Line Segment-based Aerial Image Registration

Youngwook Paul Kwon

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-121
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-121.html

May 20, 2014

Line Segment-based Aerial Image Registration

by

Youngwook Paul Kwon

A thesis submitted in partial satisfaction of the
requirements for the degree of
Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sara McMains, Chair
Professor Alexei A. Efros, Co-chair

Spring 2014

Abstract

Line Segment-based Aerial Image Registration

by

Youngwook Paul Kwon

Master of Science in Computer Science

University of California, Berkeley

Professor Sara McMains, Chair

Professor Alexei A. Efros, Co-chair

We propose a new segment-based registration system for aerial images of the same scene taken at different times, from different view points, and/or by different sensors. We introduce a quantitative characterization of the *registration difficulty* for a given pair of images. Targeting high registration difficulty input, we exploit on linear edges in images. In the first step of our registration process, we detect line segments in each image. Next we conduct a merging step on the detected line segments. Finally, using the merged line segments as input, we generate possible hypothesis transformations by choosing three segments in each image. Our collinearity score metric for the transformations balances considerations of angular and perpendicular distances. After scoring each hypothesis transformation, the highest-scoring one is selected. For high registration difficulty image pairs, our algorithm shows significant improvement compared to publicly accessible image registration codes.

Contents

List of Figures

Acknowledgments

I would like to express the deepest appreciation to my research advisor Sara McMains, who has always shown warm and kind attitude and a strong spirit of scholarship. It was great to see genuine smile and scholarly integrity at the same time. Without her supervision, this thesis would not have been possible. I would like to thank Alexei A. Efros and Jitendra Malik, who demonstrated excitement in regard to teaching and passion in regard to research. It was exciting to see two amicable and respectable professors in the research field that I am interested in.

I also want to thank to Lawrence Livermore National Laboratory for their internship opportunity, research dataset, and financial support. In addition, great thank you to Sheila Vaidya, who is an insightful supervisor and guided me into this practical research field. My mentor, Goran Kojevod suggested various research topics and ideas in friendly conversations, and it was a pleasure to work with someone so nice. A sincere thanks to Hyojin Kim who introduced me to the computer vision world and advised me with deep knowledge and clear explanations.

Chapter 1

Introduction

Image registration is the process of estimating a transformation to overlay and align two images of the same scene taken at different times, from different viewpoints, and/or by different sensors. As Zitová and Flusser [2003] argue, image registration is a crucial step in diverse fields such as weather forecasting, integrating information into geographic information systems (GIS), medicine, cartography, computer vision, etc.

The general automatic registration process consists of four steps: *feature detection, feature description, feature matching, and transformation estimation.* The first two steps are to extract a certain type of feature from each image (feature detection), and to describe each feature in a quantitative manner (feature description). Then, using the detected features (control points, CPs) and their descriptions, the correspondence matching (control point pairs, CPPs) information is extracted by comparing how similar the feature descriptions are (feature matching). Therefore, a good feature description will retain the same values after a registration transformation is applied. Finally, transformation can be calculated from the correspondence matching information. Since the matching information (CPPs) usually includes outliers, many methodologies use additional information such as spatial constraints to filter those outliers before calculating the transformation. In other words, the performance of feature-based algorithms is determined by designing a good local descriptor and distinguishing between true matches (inliers) and false matches (outliers). Several techniques will be discussed in Chapter 2.

Aerial image registration has unique characteristics that make it difficult. First, due to the limited range of sensors (cameras), the same regions are captured from arbitrarily different positions and angles of view. Second, two input images to register may be taken at different times. Some objects/buildings may appear or disappear over time (Fig 1.1). Images taken during day time and night time may have very different signals (color or gray intensity) due to light and shadows. Third, images taken with different types of sensors (i.e.,

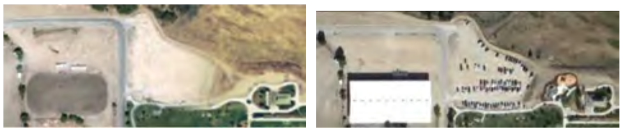
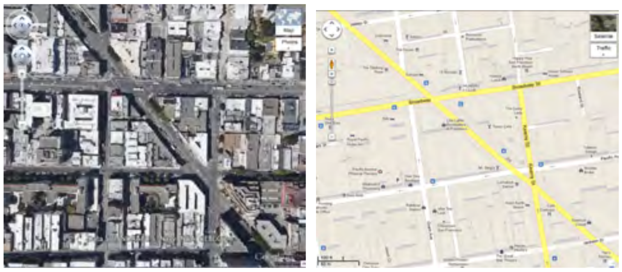


Fig. 1.1. Changes in images at different times (here, different years)



(a) Google Satellite (b) Google Map

Fig. 1.2. Example of different looking images (satellite and abstract drawing)

infra-red vs. visible color spectrum) also may have significant changes. In addition, one may want to register a sensed image of some terrain with a reference image from a different database. The more severe changes an input image pair has, the more poorly an automatic registration algorithm works. Moreover, since aerial images deal with large areas, it is hard to expect and define salient objects a priori. Some regions like the sea are too monotonous to find any special edges/corners/keypoints. Some grainy regions like forests or waves are likely to include too many random edges/corners/keypoints. In this case, one may fail to detect common features in the input images.

Nevertheless, humans are remarkably good at detecting correspondences and registering images that include partial changes. Humans can register an actual Google satellite image with an abstract and simplified Google map (Fig 1.2). This implies that humans do not only rely on local pixel-wise signal processing and matching, since the two images have different pixel-level information such as colors and oriented gradients. We propose that one of the important higher-level information that humans utilize is conspicuous boundaries and linear information such as the unique linear alignment of road networks and the boundary orientations of buildings.

Then, how can we express the *registration difficulty* of a certain pair of input images? Figure 1.3 illustrates a novel way to quantify and visualize the registration difficulty. We collect pairs of images ranging from ones that are almost the same or include only slight changes, to ones that include severe changes or monotonous regions and require humans' ability to register. These pairs are shown in Chapter 3 (for convenience, we name each input pair of images.) We acquire the CPPs between each image pair based on the well-known local descriptor "scale invariant feature transform" (SIFT), which many algorithms take as a building block due to its good performance (Chapter 2). The SIFT descriptor is a 128-tuple

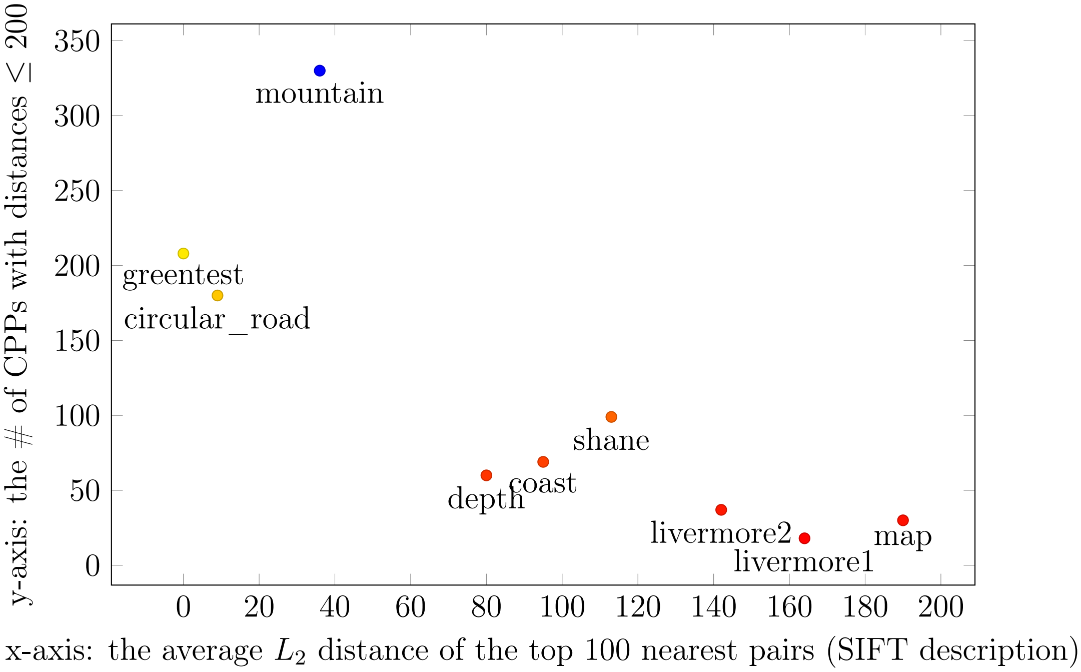


Fig. 1.3. Characteristics of input dataset

vector, and the detected features are matched basically based on the nearest descriptor using Euclidean L_2 distance as the metric. The x-axis in Fig 1.3 represents the average distance of the top 100 nearest pairs. An analogous question about the x-axis would be "how similar are the most similar CPPs?" The y-axis is the number of CPPs whose distances are less than 200. An analogous question for y-axis would be "how many CPPs are within a certain similarity threshold?" Thus, an image pair more in the lower right region means that SIFT fails to find similar CPPs (matches are highly likely to be outliers), and the number of useful CPPs is less. For the left upper datasets, SIFT returns a number of similar CPPs. Simply speaking, as an input image pair places more towards the lower right region of the graph, the registration difficulty increases.

As far as our scope of experience, we found that many algorithms work well only within a very limited range of input data and fail to correctly register the "difficult" images. Much previous research reports results only for registration performance experiments where evaluation is only on input image sets where a counterpart image is artificially made by adding some filters such as noise and blurring, and/or applying an affine transformation to an original image. These kinds of image pairs are highly likely to place in the upper left region.

In this report, we propose a registration system that covers aerial images of diverse difficulty levels, and estimates an affine transformation. To emulate humans' approach of using higher-level information, we focus on line segments (or their extended full lines) of the image. Lines are robust to occlusions and small changes, and man-made regions such as towns and airports have many feature lines.

After discussing related work (Chapter 2) and presenting our target dataset (Chapter 3), we introduce our aerial image registration algorithm in Chapter 4 and then show the results in Chapter 5.

