)

  orig_90m_image=TIR_90m[which_band,*]
  orig_90m_image=reform(orig_90m_image, X_15m/mag_ratio, Y_15m/mag_ratio)

  ;do our convols
  convol_15m=frebin(orig_15m_image, X_15m / mag_ratio, Y_15m / mag_ratio)
  convol_15m=convolve(convol_15m, PSF_T)

  ;calculate alpha. think of alpha as offset per maha distance
  ;we convolve both the distance map and the hires data to lores
  ;find the difference from the original image, and divide that by
  ;the convolved distance amount.

  ;ok, trying to clean up code 22 Sep 07. We only multiply and divide by
  ;the distance map. If our distance is 0, things become upset. But, if
  ;distance is 0, we really don't need any sort of correction.
  bs=where(convol_distance_map eq 0.0)
```

```
  if (bs[0]+1) then convol_distance_map[bs]=1.0

  alpha=((orig_90m_image - convol_15m)/convol_distance_map)

  ;any place that had a convol_distance_map value of 0 shouldn't
  ;be corrected. Set all my bs values to 0. Do same for distance
  if (bs[0]+1) then alpha[bs]=0.0
  if (bs[0]+1) then convol_distance_map[bs]=0.0

  ;at this point some of alpha is set to 'Inf' - reset those to 0
  infinites=where(alpha eq !VALUES.F_INFINITY)
  nans=where(alpha eq !VALUES.F_NAN)

  if (infinites[0] ne -1) then alpha[infinites]=0.0
  if (nans[0] ne -1) then alpha[nans]=0.0

  ;alpha is 90m - sextuple it to get to 15m sizing for multiplication

  alpha=frebin(alpha,X_15m, Y_15m)

  ;correction is alpha X d_15m in paper - assume multiply??
  correction[0,0,which_band]=alpha*distance_map

endfor; which_band

;now that we've calculated our correction - add it to our TIR_15m

;ok - convert the correction to columnar form
correction=rotate(reform(correction,X_15m*Y_15m,tirbands),4)

;now add the correction to the columns that are TIR bands
TIR_15m[0:tirbands-1,*]+=correction

;round it - DNs are only integers - byte for VNIR, SWIR and UINT for TIR
TIR_15m[0:tirbands-1,*]=round(TIR_15m[0:tirbands-1,*])

;record our correction as well
```

```
;TIR_15m 7-11 are our correction (for ASTER)
TIR_15m[tirbands+2:((2*tirbands)+1),*]=correction

return, TIR_15m

END
```

A.15 POSTSTEP10.PRO

```
PRO WriteOutASTERData, tir_15m, homog_90m, swir_15m, homog_30m, $
                basestring, Orig_Variable_Value, Orig_Variable_Name
;write out the data we want to keep 'permanently'


  ;when we get around to looping and need to put the loop val in number
  ;use strtrim(loopcount,2) to get it as a string with no spaces

  ;data
  tirfile=basestring+'--tir_15m.dat'
  openw, lun, tirfile, /get_lun
  writeu, lun, tir_15m
  free_lun, lun

  ;header
  header_tirfile=basestring+'--tir_15m.hdr'
  openw, lun, header_tirfile, /get_lun
  printf, lun, "ENVI"
  printf, lun, "description = {"
  printf, lun, "     super-res loop "+StrTrim(Orig_Variable_Value[94],2)+$
            " of "+StrTrim(round(Orig_Variable_Value[95]),2)+". Variable "+$
            Orig_Variable_Name[Orig_Variable_Value[99]]+" has value "+$
            StrTrim(Orig_Variable_Value[Orig_Variable_Value[99]],2)+". }"
  printf, lun, "samples = "+StrTrim(round(Orig_Variable_Value[2]),2)
  printf, lun, "lines = "+StrTrim(round(Orig_Variable_Value[3]),2)
```

