IMAGE REGISTRATION USING A NEW EDGE–BASED APPROACH

Jun–Wei Hsieh‡, Hong–Yuan Mark Liao† [1],
Kuo–Chin Fan‡, Ming–Tat Ko†, and Yi–Ping Hung†
†Institute of Information Science, Academia Sinica, Taiwan
‡Institute of Computer Science and Electronic Engineering
National Central University, Chung–Li, Taiwan
email: liao@iis.sinica.edu.tw
Tel: +886(2)788–3799 ext. 1811, Fax: +886(2)782–4814

Keywords: Image registration, Wavelet transform.

Running head: Edge–Based Image Registration

Corresponding author's address —

Hong–Yuan Mark Liao
Associate Research Fellow
Institute of Information Science, Academia Sinica
Nankang, Taipei, Taiwan
TEL: 886–2–788–3799 ext. 2413
FAX: 886–2–782–4814
E–mail: liao@iis.sinica.edu.tw

---

[1] To whom all correspondence should be addressed.

[2] This work was supported by Institute of Information Industry and National Science Council of Taiwan under great no. NSC84–2213–E–001–008.

Abstract

A new edge–based approach for efficient image registration is proposed. The proposed approach applies wavelet transform to extract a number of feature points as the basis for registration. Each selected feature point is an edge point whose edge response is the maximum within a neighborhood. By using a line–fitting model, all the edge directions of the feature points are estimated from the edge outputs of a transformed image. In order to estimate the orientation difference between two partially overlapping images, a so–called ''angle histogram'' is calculated. From the angle histogram, the rotation angle which can be used to compensate for the difference between two target images can be decided by seeking the angle that corresponds to the maximum peak in the histogram. Based on the rotation angle, an initial matching can be performed. During the real matching process, we check each candidate pair in advance to see if it can possibly become a correct matching pair. Due to this checking, many unnecessary calculations involving cross–correlations can be screened in advance. Therefore, the search time for obtaining correct matching pairs is reduced significantly. Finally, based on the set of correctly matched feature point pairs, the transformation between two partially overlapping images can be decided. The proposed method can tolerate roughly about 10% scaling variation and does not restrict the position and orientation of images. Further, since all the selected feature points are edge points, the restriction can significantly reduce the search space and, meanwhile, speed up the matching process. Compared with conventional algorithms, the proposed scheme is a great improvement in efficiency as well as reliability for the image registration problem.

List of Symbols

Symbol	Explanation
f_1 and f_2	the input images.
$W_{2^j}^1 f$ and $W_{2^j}^2 f$	the 2–D wavelet transforms for an image *f(x,y).*
$M_{2^j} f$	the modulus of $W_{2^j}^1 f$ and $W_{2^j}^2 f$.
R_n	the edge correlation.
N_p	a neighborhood used to determine a feature point *P.*
C_{f_1,f_2}	the cross–correlation metric.
$\overline{C}_{f_1,f_2}$	the modified cross–correlation metric.
μ_i	the local mean of an image f_i.
σ_i	the local variance of an image f_i.
Ω_e	a neighborhood of a feature point *P* used to estimate its orientation.
FP_{f_1} and FP_{f_2}	two sets of feature points extracted from f_1 and f_2.
N_{f_1}	the number of elements in FP_{f_1}.
N_{f_2}	the number of elements in FP_{f_2}.
MP	a set of possible matching pairs.
N_m	the number of elements in MP.
M_p^c	the set of all correct matching pairs.
N_c	the number of elements in M_p^c.
$A(u)$	the orientation of an edge (or feature) point u.
$\theta_{i,j}$	the orientation difference between two feature points p_i and q_j.
$\overline{\theta}$	the orientation difference between f_1 and f_2.
$\hat{f}_1$	the image f_1 after rotating with the angle $\overline{\theta}$.
$\hat{\theta}$	the orientation difference between $\hat{f}_1$ and f_2.
$H(\theta)$	the angle histogram used to estimate the angle $\overline{\theta}$.
s	a scaling parameter.
T	a translation vector.
R	a rotation matrix.
$\hat{s}$, $\hat{T}$, and $\hat{R}$	the corrections for s, T, and R, respectively.