Chapter 2

Related work

2.1 Feature detection and matching

Zitová and Flusser broadly divide feature matching techniques into area-based and feature-based approaches [2003]. Area-based methods focus more on feature matching than feature detection (Zitová and Flusser [2003]). Area-based methods usually do not attempt to detect salient objects in images, but try to find correspondences by comparing fixed certain windows (windows of predefined size or even entire images) in the reference image with sliding windows in the other image. We can compare the two windows using correlation-like methods, Fourier methods, etc. (Zitová and Flusser [2003]). Area-based methods have several limitations and are especially vulnerable to changes in images. For example, since even the shape of a window (usually a rectangle) in the reference image can be distorted by a transformation, then it may be naive to expect to find a good correspondence with the window of the original shape in the sensed image. Loveless [2013] states that the following conditions are required for successful registration: the images from the same or similar sensors, the same or similar viewpoints, similar lighting conditions, and enough overlapping area.

Feature-based approaches put more emphasis on the feature detection stage than feature matching, and make an effort to detect distinctive features, called control points (CPs), that can provide useful clues about correspondences. Feature descriptors and similarity measures between the descriptors are used for the feature matching stage. Many descriptors have been proposed. Rahtu et al. [2005] propose an area-based affine invariant descriptor using multiscale autoconvolution (MSA). A comprehensive review on the comparison of affine region detectors may be found in Mikolajczyk et al. [2005]. Mikolajczyk and Schmid [2005] compare local descriptor performances for test cases generated using affine transformations, scale changes, rotation, blur, jpeg compression, and illumination changes. They have found that the scale invariant feature transform (SIFT), proposed by Lowe [2004], has the best results for most of the tests, with lower performance for textured scenes or when edges are not reliable.

Due to the good performance of SIFT, many image registration algorithms take SIFT as their feature descriptor. To filter incorrect matches, they refine the feature description and propose scale–orientation joint restriction criteria for estimating a similarity transformation. Ma et al. [2010] obtain CPs not only from SIFT but also from normalized cross correlation, which is an area-based method. They conduct outlier screening procedures for the two types of CPs individually. Liu et al. [2012] propose an illumination and affine invariant descriptor by focusing on the pros and cons of SIFT and MSA. Li et al. [2009] experimented and showed the limitations of SIFT on multidate, multispectral, and multisensor remote images.

Other papers focus on the lines in images. Line matching is more difficult than point matching [Dubrofsky and Woodham, 2008]. First, the end points of detected line segments are not reliable. Second, the orientations of feature lines are not invariant under perspective transformation. Chou and Tsai [1993] propose an iterative scheme to match line segments for the purpose of stereo vision matching. Schmid and Zisserman [1997] propose a line matching method using cross-correlation-based matching score. Similar to Chou and Tsai, they use epipolar geometry of the views under the situation that the camera positions are known. Coiras et al. [2000] propose a line-segment-based image registration technique for multisource images by defining a similarity function for the two sets of line segments. Habib and Alruzouq [2004] use line segments as features and propose a matching strategy using a modified iterated Hough transform. For the purpose of stereo matching, Bay et al. [2005] detect line segments using Canny edge detection, identify initial candidate matches using color information, and filter outlier matches using topological information. Dubrofsky and Woodham [2008] describe a methodology to utilize point correspondences and line correspondences together for homography (Section 2.2) estimation.

2.2 Transform model estimation

From the control point pairs information, a mapping function between the sensed image and reference image is determined. There are various types of mapping functions such as similarity transforms, affine transforms, projective transforms, and transforms with higher degrees of freedom such as polynomial models.

Similarity transforms include translation and rotation, uniform scaling, and any compositions of them. Using homogeneous coordinates, a similarity transform can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.1}$$

where (x', y') is a CP in the reference image, (x, y) is its corresponding CP in the sensed image, θ is the rotation angle, (t_x, t_y) is the translation vector, and s is the scale factor. A non-rigid transformation that preserves distances after mapping can be regarded as a special case of the similarity transform with scale of 1. Under similarity transforms, shapes of objects are preserved.

Affine transformations are a superset of similarity transforms that also include scaling with different aspect ratios, reflection, skew, and compositions of them. Sets of parallel lines remain parallel after an affine transformation. Using homogeneous coordinates, an affine transform can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.2}$$

$$= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.3}$$

where s_x and s_y are the scale factors in the x and y axes respectively, and k is the skew factor. Because this model has six degrees of freedom, at least three non-collinear CP pairs are required to solve the system.

Projective transformations (called homography) can represent any mapping in projective space. Using homogeneous coordinates, a projective transformation can be written as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{2.4}$$

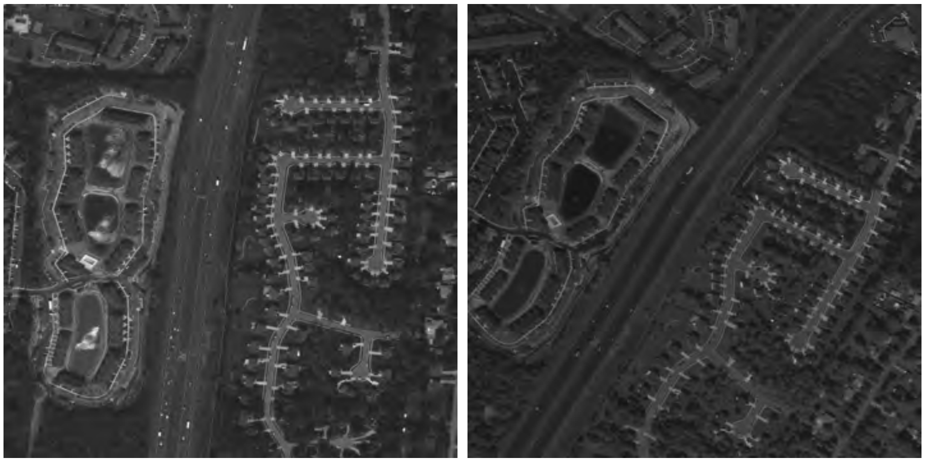
Note that, because the projective transformation matrix is a homogeneous matrix, it has eight degrees of freedom. At least four linearly independent CP pairs are required to solve the system.

While these *global mappings* such as similarity, affine, and projective transformation can cover a large range of geometric transformations with a few parameters in a simple and global manner, sometimes we need *local mappings* when, for example, objects in the scene are flexible and have some distortion, as in medical imaging application, and they can not be registered globally in nature. These types of transformation are beyond the scope of our current work. Refer to Zitová and Flusser [2003] for more details.

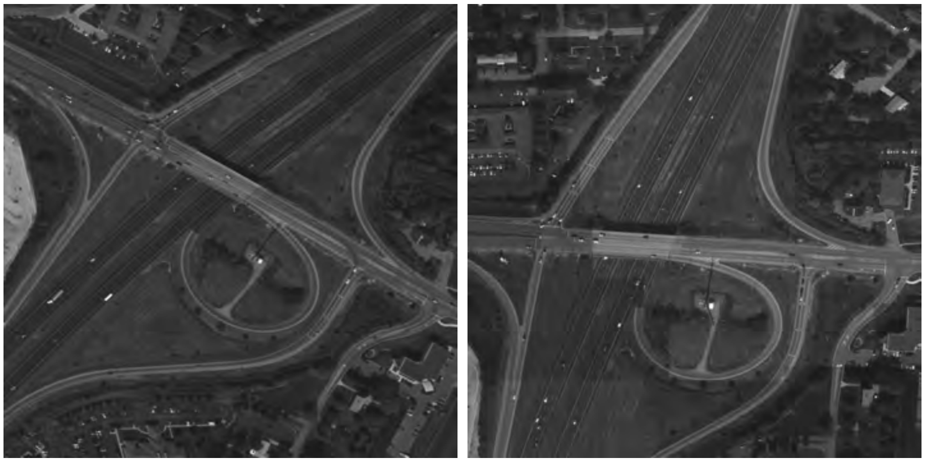
In aerial image applications, because the sensors are generally very far from objects, the input images can be regarded as flat planes. Since affine transformations are sufficient to represent transformations between general aerial images, they are commonly assumed, and we also assume affine transformation for our aerial image registration problem.