```
printf, lun, "bands = "+StrTrim(round(((3*Orig_Variable_Value[5])+2)),2)
printf, lun, "file type = ENVI Standard"
printf, lun, "data type = 4"
printf, lun, "interleave = bip"
free_lun, lun

tirhomogfile=basestring+'--tir_homog_map.dat'
openw, lun, tirhomogfile, /get_lun
writeu, lun, byte(homog_90m)
free_lun, lun

swirfile=basestring+'--swir_15m.dat'
openw, lun, swirfile, /get_lun
writeu, lun, swir_15m
free_lun, lun

header_swirfile=basestring+'--swir_15m.hdr'
openw, lun, header_swirfile, /get_lun
printf, lun, "ENVI"
printf, lun, "description = {"
printf, lun, "     super-res loop "+StrTrim(round(Orig_Variable_Value[94]),2)+$
            " of "+StrTrim(round(Orig_Variable_Value[95]),2)+". Variable "+$
            Orig_Variable_Name[Orig_Variable_Value[99]]+" has value "+$
            StrTrim(Orig_Variable_Value[Orig_Variable_Value[99]],2)+". }"
printf, lun, "samples = "+StrTrim(round(Orig_Variable_Value[2]),2)
printf, lun, "lines = "+StrTrim(round(Orig_Variable_Value[3]),2)
printf, lun, "bands = "+StrTrim(round(((3*Orig_Variable_Value[5])+2)),2)
printf, lun, "file type = ENVI Standard"
printf, lun, "data type = 4"
printf, lun, "interleave = bip"
free_lun, lun


swirhomogfile=basestring+'--swir_homog_map.dat'
openw, lun, swirhomogfile, /get_lun
writeu, lun, byte(homog_30m)
free_lun, lun
```

```
END; WriteOutASTERData

PRO WriteOutTHEMISData, tir_v, homog, VNIRClusters, TIRClusters, $
                basestring, Orig_Variable_Value, Orig_Variable_Name
;write out the data we want to keep 'permanently'



  ;when we get around to looping and need to put the loop val in number
  ;use strtrim(loopcount,2) to get it as a string with no spaces


  tirfile=basestring+'--tir_15m.dat'
  openw, lun, tirfile, /get_lun
  writeu, lun, tir_v
  free_lun, lun


  q=tir_v
  q_bands=size(tir_v)
  q_bands=(q_bands[1]/2)-1
  ;set the band values back to original scale
  q[0:q_bands-1,*]/=10000000.0
  ;band q[q,*] should be source, and not need correcting
  ;set the correction bands to the same scale as original data
  q[q_bands+1:(2*q_bands)+1,*]/=10000000.0
  tirfile=basestring+'--tir-mod_15m.dat'
  openw, lun, tirfile, /get_lun
  writeu, lun, q
  free_lun, lun

;header
  header_tirfile=basestring+'--tir-mod_15m.hdr'
  openw, lun, header_tirfile, /get_lun
  printf, lun, "ENVI"
  printf, lun, "description = {"
```

```
printf, lun, "     super-res loop "+StrTrim(round(Orig_Variable_Value[94]),2)+$
          " of "+StrTrim(round(Orig_Variable_Value[95]),2)+". Variable "+$
          Orig_Variable_Name[Orig_Variable_Value[99]]+" has value "+$
          StrTrim(Orig_Variable_Value[Orig_Variable_Value[99]],2)+". }"
printf, lun, "samples = "+StrTrim(round(Orig_Variable_Value[2]),2)
printf, lun, "lines = "+StrTrim(round(Orig_Variable_Value[3]),2)
printf, lun, "bands = "+StrTrim(round(((2*Orig_Variable_Value[5])+2)),2)
printf, lun, "file type = ENVI Standard"
printf, lun, "data type = 4"
printf, lun, "interleave = bip"
free_lun, lun


;when I pass V_T_Homog, its all 0s and 1s - make 0s and 255s
q=homog
q*=255
tirhomogfile=basestring+'--tir_homog_map.dat'
openw, lun, tirhomogfile, /get_lun
writeu, lun, byte(q)
free_lun, lun