I. Introduction

Image registration is an important technique for a great variety of applications such as aerial image analysis [1], [2], [3], stereo vision [4], [5], automated cartography [6], motion analysis[7], [8], and the recovery of the 3–D characteristics of a scene [9]. There are two tasks which need to be handled during an image registration process. They are feature selection and correspondence establishment. Typically, feature points can be selected by manual or automatic methods [1], [6], [10]. However, automatic selection of feature points is always preferable. As to the correspondence problem, algorithms for determining correspondences between feature points can be classified into two categories: feature–based and area–based methods. The former is used to extract common features such as curvatures, moments, areas, or line segments to perform accurate registration [12], [14], [15]. Since most of the proposed features do not depend on the gray–level characteristics, the feature–based method has been shown to be more suitable for the problems of multi–sensor image registration. For example, Li *et al.* [12] proposed a contour–based approach to register images from multiple sensors. The success of their method depends on the assumption that the common structures of images must be preserved well. Therefore, their method is efficient but works well only on cases where the contour information is well preserved. On the other hand, the area–based method usually adopts a window of points to determine a matched location using the correlation technique [1], [3] . The most commonly used measure is normalized cross–correlation. This method is more robust than the feature–based method in some situations. However, if the orientation difference between the two images is large, the value of cross–correlation will be greatly influenced and the correspondences between feature points, thus, hard to derive. Therefore, De Castro and Morandi [13] proposed an elegant method, called phase correlation, to overcome this problem. However, when the overlapping area between images is small, their method becomes unreliable. In order to solve the problem, it is necessary to develop a method to estimate the rotation parameter in advance. In [3], Zheng and Chellappa proposed a method for determining the rotation parameter. They used a Lambertian model to model an image. Under the assumption that the illumination source is stationary, they use a shape–from–shading technique to estimate the illuminant directions of images. By taking the difference between the illuminant directions, the rotation angle between images is obtained. After obtaining the rotation angle, one of the two images is then rotated such that the orientation difference between the two images becomes very small. By adopting the method proposed by Manjunath *et al.* [11], a number of feature points are extracted from the image pair. Then, these feature points are matched by using an area–based method in a hierarchical image structure. In Zheng–Chellappa's approach, the technique for estimating the rotation angle works well for most cases.

However, if a scene includes many buildings and objects, the method will fail due to the fact that the illumination conditions in one image may not be equivalent to those in the other. In general, the estimation of a rotation angle in their approach is rough. Further, their approach requires a Gabor function decomposition in the feature extraction process. This decomposition is computationally intensive. Another drawback of their approach is that when false matches emerge, their method can not handle them.

In this paper, we propose a new method to tackle the above mentioned problems. The domain of the images under consideration is aerial images; however the proposed method is also suitable for other types of images. The proposed method is based on the following assumptions. First, since the distance between the camera on an aircraft and the target objects on the ground is very far, it is reasonable to assume that the images are taken by cameras whose optical axes are parallel. Further, the variations of the intensity characteristics between images are assumed to be small. The proposed approach applies wavelet transform to extract a number of feature points as the basis for registration. Each selected feature point is an edge point whose edge response is the maximum within a neighborhood. By using a line–fitting model, all the edge directions of the edge points are estimated from the edge outputs of a transformed image. In order to estimate the orientation difference between the images, a so–called ''angle histogram'' is calculated . From the angle histogram, the rotation angle which can be used to compensate for the difference between the two target images can be decided by seeking the angle that corresponds to the maximum peak in the histogram. Based on the rotation angle, an initial matching can be performed. During the real matching process, we check each candidate pair in advance to see if it can possibly become a correct matching pair. Due to this checking, many unnecessary calculations involving cross–correlations can be screened in advance. Therefore, the search time for obtaining correct matching pairs is reduced significantly. Once all the correct matching pairs are found, they are then used to derive the correct registration parameters. In this work, we apply an iterative scheme to make the registration result more reliable. Since only three or fewer iterations are needed, and only a few feature points are involved in the matching process, the whole procedure can be accomplished very efficiently. Furthermore, the proposed method can tolerate approximately 10% scaling variation and does not have to restrict the position and orientation of images. Compared with conventional algorithms, the proposed scheme offers great improvement in efficiency as well as reliability for the image registration problem.

