BAR-ILAN UNIVERSITY

A SIFT-Based Image Registration Algorithm for Remotely Sensed Data

Benny Kupfer

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Mathematics, Bar-Ilan University.

Ramat-Gan, Israel 2013

BAR-ILAN UNIVERSITY

A SIFT-Based Image Registration Algorithm for Remotely Sensed Data

Benny Kupfer

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Mathematics, Bar-Ilan University.

Ramat-Gan, Israel 2013

This thesis was carried out under the supervision of Prof. Nathan S. Netanyahu of the Department of Computer Science at Bar-Ilan University and Prof. Ilan Shimshoni of the Department of Information Management at Haifa University.

Acknowledgments

First, I would like to thank my dear parents, Ezekiel and Sara, who have brought me thus far. Also, I would like to thank my academic advisor, Prof. Nathan Netanyahu, who introduced me to the problem of image registration and has guided me along the way, and my academic co-advisor, Prof. Ilan Shimshoni, for his insightful and useful suggestions.

Thank you!

Benny

Contents

Abstract

Image registration, which is used to align two images onto a common coordinate system, is an important task in a variety of fields, such as remote sensing, medical imaging, quality assurance and more. The images may differ in acquisition time, view point, sensor, etc., so the problem of image registration is quite involved. Many registration methods use algorithms that discover specific key-points in both the reference and sensed images. Thus the main task is to determine the correspondence between resulting feature sets in some distance sense. After correspondences are established, it is essential to distinguish between true and false matches (i.e., between inlying and outlying correspondences). A transformation model can then be determined with the aid of inlying correspondences , and the quality of the transformation should be assessed in some sense. If it is unsatisfactory, an iterative search should be applied to find a better transformation. Most of these algorithms require exhaustive search for finding an appropriate registration transformation.

In distinction from the above, we propose an efficient algorithm based on features extracted by the *scale invariant feature transform* (SIFT) algorithm. The main difference between our approach and other existing SIFT-based algorithms is the way of distinguishing between true correspondences ("inliers") and false ones ("outliers"). Our determination is based on a mode seeking approach; specifically, we compute the modes of scale ratio, orientation difference, and translation difference histograms resulting in a quadruple of scale, orientation, and horizontal and vertical translations that serves as an initial guess for the desired similarity

transformation. Assuming that a better transformation is in close vicinity to this guess and that most of the correspondences in this vicinity are inliers, we construct an outlier filter, which is a box in 4D centered at the above modes. All correspondences inside the box are considered inliers and are retained, while all other correspondences are considered outliers and are thus rejected. An *ordinary least squares* (OLS) algorithm is then employed with respect to all of the above inliers for computing a refined transformation.

Verification of the transformation's quality is based on a manual procedure in which *ground truth* is provided manually by two sets of corresponding points in the reference and sensed images. We apply the computed transformation to the transformed image set of points and compute the *root mean square error* (RMSE). A registration result is satisfactory if its RMSE is less than one pixel. We have implemented the above algorithm in MATLAB and C and tested it on dozens of remotely sensed image pairs. We observed more than 80% of satisfactory registration results. In addition, we show how the number of resulting inliers may indicate if the registration process results in a success or a failure. We also demonstrated how to improve our results (i.e., recover from registration failures given a-priori information on the image-acquiring sensors) using basic image processing enhancement techniques.

Chapter 1

1. Introduction

1.1 Background and Motivation

Image registration, as the term implies, refers to the alignment of two (or more) images of the same scene or area taken at different times from different views and possibly by different sensors or spectral bands. The meaning of alignment is to overlay the two images onto a common coordinate system, where both images share the same origin. Image registration is important whenever multi-source data need, in some sense, to be fused. Typical applications of image registration involve, among others, the following areas [1], [2]:

- Computer Vision:

A computerized imaging system constantly samples scenes over time; in addition, if the system moves over time or just changes its aspect, then the manufactured images will also have different angles of the scene image; therefore, image registration in this field involves aspects of different times and viewpoints. Typical applications include monitoring time varying images from surveillance/security cameras in order to detect suspected objects (baggage, suitcases, etc.) or human suspects, comparing images of manufactured electronic chips to an image of a reference chip for quality assurance in a production line, comparing images of a predefined route by a fully automated robot moving in the existing route in order to avoid different obstacles, etc.

- Remote Sensing:

  Satellites take images at different times of the Earth and other planets, usually from different views; in addition, different satellites use different spectral bands for imaging. Thus, image registration for remote sensing involves the alignment of images taken by different sensors at different times. Specific applications for remote sensing image registration include environmental monitoring, image mosaicking, multispectral classification, change detection, integrating information into geographic information systems (GIS), etc.

- Medical Imaging:

  Medical images are taken at different times by different devices (i.e., different sensors) as Magnetic Resonance Imaging (MRI), Computerized Tomography (CT), Positron Emission Tomography (PET) and Single-Photon Emission Computed Tomography (SPECT). In addition to the above, medical images are usually taken from different angles in order to create a 2D or 3D perspective; therefore, image registration for medical imaging also involves similar aspects as remote sensing image registration. Typical applications for registration of medical images are combining data about a patient from different sensors in order to obtain their detailed, comprehensive medical condition, monitoring a tumor's growth, treatment verification and comparing a patient's anatomy with theoretical models.

- Cartography (map making):

  Creating and updating maps requires comparing images of the same scene taken at different times from different views. This enables to track changes in roads, borders, coast lines and lakes. The main application in this field is to update existing maps according to such geographical changes.

1.2 Image Registration Methodology

1.2.1 Image Registration Categories

As explained above, image registration involves the alignment of two images that may differ in, possibly, four aspects; each aspect represents a different type of registration, as follows:

- *Multi-view images*: The images were taken from different viewpoints of the scene; the purpose is to enlarge the 2D or 3D view of the scene. Relevant applications include computer vision (a robot navigating automatically in a maze), remote sensing (a satellite sampling a specific scene from different angles), medical imaging (a CT system which scans a patient's body from different viewpoints), etc.

- *Multi-temporal images*: The images of the scene were taken at different times (possibly on a regular basis); the purpose is to monitor changes in a specific scene. Relevant applications include remote sensing (a satellite sampling the same scene from the same viewpoint but at different times), cartography (tracking changes in roads and lakes), etc.

- *Multi-modal/Multi-spectral images*: The images were taken by different sensors (in different spectral bands); the purpose is to increase the amount of data by fusing images from different sensors. (Some details in medical imaging, for example, are transparent in a specific band and visible in others and vice-versa.) Relevant applications include remote sensing (different satellites operate in different bands), medical imaging (different medical imaging systems operate in different bands), etc.

- *Scene to model registration*: The "sensed" image is registered to a model "reference" image; the purpose is to compare a specific scene or a tested object to their desired counterparts. Relevant applications include computer vision (inspection in production lines), medical imaging (e.g., finding pathologies), etc.

1.2.2 Image Registration Steps

First we should emphasize the fact that due to the variety of applications and the diversity of registration types it is impossible to develop one universal method for all image registration problems. Nevertheless, most image registration methods usually consist of the following four generic steps (see surveys [1], [14], [15]):

- *Feature extraction*: Distinctive image features are automatically detected and extracted; these features can be objects, corners, boundaries (closed and open), edges or simply picture elements (*pixels*). Detected features are globally called key-points. Each key-point is associated with a specific

descriptor which depends directly on the key-point's parameters (location, intensity, etc.) and possibly its neighborhood.

- *Feature matching*: This step concerns the establishment of correspondences between the features of the first image (called the *reference* image) to the features of the second image (called the *sensed* image).

- *Transform model estimation*: The correspondences between the two images found in the previous step are used to estimate the transformation parameters needed to align the sensed image to the reference image. Of course, this process could be iterative.

- *Image resampling and transformation*: The computed transformation is applied to the sensed image and some interpolation method is used to determine the gray levels at the discrete coordinates of the sensed image.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 gives extensive insight on related research concerning image registration for remote sensing. In Chapter 3 we give a complete formulation and an outline of our proposed algorithm. Chapter 4 presents our detailed experimental results. We conclude our work in Chapter 5.

Chapter 2

2. Related Work

The purpose of this research thesis is to develop a new algorithm for image registration of remotely sensed images. As explained in the Introduction, remote sensing image registration deals with three out of the four image registration types; by that we mean that remotely sensed (satellite) images are usually taken at different times (multi-temporal), from different views of the same area (multi-view), and possibly by different sensors (multi-resolution) or by the same sensor but with different imaging spectral bands (multi-spectral). Thus, the image registration problem for remotely sensed images is still very challenging and evolving.

2.1 Image Registration for Remote Sensing

Image registration in general employs algorithms from two categories. The first category is of area-based algorithms where complete areas in the reference and sensed images are matched according to differences in pixel intensities or by Fourier coefficients; usually these algorithms do not use either correspondences between specific points or preprocessing of the raw images (to obtain such points of interest). The second category consists of feature- based algorithms, where preprocessing is applied in order to extract specific features (e.g., edges, corners, etc.); matching is done between these features in order to compute an appropriate transformation. It is possible to devise an algorithm which combines area-based and feature-based concepts.

Registration methods suited, among others, to remote sensing are:

(a) Manual registration, where one chooses manually corresponding features in the reference and sensed images followed by a transformation computation;

(b) Correlation methods in which the correlation between intensity values is computed in order to minimize some distance measure between areas in the reference and sensed images. Since these methods use brute-force search for the optimal transformation they are computationally expensive and have relatively long run-times;

(c) Transform methods that use similarity between the transform coefficients (e.g., Fourier) rather than pixels; these methods also have drawbacks like correlation methods due to their brute-force nature;

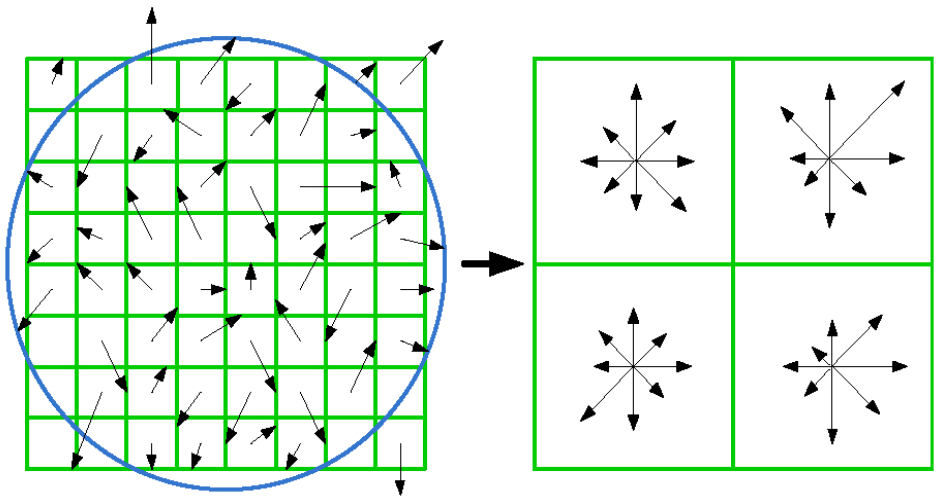
(d) Feature point methods that extract highly distinctive features from the reference and sensed images and match between them according to local image properties (the transformation computation is carried out upon those correspondences); and

(e) Contour- and area-based methods that use feature groups extracted from both images (e.g., contours, lines, edges, corners, etc.) for matching and transformation computation.

2.2 Prior Work

In this section we give a brief overview of several methods for image registration for remotely sensed data that rely on the *scale-invariant feature transform* (SIFT) [3] for the detection and description of key-points. The SIFT algorithm approximates the well-known Laplacian of Gaussian (LoG)

operator in order to detect local extreme points (which are the desired key-points) in different scales of an image; these extreme points are associated with the 4D vector, $(x, y, s, \theta)^T$, where (x, y) are the key-point coordinates, s is the key-point scale (the image scale for which this key-point was detected), and θ is the key-point orientation which is determined from peaks in the key-point's gradient orientation histogram. In addition, the SIFT algorithm also assigns a 128-element vector of reals to each key-point (i.e., a *descriptor*). As explained earlier, this descriptor is used for key-point matching in image pairs, based on its gradient magnitudes at different orientations in the key-point neighborhood (see Figure 2.1). A detailed explanation about the SIFT algorithm, which is widely used in object recognition, matching and image registration, can be found in [3].



(a) (b)

Figure 2.1: SIFT descriptor of a key-point: (a) Gradient values of 8×8 neighborhood, and (b) gradient histograms which form the descriptor (source: [3], p. 15).

2.2.1 Robust Scale-Invariant Feature Matching for Remote Sensing Image Registration

Li *et al.* [4] were among the first to propose a method based on the SIFT algorithm. They begin by refining the key-point orientation θ in the following manner:

$$\alpha = \begin{cases} \theta & \theta \in [0,180] \\ 360 - \theta & \theta \in (180,360) \end{cases} \tag{2.1}$$

This refinement is meant to compensate for gradient angle inversion, which can be caused by different illumination or spectral bands between two images to be registered. The next stage is to assign additional orientations to each key-point, besides the one assigned by SIFT in order to enlarge the transformation space, as the original SIFT orientation is not always accurate and reliable. Assigning these orientations is made by computing the histogram of the refined gradient orientations in the neighborhood of the examined key-point; all orientations whose frequency exceeds the predefined threshold:

$$T_h = \sum_{k=1}^{180} \frac{h_k}{180} \tag{2.2}$$

where h_k is the frequency of the k^{th} orientation, are defined as "main orientations" and assigned to the key-point in question. In order to match between descriptors from two images, a distance measure is defined. Let $C = \{c_1, c_2, ..., c_N\}$ and $C' = \{c'_1, c'_2, ..., c'_M\}$ be the key-point sets, respectively, from the reference and transformed images that need to be

registered. Each key-point is assigned its position, scale and set of main orientations. The relative main orientation between the main orientations of two different key-point sets is defined as $r_i = \alpha_i - \alpha'_i$, where α_i and α'_i are the main orientations of the key-point pair from the reference and sensed images, respectively. The scale error is then defined as

(2.3) $$\varepsilon_s(i) = |1 - s^* \frac{s_i}{s'_i}|$$

where s_i and s'_i are scales of the corresponding key-points, and s^* is the current scale estimate. In the same manner, the *relative main orientation* (RMO) between the corresponding key-points is defined as

(2.4) $$\varepsilon_r(i) = |r_i - r^*|$$

where r^* denotes the current rotation difference estimate. Since $\varepsilon_s(i)$ and $\varepsilon_r(i)$ represent distance measures for the scale and orientation, respectively, they can be combined to define the joint distance:

(2.5) $$J = [1 + \varepsilon_r(i)][1 + \varepsilon_s(i))]E(i)$$

where $E(i)$ is the Euclidean distance between the key-points' descriptors. Matching between key-points is done as follows: First key-points are matched according to nearest neighbor (NN) Euclidean distances between their corresponding 128×1 SIFT descriptors; false matches are excluded according to the ratio between the first and second nearest neighbors. If this ratio exceeds a predefined threshold, the match should be excluded.

(The rationale behind this ratio test is that for true matches the NN distance will be much smaller than the second NN distance, as opposed to false matches where both the first and second NN distances are quite large.) The above yields a set of key-point pairs, for which the scale and RMO histograms of peak values are computed along with the joint set $\{s^*, r^*\}_{k=1,2,...,K}$ where s^* and r^* are the peaks in scale and RMO, and K is the number of peak combinations. Now, for each $\{s^*, r^*\}_k$ key-points are matched according to the nearest neighbor joint distance. Again, matches are excluded by thresholding the ratio between the first and second nearest neighbors; in addition, the average joint distance of the tested $\{s^*, r^*\}_k$, d_k, is computed. The optimized point set is the one for which d_k is minimized.

2.2.2 High-Resolution Multispectral Satellite Image Matching Using Scale Invariant Feature Transform and Speeded Up Robust Features

Teke *et al.* [5] proposed a modified SIFT algorithm, the *orientation-restricted SIFT* (OR-SIFT), to increase the correct feature matching ratio. First, bins in the gradient orientation histogram with opposite directions (e.g., 0^0-45^0 and 180^0-225^0) are accumulated into one bin. The purpose is to compensate for inversion in gradient orientations, much like the orientation refinement described in (2.1). This accumulation results in a feature vector of half size relative to the original feature vector (i.e., 64 elements instead of 128). As explained in Sub-section 2.2.1, matches are based on the NN

Euclidean distance. In order to exclude false matches, the scale difference between two key-points P_1 and P_2 is computed by

(2.6) $$SD(P_1, P_2) = SD = |\sigma_1 - \sigma_2|$$

where σ_1 and σ_2 are the corresponding key-point scales. Next the authors define the scale restriction (SR) criterion:

(2.7) $$|SD - \overline{SD}| < W$$

where $\overline{SD}$ is the peak value of the histograms of all *SDs* and W is an empiric, image dependent parameter. Matches that do not satisfy the above SR criterion are rejected. In order to compensate for intensity differences, histogram equalization is applied to both the reference and sensed images. (Contrast stretching may be applied instead.)

2.2.3 Uniform Robust Scale-Invariant Feature Matching for Optical Remote Sensing Images

Sedaghat *et al.* [6] proposed another SIFT based algorithm, the *uniform robust SIFT* algorithm (UR-SIFT), which provides an adequate number of key-points uniformly distributed in both the image and scale spaces. The quality of the key-points is determined according to the following criteria:

(a) Stability, i.e., the strength of a feature's presence under different image acquisition conditions.

(b) Distinctiveness, i.e., the degree of feature uniqueness and that of their descriptors.

The first step in the algorithm is to determine the total number of key-points, N. Next, the SIFT algorithm is applied to create the scale space composed of N_O octaves, each contains N_L scale layers with an initial scale factor σ_0. The next steps are taken for each layer, l, of the octave, o. The initial key-points are computed by the SIFT algorithm; key-points whose contrast is within the bottom 10% of the contrast range are rejected. The number of required features is determined according to

(2.8) $$N_{ol} = NF_{ol}$$

where F_{ol} is the proportion of features in the current scale layer so that the next normalization condition holds, i.e.,

(2.9) $$\sum_{o=1}^{N_O}\sum_{l=1}^{N_L} F_{ol} = 1$$

Next, the authors compute the entropy in the local region near each key-point by

(2.10) $$H = \sum_j P_j \log_2 P_j$$

where P_j is the probability of the j^{th} pixel within the region. This entropy represents the amount of "data" (i.e., pixel intensities) in the region. The smoothed image of the scale of interest is divided into grid cells. The average entropy E_i of the i^{th} cell is then computed by (2.10). In addition, n_i and MC_i (i.e., the number and mean contrast of the key-points in the

cell, respectively) are also computed. Finally, the number of required features in each cell is determined by

$$Ncell_i = N_{ol}\left[\frac{W_E E_i}{\sum_i E_i} + \frac{W_n n_i}{\sum_i n_i} + \frac{(1-W_E-W_n)MC_i}{\sum_i MC_i}\right] \tag{2.11}$$

where W_E and W_n are the entropy and feature number weight factor, respectively. For each grid cell $3\times Ncell_i$ key-points with the highest contrast are reserved and all other key-points are rejected. The accurate position and scale for each key-point is computed by the regular SIFT algorithm; key-points with low principal curvature are also rejected since those usually represent unreliable key-points along edges. The entropy of all remaining key-points is computed by (2.10) and finally $Ncell_i$ key-points with the highest entropy are chosen for this specific cell. The key-point orientation is determined as in the SIFT algorithm. The pre-matching between key-points is done by a cross-matching constraint to confirm the matching by reverse certification. This gives rise to some false matches which are excluded by checking key-point pairs in a global transformation between the reference and transformed images.