;and put out cluster maps
homogaddresses=where(homog ne 0)
q=float(homog)
vclus=VNIRClusters
tclus=TIRClusters
vclus+=1.0
tclus+=1.0

q[homogaddresses]=1.0 ;just to be certain
q[homogaddresses]=vclus

vclusfile=basestring+'--vnir_cluster_map.dat'
openw, lun, vclusfile, /get_lun
writeu, lun, byte(q)
free_lun, lun
```

```
  q[homogaddresses]=1.0 ;just to be certain
  q[homogaddresses]=tclus

  tclusfile=basestring+'--tir_cluster_map.dat'
  openw, lun, tclusfile, /get_lun
  writeu, lun, float(q)
  free_lun, lun


END; WriteOutTHEMISData

FUNCTION CalibrateTIRRadiance, tir_img
;convert the super-resolved DN to calibrated radiance
;we're going to use the values given on the ASTER website
;as of 24 Sep 07
;
;Radiance=(DN value - 1) * Unit Conversion Coefficient
;
;band 10 - 6.882 x 10^-3
;band 11 - 6.780 x 10^-3
;band 12 - 6.590 x 10^-3
;band 13 - 5.693 x 10^-3
;band 14 - 5.225 x 10^-3

  tir_img-=1

  tir_img[0,*]*=0.006882
  tir_img[1,*]*=0.006780
  tir_img[2,*]*=0.006590
  tir_img[3,*]*=0.005693
  tir_img[4,*]*=0.005225


  return, tir_img

END; CalibrateTIRRadiance
```

```
FUNCTION CalibrateSWIRRadiance, swir_img
;convert the super-resolved DN to calibrated radiance
;we're going to use the values given on the ASTER website
;as of 24 Sep 07
;
;Radiance=(DN value - 1) * Unit Conversion Coefficient
;
;band   high     normal   low-1   low-2
;4      0.1087   0.2174   0.290   0.290
;5      0.0348   0.0696   0.0925  0.409
;6      0.0313   0.0625   0.0830  0.390
;7      0.0299   0.0597   0.0795  0.332
;8      0.0209   0.0417   0.0556  0.245
;9      0.0159   0.0318   0.0424  0.265

  swir_img-=1

  ;assume normal gains for now - need some way to check which gain to use
  swir_img[0,*]*=0.2174
  swir_img[1,*]*=0.0696
  swir_img[2,*]*=0.0625
  swir_img[3,*]*=0.0597
  swir_img[4,*]*=0.0417
  swir_img[5,*]*=0.0318