The rest of this paper is organized as follows. In the next section, we discuss how feature point extraction is performed using wavelet transforms. The procedure for finding correct matching pairs be-

tween two partially overlapping images is described in detail in Section III. Based on the correct matching pairs found by the method described in Section III, the procedure for deriving correct transformations between two target images is reported in Section IV. Section V summarizes the whole matching algorithm, and experimental results are reported in Section VI. Finally, a conclusion will be presented in Section VII.

II. Feature Point Extraction Using Wavelet Transforms

In this section, we shall describe in detail the process for applying wavelet transforms in the detection of feature points. Wavelet transform (WT) [27] for multiresolution local analysis on signals has been proved to be very effective. It has been successfully applied to many image analysis tasks such as edge detection [21], [33], corner detection [22], texture classification[23], object recognition [24], image segmentation [25], and shape recovery [26]. In this section, we shall introduce how this technique is applied to the problem of scene registration.

First of all, let *S(x,y)* be a 2–D smoothing function. Two wavelets, $\psi^1(x,y)$ and $\psi^2(x,y)$, are the partial derivatives of the smoothing function *S(x,y)* in the x and y directions, respectively, where

$$\psi^1(x,y) = \frac{\partial S(x,y)}{\partial x} \text{ and } \psi^2(x,y) = \frac{\partial S(x,y)}{\partial y}.$$

The above smoothing function *S(x,y)* and its corresponding wavelets are the same with the ones in Mallat's paper [21]. Let $\psi^1_{2^j}(x,y) = \frac{1}{4^j}\psi^1(\frac{x}{2^j},\frac{y}{2^j})$ and $\psi^2_{2^j}(x,y) = \frac{1}{4^j}\psi^2(\frac{x}{2^j},\frac{y}{2^j})$. At each scale, 2^j, the 2–D wavelet transform of a function $f(x,y)$ in $L^2(R^2)$ can be decomposed into two independent directions as follows:

$$W^1_{2^j}f(x,y) = f * \psi^1_{2^j}(x,y),$$

and

$$W^2_{2^j}f(x,y) = f * \psi^2_{2^j}(x,y).$$

Basically, these two components are equivalent to the gradients of *f(x,y)* smoothed by *S(x,y)* at scale 2^j in the x and y directions. At a specific scale $s = 2^j$, the modulus of the gradient vector of *f(x,y)* can be calculated [21]:

$$M_{2^j}f(x,y) = \sqrt{|W^1_{2^j}f(x,y)|^2 + |W^2_{2^j}f(x,y)|^2}. \quad (1)$$

If the local maxima of $M_{2^j}f(x,y)$ are located and thresholded with a preset value, then all the edge points of *f(x,y)* at scale 2^j can be detected. Since we are interested in some specific feature points for scene registration, additional constraints have to be introduced. In general, noise is the main cause of false detection of edge points. In order to suppress the effect of noise, a criterion called edge correlation is introduced [31]:

$$R_n(j,x,y) = \prod_{i=0}^{n-1} M_{2^{j+i}} f(x,y), \tag{2}$$

where n is a positive integer indicating the number of scales involved in the multiplication, and j represents the initial scale for edge correlation. Fig. 1 illustrates an example of one dimensional R_n. Fig. 1(b)–(d) show the results after applying wavelet transform to *f(x)* at scales j=1, 2, 3, respectively. Fig. 1(e) shows the result of $R_2(1,x) = |W_{2^1} f(x) W_{2^2} f(x)|$. From the example, it is apparent that R_2 reveals a peak whenever a true edge exists. On the other hand, if a point at location x is not a true edge, it is suppressed by the multiplication process. Therefore, based on $R_n(j,x,y)$, the noise in an image can be suppressed while the true edges can be retained. In this paper, the number of scales for multiplication is chosen to be 2. This is because if the number is larger, the edge delocalization problem of wavelet transforms will become more serious. In order to conserve the energy level, $R_n(j,x,y)$ has to be normalized as follows:

$$\overline{R}_n(j,x,y) = R_n(j,x,y) \sqrt{\frac{MP(j)}{RP_n(j)}},$$

where $MP(j) = \sum_{x,y} |M_{2^j} f(x,y)|^2$ and $RP_n(j) = \sum_{x,y} |R_n(j,x,y)|^2$. In the feature point selection process, an edge point is recognized as a candidate if its corresponding normalized edge correlation $\overline{R}_2(1,x,y)$ is larger than its corresponding modulus value. Basically, the above mentioned process is equivalent to detecting an edge point whose edge response is the strongest within a local area. In what follows, we summarize the three conditions adopted in our approach which will be used to judge whether a point *P(x,y)* is a feature point or not.

Condition 1: $P(x,y)$ must be an edge point of the image *f(x,y)*. This means that $P(x,y)$ is a local maxima of $M_{2^1} f(x,y)$, and $M_{2^1} f(x,y) >$ a threshold,

Condition 2: $\overline{R}_2(1,x,y) > M_{2^1} f(x,y)$,

Condition 3: $$M_{2^1} f(x,y) = \max_{(x',y') \in N_p} \left\{ M_{2^1} f(x',y') \right\}, \tag{3}$$

where Np is the neighborhood of $P(x,y)$.

III. Finding Correct Matching Pairs

In the previous section, we have presented a systematic way to extract important features from two partially overlapping images. In this section, we shall show how to find a set of correct matching pairs between the above images. In what follows, the procedure will be elaborated in detail step by step.

A. Defining a Matching Metric

In this subsection we shall define a metric and then use it to evaluate the similarity between any two feature points. Let $p=(p_x,p_y)^t$ and $q=(q_x,q_y)^t$ be two feature points located, respectively, in $f_1(x,y)$ and $f_2(x,y)$, where $f_1(x,y)$ and $f_2(x,y)$ are two partially overlapping images. A cross–correlation which can be used to measure the similarity degree between p and q is defined as follows [1], [3]:

$$C_{f_1,f_2}(p;q) = \frac{1}{\sigma_1\sigma_2(2M+1)^2}\sum_{x,y=-M}^{x,y=M}[f_1(x+p_x,y+p_y)-\mu_1][f_2(x+q_x,y+q_y)-\mu_2], \qquad (4)$$

where μ_i and σ_i are the local mean and variance of $f_i(x,y)$, respectively; $(2M+1)^2$ represents the area of matching window. In general, the format of the similarity measure defined in Eq. (4) is very sensitive to rotation. Therefore, if the rotation effect is important in an application, Eq. (4) should be updated as follows:

$$\overline{C}_{f_1,f_2}(p;q;\theta) = \frac{1}{\sigma_1\sigma_2(2M+1)^2}\sum_{\hat{x},\hat{y}=-M}^{M}[f_1(x+p_x,y+p_y)-\mu_1][f_2(\hat{x}+q_x,\hat{y}+q_y)-\mu_2], \qquad (5)$$

where $x=\hat{x}\cos\theta-\hat{y}\sin\theta$ and $y=\hat{x}\sin\theta+\hat{y}\cos\theta$. If the angle θ in Eq. (5) can be estimated in advance, then no matter how $f_1(x,y)$ or $f_2(x,y)$ are rotated, finding the correct matching between the two images is always easier.