2.2.4 Multispectral Image Matching Using Rotation-Invariant Distance

Li *et al.* [7] proposed a registration method which is invariant to position and orientation but not to scale. The method arranges the SIFT descriptors in a way which makes it easy to find a correlation between the reference and sensed images and subsequently find the translation and rotation

needed for the desired transformation. First, the regular rectangular grid used for computing the SIFT descriptor is replaced by a polar grid. The polar region around each key-point is divided to inner and outer rings with corresponding radii of $R/2$ and R, for some predefined parameter R. Each ring is divided into N sectors and each refined orientation histogram (defined in (2.1)) is divided into N bins. This partition gives rise to two descriptor vectors (one for each ring), each with length N^2. These vectors are denoted as V_1 and V_2, respectively, and are ordered as two $N \times N$ matrices H_1 and H_2, respectively, where the rows correspond to the bins and the columns correspond to the sectors. The complete key-point feature matrix is defined as $H = H_1 + H_2$, as opposed to the regular 128×1 SIFT descriptor. The matrix H is arranged such that each right circular shift by one column and down by two rows corresponds to rotating the image counter-clockwise by $2\pi / N$. Let H and H' denote a pair of key-points; the correlation between the features can be computed with the aid of the fast Fourier transform (FFT) by

$$C(H, H') = F^{-1}(F(H)\overline{F(H')}) \tag{2.12}$$

where F^{-1} and F are the IFFT and FFT, respectively, and $\overline{F(H')}$ is the FFT complex conjugate of H'. We note that the element of the correlation matrix $c(i) = c[\mathrm{mod}(2i-1, N)][i]$, where mod stands for computing the remainder, corresponds to rotation angle of $\theta(i) = \frac{(i-1)\cdot 2\pi}{N}$, $i = 1, 2, ..., N$. Denoting the index of the maximal element $c(i)$ by i^* and letting V and

V' be the corresponding feature vectors, the rotation invariant distance between V and V' is defined as

(2.13) $$\mathrm{RID}(V,V') = D\left(V, circshift\left(V', [\mathrm{mod}(2i^* - 1, N), i^*]\right)\right)$$

where $circshift\left(V', [a,b]\right)$ stands for a circular shift of the corresponding matrix H' to the right by a and down by b and D stands for regular Euclidean distance. Matching between feature vectors is done by finding the nearest neighbor according to the rotation invariant distance which becomes minimal once the correlation is maximal.

2.2.5 Multispectral Remote Sensing Image Registration via Spatial Relationship Analysis on SIFT Key-Points

Hasan *et al.* [8] proposed a method for automatic registration by inflating the number of SIFT key-points with the aid of original key-point area. First, feature points are found via SIFT in both the reference and sensed images; feature points are rejected according to the ratio between the first and second NN, where a distance ratio threshold of 0.8 is used to reject outlying feature points. In order to remove outliers that survive the above criterion, the RANSAC algorithm [9] is applied to find a global affine transformation. (The latter considers several random samples from the set of correspondences obtained and tries to compute a transformation that would be appropriate (i.e., within some pre-defined threshold) with respect to the whole set.) The resulting feature points are called primary matched feature points. Next, the procedure locates, for each primary matched feature point, all of its neighboring feature points within a W - pixel wide

grid; this is done in order to increase the amount of correspondences in the primary feature's neighborhood, where the region is assumed to be "good" in terms of SIFT matching. Once again, the ratio of the first and second NN is used to match between all the feature points in the above grid. (A threshold of 0.9 is picked here to reject outliers.) All feature point matches whose positional difference relative to that of the affine transformation is greater than a predefined threshold T are rejected, and all other accepted correspondences are called secondary matched feature points. Finally, both primary and secondary feature points are used as key-points to register the images.

2.2.6 Modified SIFT for Multi-Modal Remote Sensing Image Registration

Hasan *et al.* [10] proposed numerous modifications in the SIFT algorithm to improve the results of matching and registration. These include the following:

1. Preserve every key-point instead of eliminating key-points along edges; this is achieved by setting the SIFT threshold to infinity. The rationale behind this step is to enlarge the key-point sample space in order to achieve better correspondences.

2. Reduce the effect of strong edges; to prevent cases where a key-point in the reference image lies on a strong edge while the corresponding key-point in the sensed lies on a weak edge, it is proposed to limit the gradient values to a predefined threshold (e.g., 0.08 for normalized pixel values).

3. Enlarge the SIFT descriptor window; originally, the squared window is of size $(16s)\times(16s)$ pixels, where s is the current scale. A window which is 1.67 times larger than the above is considered an optimal choice. This will give rise to SIFT descriptors which will be "richer" in content and therefore more reliable in the matching step.

4. Enlarge the sub-regions for the SIFT descriptors; due to the above, it is proposed to use 6×6 sub-regions instead of the classical 4×4 SIFT sub-regions. This will result in dimensionality of 288 for the SIFT descriptors (instead of 128).

5. Overlook the largest difference; it is proposed to ignore the largest difference (out of 288 dimensions) between the descriptors of each key-point pair candidate.

6. Three-level matching: In order to increase the number of true matches, it is proposed to match first the first 20 dimensions, followed by the first 64 dimensions and finally all 288 dimensions to achieve a refined matching.

Chapter 3

3. Research Outline

3.1 Preliminary Approach

Netanyahu *et al.* [11] proposed a remote sensing image registration algorithm in which key-points are actually image features (e.g., edges, corners, etc.) extracted by a wavelet-based algorithm (Simoncelli's steerable filters [12]). Pixel intensities within the top 10% of all intensities are considered as features to be matched; the relatively high threshold for feature detection was meant to match fewer and more meaningful features and thereby improve the computational complexity.

Feature matching is done iteratively at various levels of resolution. The initial target transformation is $(\theta, t_x, t_y) = (0°, 0, 0)$, where θ, t_x and t_y denote the rotation angle and vertical and horizontal translations of the hypothesized similarity transformation, respectively; in addition, an initial bounding box $(\delta\theta, \delta t_x, \delta t_y) = (16°, 32, 32)$ is chosen to avoid the need for an elaborate search at the higher resolution levels. The search is done exhaustively at the coarsest level, followed by search in the finer levels with relatively small bounding boxes. Denoting by (θ, t_x, t_y) the current transformation, the initial transformation of the next iteration becomes $(\theta, 2t_x, 2t_y)$. The process iterates until convergence is reached. In order to estimate the transformation accuracy, the *partial Hausdorff distance* (PHD) similarity measure is used. It is defined as

(3.1) $$\text{med}_{a\in A}(\min_{b\in B}|a-b|)$$

where med stands for the median value, A and B are the reference and transformed feature sets, respectively, and $|a-b|$ is the Euclidean distance between features from the reference and transformed feature sets, respectively. Convergence is achieved ideally when $\text{PHD}<1$, i.e., when the PHD between the feature sets is smaller than 1 pixel. The matching algorithm returns a set of pixel pair correspondences $\{(x_i,y_i),(x'_i,y'_i)\}_{i=1,\ldots,N}$, where (x_i,y_i) and (x'_i,y'_i) are the pixel locations of the corresponding control points in the transformed and reference images, respectively. Defining the correspondence error under the similarity transformation assumed for each point pair as

(3.2) $$E_i=[x'_i-(x_i\cos\theta-y_i\sin\theta)-t_x]^2+[y'_i-(x_i\sin\theta+y_i\cos\theta)-t_y]^2$$

the classical *ordinary least squares* (OLS) method finds the optimal transformation parameters by minimizing the sum of errors E_i. Namely, defining $E(\theta,t_x,t_y)=\sum_i E_i$, the optimal transformation in an OLS sense is the triplet (θ,t_x,t_y) that minimizes $E(\theta,t_x,t_y)$. The main problem of this method is its sensitivity to noisy samples (i.e., false correspondences). Therefore, instead of minimizing $E(\theta,t_x,t_y)$, one may minimize the median of the correspondence errors, defined as $E(\theta,t_x,t_y)=\text{med}_i(E_i)$. Since there is no closed-form solution to this minimization problem, an approximation

[13] is used to find the optimal transformation of the *least median of squares* (LMS).

3.2 Our New Approach

The main rationale of our proposed method is to avoid the intensive search for the optimal transformation based on extracted features as explained above. Instead, we use correspondences between the images, based on the SIFT descriptors. Using these correspondences we can compute an initial transformation which is expected to be sufficiently close to the optimal similarity transformation between the reference and sensed images, provided that the correspondences used are true. Further refinement could be carried out to find a more accurate transformation (if needed). A block diagram of the proposed method is shown in Figure 3.1. First, the SIFT descriptors are

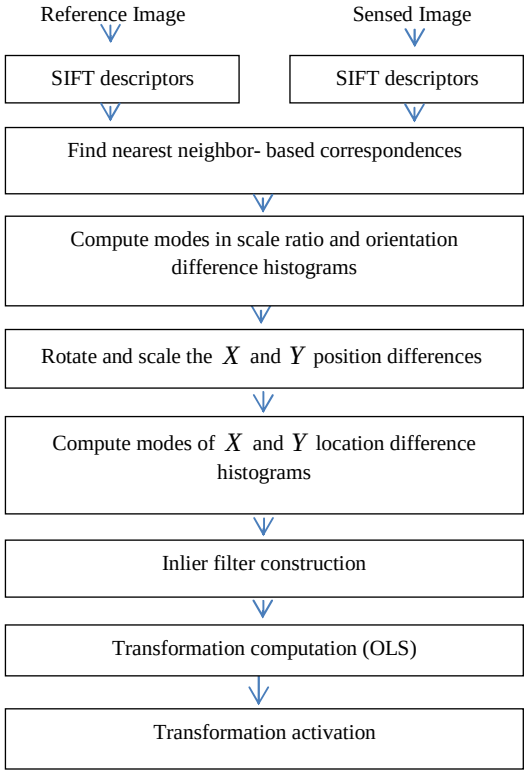


Figure 3.1: Proposed registration algorithm.

extracted for both the reference and sensed images. For matching, instead of using the standard ratio test between the first and second nearest neighbors, we use a Hough-like approach for mode seeking (MS) as follows. First we find for each key-point in the reference image its nearest neighbor in a Euclidean distance sense in the sensed image. Let us denote the set of the resulting correspondences by $\{(x_n, y_n) \leftrightarrow (x_n', y_n')\}_{n=1,2,\ldots,N}$, where (x_n, y_n) and (x_n', y_n') are the spatial locations of the SIFT key-points in the reference and transformed images, respectively. The next stage is to form histograms of scale ratios and orientation differences between the correspondence pairs found in the previous stage. We find the maximum value of each histogram and compute the corresponding modes s_{mode} and $\Delta\theta_{\text{mode}}$ by a weighted average of the maximum value and its two adjacent bins (i.e., the bins to its left and right). We use these modes to rotate and scale the position differences, in both the X and Y directions, between nearest neighbor pairs as follows:

$$\begin{aligned} \Delta x &= x - s_{\text{mode}}\left(x'\cos(\Delta\theta_{\text{mode}}) - y'\sin(\Delta\theta_{\text{mode}})\right) \\ \Delta y &= y - s_{\text{mode}}\left(x'\sin(\Delta\theta_{\text{mode}}) + y'\cos(\Delta\theta_{\text{mode}})\right). \end{aligned} \tag{3.3}$$

Now we can compute the histograms of these differences and find their modes which we denote by Δx_{mode} and Δy_{mode}, respectively. Obtaining the quadruple $(s_{\text{mode}}, \Delta\theta_{\text{mode}}, \Delta x_{\text{mode}}, \Delta y_{\text{mode}})^T$, we can now filter outliers with respect to the initial correspondences. First we define the following two logical filters:

(3.4)
$$F_1 : \left| \Delta x - \Delta x_{\text{mode}} \right| \geq \Delta x_{\text{thresh}}$$
$$F_2 : \left| \Delta y - \Delta y_{\text{mode}} \right| \geq \Delta y_{\text{thresh}}$$

where Δx_{thresh} and Δy_{thresh} denote, respectively, predefined thresholds of horizontal and vertical differences, in terms of corresponding histogram bin widths (measured in pixels). Our outlier filter will reject all correspondences for which F_1 or F_2 holds. All remaining correspondences are considered inliers, i.e., they are assumed to be very reliable, so the next stage is to compute the similarity transformation resulting from these correspondences by OLS. (An exact derivation of the OLS procedure we have used can be found in Appendix A.)

In order to assess the correctness of the final transformation, we will choose manually N points (pixels) in the reference image and their corresponding points in the sensed image. Typically, N will be between 10 to 20 points. Again, we denote these sets by $\{(x_i, y_i)\}_{i=1,\ldots,N}$ and $\{(x'_i, y'_i)\}_{i=1,\ldots,N}$, respectively. We refer to these sets as *ground truth (*GT), in the sense that they represent the most accurate transformation possible. Next we apply the transformation to each point in $\{(x'_i, y'_i)\}_{i=1,\ldots,N}$ and compute the *root mean square error* (RMSE) defined as

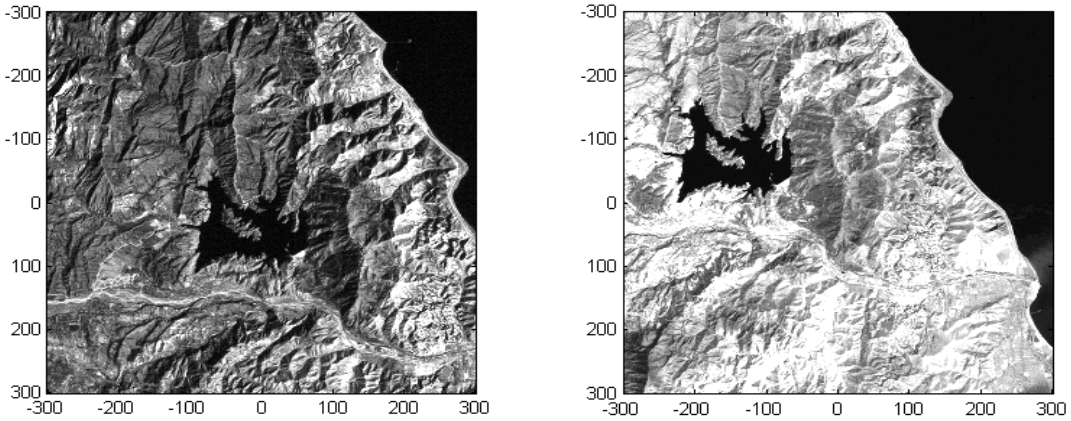
(3.5)
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2}$$

where $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=1,\ldots,N}$ is the set obtained by applying the transformation to $\{(x'_i, y'_i)\}_{i=1,\ldots,N}$. As a rule of thumb, we will assume that an RMSE value of at most 1 pixel represents a good registration.

As for comparison with other approaches, we focus on simplicity and fast performance rather than exhaustive, time consuming search algorithms. The approach described in [4], for example, involves computing joint distances for all possible correspondences (which pass an initial standard rejection filter) for each pair of peaks in the RMO and scale histograms; this means that the computation complexity is greater by, at least, an order of magnitude than that of our method. In the same manner, the approach described in [7] involves computing a correlation matrix between all the key-points; this is followed by rotating the feature matrix of minimum correlation by the appropriate orientation. Again, these are very expensive computational operations. In addition, this method is not scale invariant. Our approach also eliminates the search for an optimal transformation as in [11] and is thus far simpler.

To further explain our algorithm, we illustrate its complete execution on a simple example. Figure 3.2 shows a 600×600 reference image (left) and a sensed image (right) of the same size. Both images were acquired by Landsat; the reference image (band5) in 1984 and the sensed image (band7) in 1986 (images source: UCSB site: http://vision.ece.ucsb.edu /registration /satellite/testimag/index.htm). This pair is thus a multi-temporal, multi-spectral, and multi-view. As we can see, the reference image is much brighter than the sensed one due to the multi-spectral nature of the image pair; we also note a slight rotation and horizontal and vertical translations between the images. The upper part of Figure 3.3 shows the scale ratio and orientation difference histograms as explained in the previous section. In the

same manner, the lower part of Figure 3.3 shows the X and Y position difference histograms after being scaled and rotated by the mode values of the scale ratio and orientation difference histograms. As explained above, these modes are used to construct an inlier filter which resulted in this case in 82 correspondences (out of 797 initial correspondences) from which the



(a) (b)

Figure 3.2: Landsat images over Casitas Lake: (a) Reference (band5, 1984), and (b) sensed (band7, 1986) ; source: UCSB site as given above.

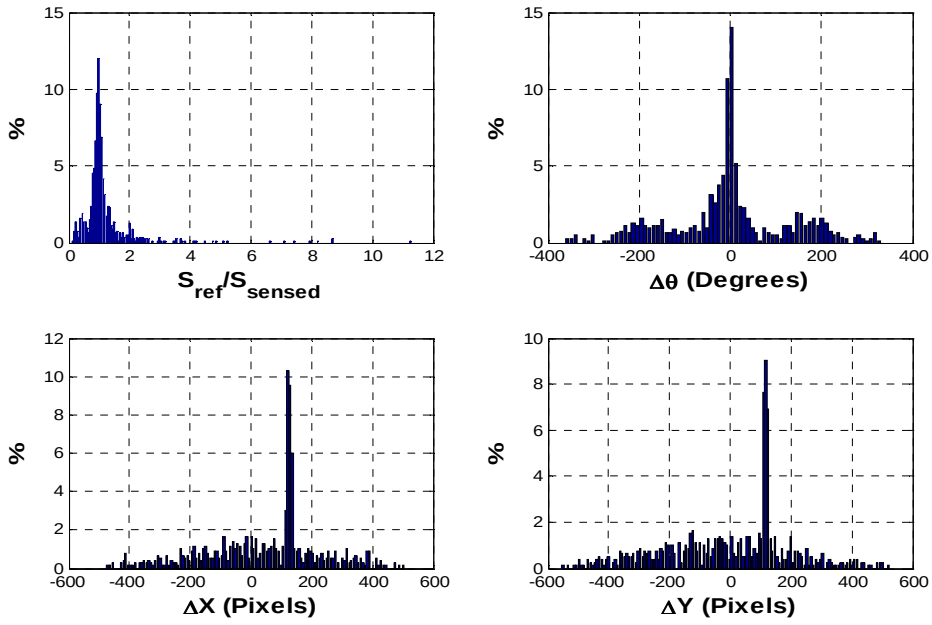


Figure 3.3: Histograms of key-point correspondences.

similarity transformation is computed (via OLS) to register the sensed image onto the reference image. The registered images are shown in Figure 3.4. We can see that the registration is quite good and the sensed image is now aligned with the reference image. This visual perception is also confirmed by computing the RMSE which is 0.75 pixels.

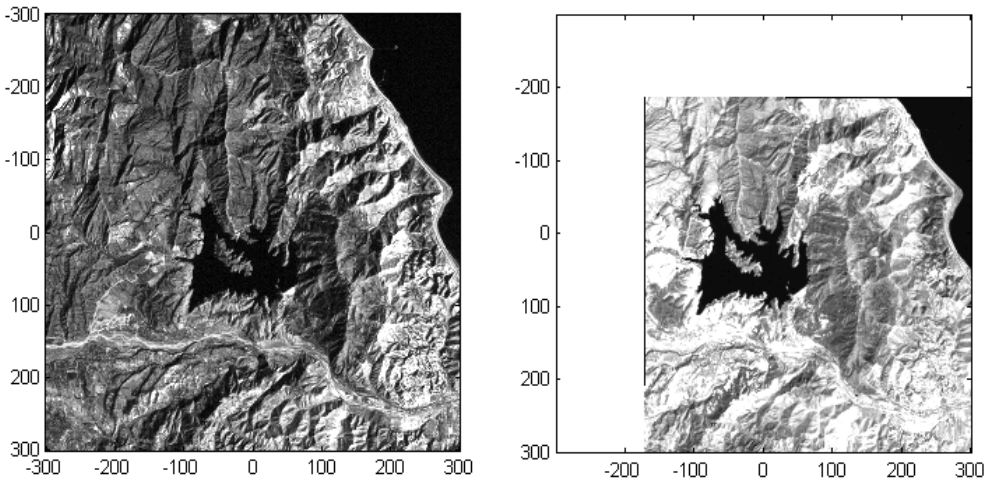


Figure 3.4: Registered image pair of Figure 3.2.