Chapter 3

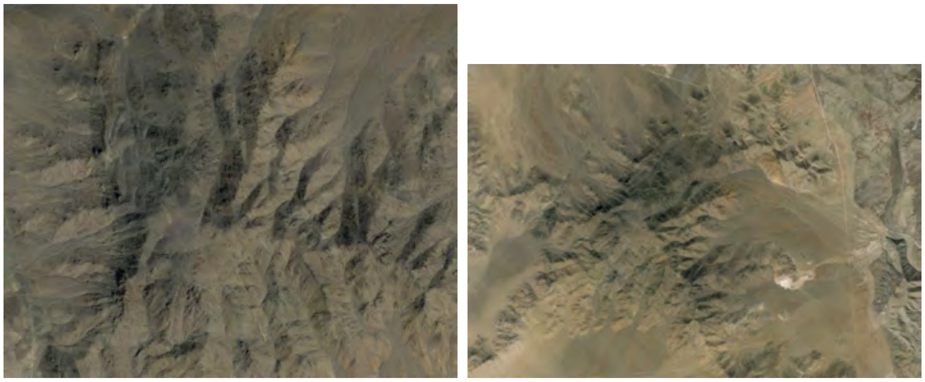
Dataset



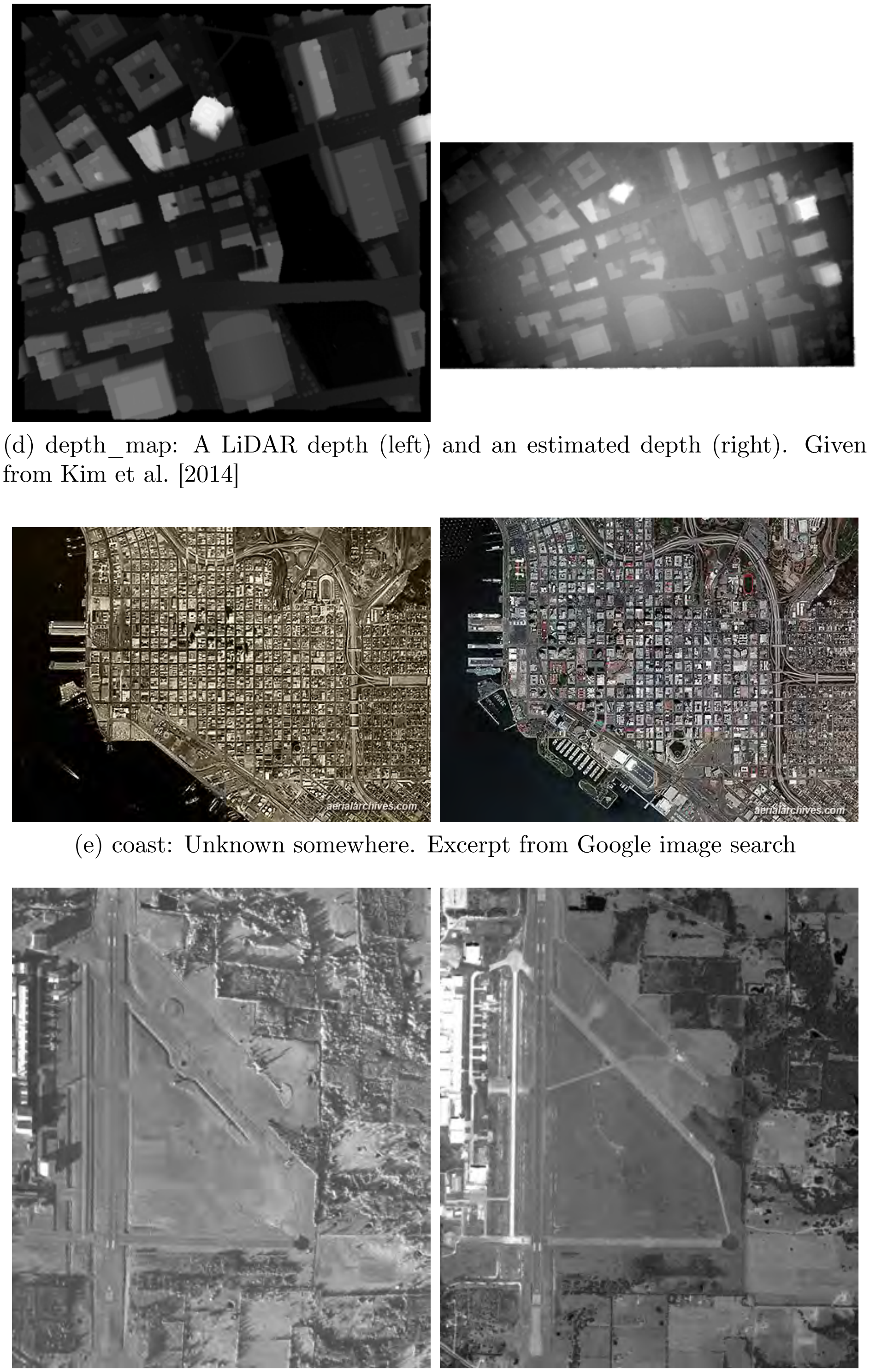
(a) greentest: Different frames from an aerial video. Given from LLNL.



(b) circular_road: Different frames from an aerial video. Given from LLNL.



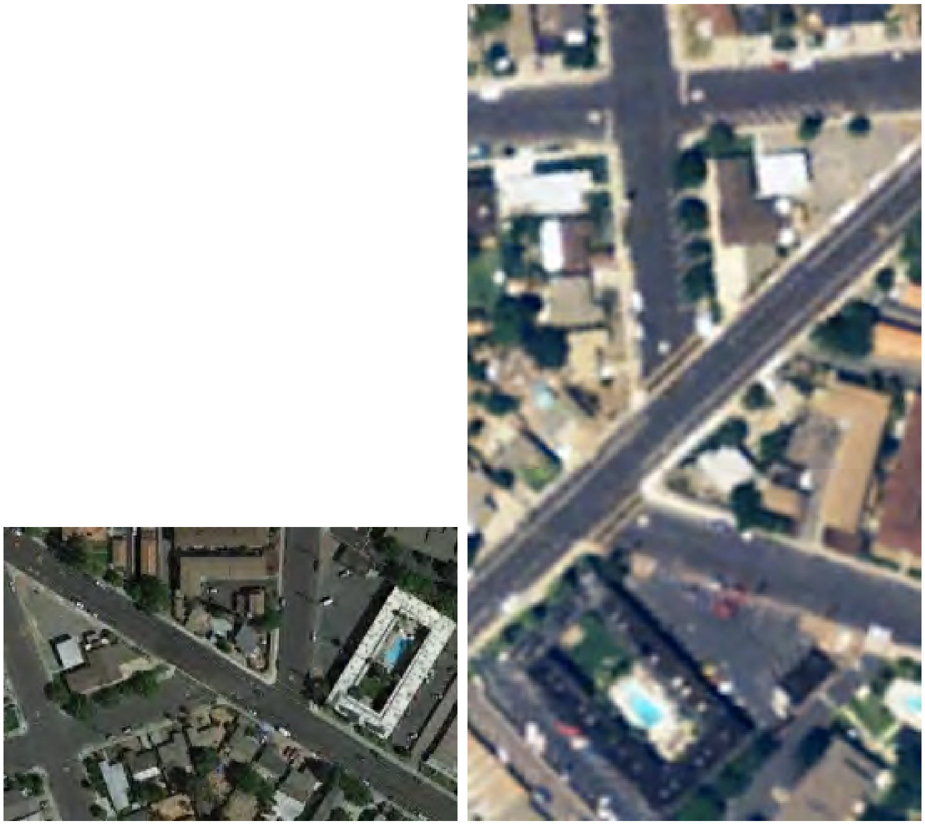
(c) mountain: Google Satellite (left) and Google Earth (right)



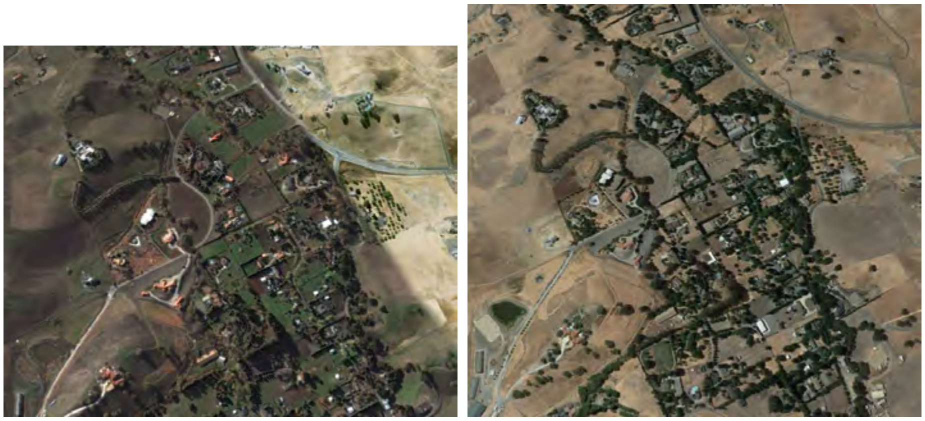
(d) depth_map: A LiDAR depth (left) and an estimated depth (right). Given from Kim et al. [2014]

(e) coast: Unknown somewhere. Excerpt from Google image search

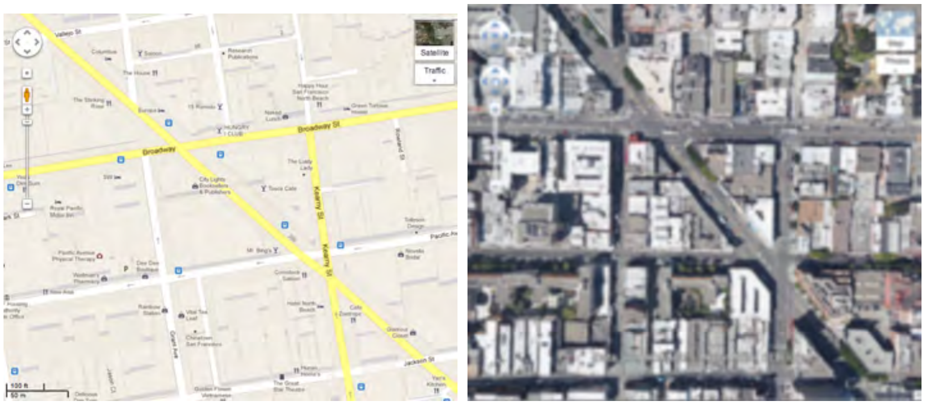
(f) shane: Photo (left) and infra-red image (right). Given from Lawrence Livermore National Laboratory (LLNL).



(g) livermore2: A random place in Livermore in different years. Excerpt from Google Earth.



(h) livermore1: Somewhere in Livermore, CA, 2004 and 2012. Excerpt from Google map.



(i) map: Google map (left) and Google Satellite (right) of a random place in San Francisco

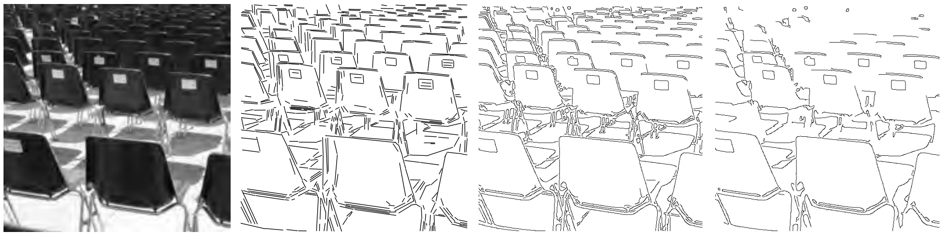
Fig. 3.-2. Our input dataset. We order the input image pairs in registration difficulty order in Fig 1.3, from the easiest to highest.