  return, swir_img

END; CalibrateSWIRRadiance
```

BIBLIOGRAPHY

Adams, J. B., Smith, M. O., Johnson, P. E., 1986. Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander I site. Journal of Geophysical Research, 91, 8098-8112.

Arvidson, R. E., Greeley, R., Malin, M. C., Saunders, R. S., Izenberg, N., Plaut, J. J., Stofan, E. R., Shepard, M. K., 1992. Surface Modification of Venus as Inferred from Magellan Observations of Plains. Journal of Geophysical Research-Planets. 97**,** 13303-13317.

Baldridge, A. M., Farmer, J. D., Moersch, J. E., 2004. Mars remote-sensing analog studies in the Badwater Basin, Death Valley, California. J. Geophys. Res. 109**,** E12006.

Baldridge, A. M., Hook, S. J., Crowley, J. K., Marion, G. M., Kargel, J. S., Michalski, J. L., Thomson, B. J., de Souza, C. R., Bridges, N. T., Brown, A. J., 2009. Contemporaneous deposition of phyllosilicates and sulfates: Using Australian acidic saline lake deposits to describe geochemical variability on Mars. Geophysical Research Letters. 36**,** -.

Ball, G. H., Hall, D. J., 1967. A clustering technique for summarizing multivariate data. Behavioral Science. 12**,** 153-155.

Bandfield, J. L., 2008. High-silica deposits of an aqueous origin in western Hellas Basin, Mars. Geophysical Research Letters. 35**,** -.

Bandfield, J. L., 2009. Effects of surface roughness and graybody emissivity on martian thermal infrared spectra. Icarus. 202**,** 414-428.

Bandfield, J. L., Hamilton, V. E., Christensen, P. R., 2000. A global view of Martian surface compositions from MGS-TES. Science. 287**,** 1626-1630.

Bandfield, J. L., Rogers, D., Smith, M. D., Christensen, P. R., 2004. Atmospheric correction and surface spectral unit mapping using Thermal Emission Imaging System data. Journal of Geophysical Research-Planets. 109**,** Doi 10.1029/2004je002289.

Baumgartner, E., In-situ exploration of mars using rover systems. Citeseer, pp. 2000-5062.

Benison, K. C., LaClair, D. A., 2003. Modern and Ancient Extremely Acid Saline Deposits: Terrestrial Analogs for Martian Environments? Astrobiology. 3**,** 609-618.

Bibring, J. P., Langevin, Y., Gendrin, A., Gondet, B., Poulet, F., Berthe, M., Soufflot, A., Arvidson, R., Mangold, N., Mustard, J., Drossart, P., Team, O., 2005. Mars surface diversity as revealed by the OMEGA/Mars Express observations. Science. 307**,** 1576-1581.

Bishop, J., Dobrea, E., McKeown, N., Parente, M., Ehlmann, B., Michalski, J., Milliken, R., Poulet, F., Swayze, G., Mustard, J., 2008. Phyllosilicate diversity and past aqueous activity revealed at Mawrth Vallis, Mars. Science. 321**,** 830.

Bridges, J. C., Grady, M. M., 2000. Evaporite mineral assemblages in the nakhlite (martian) meteorites. Earth and Planetary Science Letters. 176**,** 267-279.

Bruegge, C. J., Chrien, N. L., Ando, R. R., Diner, D. J., Abdou, W. A., Helmlinger, M. C., Pilorz, S. H., Thome, K. J., 2002. Early validation of the multi-angle Imaging SpectroRadiometer (MISR) radiometric scale. IEEE Transactions on Geoscience and Remote Sensing. 40**,** 1477-1492.

Christensen, P., Gorelick, N., Anwar, S., Dickenshied, S., Edwards, C., Engle, E., 2007. New Insights About Mars From the Creation and Analysis of Mars Global Datasets. AGU Fall Meeting Abstracts. 11**,** 01.

Christensen, P. R., Bandfield, J. L., Hamilton, V. E., Howard, D. A., Lane, M. D., Piatek, J. L., Ruff, S. W., Stefanov, W. L., 2000. A thermal emission spectral library of rock-forming minerals. Journal of Geophysical Research-Planets. 105**,** 9735-9739.

Christensen, P. R., Jakosky, B., Kieffer, H. H., Malin, M. C., McSween, H. Y., Nealson, K., Mehall, G. L., Silverman, S. H., Ferry, S., Caplinger, M., Ravine, M., 2004. The Thermal Emission Imaging System (THEMIS) for the Mars 2001 Odyssey Mission. Space Science Reviews. 110**,** 85-130.