Chapter 4

4. Experimental Results

4.1 Experimental Methodology

We have implemented our SIFT-based MS algorithm in MATLAB and C and applied it to dozens of multi-temporal, multi-spectral, and multi-sensor image pairs. For each type of image pair, the results below are arranged in the following manner. First, we present the image pairs (including sensor used, size, time, and area of acquisition); this is followed by visual results of the registration process, i.e., the reference and sensed images of their overlapping area are mosaicked onto a common coordinate system. Finally, we provide numerical results which include the number of initial correspondences and the number of key-points which survived the filtering process, the transformation parameters, an RMSE value, and the run-time measured within the C-style implementation on a regular PC (Intel Q8200 with 3 [GB] of RAM and Vista OS). We point out that in some cases the registration process failed because the number of correspondences after filtering was insufficient to compute a meaningful transformation; in these cases, we introduce black "images" instead of visual registration results. We used consistently the bin widths $q_s = 0.075, q_{\Delta\theta} = 9^\circ$ and $q_{\Delta X} = q_{\Delta Y} = 7.5[pixels]$ for the histograms of the scale ratio, orientation difference, and position difference, respectively.

4.2 Chip Extraction

Our first two datasets are the ones described in [11]. First, we used several 256×256 sub-images with known geographical positions to serve as our reference chips. Next we used several 2048×2048 Landsat-7 and Landsat-5 images sensed at different times, from which 256×256 "windows" partially overlapping these reference chips were extracted (see [11] for details). We registered these windows against the reference chips. Figure 4.1 shows these concepts.

4.2.1. Data over Washington, DC

Our first set of chips is from the Washington, DC area. Eight reference chips were extracted from a Landsat-7/ETM image acquired on 28/07/99; these chips are depicted in Figure 4.2. Our 8-window sets were extracted from

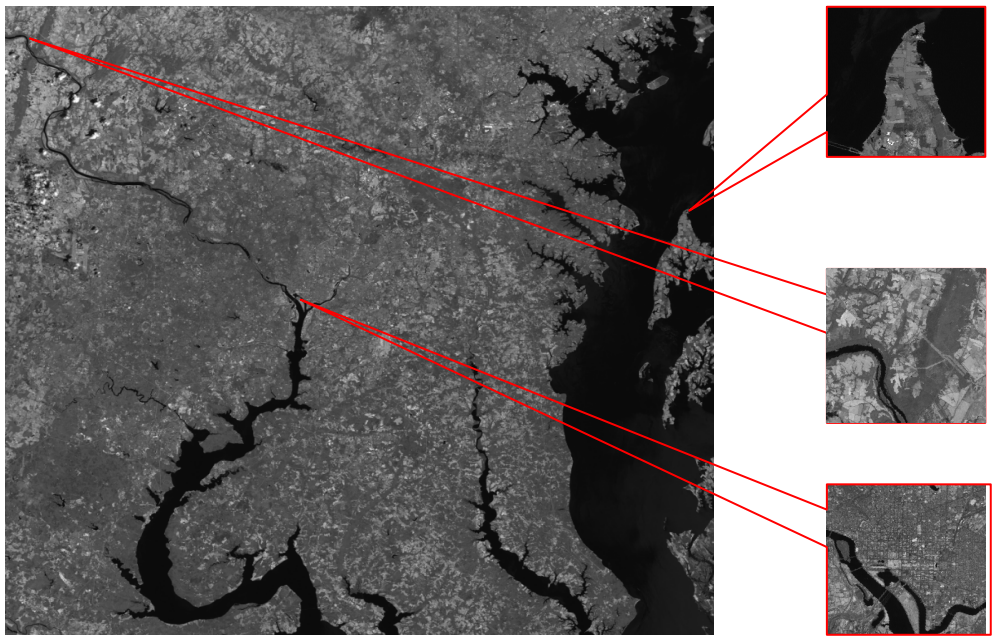
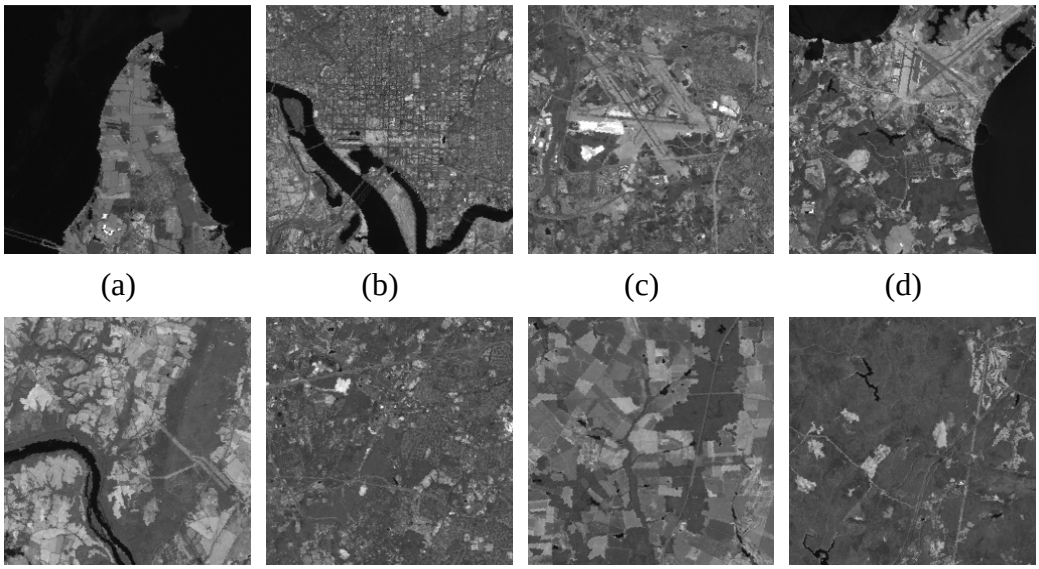


Figure 4.1: Extraction of 256×256 windows from 2048×2048 Landsat scene.

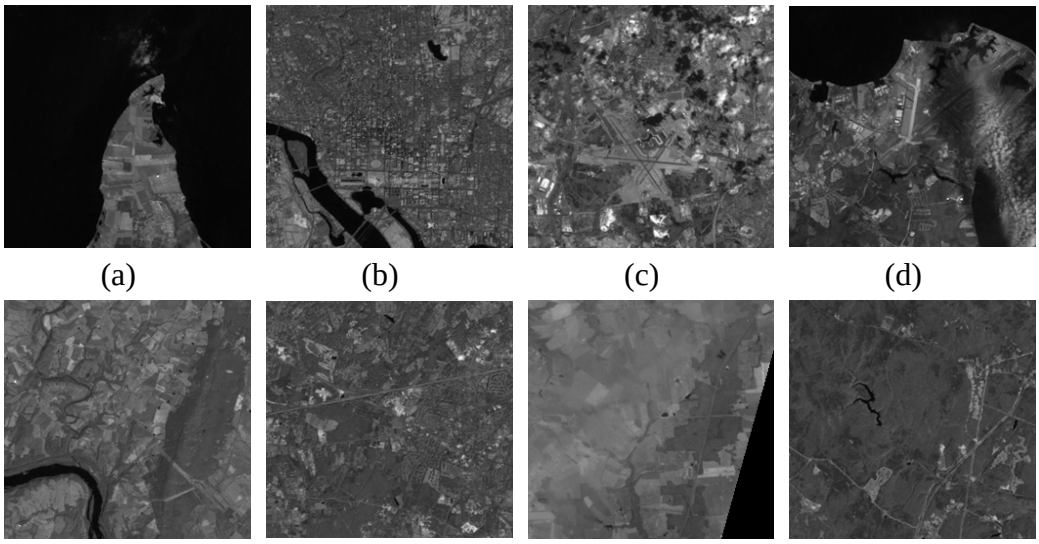
Landsat-5/TM images taken on 27/08/84, 16/05/87, 12/08/90, and 11/07/96 (abbreviated as 840827, 870516, 900812, and 960711, respectively). These windows are shown in Figures 4.3, 4.5, 4.7, and 4.9, the corresponding registration results are shown in Figures 4.4, 4.6, 4.8, and 4.10 and in Tables 4.1-4.4 (N/A stands for non-applicable registration due to an insignificant number of inliers after filtering).



(a) (b) (c) (d)

(e) (f) (g) (h)

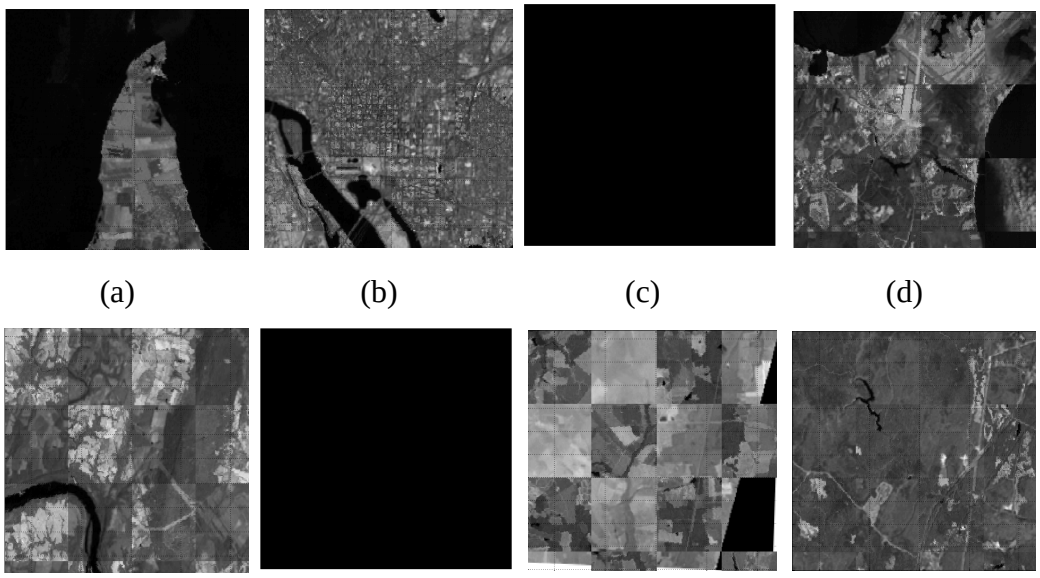
Figure 4.2: Eight reference 256×256 chips from the Washington, DC area.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.3: Extracted windows from Landsat scene (840827) over the DC area.



(a) (b) (c) (d)

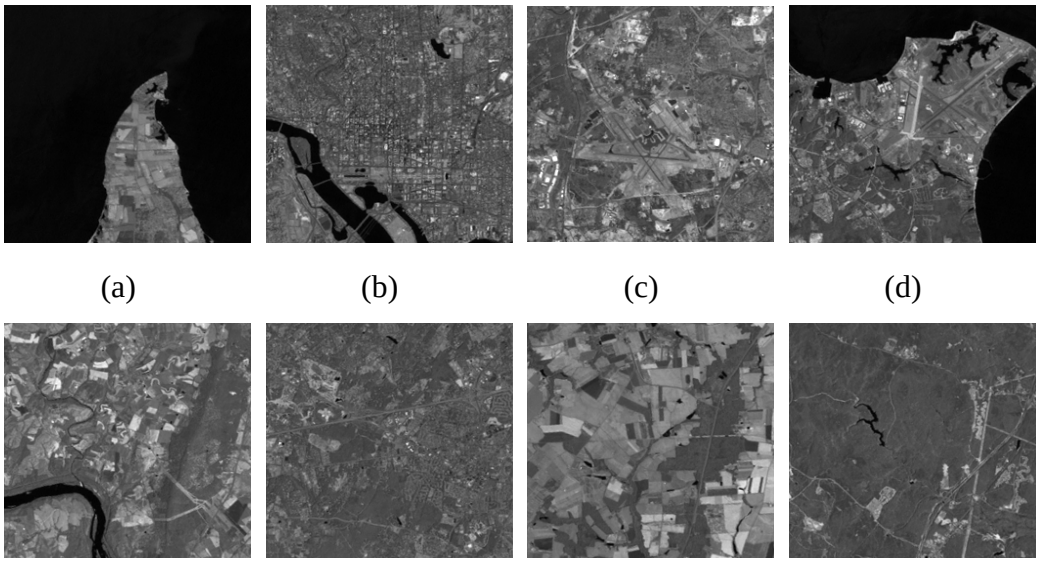
(e) (f) (g) (h)

Figure 4.4: Registration results for Landsat windows (from 840827 over the DC area) vs. reference chips.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	68	12 (17%)	1.001	-0.24	-5.76	-46.39	0.74	0.658
b	93	19 (20%)	1.006	-0.27	-4.75	-47.61	0.86	0.693
c	134	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.742
d	105	13 (12%)	1.014	0.04	-4.22	-45.74	0.73	0.699
e	120	18 (15%)	1.003	-0.47	-4.55	-48.73	0.63	0.725
f	99	2 (2%)	N/A	N/A	N/A	N/A	N/A	0.710
g	139	9 (6%)	1.038	2.38	-6.36	-46.18	0.76	0.751
h	98	19 (19%)	1.002	-0.006	-5.40	-47.70	0.60	0.698

Table 4.1: Registration results for Landsat over the DC area (840827)

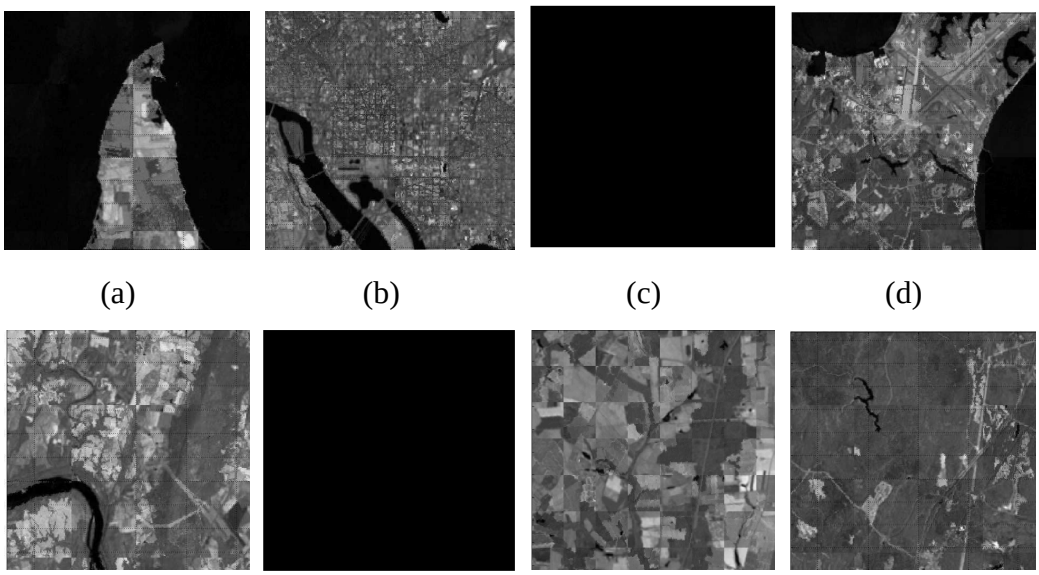
We failed to register windows (c) and (f) due to a low number of inliers that survived the filtering process. Both failures resulted from substantial differences between the reference and sensed images (due to clouds mainly). It seems that intensity distributions for these windows were different in a manner which confused the SIFT algorithm and led to irrelevant correspondences. To further validate this assumption, we picked manually, for both failed image pairs, eight ground truth correspondences and computed the resulting transformations, $< s, \theta, t_x, t_y > = < 1.01, 0.81, -6.02, -47.29 >$ (for image pair (c)) and $< s, \theta, t_x, t_y > = < 0.999, 0.3, -4.92, -47.92 >$ (for image pair (f)). Next, we computed the Euclidean distance between the SIFT key-points of the reference chip and the corresponding transformed key-points of the sensed window. We consider a correspondence to be true if the above distance is smaller than 2 pixels and false otherwise. This stems from the noisy nature of our algorithm, e.g., the histogram quantization. Only 3 true correspondences were obtained for window (c) and none for window (f). Obviously, these are insufficient numbers of inliers for our algorithm. As for a comparison to [11], the original algorithm failed on windows (c) and (g) but succeeded on window (f). We should emphasize that run-times of the latter for both failures and successes were greater by an order of magnitude, at least.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.5: Extracted windows from Landsat scene (870516) over the DC area.



(a) (b) (c) (d)

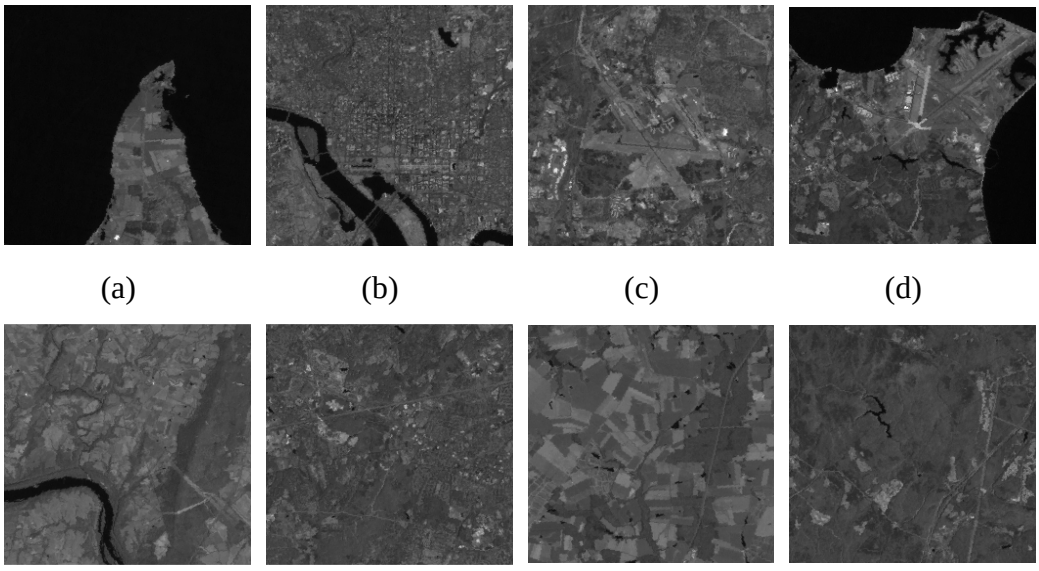
(e) (f) (g) (h)

Figure 4.6: Registration results for Landsat windows (from scene 870516 over the DC area) vs. reference chips.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	68	11 (16%)	0.983	0.12	-8.10	-47.21	0.90	0.699
b	87	17 (19%)	0.972	0.42	-8.06	-44.7	0.80	0.692
c	133	2 (1%)	N/A	N/A	N/A	N/A	N/A	0.732
d	102	8 (8%)	0.996	-0.35	-8.43	-45.08	0.71	0.695
e	130	8 (6%)	1.002	1.26	-6.85	-47.89	0.81	0.733
f	99	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.700
g	153	14 (9%)	0.990	-0.26	-8.40	-42.95	0.82	0.759
h	80	7 (8%)	0.991	-0.07	-8.30	-47.92	0.68	0.676

Table 4.2: Registration results for Landsat over the DC area (870516)

Again, we failed to register windows (c) and (f) for the same reasons explained above; specifically, comparing SIFT correspondences against ground truth as explained above resulted in 4 true correspondences for window (c) and none for window (f). Comparing to [11], the original algorithm failed to register windows (d), (e), (f), and (g).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.7: Extracted windows from Landsat over the DC area (900812).