Chapter 4

Algorithm

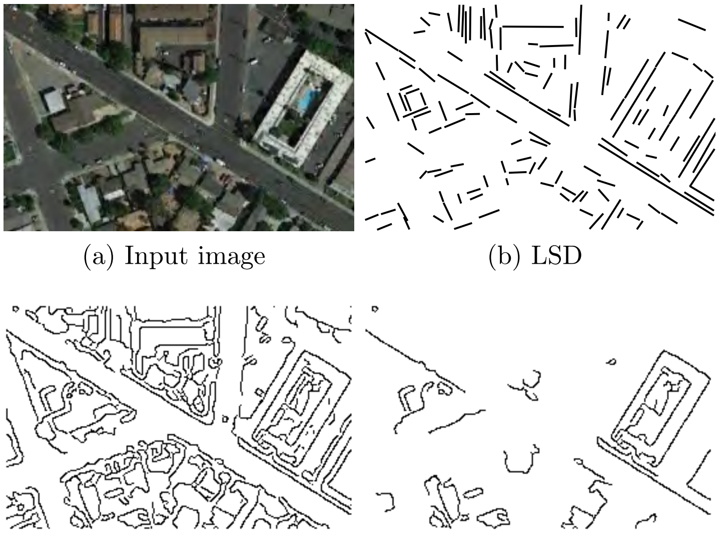
4.1 Line Segment Detection

To detect line segments in images, we choose von Gioi et al.'s Line Segment Detector (LSD) algorithm [2012], while others use Hough transform [Dubrofsky and Woodham, 2008; Habib and Alruzouq, 2004] or Canny edge extractor [Coiras et al., 2000; Schmid and Zisserman, 1997]. Hough transform extracts not line segments but full lines in images, from which the line segments would need to be extracted. The Canny edge extractor requires a threshold parameter δ [Canny, 1986], and the detected edges are not guaranteed to be linear. The LSD algorithm detects line segments and runs in linear time without parameter tuning. Figures 4.1 and 4.2 show the results of the LSD compared to the Canny edge extractor.



(a) Input image (b) LSD (c) Canny, $\delta = 0.2$ (d) Canny, $\delta = 0.5$

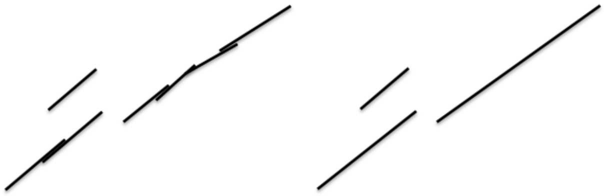
Fig. 4.1. Line segment detector and Canny edge extractor (example 1).



(a) Input image (b) LSD

(c) Canny, $\delta = 0.2$ (d) Canny, $\delta = 0.5$

Fig. 4.2. Line segment detector and Canny edge extractor (example 2).



(a) Before merging (b) After merging

Fig. 4.3. Line segment merging step.

4.2 Line Segment Merging

The line segments detected using LSD are incomplete in the sense that a single salient and useful line segment may be detected as a few broken and/or overlapped segments. To deal with this incompleteness, we add a merging step. Fig 4.3 illustrates the results of our procedure. We merge two line segments if they are (i) collinear (within a threshold) and (ii) overlapped. We describe as follows how to check the collinearity and overlap for two given line segments s and s'.

First, we define a distance function $D_{\theta d}(s, s')$ over two line segments s and s'. This definition is based on that of Coiras et al.'s work [2000], but modified from a non-symmetric distance to a symmetric distance. As in Fig 4.4, for two given line segments s and s', we can calculate the angular difference $\Delta\theta$, and the perpendicular distance Δd, where $\Delta\theta$ is the acute angle that s and s' (or their extended lines) make, and Δd is the maximum of two

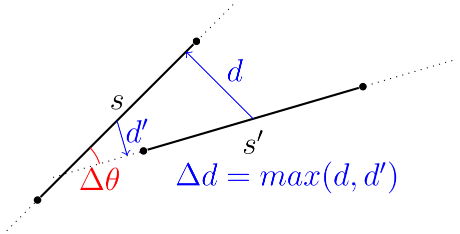


Fig. 4.4. The angular distance $\Delta\theta$ and the perpendicular distance Δd

perpendicular distances: that from the midpoint of s to the line containing s' (d' in Fig 4.4), and the that from the midpoint of s' to the line containing s (d in Fig 4.4). Then we define the distance function $D_{\theta d}(s, s')$ between the two line segments s and s' as:

$$D_{\theta d}(s, s') = \frac{1}{\sqrt{2}}\sqrt{(\frac{\Delta\theta}{\theta_\delta})^2 + (\frac{\Delta d}{d_\delta})^2}, \tag{4.1}$$

where θ_δ and d_δ are thresholds for angular difference and perpendicular distance, respectively. We set $\theta_\delta = 5°, d_\delta = 5$. For example, if s and s' are collinear, then $\Delta\theta = 0$, and $\Delta d = 0$, and therefore, $D_{\theta d}(s, s') = 0$ (and vice versa). As $\Delta\theta$ or Δd increases, $D_{\theta d}(s, s')$ increases. If s and s' differ by $\Delta\theta = \theta_\delta$, and $\Delta d = d_\delta$, then $D_{\theta d}(s, s') = 1$.

Then we define a collinearity score $Col(s, s')$, which represents how collinear two given line segments s and s' are, as below:

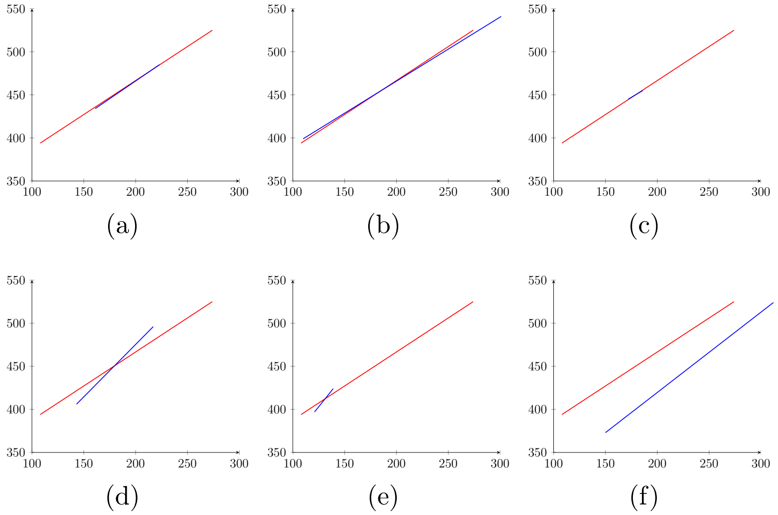
$$Col(s, s') = \begin{cases} 1 - D_{\theta d}(s, s'), & \text{if } D_{\theta d}(s, s') \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, if the distance between s and s' is zero (or, s and s' are collinear), the collinearity score is equal to 1. If the distance is greater than 1, the collinearity score is zero. Fig 4.5 shows example scores of different line segment pairs. We regard s and s' as collinear for the merging step if $Col(s, s')$ is non-zero. The strictness of the collinearity test can be adjusted by the thresholds θ_δ and d_δ.

The merging step is done as follows. Let S be the entire set of detected segments from an image. For a given line segment s, let $S_{col}(s)$ be the set of all segments collinear with s:

$$S_{col}(s) = \{s_i \mid Col(s, s_i) \neq 0, \forall s_i \in S\}. \tag{4.2}$$

To check the overlap condition, we project all segments in S_{col} onto a virtual parallel line as illustrated in Fig 4.6. The two endpoints of each segment $s_i \in S_{col}$ can be represented as two parametric real numbers, $left(s_i)$ and $right(s_i)$ ($left(s_i) < right(s_i)$). The smaller parametric number is considered to be "lexicographically left." We examine overlaps using the parametric numbers. When the left point of a segment s_j starts before the right point of a collinear segment s_i, or $left(s_j) \leq right(s_i)$, we merge s_i and s_j. The two end points of the new merged line are $\min(left(s_i), left(s_j))$ and $\max(right(s_i), right(s_j))$ (the lexicographically leftmost and rightmost points). Fig 4.7 illustrates the result of the merging step.



(a) (b) (c)

(d) (e) (f)

	$Col(s, s')$	$D_{\theta d}(s, s')$	Δd	$\Delta\theta$
(a)	0.885	0.115	0.635	1.2°
(b)	0.854	0.146	0.740	1.6°
(c)	0.800	0.200	0.756	2.7°
(d)	0.124	0.876	3.098	12.3°
(e)	0	3.515	23.575	18°
(f)	0	5.003	35.320	4.7°

Fig. 4.5. Examples of the scores $Col(s, s')$ for six sample input pairs

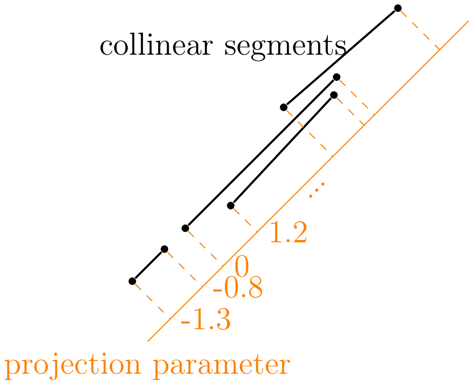
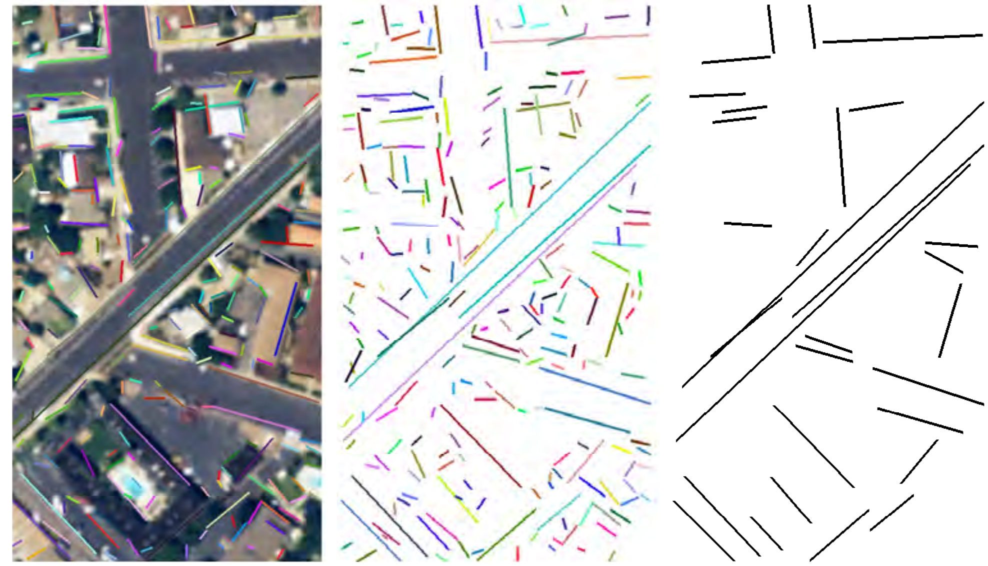


Fig. 4.6. Projection to 1D parametric space.