Christensen, P. R., McSween, H. Y., Bandfield, J. L., Ruff, S. W., Rogers, A. D., Hamilton, V. E., Gorelick, N., Wyatt, M. B., Jakosky, B. M., Kieffer, H. H., Malin, M. C., Moersch, J. E., 2005. Evidence for magmatic evolution and diversity on Mars from infrared observations. Nature. 436**,** 504-509.

Ehlmann, B., Mustard, J., Fassett, C., Schon, S., Head III, J., Des Marais, D., Grant, J., Murchie, S., 2008. Clay minerals in delta deposits and organic preservation potential on Mars. Nature Geoscience. 1**,** 355-358.

Ehlmann, B. L., Mustard, J. F., Swayze, G. A., Clark, R. N., Bishop, J. L., Poulet, F., Marais, D. J. D., Roach, L. H., Milliken, R. E., Wray, J. J., Barnouin-Jha, O., Murchie, S. L., 2009. Identification of hydrated silicate minerals on Mars using MRO-CRISM: Geologic context near Nili Fossae and implications for aqueous alteration. Journal of Geophysical Research-Planets. 114**,** Doi 10.1029/2009je003339.

Ekren, E. B., Quinlivan, W. D., Snyder, R. P., Kleinhampl, F. J., 1974. Stratigraphy, structure, and geologic history of the Lunar Lake caldera of northern Nye County, Nevada. US Geological Survey Journal of Research. 599 - 608.

Fishbaugh, K. E., Poulet, F., Chevrier, V., Langevin, Y., Bibring, J. P., 2007. On the origin of gypsum in the Mars north polar region. Journal of Geophysical Research-Planets. 112**,** -.

Fujisada, H., Sakuma, F., Ono, A., Kudoh, M., 1998. Design and preflight performance of ASTER instrument protoflight model. IEEE Transactions on Geoscience and Remote Sensing. 36**,** 1152-1160.

Gendrin, A., Mangold, N., Bibring, J. P., Langevin, Y., Gondet, B., Poulet, F., Bonello, G., Quantin, C., Mustard, J., Arvidson, R., LeMouelic, S., 2005. Sulfates in martian layered terrains: the OMEGA/Mars Express view. Science. 307**,** 1587-1591.

Gillespie, A., Rokugawa, S., Matsunaga, T., Cothern, J. S., Hook, S., Kahle, A. B., 1998. A temperature and emissivity separation algorithm for Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) images. IEEE Transactions on Geoscience and Remote Sensing. 36**,** 1113-1126.

Glotch, T. D., Bandfield, J. L., Tornabene, L. L., Jensen, H. B., Seelos, F. P., 2010. Distribution and formation of chlorides and phyllosilicates in Terra Sirenum, Mars. Geophys. Res. Lett. 37**,** L16202.

Glotch, T. D., Rogers, A. D., 2007. Evidence for aqueous deposition of hematite- and sulfate-rich light-toned layered deposits in Aureum and Iani Chaos, Mars. Journal of Geophysical Research-Planets. 112**,** Doi 10.1029/2006je002863.

Gooding, J. L., Zolensky, M. E., Wentworth, S. J., 1991. Aqueous alteration of the Nakhla meteorite. Journal Name: Meteoritics; (United States); Journal Volume: 26. Medium: X; Size: Pages: 135-143.

Green, R. O., Eastwood, M. L., Sarture, C. M., Chrien, T. G., Aronsson, M., Chippendale, B. J., Faust, J. A., Pavri, B. E., Chovit, C. J., Solis, M. S., Olah, M. R., Williams, O., 1998. Imaging spectroscopy and the Airborne Visible Infrared Imaging Spectrometer (AVIRIS). Remote Sensing of Environment. 65**,** 227-248.

Hamilton, V. E., Wyatt, M. B., McSween, H. Y., Christensen, P. R., 2001. Analysis of terrestrial and Martian volcanic compositions using thermal emission spectroscopy - 2. Application to Martian surface spectra from the Mars Global Surveyor Thermal Emission Spectrometer. Journal of Geophysical Research-Planets. 106**,** 14733-14746.

Hughes, C. G., Ramsey, M. S., 2010. Super-resolution of THEMIS thermal infrared data: Compositional relationships of surface units below the 100 meter scale on Mars. Icarus. 208**,** 704-720.

Hurowitz, J. A., Fischer, W., Tosca, N. J., Milliken, R. E., 2010. Origin of acidic surface waters and the evolution of atmospheric chemistry on early Mars. Nature Geoscience. 3**,** 323-326.