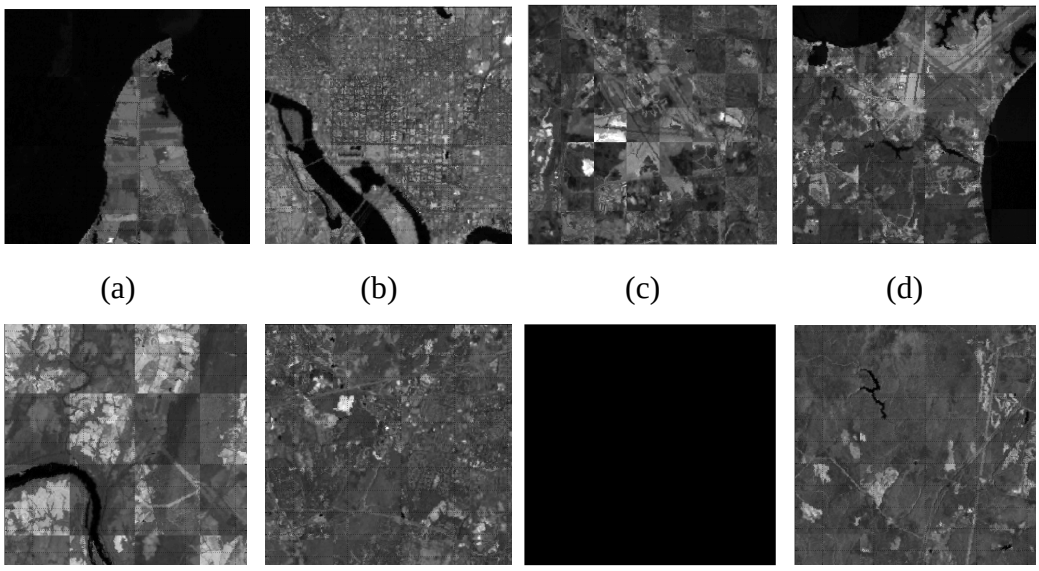
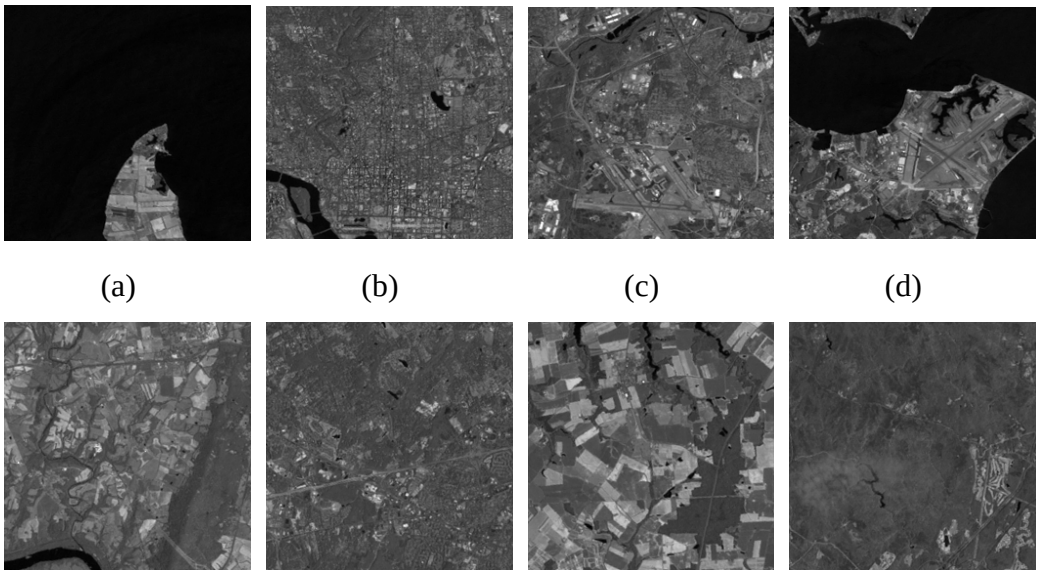


Figure 4.8: Registration results for Landsat windows (from scene 900812 over the DC area) vs. reference chips.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	68	16 (23%)	1.029	-0.95	-16.45	-30.63	0.75	0.679
b	85	21 (24%)	1.002	0.14	-15.64	-33.70	0.76	0.682
c	111	13 (11%)	1.01	0.69	-14.43	-32.07	0.79	0.737
d	101	25 (25%)	1.001	0.04	-16.10	-31.33	0.82	0.694
e	128	26 (20%)	0.997	0.06	-16.32	-35.78	0.87	0.749
f	99	13 (13%)	1.004	-1.37	-16.99	-35.82	0.71	0.704
g	133	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.737
h	98	17 (17%)	1.005	-0.06	-15.52	-34.59	0.66	0.702

Table 4.3: Registration results for Landsat over the DC area (900812)

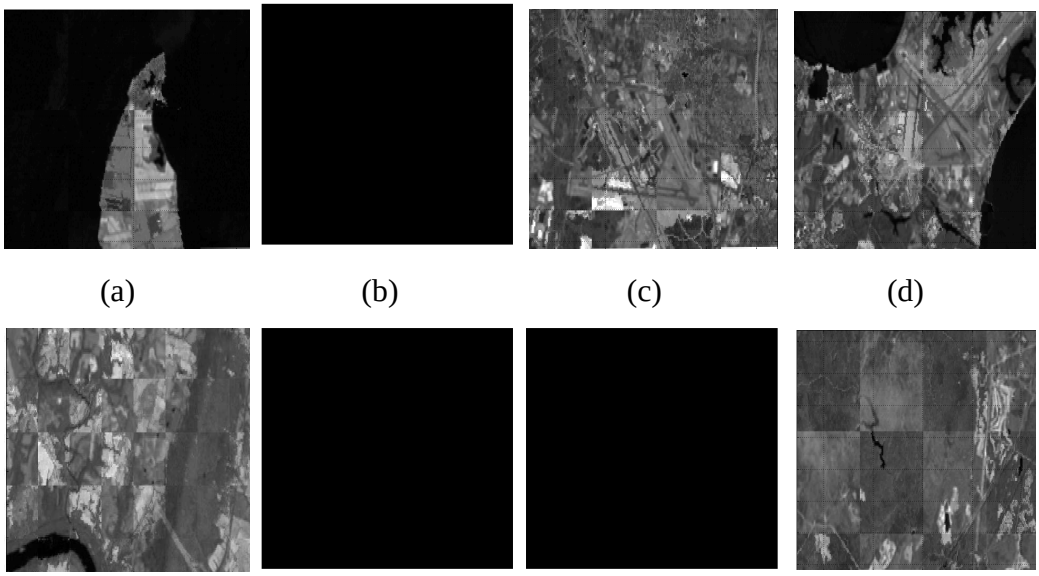
In this case we only failed to register window (g) which yielded only 4 true correspondences. These results are consistent with those of [11], with the main advantage of significantly shorter run-times.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.9: Extracted windows from Landsat scene 960711 over the DC area.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.10: Registration results for Landsat windows (from scene 960711 over the DC area) vs. reference chips.

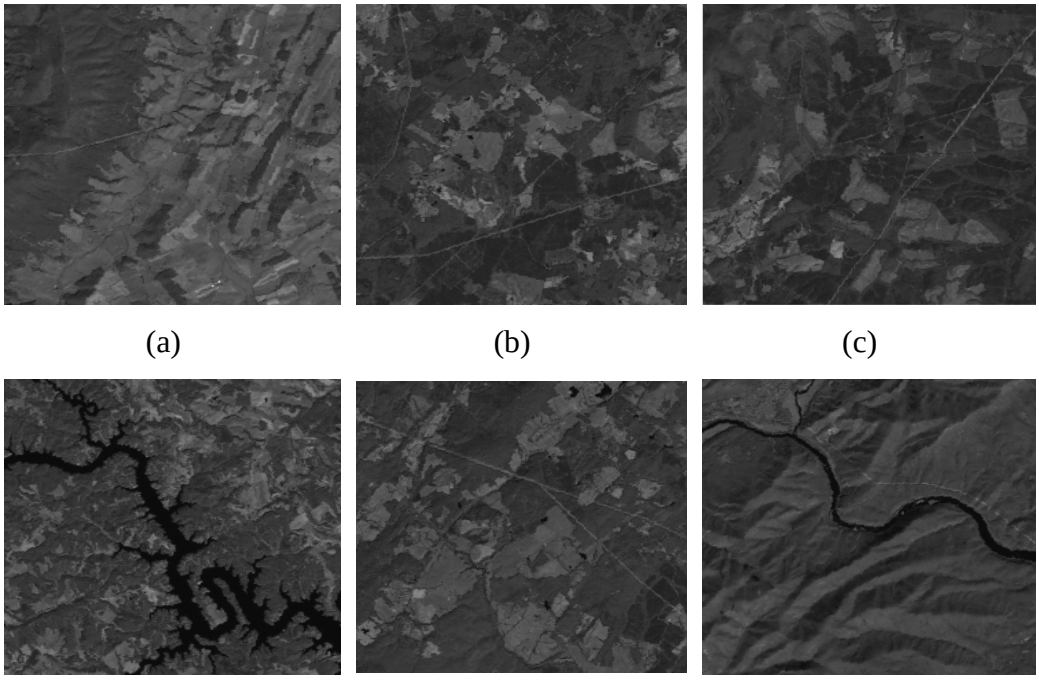
wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	68	18 (26%)	0.99	-0.33	-8.80	-102.73	0.72	0.658
b	73	1 (1%)	N/A	N/A	N/A	N/A	N/A	0.673
c	116	9 (7%)	1.002	-0.50	-7.69	-103.08	0.75	0.725
d	107	19 (17%)	0.998	0.17	-8.72	-101.4	0.95	0.710
e	126	12 (9%)	0.988	0.05	-8.31	-106.15	0.79	0.742
f	97	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.677
g	145	1 (1%)	N/A	N/A	N/A	N/A	N/A	0.738
h	102	12 (12%)	1.003	0.08	-8.64	-104.36	0.63	0.704

Table 4.4: Registration results for Landsat over the DC area (960711)

In this case we had 3 failures. Ground truth testing for these cases gave rise to 6, 0 and 3 true correspondences for windows (b), (f) and (g), respectively. Results for this scene in [11] show failures for windows (a), (b), (e) and (g).

4.2.2. Data over Virginia

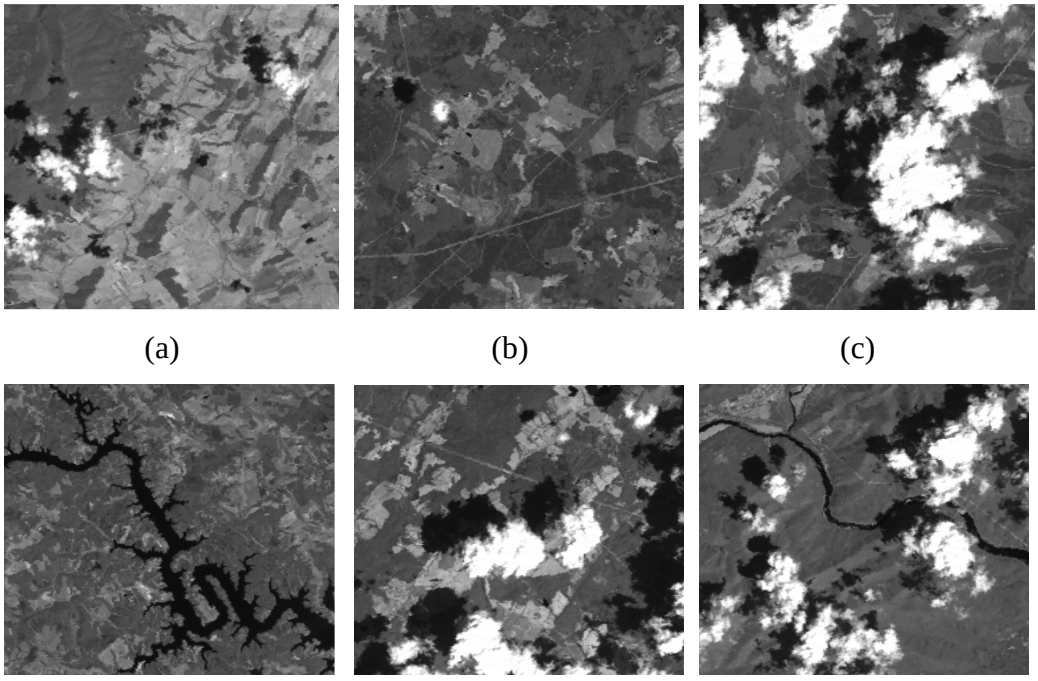
Our second set of chips is from the Virginia area. Six reference chips were extracted from a Landsat-7/ETM image taken on 07/10/99; these chips are depicted in Figure 4.11. Our 6-window sets were extracted from the same sensor on 04/08/99, 08/11/99, 28/02/00, and 22/08/00 (abbreviated as 990804, 991108, 000228 and 000822, respectively). Figures 4.12, 4.14, 4.16, and 4.18 show the corresponding window sets. Figures 4.13, 4.15, 4.17, and 4.19 and Tables 4.5-4.8 give the registration results (all images in this case are of size 250×250 pixels).



(a) (b) (c)

(d) (e) (f)

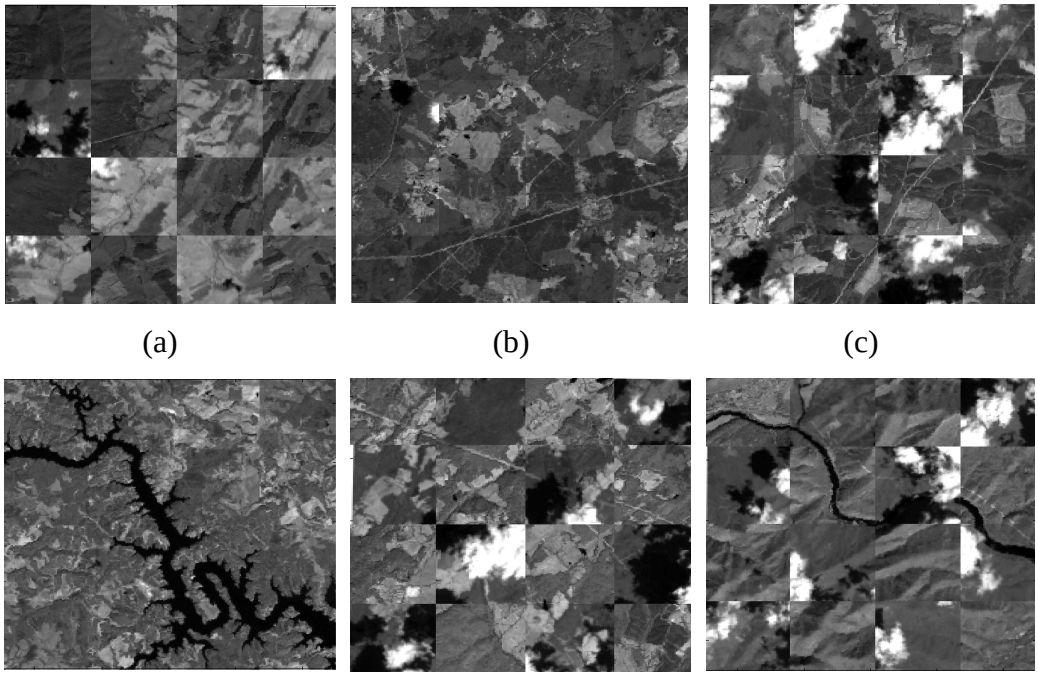
Figure 4.11: Six 250×250 reference chips from Virginia.



(a) (b) (c)

(d) (e) (f)

Figure 4.12: Extracted windows from Landsat scene 990804 over Virginia.



(a) (b) (c)

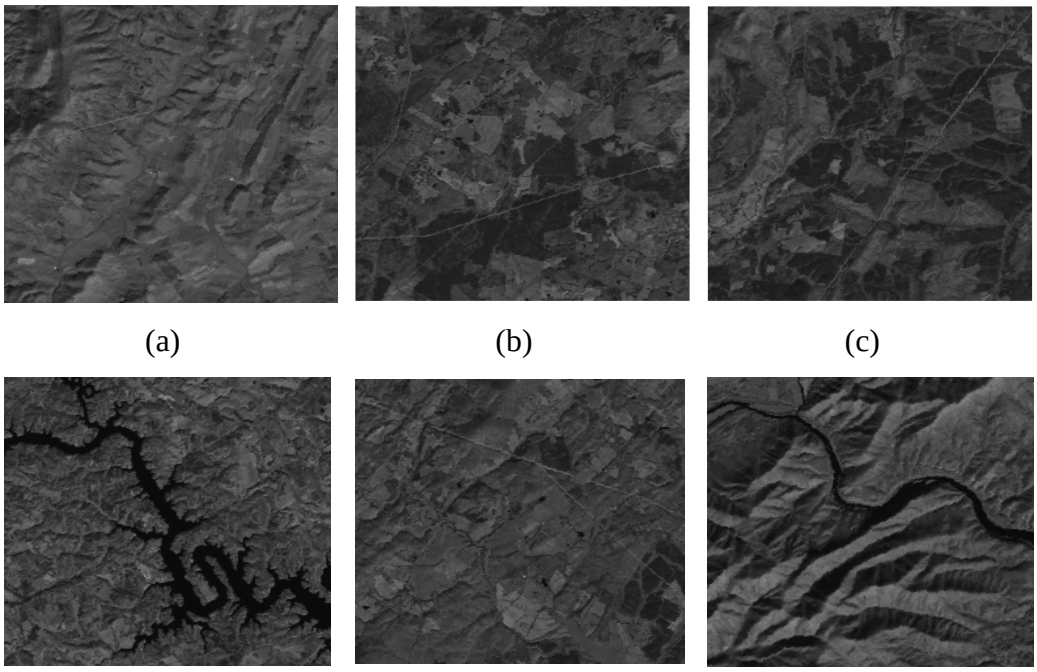
(d) (e) (f)

Figure 4.13: Registration results for Landsat windows (from scene 990804) vs. reference chips over Virginia.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	128	13 (10%)	1.000	-0.43	-0.06	3.66	0.77	0.747
b	117	55 (47%)	0.996	-0.05	-0.19	3.98	0.83	0.711
c	123	9 (7%)	1.002	-0.11	-0.35	3.95	0.69	0.740
d	156	102 (65%)	0.999	0.06	0.00	4.21	0.20	0.813
e	123	24 (19%)	1.00	-0.02	0.141	4.14	0.93	0.742
f	140	15 (10%)	1.01	0.82	-0.67	4.76	0.81	0.804

Table 4.5: Registration results for Landsat over Virginia (990804)

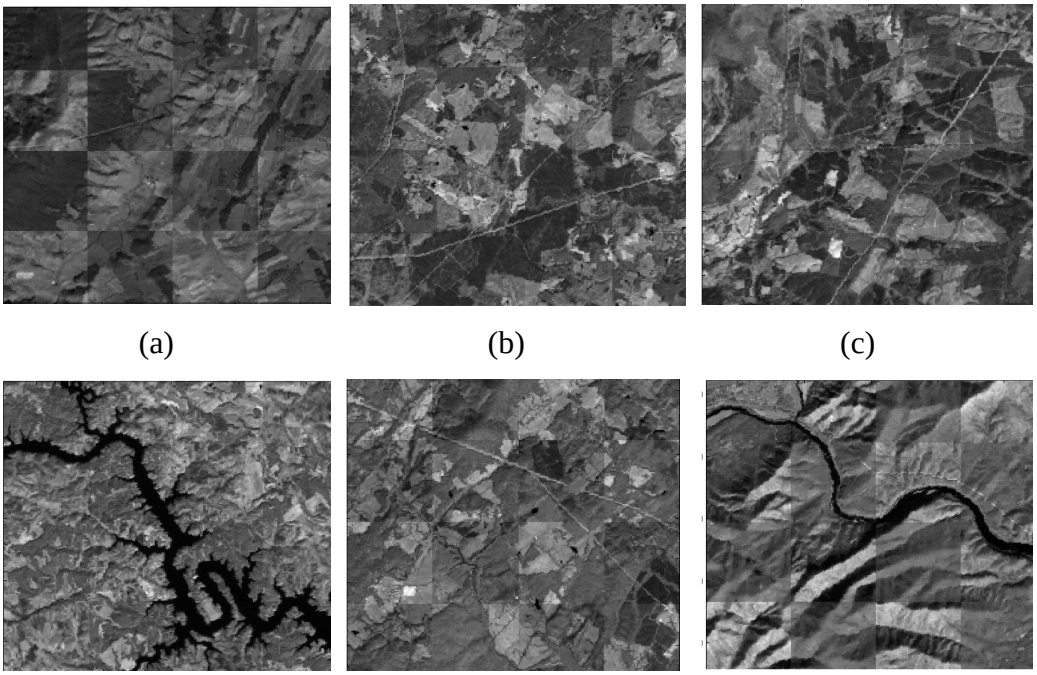
As we can see, no registration failures were observed for these windows as opposed to [11] were failures were observed for windows (c), (e), and (f) with substantial run-times (at least 14 seconds).



(a) (b) (c)

(d) (e) (f)

Figure 4.14: Extracted windows from Landsat scene 991108 over Virginia.