(a) Line segments with image (b) Segments only (c) Segments after merging

Fig. 4.7. The (a) and (b) show the detected lines using LSD. The coloring is just for the visual help to distinguish individual line segment. The (c) shows the 30 longest segments after the merging step.

4.3 Transformation

We follow Hartley and Zisserman [2003] to solve the affine transformation between two images. Because an affine transformation has six degrees of freedom, one can solve an affine transformation from three control point pairs (CPPs) as follows. Given that a CPP of a point $\mathbf{p}$ in image 1 and $\mathbf{p}'$ in image 2 ($\mathbf{p}$ and $\mathbf{p}'$ are represented as 3×1 vectors in homogeneous coordinates), the 3×3 affine transformation $T : \mathbf{p} \to \mathbf{p}'$ satisfies:

$$\mathbf{p}' = T\mathbf{p}. \tag{4.3}$$

Then the affine transformation T that maps the points $\{\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}\}$ to $\{\mathbf{p'_1}, \mathbf{p'_2}, \mathbf{p'_3}\}$ respectively is uniquely solved by:

$$P' = TP \tag{4.4}$$

or equivalently

$$T = P'P^{-1}. \tag{4.5}$$

where

$$P := \left[\mathbf{p_1} \mid \mathbf{p_2} \mid \mathbf{p_3}\right] \tag{4.6}$$

$$P' := \left[\mathbf{p'_1} \mid \mathbf{p'_2} \mid \mathbf{p'_3}\right]. \tag{4.7}$$

Note that P, P' and T are all 3 × 3 matrices. In order for T to exist, P must be full rank, or in other words, $\{\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}\}$ should not be collinear.

We check all the possible transformations that can be acquired from the detected line segments information, instead of estimating the transformation based on feature description, because we assume the two input images may include severe changes, in which case there is no appropriate point-level descriptor.

Let S and S' be the sets of detected line segments from image 1 and image 2, respectively. We choose any three non-parallel segments $\{s_i, s_j, s_k\} \in S$ and $\{s'_l, s'_m, s'_n\} \in S'$ from each set. Both sets of three non-parallel segments (or their extended full lines) yield three intersection points, which we order counter-clock wise and denote $(\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3})$ and $(\mathbf{p'_1}, \mathbf{p'_2}, \mathbf{p'_3})$, respectively. In this setting, note that we do not know which point corresponds to which; there are three possible cases: the three points $(\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3})$ may correspond to $(\mathbf{p'_1}, \mathbf{p'_2}, \mathbf{p'_3})$, $(\mathbf{p'_2}, \mathbf{p'_3}, \mathbf{p'_1})$ or $(\mathbf{p'_3}, \mathbf{p'_1}, \mathbf{p'_2})$. Because we exclude the reflection transformations (aerial images are always collected above the ground), the counter-clock-wise order itself should be preserved.

Denote a case C: $(\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3})_C$ and $(\mathbf{p'_1}, \mathbf{p'_2}, \mathbf{p'_3})_C$. Hypothesizing that these are true CPPs, we solve for the affine transformation T_C using equation 4.5. We repeat this process, testing every combination of three segments chosen from S and S'. We will discuss how to evaluate a matching score of $score(T_C)$ in section 4.4. The optimal transformation T^* is the one with the best matching score:

$$T^* = \arg\max_C score(T_C). \tag{4.8}$$

This process requires a huge number of calculations because the number of possible combinations is so large. In order to reduce computation, we limit the search to the most promising candidates by selecting the triples of segments only from the longest 30 line segments each in S and S'. Since the evaluation process is independent, parallel computation can be easily applied.

4.4 Matching Score

In this section, we define the matching score function $score(T_C)$ for a given hypothesis transformation T_C. Let S_{T_C} be the transformation of S by T_C. The idea is that if T_C is a true transformation, S_{T_C} and S' will be in harmony. In other words, many common segments in S_{T_C} and S' will be collinear.

For a segment $s \in S_{T_C}$, define a function $best(s, S')$ as:

$$best(s, S') = \max_{s' \in S'}(Col(s, s')). \tag{4.9}$$

This function searches for the segment $s' \in S'$ that is most collinear with s. Then we define a matching score function:

$$score_{S \to S'}(T_C) = \frac{1}{|S_{T_C}|} \sum_{s \in S_{T_C}} (best(s, S')), \tag{4.10}$$

where $|S_{T_C}|$ is the cardinality of S_{T_C}.

Note that S can be transformed into the coordinate system of S' by T_C, and S' can be transformed into the coordinate system of S by ${T_C}^{-1}$. Since the mapping score is not symmetric, we calculate the mapping score for the latter direction as well.

$$score_{S' \to S}({T_C}^{-1}) = \frac{1}{|S'_{{T_C}^{-1}}|} \sum_{s' \in S'_{{T_C}^{-1}}} (best(s', S)) \tag{4.11}$$

The final mapping score is their average:

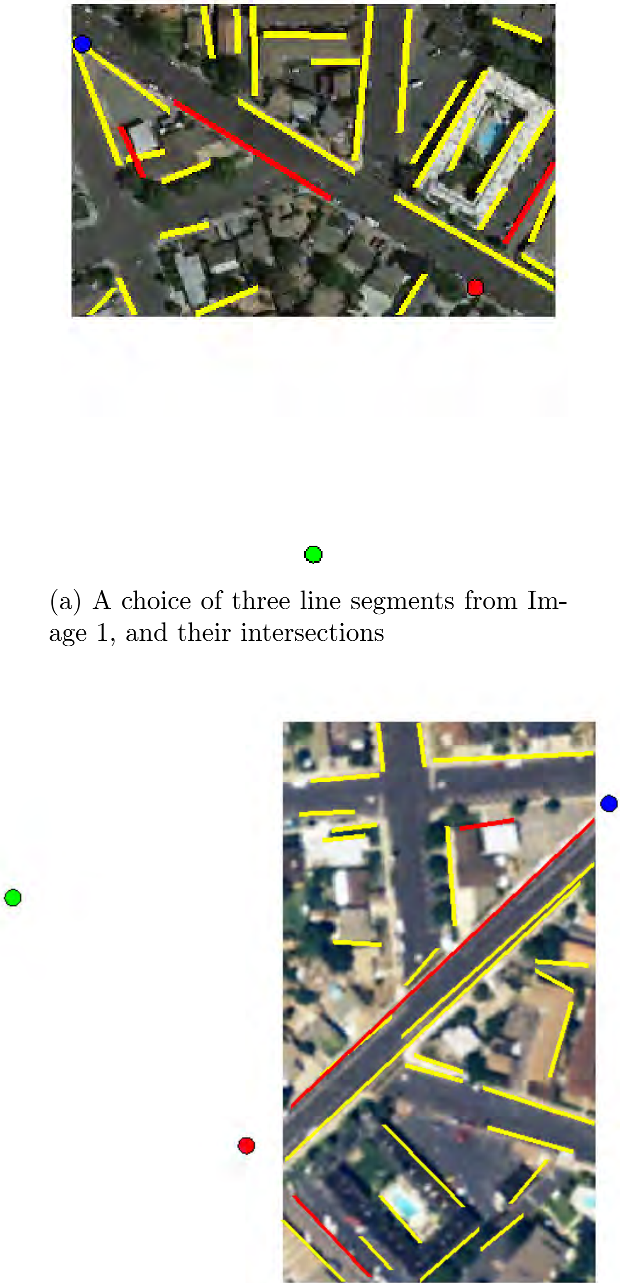
$$score(T_C) = \frac{1}{2} score_{S \to S'}(T_C) + \frac{1}{2} score_{S' \to S}({T_C}^{-1}). \tag{4.12}$$

Fig 4.8 visualizes the choice of the three line segments (per image) that yields the transformation with the best matching score. It also shows the three intersections in each image, for which there are three possible ways to pair them. The color of these points (red, green, and blue) represents the pairing relationship. Fig 4.9 shows the registration result and how the S and S' are finally matched. In Fig 4.10, we compare our result with Fedorov et al.'s work [2003a] (online demo: Fedorov et al. [2003b]), and image registration using SIFT (online open source: **?**), since these are the only algorithms that we can access.

4.5 Filtering

For the purpose of reducing unnecessary computation, before measuring a score of the hypothesis transformation T_C, we examine T_C by decomposing it into the six elements of the affine transformation: t_x (translation in x), t_y (translation in y), s_x (scale in x), s_y (scale in y), θ (rotation angle), k (shear). The "true" transformation should have these elements in certain limited ranges. For example, a t_x or t_y so large that it means that the two input images are not even overlapped is not appropriate. If T_C has a very large s_x or s_y, it is highly likely to be a wrong answer. The rules we use are as follows, considering T_C to be valid only when:

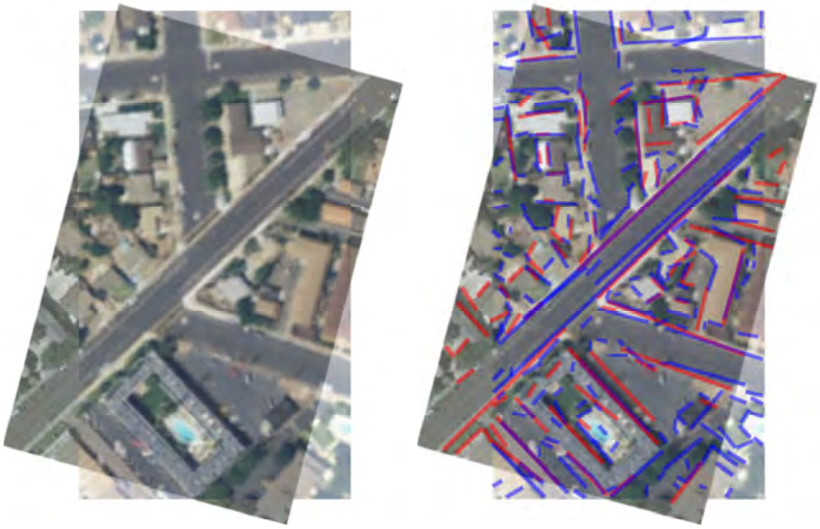
- translation: $|t_x|, |t_y| <$ min(image height, image width);
- scale: $1/3 < s_x, s_y < 3$;