Iwasaki, A., Tonooka, H., 2005. Validation of a Crosstalk Correction Algorithm for ASTER/SWIR. IEEE Transactions on Geoscience and Remote Sensing. 43, 2747 -2751.

Langevin, Y., Poulet, F., Bibring, J. P., Gondet, B., 2005. Sulfates in the north polar region of Mars detected by OMEGA/Mars express. Science. 307**,** 1584-1586.

Marion, G. M., Farren, R. E., Komrowski, A. J., 1999. Alternative pathways for seawater freezing. Cold Regions Science and Technology. 29**,** 259-266.

McConnochie, T. H., Bell, J. F., Savransky, D., Mehall, G., Caplinger, M., Christensen, P. R., Cherednik, L., Bender, K., Dombovari, A., 2006. Calibration and in-flight performance of the Mars Odyssey Thermal Emission Imaging System visible imaging subsystem (THEMIS VIS). Journal of Geophysical Research-Planets. 111**,** Doi 10.1029/2005je002568.

Michalski, J. R., Reynolds, S. J., Sharp, T. G., Christensen, P. R., 2004. Thermal infrared analysis of weathered granitic rock compositions in the Sacaton Mountains, Arizona: Implications for petrologic classifications from thermal infrared remote-sensing data. J. Geophys. Res. 109**,** E03007.

Milam, K. A., McSween, H. Y., Jr., Moersch, J., Christensen, P. R., 2010. Distribution and variation of plagioclase compositions on Mars. J. Geophys. Res. 115**,** E09004.

Milliken, R. E., Bish, D. L., 2010. Sources and sinks of clay minerals on Mars. Philosophical Magazine. 90**,** 2293-2308.

Milliken, R. E., Fischer, W. W., Hurowitz, J. A., 2009. Missing salts on early Mars. Geophysical Research Letters. 36**,** -.

Mimmack, G. M., Mason, S. J., Galpin, J. S., 2001. Choice of distance matrices in cluster analysis: Defining regions. Journal of Climate. 14**,** 2790-2797.

Murchie, S. L., Mustard, J. F., Ehlmann, B. L., Milliken, R. E., Bishop, J. L., McKeown, N. K., Dobrea, E. Z. N., Seelos, F. P., Buczkowski, D. L., Wiseman, S. M., Arvidson, R. E., Wray, J. J., Swayze, G., Clark, R. N., Marais, D. J. D., McEwen, A. S., Bibring, J. P.,

2009. A synthesis of Martian aqueous mineralogy after 1 Mars year of observations from the Mars Reconnaissance Orbiter. Journal of Geophysical Research-Planets. 114**,** -.

Mustard, J., Murchie, S., Pelkey, S., Ehlmann, B., Milliken, R., Grant, J., Bibring, J., Poulet, F., Bishop, J., Dobrea, E., 2008. Hydrated silicate minerals on Mars observed by the Mars Reconnaissance Orbiter CRISM instrument. Nature. 454**,** 305-309.

Osterloo, M. M., Hamilton, V. E., Bandfield, J. L., Glotch, T. D., Baldridge, A. M., Christensen, P. R., Tornabene, L. L., Anderson, F. S., 2008. Chloride-bearing materials in the southern highlands of Mars. Science. 319**,** 1651-1654.

Pagnutti, M., Ryan, R. E., Kelly, M., Holekamp, K., Zanoni, V., Thome, K., Schiller, S., 2003. Radiometric characterization of IKONOS multispectral imagery. Remote Sensing of Environment. 88**,** 53-68.

Petroy, S., Arvidson, R., Spectral Emissivity of the Silver and Lunar Lake Playas-Relevance to Analyses of Mars TIR Data. Vol. 1, 1990, pp. 237.

Poulet, F., Bibring, J. P., Mustard, J. F., Gendrin, A., Mangold, N., Langevin, Y., Arvidson, R. E., Gondet, B., Gomez, C., Team, O., 2005. Phyllosilicates on Mars and implications for early martian climate. Nature. 438**,** 623-627.

Ramsey, M., Dehn, J., 2004. Spaceborne observations of the 2000 Bezymianny, Kamchatka eruption: the integration of high-resolution ASTER data into near real-time monitoring using AVHRR. Journal of Volcanology and Geothermal Research. 135**,** 127-146.

Ramsey, M. S., Fink, J. H., 1999. Estimating silicic lava vesicularity with thermal remote sensing: A new technique for volcanic mapping and monitoring, Bull. Volcanol., 61, 32–39.