(a) (b) (c)

(d) (e) (f)

Figure 4.15: Registration results for Landsat windows (from scene 991108) vs. reference chips over Virginia.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	117	20 (17%)	0.996	-0.15	-0.57	13.45	0.98	0.712
b	123	52 (42%)	0.995	-0.02	-1.58	13.88	0.94	0.711
c	146	63 (43%)	1.000	-0.07	-1.29	13.47	0.69	0.780
d	136	69 (50%)	1.000	0.15	-1.13	13.23	0.54	0.792
e	101	25 (25%)	1.002	0.07	-1.52	13.86	0.73	0.738
f	163	88 (54%)	0.998	-0.03	-1.25	13.44	0.54	0.809

Table 4.6: Registration results for Landsat over Virginia (991108)

Again, no registration failures were observed for these windows, exactly as in [11], where run-times were at least 12 seconds.

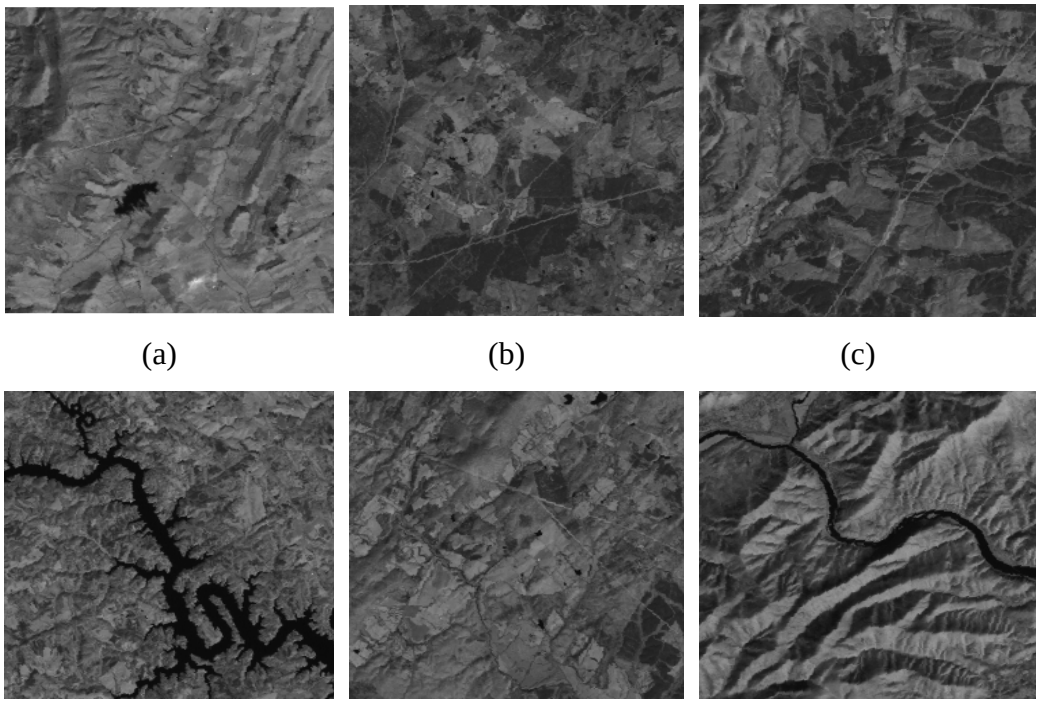
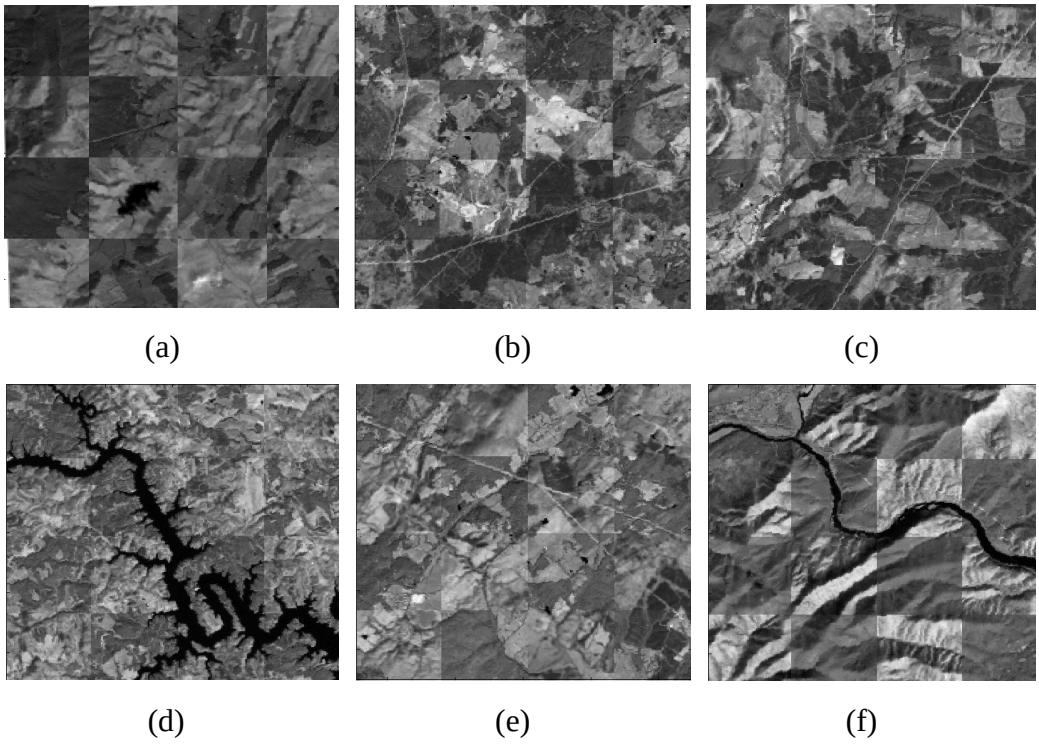


Figure 4.16: Extracted windows from Landsat scene 000228 over Virginia.



(a) (b) (c)

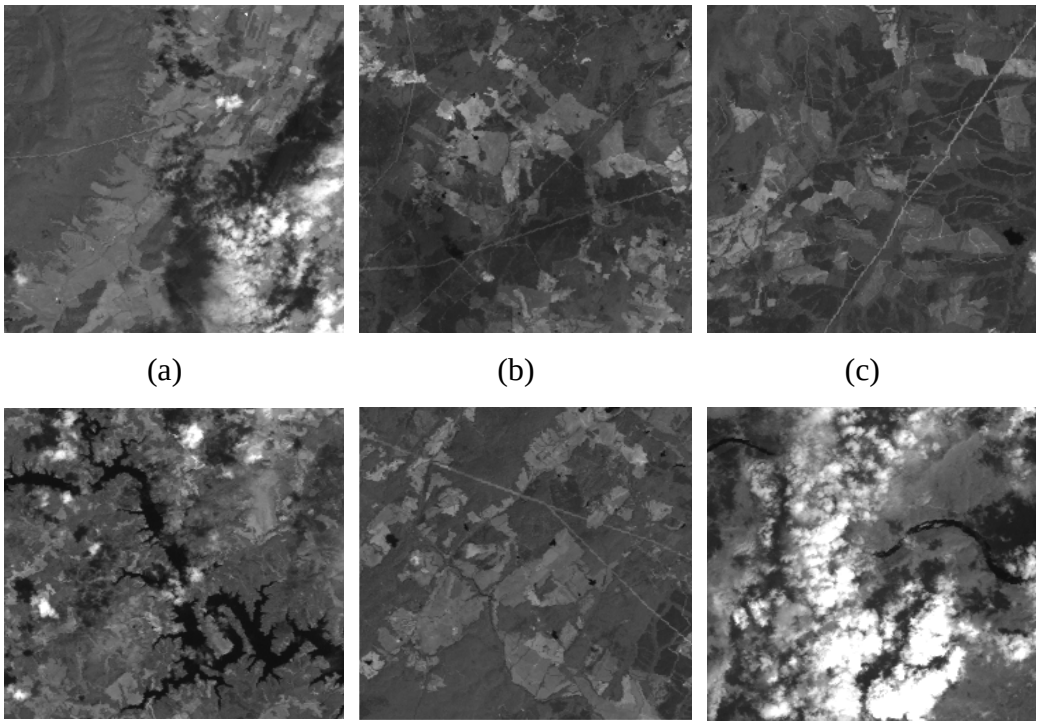
(d) (e) (f)

Figure 4.17: Registration results for Landsat windows (from scene 000228) vs. reference chips over Virginia.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	128	20 (15%)	1.000	-0.11	0.55	2.45	0.66	0.740
b	124	54 (43%)	0.999	-0.05	0.97	2.88	0.58	0.712
c	146	34 (23%)	0.996	0.07	1.09	2.57	0.82	0.777
d	147	75 (51%)	1.000	0.06	0.988	2.15	0.53	0.805
e	125	35 (28%)	0.997	-0.13	1.03	2.66	0.87	0.757
f	174	103 (59%)	0.998	0.07	0.85	2.30	0.77	0.825

Table 4.7: Registration results for Landsat over Virginia (000228)

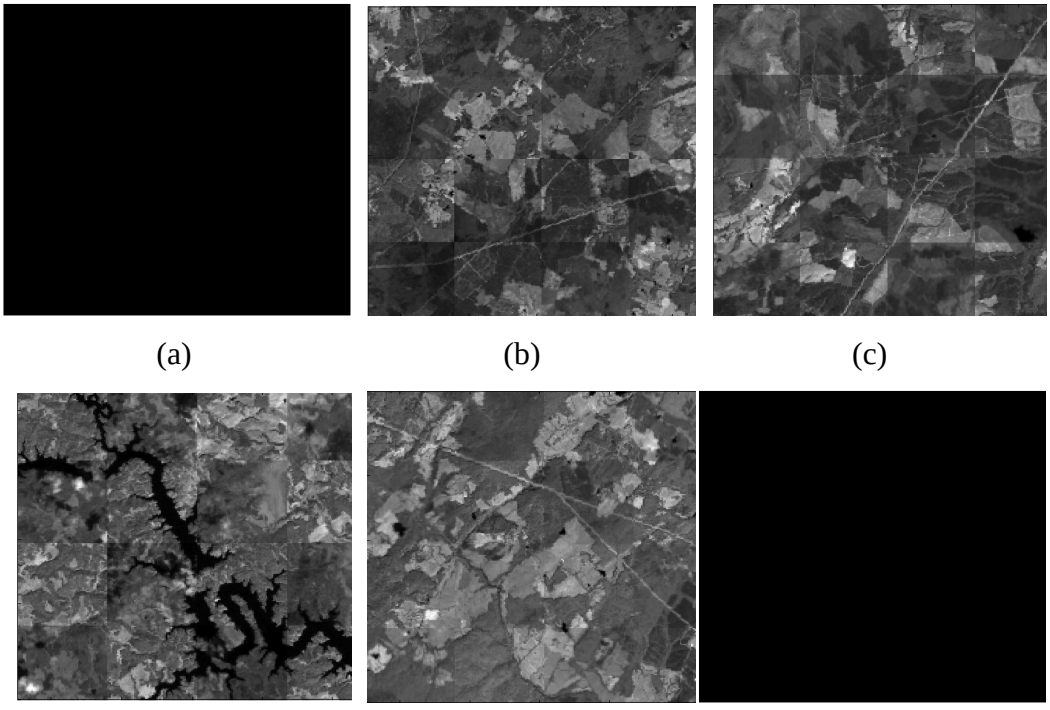
Once again, registration results are identical to those specified in [11].



(a) (b) (c)

(d) (e) (f)

Figure 4.18: Extracted windows from Landsat scene 000822 over Virginia.



(a) (b) (c)

(d) (e) (f)

Figure 4.19: Registration results for Landsat windows (from scene 000822) vs. reference chips over Virginia.

wind	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	115	2 (1%)	N/A	N/A	N/A	N/A	N/A	0.725
b	124	44 (35%)	0.999	-0.09	-0.05	9.86	0.89	0.733
c	146	59 (40%)	0.997	-0.23	-0.04	9.63	0.88	0.769
d	138	31 (22%)	1.004	0.09	-0.63	8.92	0.76	0.777
e	125	58 (46%)	1.001	0.16	-0.27	9.76	0.82	0.756
f	145	3 (2%)	N/A	N/A	N/A	N/A	N/A	0.797

Table 4.8: Registration results for Landsat over Virginia (000822)

In this case we had only two failures. Ground truth testing for these cases showed that window (a) had 5 true correspondences while window (f) had no true correspondences at all. It is quite evident that the two failures were caused due to substantial differences between the windows and the corresponding reference chips caused by large amount of clouds (especially window (f)). These explanations are similar to those offered in [11].

4.3. Multi-spectral/sensor Images

Our next 4 datasets consist of images obtained by Landsat/ETM and IKONOS in the near infra-red (NIR) and infra-red (IR) bands.

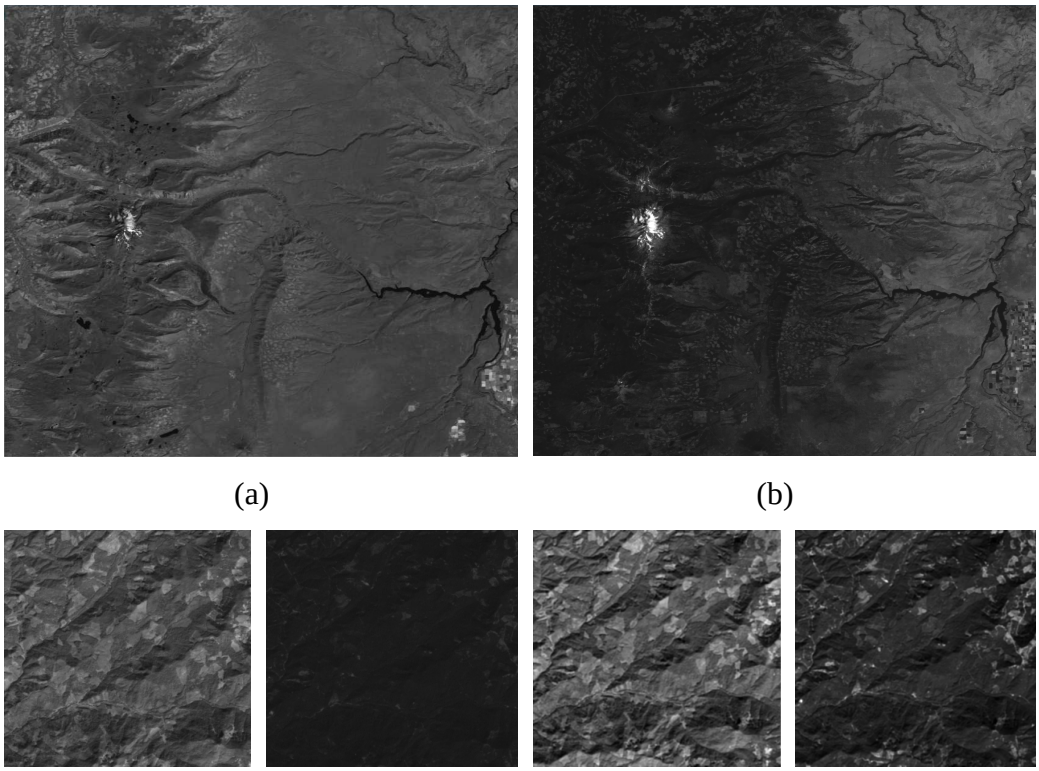
4.3.1. Cascade Area

For this area we had 6 images listed in Table 4.9 and shown in Figure 4.20. Seven image pairs were registered from this dataset; specifically, we tried to register pairs (a,b), (c,d), (c,e), (c,f), (d,e), (d,f), and (e,f). In this case we

were successful in all registration trials. The results are detailed in Table 4.10 and illustrated in Figure 4.21.

Image	Sensor	Band	Size
a	ETM	NIR	2048×2048
b	ETM	IR	2048×2048
c	ETM	NIR	312×312
d	ETM	IR	312×312
e	IKONOS	NIR	312×312
f	IKONOS	IR	312×312

Table 4.9: Images from Cascades area



(a) (b)

(c) (d) (e) (f)

Figure 4.20: Cascades area images.

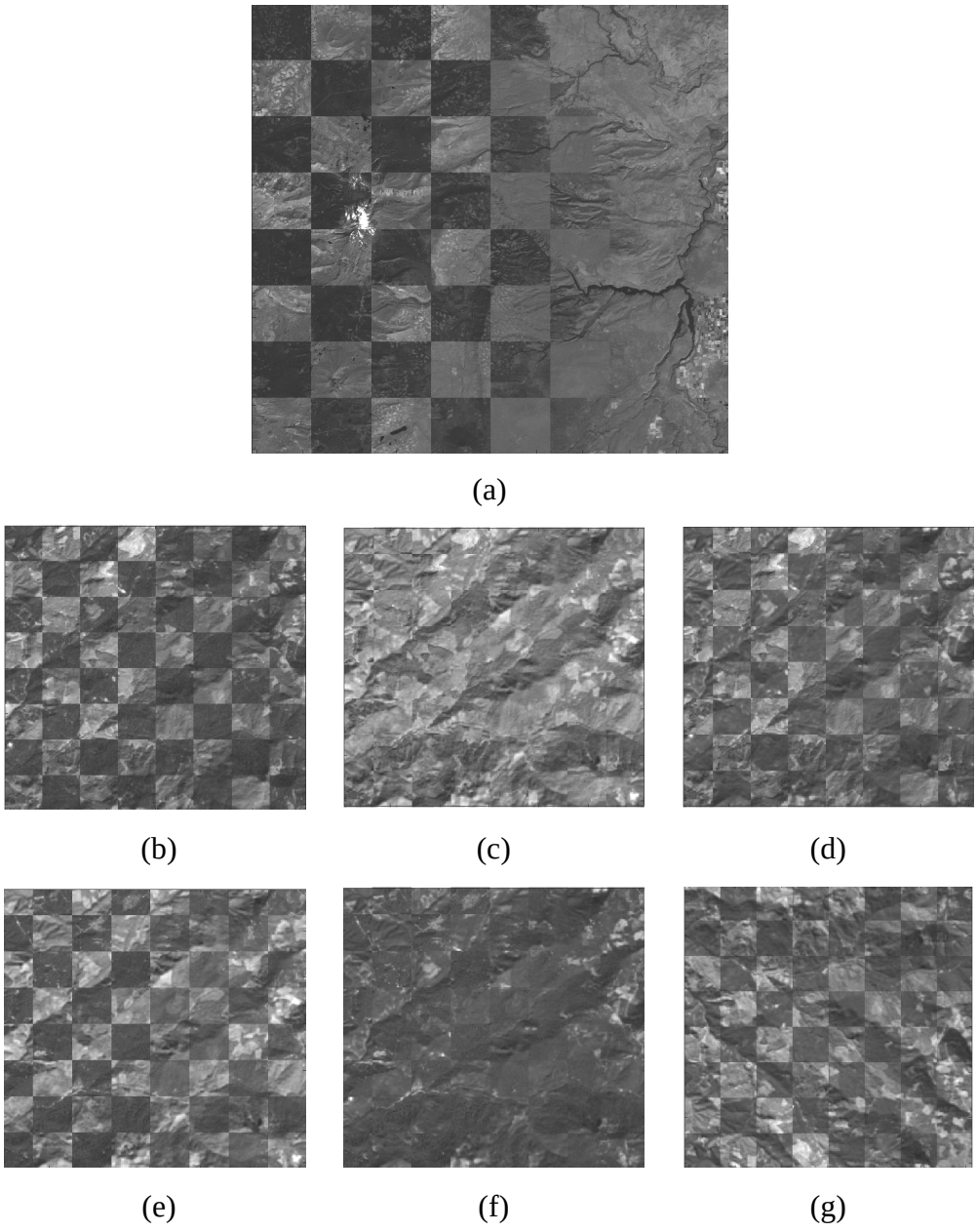


Figure 4.21: Registration results for image pairs from Cascades area.