(a) A choice of three line segments from Image 1, and their intersections

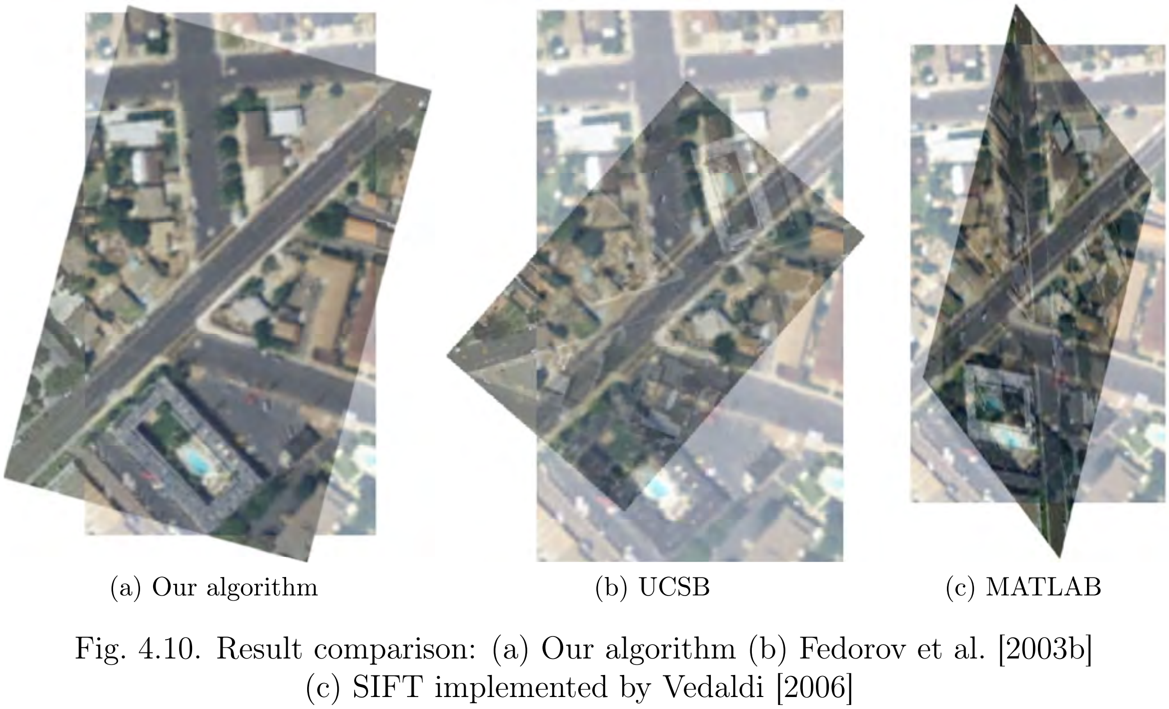
(b) A choice of three line segment choice from Image 2, and their intersections

Fig. 4.8. The three line segments highlighted in red in the two images is the choice that yields the optimal transformation T^*. The three circles in each image indicate the intersection points of the three chosen segments. The circle color represents correspondence between the points.



(a) Registered image (b) Registered line segments

Fig. 4.9. (a) shows the registration result using the optimal transformation T^*. (b) visualizes how S and S' are finally matched.



(a) Our algorithm (b) UCSB (c) MATLAB

Fig. 4.10. Result comparison: (a) Our algorithm (b) Fedorov et al. [2003b] (c) SIFT implemented by Vedaldi [2006]

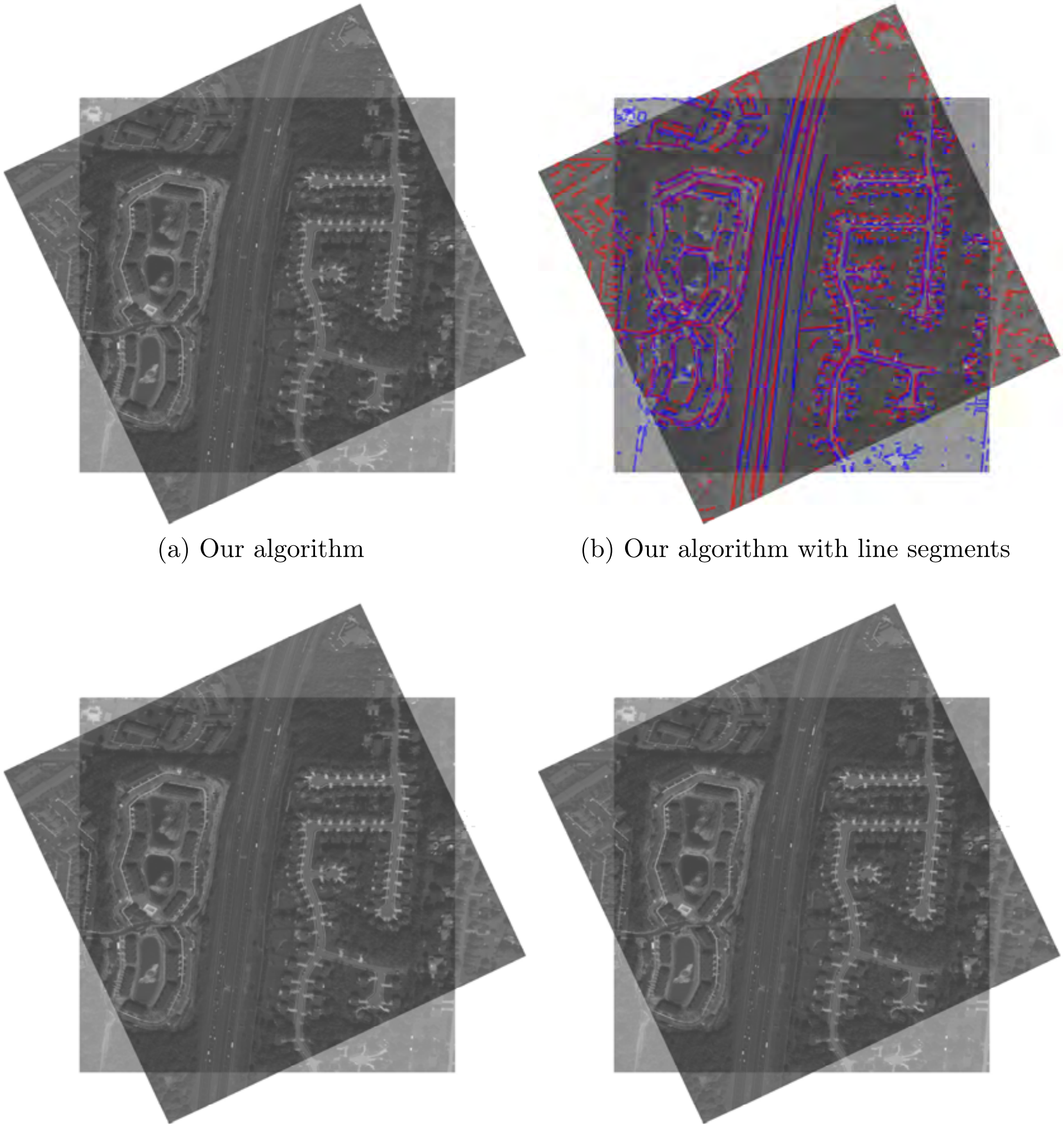
- skew: $|k| < 0.2$; and
- rotation: no constraint.

If T_C violates one of these conditions, we do not need to calculate its matching score. We discard the choice and keep repeating the process with another choice. If we have stricter rules, we can save more computations by discarding more frequently.

Chapter 5

Results

In this chapter, we show our results compared to Fedorov et al. [2003b] and Vedaldi [2006]. We order the input image pairs in registration difficulty order, from the easiest to highest. As the registration difficulty increased, other algorithms fail easily. We show that our registration algorithm is outstanding especially for the input images in high registration difficulty.