Ramsey, M. S., Christensen, P. R., 1998. Mineral abundance determination: Quantitative deconvolution of thermal emission spectra. Journal of Geophysical Research-Solid Earth. 103**,** 577-596.

Rogers, A. D., Christensen, P. R., 2007. Surface mineralogy of Martian low-albedo regions from MGS-TES data: Implications for upper crustal evolution and surface alteration. J. Geophys. Res. 112**,** E01003.

Rose, S., Ramsey, M., 2009. The 2005 eruption of Kliuchevskoi volcano: Chronology and processes derived from ASTER spaceborne and field-based data. Journal of Volcanology and Geothermal Research. 184**,** 367-380.

Ruff, S. W., Christensen, P. R., Barbera, P. W., Anderson, D. L., 1997. Quantitative thermal emission spectroscopy of minerals: A laboratory technique for measurement and calibration. Journal of Geophysical Research-Solid Earth. 102**,** 14899-14913.

Schubert, B. A., Lowenstein, T. K., Timofeeff, M. N., 2009. Microscopic Identification of Prokaryotes in Modern and Ancient Halite, Saline Valley and Death Valley, California. Astrobiology. 9**,** 467-482.

Stockstill, K. R., Moersch, J. E., McSween, H. Y., Jr., Piatek, J., Christensen, P. R., 2007. TES and THEMIS study of proposed paleolake basins within the Aeolis quadrangle of Mars. J. Geophys. Res. 112**,** E01001.

Thome, K., Schiller, S., Conel, J., Arai, K., Tsuchida, S., 1998. Results of the 1996 Earth Observing System vicarious calibration joint campaign at Lunar Lake Playa, Nevada (USA). Metrologia. 35**,** 631.

Thome, K. J., Helder, D. L., Aaron, D., Dewald, J. D., 2004. Landsat-5 TM and Landsat-7 ETM+ absolute radiometric calibration using the reflectance-based method. IEEE Transactions on Geoscience and Remote Sensing. 42**,** 2777-2785.

Thomson, J. L., Salisbury, J. W., 1993. The mid-infrared reflectance of mineral mixtures (7-14 [mu]m). Remote Sensing of Environment. 45**,** 1-13.

Tonooka, H., Resolution enhancement of ASTER shortwave and thermal infrared bands based on spectral similarity. In: Y. Yoshifumi, G. U. Stephen, Eds.), Vol. 5657. SPIE, 2005, pp. 9-19.

Tosca, N. J., McLennan, S. M., 2006. Chemical divides and evaporite assemblages on Mars. Earth and Planetary Science Letters. 241**,** 21-31.

Townshend, J. R. G., Huang, C., Kalluri, S. N. V., Defries, R. S., Liang, S., Yang, K., 2000. Beware of per-pixel characterization of land cover. International Journal of Remote Sensing. 21**,** 839-843.

Warren, J. K., 2006. Evaporites : sediments, resources and hydrocarbons. Springer, Berlin.

Warren, J. K., 2010. Evaporites through time: Tectonic, climatic and eustatic controls in marine and nonmarine deposits. Earth-Science Reviews. 98**,** 217-268.

Weitz, C. M., Milliken, R. E., Grant, J. A., McEwen, A. S., Williams, R. M. E., Bishop, J. L., Thomson, B. J., 2010. Mars Reconnaissance Orbiter observations of light-toned layered deposits and associated fluvial landforms on the plateaus adjacent to Valles Marineris. Icarus. 205**,** 73-102.

Wentworth, S. J., Gibson, E. K., Velbel, M. A., McKay, D. S., 2005. Antarctic Dry Valleys and indigenous weathering in Mars meteorites: Implications for water and life on Mars. Icarus. 174**,** 383-395.

Wray, J. J., Murchie, S. L., Squyres, S. W., Seelos, F. P., Tornabene, L. L., 2009. Diverse aqueous environments on ancient Mars revealed in the southern highlands. Geology. 37**,** 1043-1046.

Yamaguchi, Y., Kahle, A. B., Tsu, H., Kawakami, T., Pniel, M., 1998. Overview of Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER). IEEE Transactions on Geoscience and Remote Sensing. 36**,** 1062-1071.

Zhukov, B., Oertel, D., Lanzl, F., Reinhackel, G., 1999. Unmixing-based multisensor multiresolution image fusion. IEEE Transactions on Geoscience and Remote Sensing. 37**,** 1212-1226.