B. Estimating the Orientation of a Feature Point

In the previous subsection, we have mentioned that the orientation of a feature point is important for deriving a correct metric. In this subsection we shall discuss how the orientation of a feature point is estimated. In Section 2, we have mentioned that two sets of feature points are extracted, respectively, from two partially overlapping images. In order to perform accurate image registration between these two images, the corresponding feature points between the two images have to be identified. In Eq. (5), the fit measure $\overline{C}_{f_1,f_2}$ contains a rotation angle, θ, which represents the orientation difference between two selected feature points. In order to solve θ, we have to determine the orientation of each point in advance. Basically, the orientation of a feature point can be estimated by using the results of the wavelet transform described in Section II, i.e., $W_{2^j}^1 f$ and $W_{2^j}^2 f$. A standard representation of the orientation of an edge–based feature point at scale 2^j can be expressed as follows [19], [21]:

$$Arg(W_{2^j}^1 f(x,y) + iW_{2^j}^2 f(x,y)).$$

However, the above representation might be very sensitive to noise. Therefore, we adopt a line–fitting model to solve the noise problem.

Let p be a feature point and Ω_e be its neighboring edge points with a $(2M_e + 1) \times (2M_e + 1)$ neighborhood. Since p is an edge point, there should exist an edge line passing through it. By considering p as a bridging point, an edge line passing through p can be determined by searching in all the directions from p. All the edge points on the edge line are then used as candidates for determining the orientation of the edge line. In the searching process, the edge connection constraint and the direction consistency constraint have to be enforced. Let $\mathcal{P} = \{ p_1, p_2, ..., p_i = (x_i, y_i)^t, ..., p_N \}$ be the set of selected edge points in Ω_e for determining the orientation of p, where p_i is the successor of p_{i-1}. Let $q_k = (\tilde{x}_k, \tilde{y}_k)^t$ be an edge point near p_i and $q_k \notin \mathcal{P}$. Since q_k is not the successor of p_i, the connection constraint is violated. It means that $|\tilde{x}_k - x_i| > 2$ or $|\tilde{y}_k - y_i| > 2$. Conversely, if q_k is not selected due to the direction consistency constraint, it means that the direction difference between $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i q_k}$ is larger than that of $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i p_{i+1}}$. For example, in Fig. 2, q_1, q_2 and l_4 are not selected since the connection condition is not satisfied. l_3 cannot be chosen because it violates the direction consistency constraint. As to the edge line segment l_2, it violates the basic requirement that a candidate edge line should pass through point p. Of course, it is also possible that there are more than one edge lines passing through p. Under the circumstances, we adopt the first line detected to estimate the orientation. This policy is feasible because a check process will be introduced later to eliminate the false matching pairs. In what follows, we shall use a line–fitting model to estimate the direction of an edge point.

Assume that a set of N points $\{ q_i = (x_i, y_i)^t \}_{i=1,2,..,N}$ is about to fit a straight line: $y_i = mx_i + b$. In order to evaluate the goodness of a match, a cost function is defined as follows [32]:

$$\xi = \sum_{i=1}^{N} \left(\frac{y_i - b - mx_i}{\sigma_i^2} \right)^2, \tag{6}$$

where σ_i represents the weight of a point $(x_i, y_i)^t$. The best fit will be achieved whenever an estimation *(m,b)* minimizes the cost function ξ. In order to minimize ξ, we have

$$\frac{\partial \xi}{\partial b} = -2 \sum_{i=1}^{N} \frac{y_i - b - mx_i}{\sigma_i^2} = 0 \tag{7}$$

and

$$\frac{\partial \xi}{\partial m} = -2 \sum_{i=1}^{N} \frac{(y_i - b - mx_i)x_i}{\sigma_i^2} = 0. \tag{8}$$

Let $\sigma_i = 1$. By solving Eq. (7) and (8), m is obtained as follows:

$$m = \frac{N\ \overline{X}\ \overline{Y} - \sum_{i=1}^{N} x_i y_i}{N\ \overline{X}\ \overline{X} - \sum_{i=1}^{N} x_i^2}, \tag{9}$$

where $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\overline{Y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. Since only the orientation is required in our scheme, the calculation of b is not necessary. Further, in order to make the orientation of an edge range from 0 to 360 degrees, both sides of an edge have to be distinguished. Basically, this problem can be solved by calculating the signs of the extremes of $W_{2^2}^1 f$ and $W_{2^2}^2 f$. Fig. 3 shows an example demonstrating how the signs of the extremes of $W_{2^2}^1 f$ and $W_{2^2}^2 f$ can be used to uniquely decide the orientation of an edge.