Image pair	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
(a,b)	3719	70 (2%)	1.000	-0.07	1.00	0.77	0.82	9.40
(c,d)	153	33 (21%)	0.996	-0.13	-0.84	0.03	0.63	0.885
(c,e)	223	142 (63%)	1.064	-0.09	8.73	10.27	0.88	0.920
(c,f)	184	53 (28%)	1.064	-0.15	7.64	9.67	0.97	0.923
(d,e)	153	21 (13%)	1.061	-0.04	9.21	10.54	0.69	0.877
(d,f)	153	78 (50%)	1.064	-0.10	8.83	10.28	0.75	0.853
(e,f)	184	46 (25%)	1.001	-0.18	-0.29	-0.10	0.68	0.896

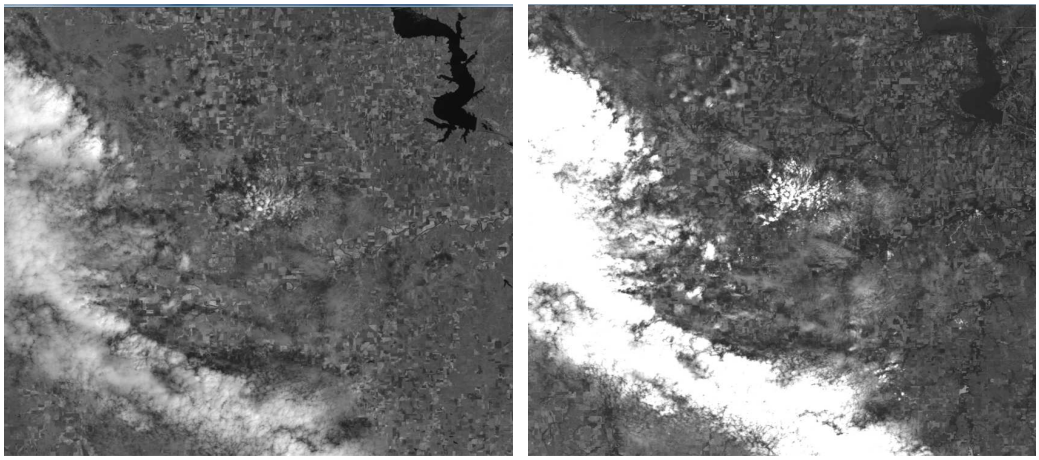
Table 4.10: Registration results for image pairs from Cascades area

4.3.2. Konza Area

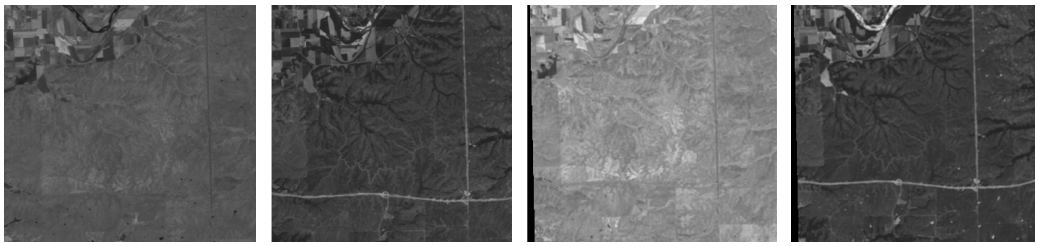
For this area we had 6 images listed in Table 4.11 and shown in Figure 4.22. Seven image pairs were registered from this dataset; specifically, we tried to register pairs (a,b), (c,d), (c,e), (c,f), (d,e), (d,f), and (e,f). In this case we succeeded to register only pairs (a,b), (c,e), and (d,f). The results are detailed in Table 4.12 and illustrated in Figure 4.23.

Image	Sensor	Band	Size
a	ETM	NIR	2048×2048
b	ETM	IR	2048×2048
c	ETM	NIR	344×336
d	ETM	IR	344×336
e	IKONOS	NIR	344×336
f	IKONOS	IR	344×336

Table 4.11: Images from Konza area



(a) (b)

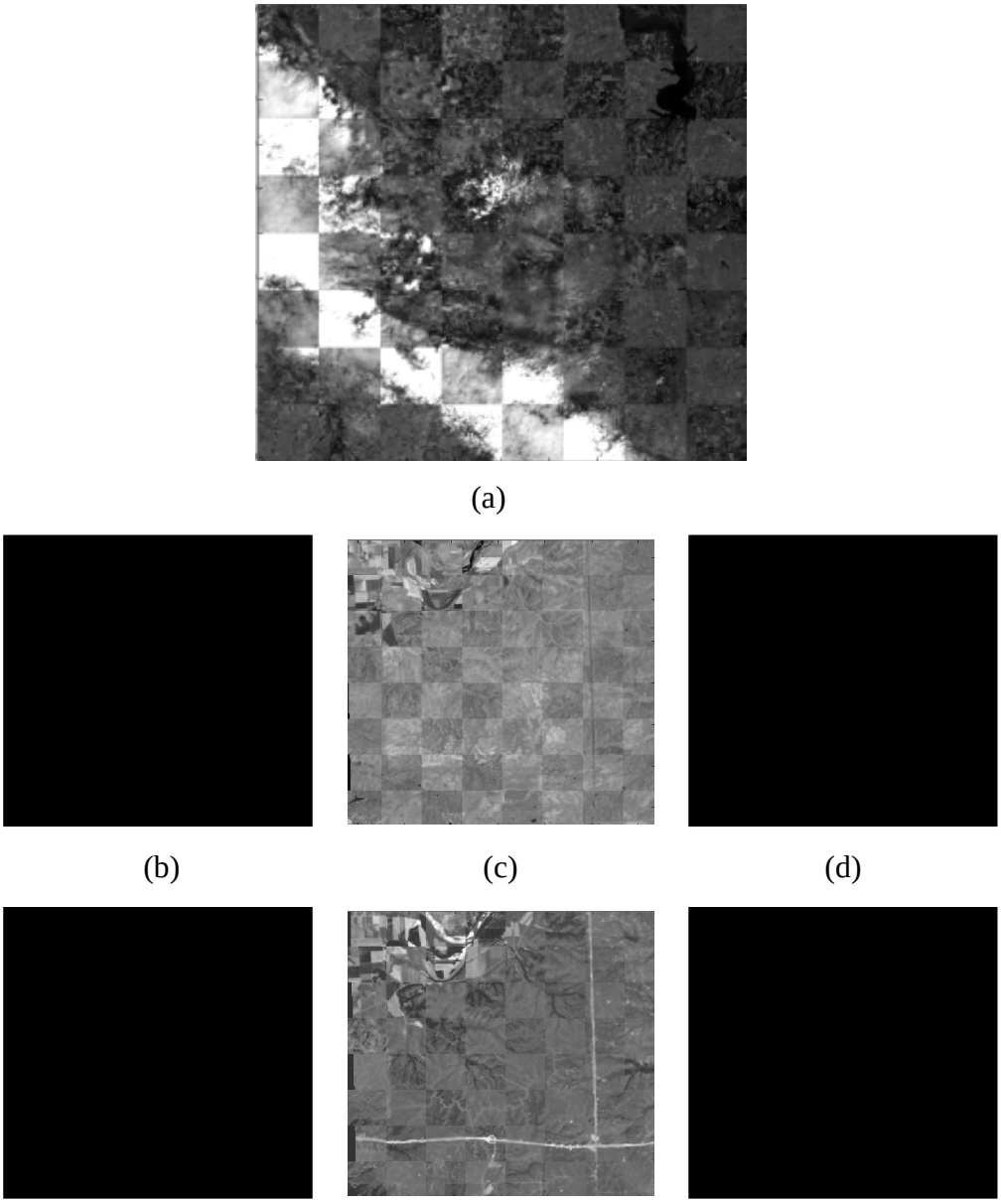


(c) (d) (e) (f)

Figure 4.22: Konza area images.

Image pair	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
(a,b)	2610	84 (3%)	1.002	0.05	-0.12	0.35	0.85	17.41
(c,d)	209	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.970
(c,e)	209	45 (21%)	1.061	0.00	13.56	12.29	0.84	0.974
(c,f)	209	1 (~0%)	N/A	N/A	N/A	N/A	N/A	0.976
(d,e)	227	0 (0%)	N/A	N/A	N/A	N/A	N/A	0.989
(d,f)	210	104 (50%)	1.064	-0.05	12.42	11.54	0.94	0.985
(e,f)	210	1(~0%)	N/A	N/A	N/A	N/A	N/A	0.987

Table 4.12: Registration results for image pairs from Konza area



(a) (b) (c) (d) (e) (f) (g)

Figure 4.23: Registration results for image pairs from Konza area.

We found no true correspondences at all for all registration failures observed for this area.

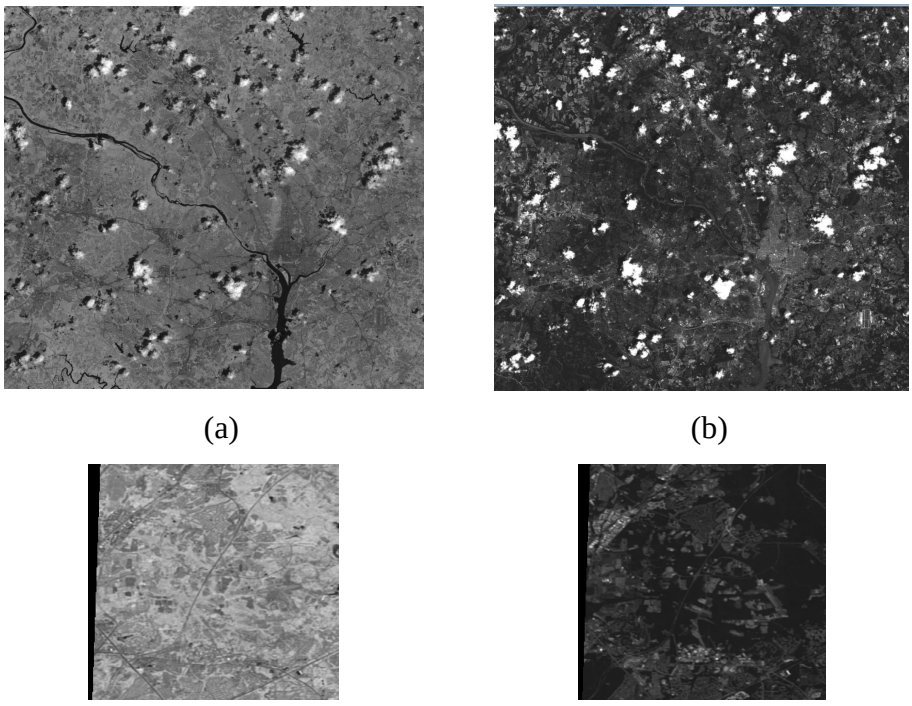
4.3.3. USDA Area

For this area we had 4 images listed in Table 4.13 and shown in Figure 4.24. Two image pairs were registered from this dataset; specifically, we tried to

register pairs (a,b) and (c,d). In this case we succeeded to register pair (a,b). The results are detailed in Table 4.14 and illustrated in Figure 4.25.

Image	Sensor	Band	Size
a	ETM	NIR	2048×2048
b	ETM	IR	2048×2048
c	IKONOS	NIR	392×296
d	IKONOS	IR	392×296

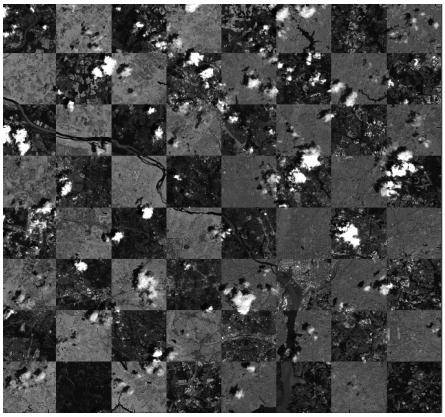
Table 4.13: Images from USDA area



(a) (b)

(c) (d)

Figure 4.24: USDA area images.



(a)

(b)

Figure 4.25: Registration results for image pairs from USDA area.

Image pair	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
(a,b)	1760	22 (1%)	1.001	0.10	0.59	1.47	0.91	15.85
(c,d)	228	0 (0%)	N/A	N/A	N/A	N/A	N/A	1.00

Table 4.14: Registration results for image pairs from USDA area

4.3.4. Virginia Area

For this area we had 6 images listed in Table 4.15 and shown in Figure 4.26. Seven image pairs were registered from this dataset; specifically, we tried to register pairs (a,b), (c,d), (c,e), (c,f), (d,e), (d,f), and (e,f). In this case we succeeded to register pairs (a,b), (c,e), (d,f), and (e,f). The results are detailed in Table 4.16 and illustrated in Figure 4.27.

Image	Sensor	Band	Size
a	ETM	NIR	2048×2048
b	ETM	IR	2048×2048
c	ETM	NIR	344×344
d	ETM	IR	344×344
e	IKONOS	NIR	344×344
f	IKONOS	IR	344×344

Table 4.15: Images from Virginia area

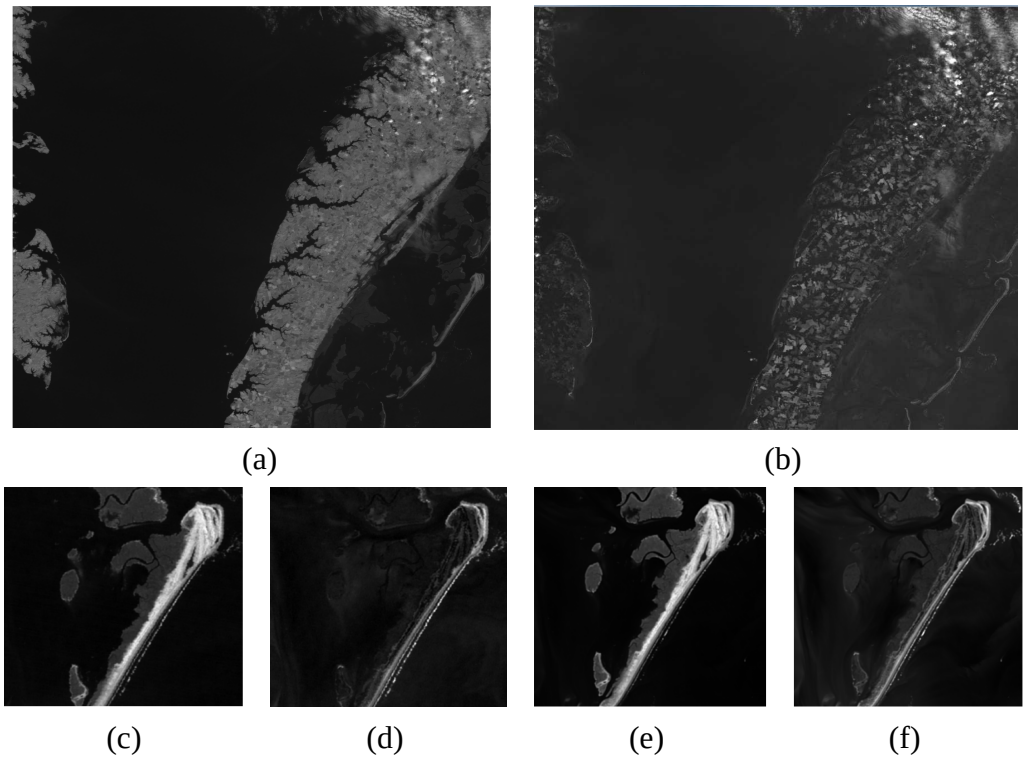
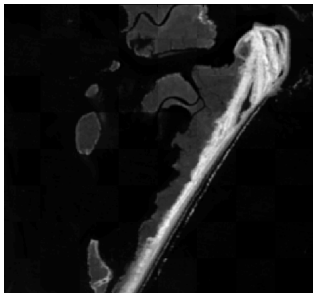


Figure 4.26: Virginia area images.

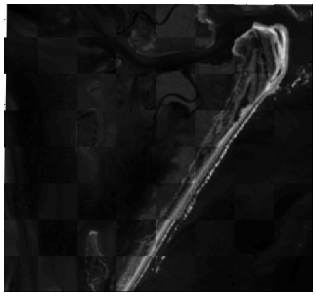
Image pair	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
(a,b)	364	34 (9%)	1.002	0.10	-2.89	0.41	0.74	6.49
(c,d)	187	0 (0%)	N/A	N/A	N/A	N/A	N/A	1.02
(c,e)	187	54 (28%)	1.07	0.14	12.67	13.56	0.88	0.97
(c,f)	187	1 (~0%)	N/A	N/A	N/A	N/A	N/A	0.99
(d,e)	197	1 (~0%)	N/A	N/A	N/A	N/A	N/A	1.02
(d,f)	197	24 (12%)	1.082	0.23	10.50	13.05	0.74	1.00
(e,f)	197	28 (14%)	0.99	0.14	-1.10	-0.78	0.68	0.98

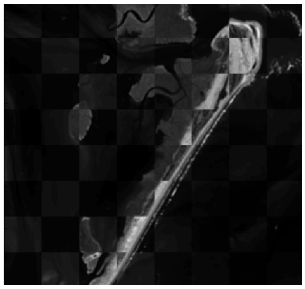
Table 4.16: Registration results for image pairs from Virginia area

(a)



(b) (c) (d)





(e) (f) (g)

Figure 4.27: Registration results for image pairs from Virginia area.

We found only 2 true correspondences for the failure in image pair (c,d) and no true correspondences at all for the other two failures.

4.4. Miscellaneous Image Pairs

Our last dataset consists of various sensors and bands (from the UCSB website: http://vision.ece.ucsb.edu /registration /satellite/testimag/index.htm). Table 4.17 presents a summary of the image pairs shown in Figure 4.28. The registration results are summarized in Table 4.18 and Figure 4.29.