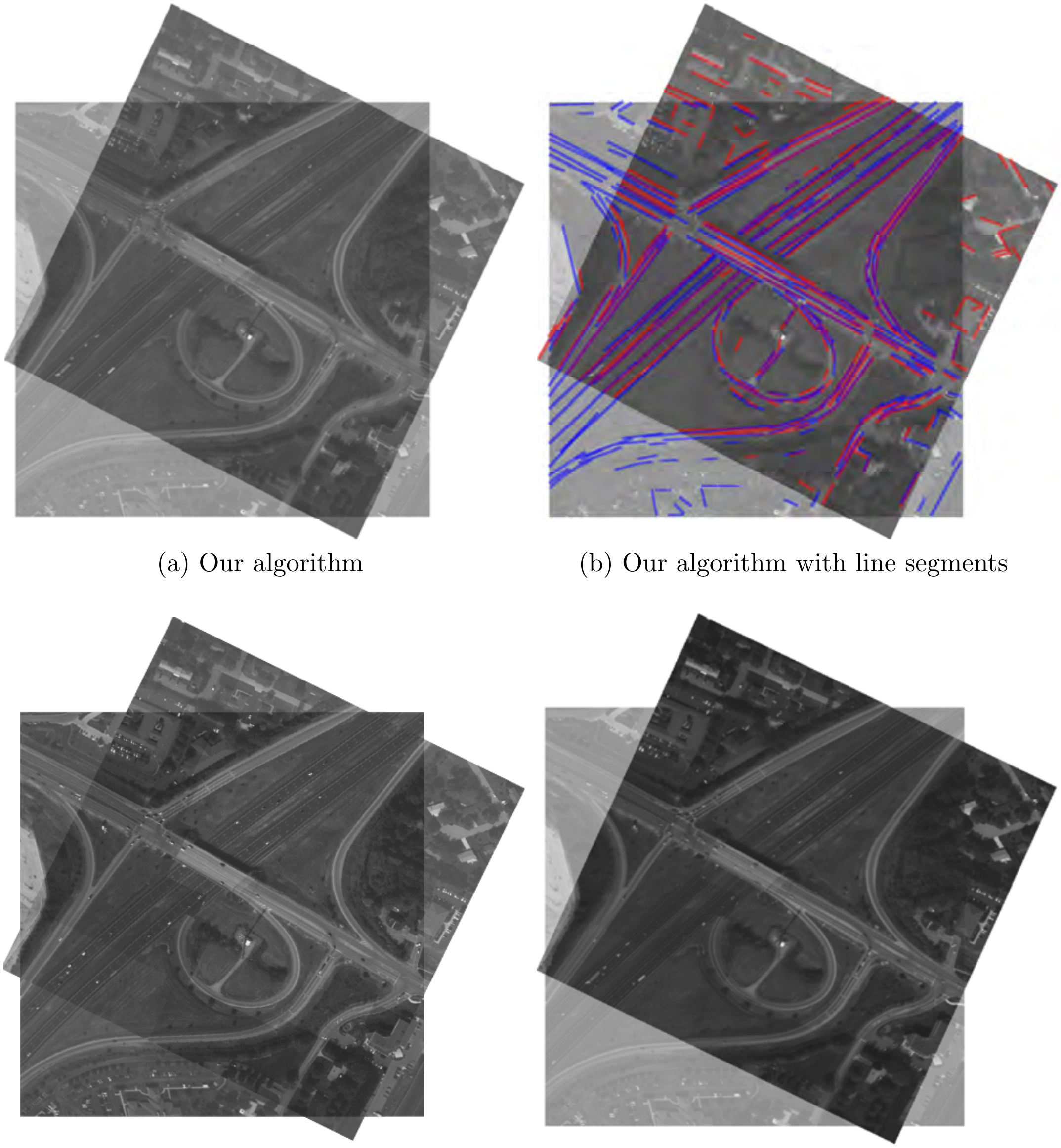
(a) Our algorithm

(b) Our algorithm with line segments

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.1. Result: greentest



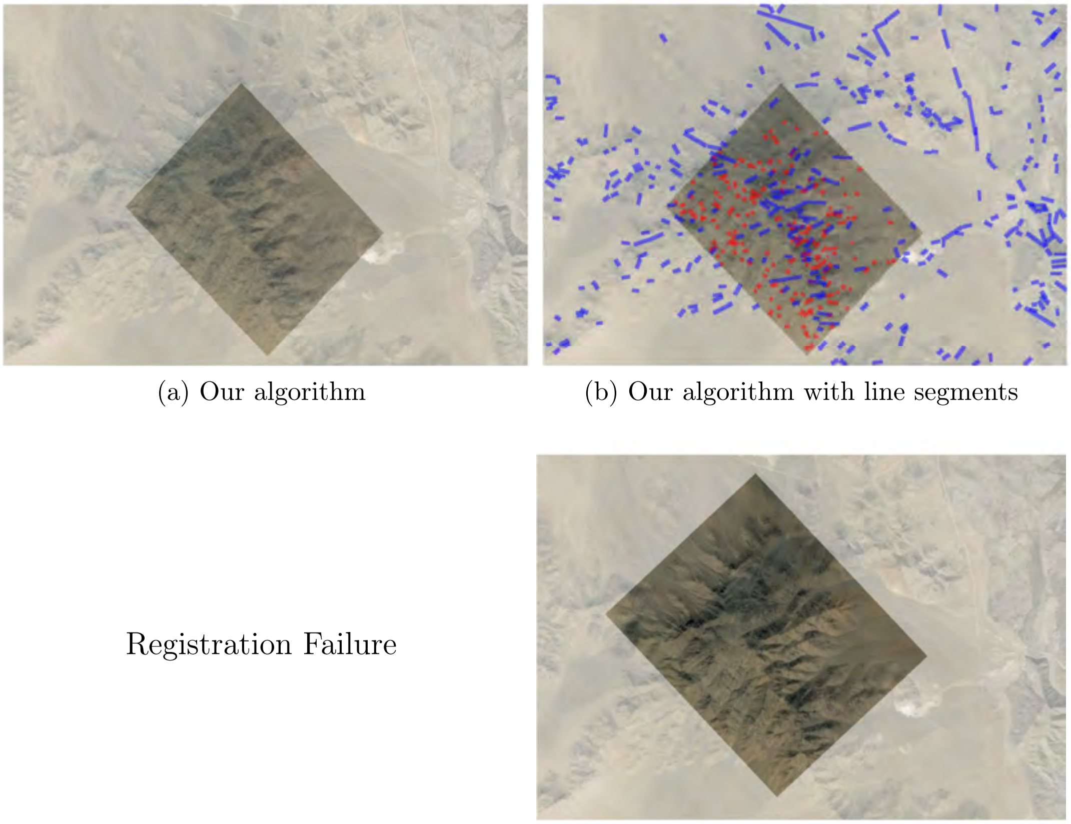
(a) Our algorithm

(b) Our algorithm with line segments

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.2. Result: circular_road



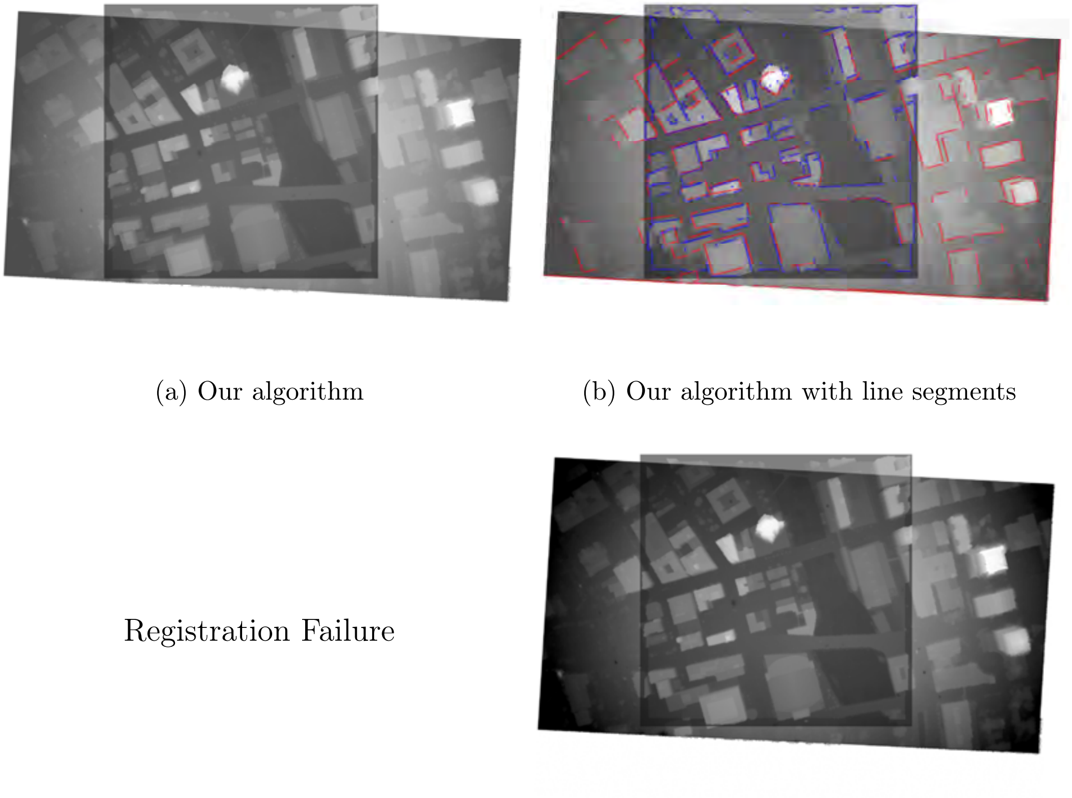
(a) Our algorithm

(b) Our algorithm with line segments

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.3. Result: mountain



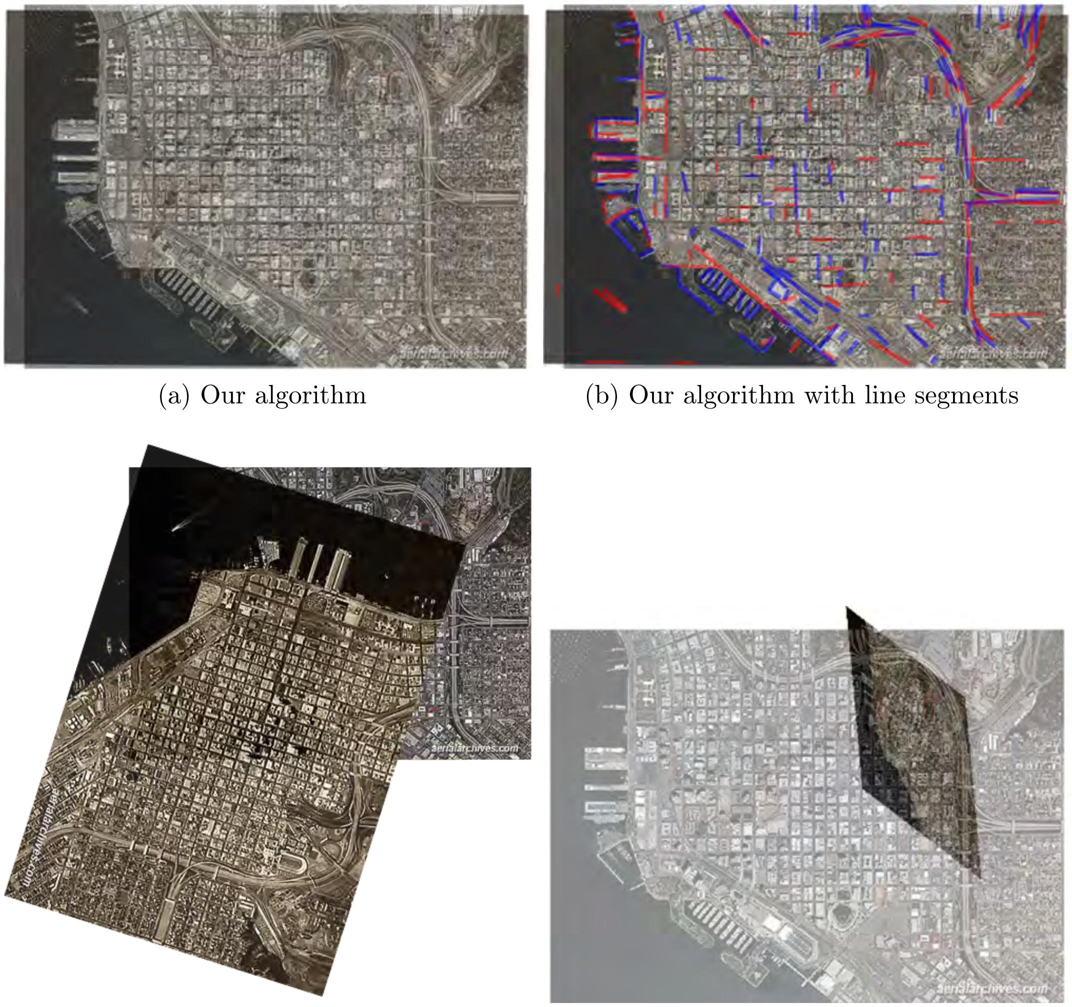
(a) Our algorithm

(b) Our algorithm with line segments

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.4. Result: depth



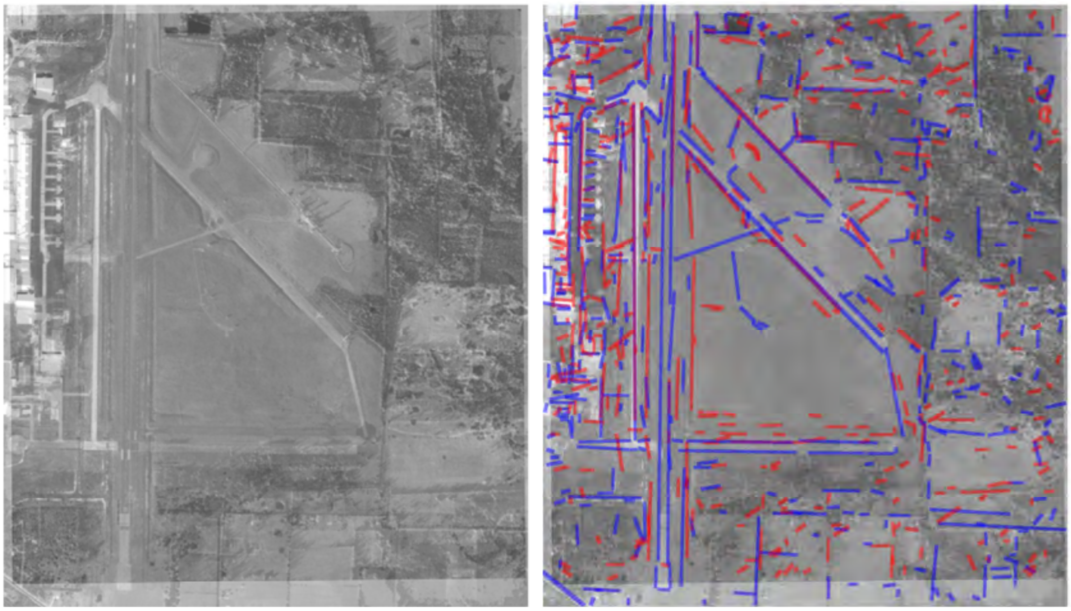
(a) Our algorithm

(b) Our algorithm with line segments

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.5. Result: coast



(a) Our algorithm

(b) Our algorithm with line segments

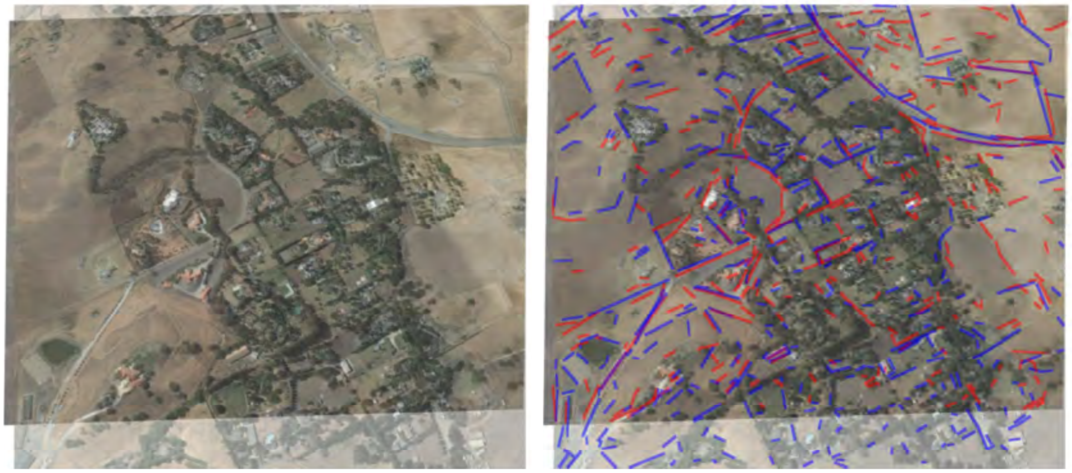
Registration Failure

Registration Failure

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.6. Result: shane



(a) Our algorithm

(b) Our algorithm with line segments

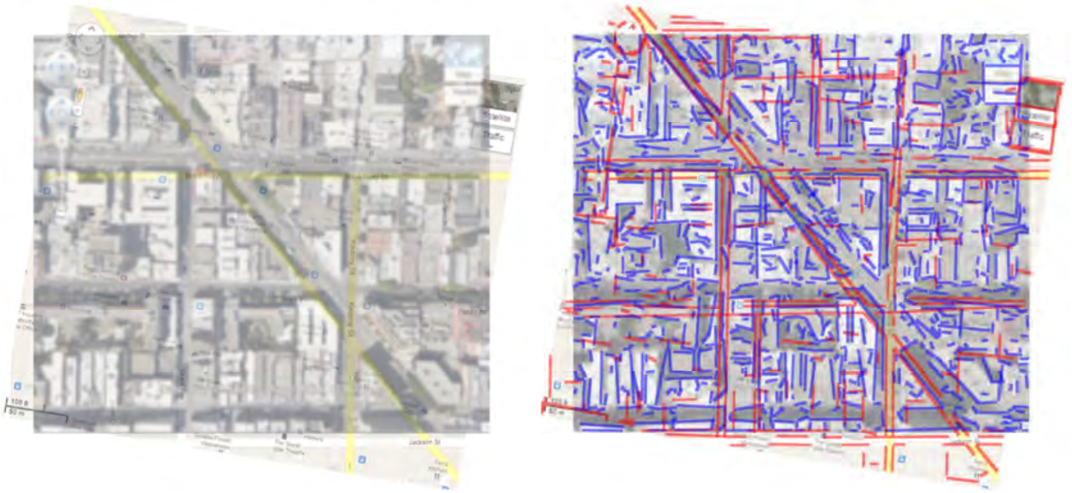
Registration Failure

Registration Failure

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.7. Result: livermore1



(a) Our algorithm

(b) Our algorithm with line segments

Registration Failure

Registration Failure

(c) Fedorov et al. [2003b]

(d) SIFT by Vedaldi [2006]

Fig. 5.8. Result: map

Chapter 6

Conclusion

We propose a new segment-based registration system for multimodal aerial images with a wide range of registration difficulties. We first visually show the registration difficulty in a quantitative manner. In the first step of our registration process, we detect line segments in each image using the LSD algorithm. Next we conduct the merging step on the detected line segments. Finally, using the merged line segments as input, we generate possible hypothesis affine transformations by choosing three segments in each image. After scoring each hypothesis transformation based on the score metric, the highest-scoring one is selected.

This algorithm can easily be extended to projective transformations or higher degrees of transformations by choosing four or more segments instead of three, and then solving a hypothesis transformation based on the choice. This is exactly the same procedure, however it will have a larger searching space and more intense computation.

Obviously, one limitation is computation time. As future work, we plan to reduce the search space. In addition, we consider taking this registration algorithm as an initial transformation estimate, and then performing additional micro-tuning afterwards to get an even more accurate final transformation result.

Bibliography

Bay, H., Ferrari, V., and Van Gool, L. (2005). Wide-baseline stereo matching with line segments. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, pages 329–336.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698.