Pair #	Scene	Image Pair	Size
a	Desert	1. Optical Image 2. Simulated Transformation	1. 512×512 2. 512×512
b	Coast line	1. Landsat from 1988 2. Landsat from 1986	1. 400×400 2. 400×400
c	Agricultural	1. Landsat Band 5 from 9/9/90 2. Landsat Band 5 from 18/7/94	1. 512×512 2. 512×512
d	Coast line	1. Landsat from 1988 2. Landsat from 1990	1. 600×600 2. 600×600
e	USCB	1. Optical Image. 2. Simulated Transformation	1. 386×306 2. 472×335
f	Coast line	1. AVIRIS Band 39 2. AVIRIS Band 39	1. 256×256 2. 256×256
g	Rain forest	1. Landsat Band 5 from 7/6/92 2. Landsat Band 5 from 15/7/94	1. 512×512 2. 512×512
h	Casitas lake	1. Landsat Band 5 from 1984 2. Landsat Band 7 from 1986	1. 600×600 2. 600×600
i	Gibraltar	1. Landsat Band 5 from 1984 2. Landsat Band 7 from 1986	1. 600×600 2. 600×600
j	Mountains	1. Landsat 2. Landsat	1. 512×512 2. 512×512
k	Coast line	1. Landsat Band 3 from 1988 2. Landsat Band 5 from 1988	1. 512×512 2. 512×512
l	Mountains	1. Landsat Band 1 from 1988 2. Landsat Band 3 from 1988	1. 512×512 2. 512×512
m	Mountains	1. Landsat Band 4 from 1988 2. Landsat Band 7 from 1988	1. 512×512 2. 512×512
n	Unknown	1. JERS 1 from 10/10/1995 2. JERS 1 from 13/8/1996	1. 256×256 2. 256×256
o	Brasilia	1. SPOT Band 3 from 8/8/95 2. Landsat Band 4 from7/6/94	1. 256×256 2. 256×256

Table 4.17: List of miscellaneous images

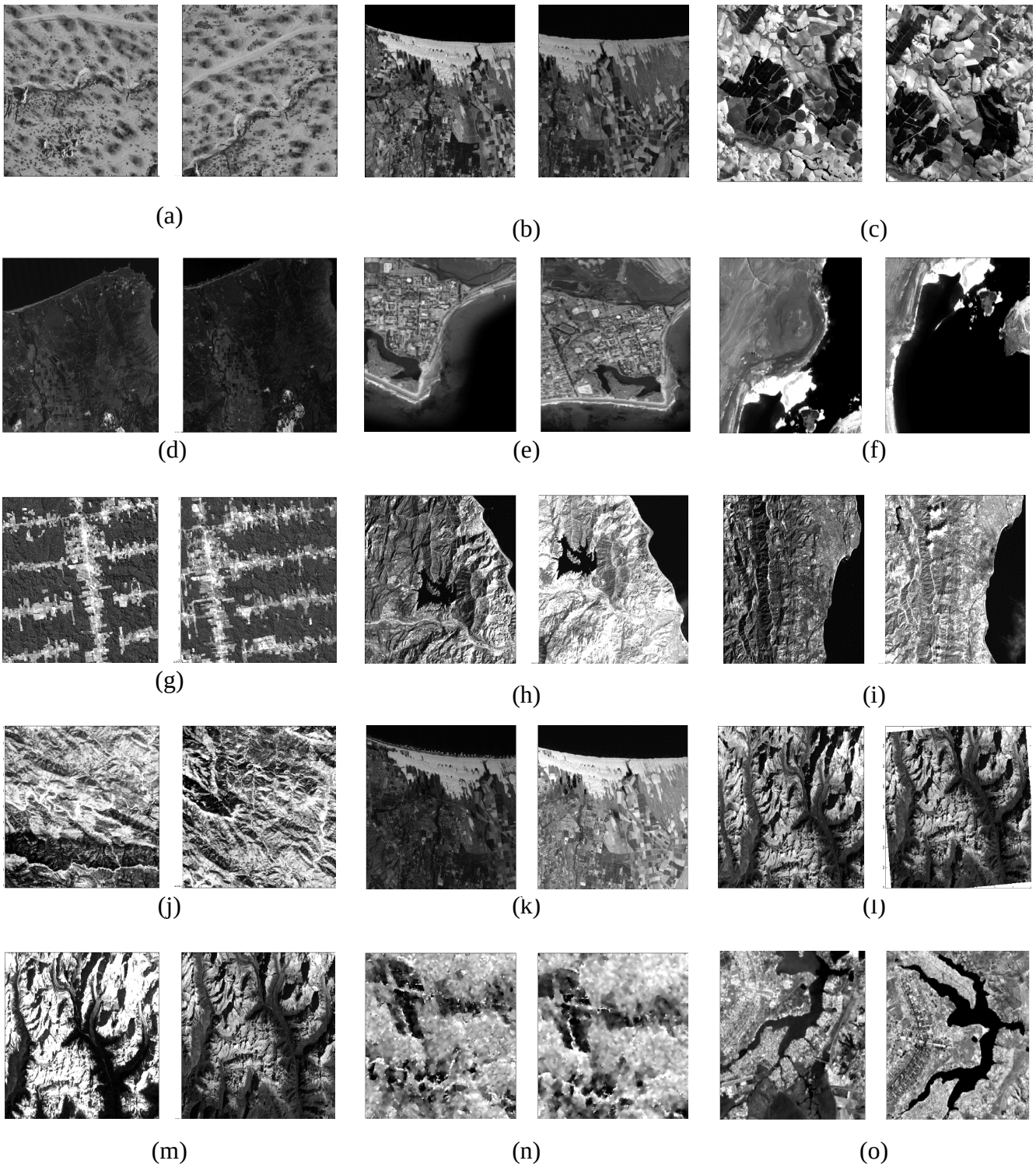
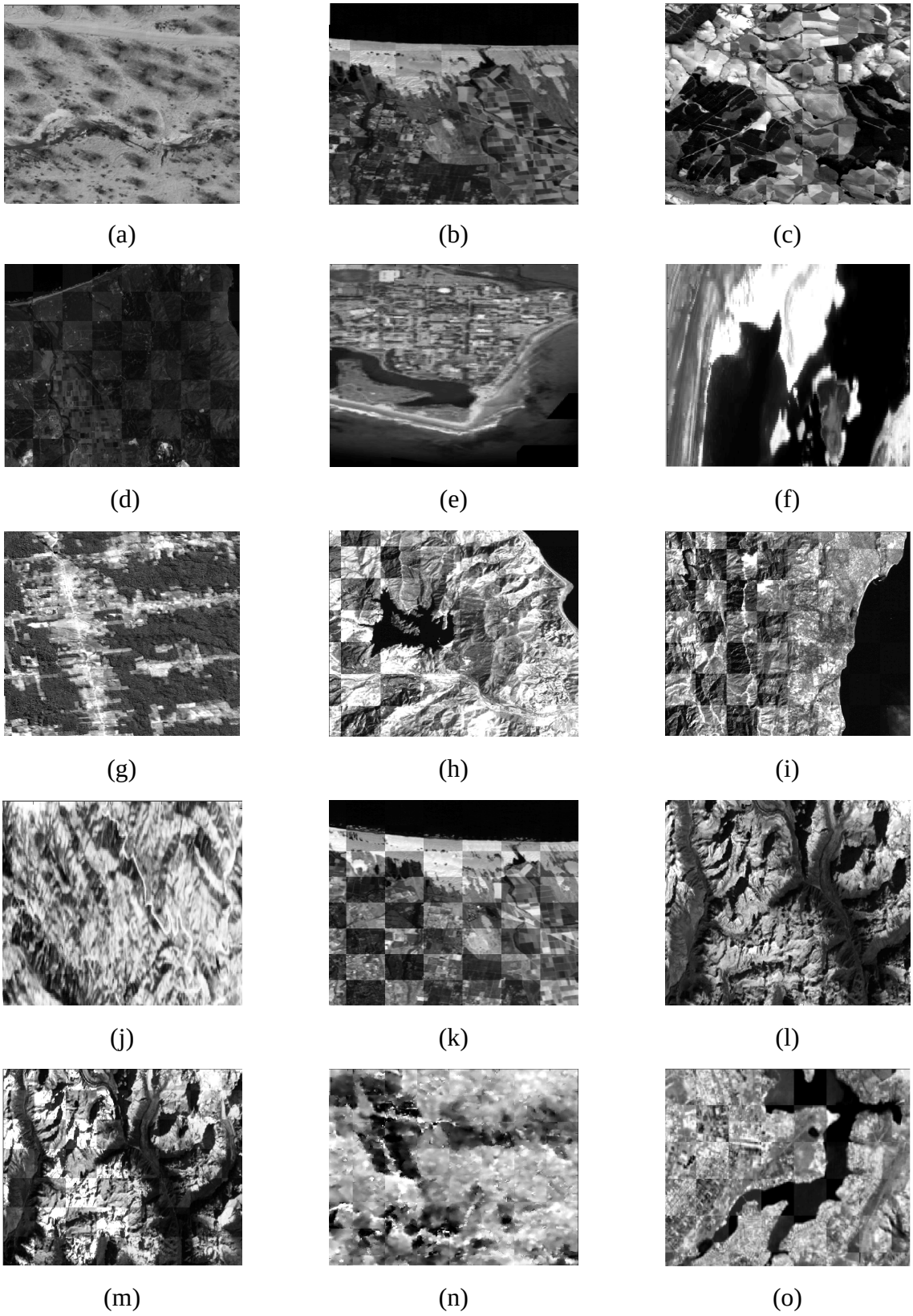


Figure 4.28: Miscellaneous image pairs : (a) Desert, (b) coast line, (c) agricultural, (d) coast line, (e) UCSB, (f) coast line, (g) rain forest, (h) Casitas lake, (i) Gibraltar, (j) mountains, (k) coast line, (l) mountains, (m) mountains, (n) unknown ,(o) Brasilia.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 4.29: Registration results for image pairs of Fig. 4.28.

Image pair	# init corres.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	531	284(53%)	0.996	-29.94	-52.15	137.47	0.58	2.00
b	310	53(17%)	1.00	0.021	-127.05	-7.15	0.57	1.25
c	562	64(11%)	0.99	-0.00	77.65	-87.22	0.67	2.05
d	211	12(5%)	1.005	-0.26	-1.80	-81.01	0.65	1.87
e	135	109(80%)	1.00	19.98	-96.93	-75.61	0.21	1.12
f	40	12(30%)	0.99	1.43	23.70	-166.66	0.59	0.66
g	553	34(6%)	1.00	-0.02	-183.28	36.98	0.66	2.14
h	797	82(10%)	0.99	0.285	127.8	112.34	0.75	2.61
i	684	102(14%)	1.00	0.17	-3.89	110.98	0.71	2.49
j	763	112(14%)	1.00	15.02	-82.05	307.49	0.31	2.31
k	252	84(33%)	0.99	0.043	78.11	-0.04	0.62	1.2
l	340	149(43%)	1.00	-5.02	-43.39	-88.08	0.43	1.37
m	342	34(10%)	1.00	-0.09	-17.95	64.69	0.75	1.34
n	153	20(13%)	1.00	0.42	10.35	-20.92	0.88	0.78
o	115	10(8%)	0.97	-0.913	-7.40	-78.67	0.57	0.72

Table 4.18: Registration results for Image pairs of Table 4.17.

4.5. Analysis

4.5.1. Algorithm's Performance

As explained above, we consider the registration as success if the RMSE is smaller than 1 pixel. On the other hand, if the transformation cannot be computed due to an insufficient number of filtered inliers, the registration fails. In accordance with the above, we observed 76 successes out a total of 94 registration trials, i.e., ~81% success rate. Also, as shown above, the algorithm's run-time varies between less than a second for 256×256 images to several seconds for very large images of size 2048×2048; these running

times can be further improved by a more powerful computer and/or more efficient coding. In any event, they are much faster than those of other similar SIFT-based algorithms which usually vary between dozens to hundreds of seconds for similar image sizes.

Another important issue is that of verification/validation of the transformation correctness due to some indication as to whether or not the algorithm finds the correct transformation (in an RMSE sense). Figure 4.30 depicts the distribution of the number of inliers for all registration trials; we can easily spot a gap between 4 to 7 inliers and exploit it for verification of our transformation as follows. If the number of inliers is larger than or equal to 7, then we accept the transformation found by the MS-SIFT algorithm; otherwise, we will assume the algorithm has failed to find a proper transformation. We should emphasize that in all of our registration trials, we did not encounter a single case where evident, concrete peaks in the scale, orientation, and translation histograms were found and the resulting transformation was wrong. Registration failures were caused only when no

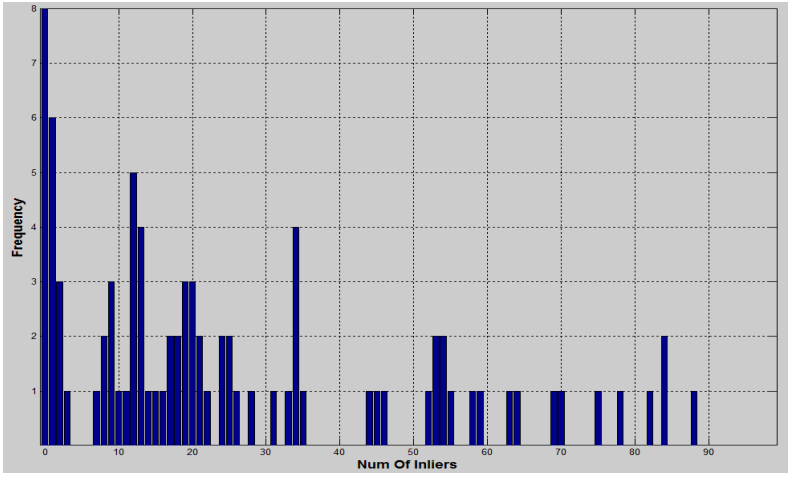


Figure 4.30: Inlier distribution.

evident peaks were found. This rule of thumb can serve as a practical indication as to the correctness of our method.

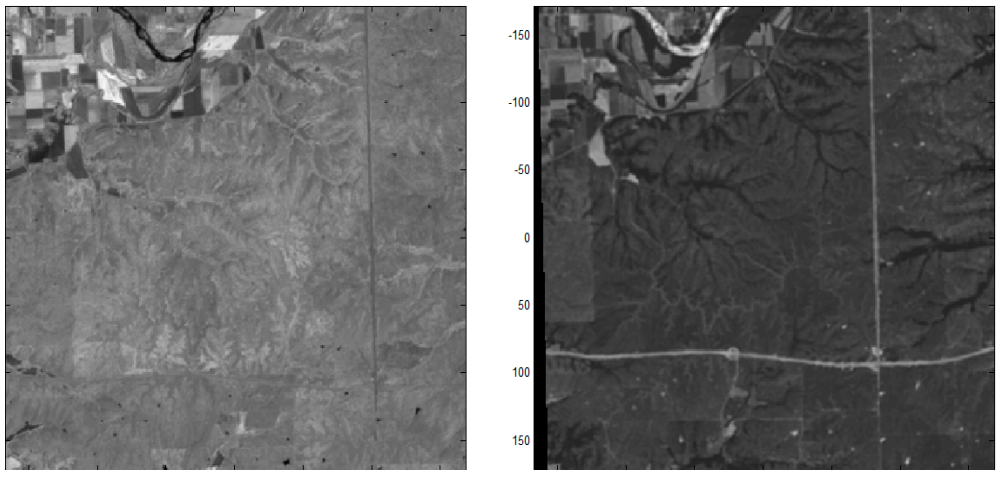
4.5.2. Failure Analysis and Enhancements

We can divide our failures into two categories:
(1) The first category consists of failures due to substantial differences between the reference and sensed images. These differences can be due to changes in land cover (e.g., deforestation, lake dehydration, etc.), insufficient overlap between the images and instantaneous differences upon acquisition of one of the images (e.g., cloud appearance). It is impossible to quantify to what extent and where the images should overlap in order for our registration algorithm to succeed. In any event, our failures from this category were caused usually due to clouds which were present in one image but not in the other.

(2) The second category is of failures due to differences in intensities between the images. Since the SIFT algorithm is based on gradient values, substantial differences in the intensity distribution between the reference and sensed images can lead to a wrong match between corresponding SIFT descriptors and eventually to no evident peaks in the distributions of the SIFT characteristics. Fortunately, differences in intensities between images can be anticipated, since these differences usually stem from the use of different sensors. We can thus exploit some basic image processing techniques to enhance the SIFT results, as well as our MS-SIFT algorithm.

Let us revisit the image pair over the Konza area shown in Figure 4.31. The reference image is the NIR band of a Landsat/ETM image and the sensed image is in the IR band of IKONOS (the size of both images is 344×336 pixels). Figure 4.32 shows the histograms of the various SIFT characteristics. As we can see, there are no evident, concrete peaks in the translation histograms; in addition, there are two possible peaks in the orientation histogram. This kind of histograms will usually cause our algorithm to fail, as only a small amount of inliers (less than five) will be found. In this specific case, only one possible inlier (out of 209 correspondences) was found. In order to improve our results, we employ some basic image processing techniques. First we enhance both images with the aid of the Laplacian operator ∇^2, defined for a digital image $f(x, y)$ as:

(1) $$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$



(a) (b)

Figure 4.31: Image pair over Konza area: (a) Reference ETM NIR and (b) sensed IKONOS IR.

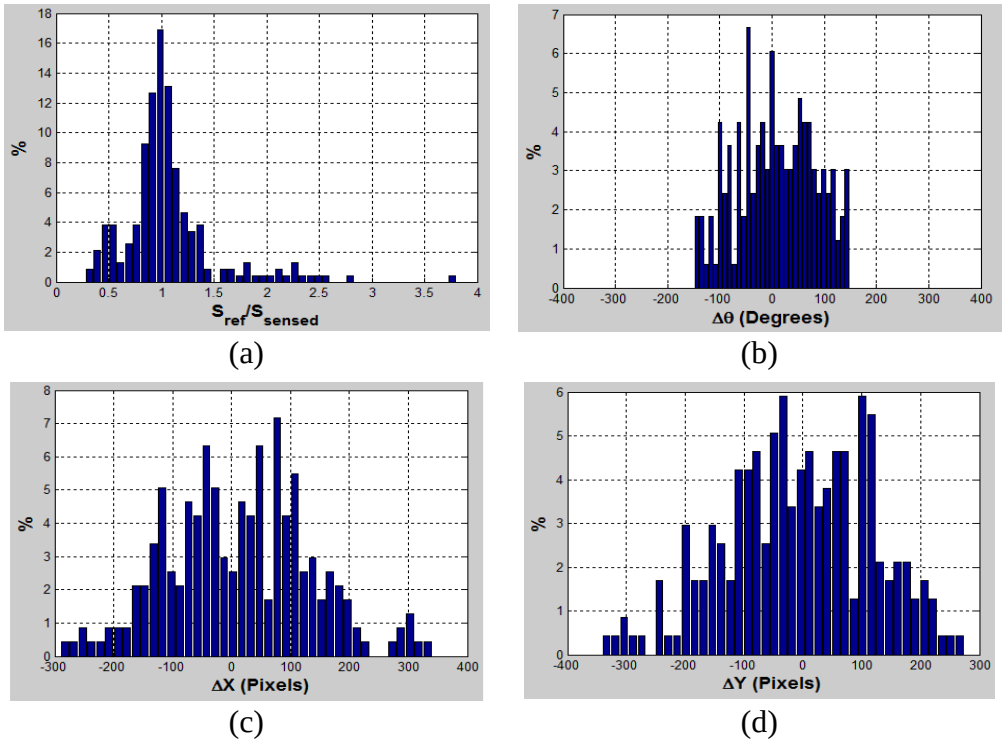


Figure 4.32: Histograms of SIFT characteristics for image pair of Fig. 4.31: (a) Scale ratios, (b) orientation differences, (c) horizontal shift and (d) vertical shift.

we have sharpened both images by taking:

(2) $$\tilde{f}(x,y) = f(x,y) - k\nabla^2 f(x,y)$$

where $\tilde{f}(x,y)$ is the sharpened image and k is a positive constant ($k = 0.05$ was picked for the reference image and $k = 0.75$ was picked for the sensed image); see [16]. In addition, we reverse the intensity level of the reference image r (i.e., $s = T(r) = 1 - r$ where we assume that $r \in [0,1]$). The reason for the sharpening procedure is that the SIFT algorithm uses the difference of Gaussians (DoG) in order to find image key-points, and since the DoG is analogous to an edge detection operator, it is reasonable to emphasize edges in both images in order to obtain reliable key-points. The purpose of intensity

reversal in the reference image is to equalize intensity distributions between the two images. Figures 4.33 and 4.34 depict the images and the resulting

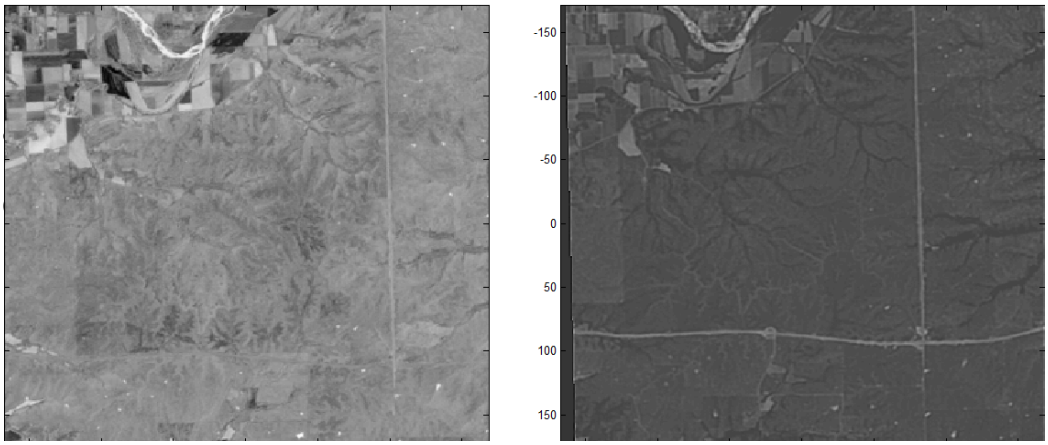
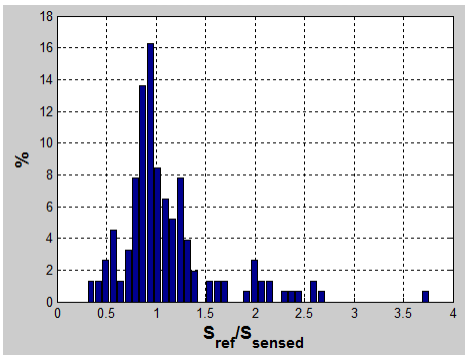
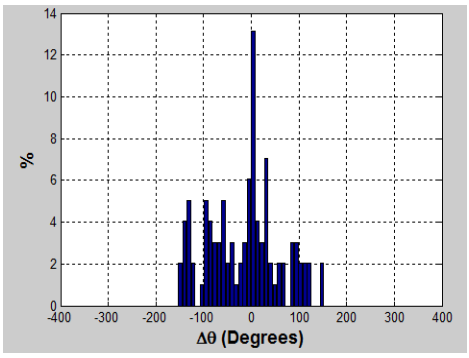


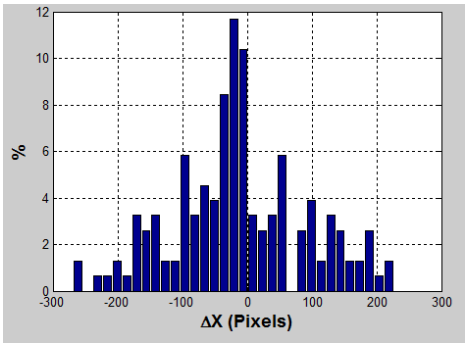
Figure 4.33: Image pair over Konza after sharpening and intensity transformation.



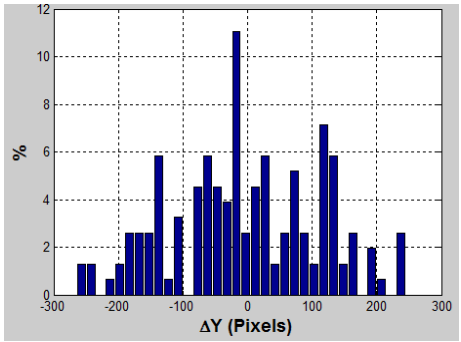
(a)



(b)



(c)



(d)

Figure 4.34: Histograms of SIFT characteristics for image pair of Fig. 4.33: (a) Scale ratios, (b) orientation differences, (c) horizontal shift, and (d) vertical shift.

histograms of the SIFT characteristics. Now, the peaks obtained are unique and can easily be spotted; in this case, we had 154 initial correspondences, 17 of which survived the filtering process (i.e., ~11%). This was sufficient to compute reliably the transformation parameters which were $< s, \theta, t_x, t_y > = < 1.05, -0.63^0, 13.06, 13.51 >$, with an RMSE of 0.76 pixels. The registration results are shown in Figure 4.35. In the same spirit (i.e., using the same values of k), Figure 4.36 and Table 4.19 summarize the successful registration results for several image pairs for which our algorithm originally failed to register. (Run-times were measured inside of MATLAB.)

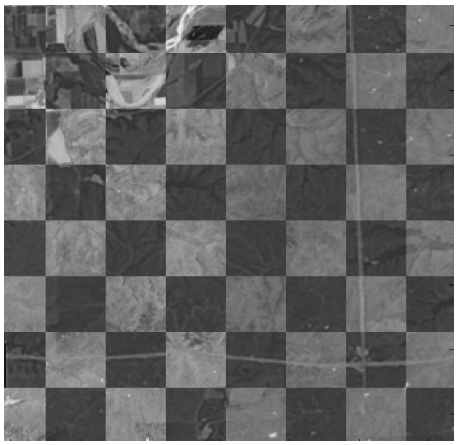
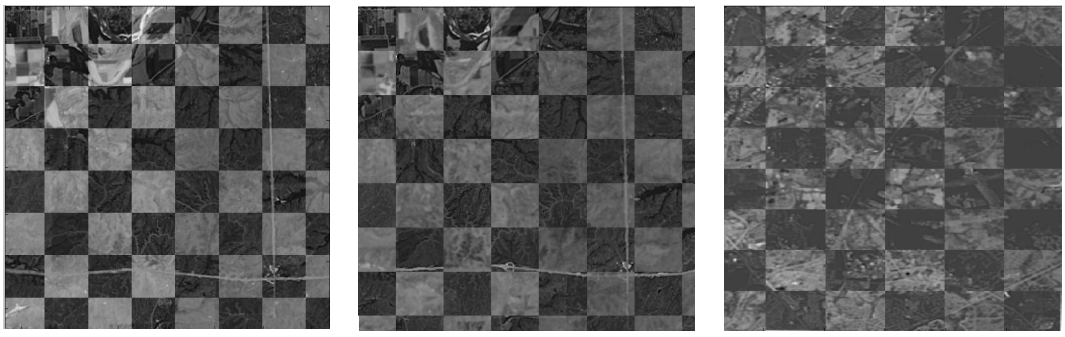


Figure 4.35: Improved registration results for image pair over the Konza area.



(a) (b) (c)

Figure 4.36: Improved registration results after preprocessing enhancement.

Img. pair	Details	# init. corresp.	# inliers	s	θ [deg]	t_x [pix]	t_y [pix]	RMSE [pix]	run-time [s]
a	Konza ETM NIR Konza ETM IR	183	13 (7%)	1.035	-0.12	3.34	2.92	0.76	0.75
b	Konza ETM IR Konza IKONOS NIR	183	14 (7%)	1.07	-0.20	11.94	14.98	0.63	0.75
c	USDA IKONOS IR USDA IKONOS NIR	184	13 (7%)	0.99	0.98	-1.45	0.03	0.67	0.77

Table 4.19: Improved registration results after preprocessing enhancement

Chapter 5

5. Conclusions

5.1 Summary of Thesis

In this thesis we presented a novel framework for registration of remotely sensed images based on the SIFT algorithm. Specifically, we used the complete data available from the SIFT key-points location vector, i.e., scale, orientation, and position as opposed to other available algorithms which tend to use only the scale and orientation. We exploited the above information in a mode-seeking fashion where we first searched for modes in the scale ratio and orientation difference histograms defined over initial correspondences obtained by the nearest neighbors between SIFT descriptors. Next, we used these values to compute the modes in the horizontal and vertical location difference histograms. The overall quadruple obtained served as an initial guess for the transformation parameters, assuming that a more suitable transformation lies in its vicinity. We constructed an outlier filter with respect to this initial guess, which was found very reliable. This is a novel approach for filtering correspondence outliers, as opposed to other algorithms which typically use the distance ratio between the first and second nearest neighbor (of SIFT descriptors) to achieve this. We applied a one-step ordinary least squares algorithm to the remaining inliers to refine the transformation parameters values.

We implemented the above algorithm in MATLAB and C and tested it on a variety of datasets including multi-temporal, multi-view, multi-sensor, and multi-band image pairs. The registration quality was measured in terms of the

RMSE value (using the criterion of $RMSE \leq 1$ pixel). Our results show over 80% success. When compared to other existing algorithms, the most prominent advantage of our algorithm is its run-time which is faster by an order of a magnitude, at least. Another important issue is that of verification, i.e., computing an RMSE value with respect to manually picked ground truth correspondences. Concerning automatic verification, we used the number of inliers after filtering to determine whether or not the transformation obtained is reliable; of course, this threshold is a trade-off between detecting true failures and classifying successful registrations as failures.

Concerning registration failures, our investigation against ground truth transformations showed that in all cases there was a very small number of true SIFT correspondences in order to find a proper transformation. The above implies that we have been taking full advantage of the data supplied by the SIFT descriptor. We also showed that by some basic image processing techniques used to enhance the images, we can improve our results; this comes mainly to enhance the SIFT algorithm outputs, in the sense that it makes its key-points and descriptors more distinctive and thus more reliable.

5.2 Future Work

We point to several issues concerning future work:

1. The usage of a mode-seeking approach can be problematic when there are multi-modes in the SIFT characteristics histograms. According to our research and relevant publications, the multi-mode problem usually occurs only in the orientation difference histogram. A simple solution to this problem would be

to use a search algorithm on all possible modes and to take the one resulting with the largest amount of inliers in the filter; of course, this approach will result in a slower algorithm.

2. Along the way we assumed a similarity transformation between the reference and sense image pair; in some applications this is not necessarily the case, and a more complicated transformation might be required. It will be a challenge to devise an algorithm, based on our MS-SIFT paradigm, to compute a transformation for such cases.

3. As noted, in some cases the SIFT descriptor does not yield a sufficient number of true correspondences, which results in registration failures. Since SIFT was first presented, several competing descriptors have been developed (e.g., SURF [17], GLOH [18], etc.). Again, it would be of interest to derive a mode-seeking registration approach based on these descriptors rather than SIFT.

References

[1] B. Zitova, and J. Flusser, Image Registration Methods: A Survey, *Image and Vision Computing*, Vol. 21, No. 11, pp. 977-1000, 2003.

[2] M. V. Wyawahare, P. M. Patil, and H. K. Abhyankar, Image Registration Techniques: An Overview, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 2, No. 3, pp. 1-5, 2009.

[3] D. G. Lowe, Distinctive Image Features from Scale Invariant Keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[4] Q. Li, G. Wang, J. Liu, and S. Chen, Robust Scale-Invariant Feature Matching for Remote Sensing Image Registration, *IEEE Geoscience and Remote Sensing Letters*, Vol. 6, No. 2, pp. 287-291, 2009.

[5] M. Teke, M. F. Vural, A. Temizel, and Y. Yardimci, High Resolution Multispectral Satellite Image Matching Using Scale Invariant Feature Transform and Speeded Up Robust Features, *Journal of Applied Remote Sensing*, Vol. 5, No. 1, pp. 053553, 2011.

[6] Sedaghat, A., M. Mokhtarzade, and H. Ebadi, Uniform Robust Scale Invariant Feature Matching for Optical Remote Sensing Images, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 49, No. 11, pp. 4516-4527, 2011.

[7] Q. Li, H. Zhang, and T. Wang, Multispectral Image Matching Using Rotation-Invariant Distance, *IEEE Geoscience and Remote Sensing Letters*, Vol. 8, No. 3, pp. 406-408, 2011.

[8] M. Hasan, X. Jia, A. Robles-Kelly, J. Zhou, and M. R. Pickering, Multi-Spectral Remote Sensing Image Registration via Spatial Relationship Analysis on SIFT Keypoints, *in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* pp. 1011-1014, 2010.

[9] M. Fischler and R. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the Association for Computed Machinery*, Vol. 24, No. 6, pp. 381–395, 1981.

[10] M. Hasan, X. Jia, and M. R. Pickering, Modified SIFT for Multi-Modal Remote Sensing Image Registration, *in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* pp. 2348-2351, 2012.

[11] N. S. Netanyahu, J. Le Moigne, and J. G. Masek, Georegistration of Landsat Data via Robust Matching of Multi-resolution Features, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 42, No. 7, pp. 1586-1600, 2004.

[12] A. Karasarisidis and E. P. Simoncelli, A Filter Design Technique for Steerable Pyramid Image Transforms, *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2389-2392, 1996.

[13] P. J. Rousseeuw and A. M. Leroy, *Robust Registration and Outlier Detection*, Wiley, 1987, New York.

[14] L. G. Brown, A Survey of Image Registration Techniques, *Association for Computed Machinery Computing Surveys*, Vol. 24, No. 4, pp. 325-376, 1992.

[15] R. D. Eastman, N. S. Netanyahu, and J. Le Moigne, Survey of Image Registration Methods, *Image Registration for Remote Sensing*, J. Le Moigne, N. S. Netanyahu, and R. D. Eastman, Eds., Cambridge University Press, Cambridge, United Kingdom, pp. 35-76, 2011.

[16] R. C. Gonzalez, and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2008, New Jersey.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, Speeded Up Robust Features (SURF), *International Journal on Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359, 2008.

[18] K. Mikolajczyk, and C. Schmid, A Performance Evaluation of Local Descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615-1630, 2005.

[19] D. Mount, Personal Communication, 2013.

[20] A. Goshtasby, G. C. Stockman, and C. V. Page, A Region-Based Approach to Digital Image Registration with Sub-Pixel Accuracy, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 24, No. 3, pp. 390-399, 1986.

Appendix A

Derivation of the 1-Step Ordinary Least Squares [19]

This procedure computes either a rigid or similarity transformation that maps a set of model points to a set of corresponding object points. Let $M = \{m_i\}_{i=1}^{n}$ denote the model points (as 2-element column vectors) and $O = \{o_i\}_{i=1}^{n}$ denote the object points. Let $(a \cdot b)$ denote vector dot product, and let $|m_i, o_i| = (m_i[x]o_i[y] - m_i[y]o_i[x])$ be the determinant of the 2×2 matrix(m_i, o_i). We do this manner to minimize the sum of squared distances between corresponding pairs. Here is some justification. It is well known that, irrespective of rotation and scale, the translation that minimizes the sum of squared distances is the one that aligns the centroids of the two sets, so if we denote m_{ctr} and o_{ctr} as the centroids of the model and object points respectively then by definition:

$$m_{ctr} = \frac{1}{n}\sum_i m_i$$
$$o_{ctr} = \frac{1}{n}\sum_i o_i \quad \text{(A.1)}$$

According to the above, the optimal translation is given by:

$$t = o_{ctr} - m_{ctr} \quad \text{(A.2)}$$

Next we let m'_i and o'_i denote the image of m_i and o_i, respectively, under the translation that maps their respective centroids to the origin.

For a given correspondence and for any scaling factor, we claim that the rotation angle that minimizes the sum of squared errors is given by:

(A.3) $$r = \arctan(x_{mdt} / x_{\text{var}})$$

where x_{mdt} is the sum of determinants $\left|m'_i, o'_i\right|$ and x_{var} is the sum of dot products $\left(m'_i \cdot o'_i\right)$. To see this, let R denote the optimal rotation matrix. Then the objective is to minimize the sum of squared distances:

(A.4) $$\sum_i \left\| Rm'_i - o'_i \right\|^2 = \sum_i ((Rm'_i)^2 - 2(Rm'_i \cdot o'_i) + (o'_i)^2)$$

where $(a)^2 = (a \cdot a)$ denotes the squared length of a vector. Observe that rotation does not alter the length of a vector, and so $(Rm'_i)^2 = (m'_i)^2$, and hence the first at last of the summation do not appear on the rotation. Thus it suffices to maximize the negation of the middle term:

(A.5) $$\sum_i (Rm'_i \cdot o'_i)$$

Letting c and s denote the cosine and sine of the optimal rotation angle we have:

(A.6)
$$\begin{aligned} &\sum_i (Rm'_i \cdot o'_i) = \sum_i (cm'_i[x] - sm'_i[y])o'_i[x] + (sm'_i[x] + cm'_i[y])o'_i[y] \\ &= c\sum_i (m'_i[x]o'_i[x] + m'_i[y]o'_i[y]) + s\sum_i (m'_i[x]o'_i[y] - m'_i[y]o'_i[x]) \\ &= cx_{\text{var}} + sx_{mdt} \end{aligned}$$

By taking the derivative with respect to the rotation angle r and setting to zero, we obtain $s/c = x_{mdt} / x_{\text{var}}$, as desired. Finally, we compute the scale factor as:

(A.7) $$sc = \frac{\sum_i (Rm'_i \cdot o'_i)}{\sum_i (o'_i \cdot o'_i)}$$

We should note that this derivation is analogous to those given in [20] without the need to inverse the least squares matrix.

הרישום כ"טובות" אם ערך זה קטן מפיקסל בודד. מימשנו את האלגוריתם לעיל בשפות MATLAB ו-C, ובדקנו את ביצועיו על עשרות זוגות של תמונות חישה מרחוק. התוצאות הניסיוניות מראות על למעלה מ-80% הצלחה (על סמך הקריטריון לעיל). בנוסף, פיתחנו דרך לקביעה האם תהליך הרישום הצליח או לא על סמך מספר "התאמות האמת" שהתקבלו ע"י האלגוריתם. עבור אותם מקרים בהם האלגוריתם נכשל, הראינו כיצד בעזרת שימוש בטכניקות עיבוד תמונה בסיסיות, ניתן לשנות את התמונות כך שהאלגוריתם יצליח בכל זאת לבצע רישום איכותי שלהן.

תקציר

רישום תמונות הינו תהליך שבו זוג תמונות מיושרות על מערכת צירים משותפת. פעולה זו מהווה כלי חשוב במגוון יישומים, ביניהם חישה מרחוק, הדמאה רפואית, אבטחת איכות בתהליכי ייצור ועוד. התמונות המיועדות לרישום יכולות להיות שונות בזמן רכישתן, בנקודת המבט, בחיישן בו נעשה שימוש, בתדר ההדמאה ועוד; לכן בעיית הרישום הינה בעלת ענין רב למחקר ופיתוח. שיטות רישום רבות עושות שימוש ב"נקודות מפתח" בתמונת הייחוס ובתמונת החישה. לכן המשימה המרכזית בתהליך הרישום היא לקבוע התאמות בין נקודות מפתח בתמונות אלה על סמך מדד דמיון המוגדר מראש. לאחר מציאת ההתאמות הדרושות יש לקבוע אילו התאמות הינן נכונות (התאמות אמת) ואלו התאמות אינן נכונות (התאמות שווא). בהנחת מודל להתמרה של תמונת החישה למערכת הצירים של תמונת הייחוס, יש למצוא את ערכי הפרמטרים של ההתמרה על סמך ההתאמות הנכונות. איכות ההתמרה משוערכת במובן כלשהו הנקבע מראש. אם איכות ההתמרה אינה מספקת, יש צורך בתהליך איטרטיבי לשיפור ההתמרה עד להתכנסות או מיצוי ניסיונות השיפור. רוב שיטות הרישום כוללות חיפוש מייגע למציאת ההתמרה הדרושה. לעומת זאת, אנו מציעים שיטה חדשה לרישום תמונות המתבססת על נקודות מפתח המושגות על-ידי שיטת ה-Scale-Invariant Feature Transform (SIFT). ההבדל העיקרי בין השיטה שפיתחנו לשיטות אחרות מבוססות SIFT הוא הדרך לקביעת אמיתות ההתאמות אשר מסתמכת על אופן (mode) של פילוגים שונים. בפרט, אנחנו מחשבים את אופני ההתפלגויות של יחסי קני-מידה, הפרשי האוריינטציות והמיקום (אופקי ואנכי). ארבעת האופנים המתקבלים מהווים ניחוש מבטיח לערכי ההתמרה הדרושה של תמונת החישה ביחס לתמונת הייחוס; בהנחה כי ההתמרה המבוקשת נמצאת בקרבת הניחוש הראשוני וכי רוב ההתאמות באזור זה (קרי, בתיבה ארבע-ממדית שמרכזה בניחוש הראשוני) הן "התאמות אמת" (בניגוד להתאמות שמחוץ לתיבה הנחשבות "התאמות שווא"), ניתן להפעיל את שיטת "הריבועים הפחותים" בכדי למצוא את ההתמרה הדרושה. איכות ההתמרה תיבדק על-ידי נקודות "אמת מוחלטת" אשר יבחרו ידנית בתמונת הייחוס ובתמונת החישה ובעזרתן יחושב ערך RMSE להערכת איכות ההתמרה. נתייחס לתוצאות

תודות

ראשית, ברצוני להודות להוריי היקרים, שרה ויחזקאל שהביאוני עד הלום. ברצוני להודות גם למנחה האקדמי שלי, פרופ' נתן נתניהו, אשר חשף בפני את הבעיה המרתקת של רישום תמונות והדריך אותי לאורך המחקר. כמו-כן תודתי נתונה למנחה הנוסף שלי, פרופ' אילן שמשוני, עבור תובנותיו והערותיו המועילות.

תודה לכולכם!

בני

עבודה זו נעשתה בהדרכתו של פרופ' נתן נתניהו מהמחלקה למדעי המחשב באוניברסיטת בר-אילן ובסיועו של מנחה נוסף, פרופ' אילן שמשוני מהמחלקה למערכות מידע באוניברסיטת חיפה.

אוניברסיטת בר-אילן

רישום מבוסס SIFT של תמונות חישה מרחוק

בני קופפר

עבודה זו מוגשת כחלק מהדרישות לשם קבלת תואר מוסמך במדעים במחלקה

למתמטיקה באוניברסיטת בר-אילן

רמת-גן תשע"ד

אוניברסיטת בר-אילן

רישום מבוסס SIFT של תמונות חישה מרחוק

בני קופפר

עבודה זו מוגשת כחלק מהדרישות לשם קבלת תואר מוסמך במדעים במחלקה

למתמטיקה באוניברסיטת בר-אילן

רמת-גן תשע"ד