Chou, S.-L. and Tsai, W.-H. (1993). Line segment matching for 3D computer vision using a new iteration scheme. *Machine Vision and Applications*, 6(4):191–205.

Coiras, E., Santamarı, J., and Miravet, C. (2000). Segment-based registration technique for visual-infrared images. *Optical Engineering*, 39(1):282–289.

Dubrofsky, E. and Woodham, R. J. (2008). Combining Line and Point Correspondences for Homography Estimation. In *Advances in Visual Computing*, pages 202–213. Springer Berlin Heidelberg, Berlin, Heidelberg.

Fedorov, D. V., Fonseca, L. M., Kenney, C., and Manjunath, B. S. (2003a). Automatic registration and mosaicking system for remotely sensed imagery. pages 444–451.

Fedorov, D. V., Kenney, C., Manjunath, B. S., and Fonseca, L. M. (2003b). Image Registration With Fit Assessment Demo. http://vision.ece.ucsb.edu/registration/imreg/. Last accessed on May 14, 2014.

Habib, A. F. and Alruzouq, R. I. (2004). Line-based modified iterated Hough transform for automatic registration of multi-source imagery. *The Photogrammetric Record*, 19(105):5–21.

Hartley, R. and Zisserman, A. (2003). Multiple View Geometry in Computer Vision. Cambridge.

Kim, H., Correa, C., and Max, N. (2014). Automatic Registration of LiDAR and Optical Imagery using Depth Map Stereo. *Computational Photography, IEEE International Conference on*, pages 205–212.

Li, Q., Wang, G., Liu, J., and Chen, S. (2009). Robust Scale-Invariant Feature Matching for Remote Sensing Image Registration. *IEEE Geoscience and Remote Sensing Letters*, 6(2):287–291.

Liu, Z., Wang, Y., Jing, Y., Zhao, J., and Wang, J. (2012). An illumination and affine invariant descriptor for aerial image registration. In Shimura, T., Xu, G., Tao, L., and Zheng, J., editors, *Photonics Asia*, page 85580Z. SPIE.

Loveless, B. E. (2013). Automated near real-time image registration. Master's thesis, San Diego State University.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints - Springer. *International journal of computer vision*.

Ma, J., Chan, J. C., and Canters, F. (2010). Fully Automatic Subpixel Image Registration of Multiangle CHRIS/Proba Data. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(7):2829–2839.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72.

Rahtu, E., Salo, M., and Heikkila, J. (2005). Affine invariant pattern recognition using multiscale autoconvolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):908–918.

Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 666–671.

Vedaldi, A. (2006). MATLAB/C implementation of the SIFT detector and descriptor. http://www.robots.ox.ac.uk/ vedaldi/code/sift.html. Last accessed on May 14, 2014.

von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2012). LSD: a Line Segment Detector. *Image Processing On Line*, 2012.

Zitová, B. and Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000.