C. Estimating the Orientation Difference between Two Overlapping Images

From the detected feature points as well as their corresponding orientations, it is not difficult to determine the matching pairs between two overlapping images. In what follows, we shall describe how the above mentioned information is used to estimate the orientation difference between two overlapping images. Let $FP_{f_1} = \left\{ p_i = (p_x^i, p_y^i)^t \right\}_{i=1,2,\ldots,N_{f_1}}$ and $FP_{f_2} = \left\{ q_j = (q_x^j, q_y^j)^t \right\}_{j=1,2,\ldots,N_{f_2}}$ be two sets of feature points extracted from two partially overlapping images, f_1 and f_2, respectively. N_{f_1} and N_{f_2}, respectively, represent the number of elements in FP_{f_1} and FP_{f_2}. Let $A(u)$ be the angle of an edge (or feature) point u. For a feature point p_i in FP_{f_1} and a feature point q_j in FP_{f_2}, the orientation difference between p_i and q_j can be calculated as follows:

$$\theta_{i,j} = A(q_j) - A(p_i). \tag{10}$$

Here, $\theta_{i,j}$ ranges from 0° to 359°, and its value has to be an integer. In order to estimate the orientation difference between f_1 and f_2, $\theta_{i,j}$ and the similarity measure $\overline{C}_{f_1,f_2}$ are used. A so–called ''angle histogram'' $H(\theta)$ reports the distribution of the number of $\left\{ p_i \Leftrightarrow q_j \right\}$ pairs at angle θ that satisfy the conditions $\theta_{i,j} = \theta$ and $\overline{C}_{f_1,f_2}(p_i; q_j; \theta_{i,j}) \geq 0.8$. Basically, from the angle histogram, the rotation angle can be decided by seeking the angle that corresponds to the maximum peak in the histogram. If the orientation difference

between f_1 and f_2 is $\overline{\theta}$, then the highest peak in $H(\theta)$ must be very close to $\overline{\theta}$. Further, it is possible to modify $H(\theta)$ so that a more accurate estimation can be obtained. The modification is

$$\overline{H}(\theta) = \sum_{i=-2}^{2} H(\theta + i). \tag{11}$$

Here, $\theta + i$ may possibly exceed the limitation of 360°. Under these circumstances, it has to be adjusted by modulation. Using $\overline{H}(\theta)$ and finding its maximum peak, a very accurate estimation for the orientation difference between two partially overlapping images can be found. For example, Fig. 4 (a) and (b) are two partially overlapping Pentagon images. The two images have a 90° orientation difference. Fig. 4 (c) and (d) show, respectively, the angle histogram $H(\theta)$ and the modified angle histogram $\overline{H}(\theta)$. In Fig. 4 (c), it is found that the maximum peak is located at 87°. For the modified histogram, the maximum peak is at 88°. In comparison with some conventional methods which require derivation of the correlations of all feature points in a window, the number of feature points used in the proposed method is very few. Therefore, the calculation of the similarity measure $\overline{C}_{f_1 f_2}$ between any $\left\{ p_i \Leftrightarrow q_j \right\}$ pair only requires very short computation time.

D. Finding the Matching Pairs

In the previous section, we have described how the orientation difference between $f_1(x, y)$ and $f_2(x, y)$ is derived. Now, we are ready to find the matching feature point pairs between $f_1(x, y)$ and $f_2(x, y)$. The first step of the process is to rotate all the points in $f_1(x, y)$ to their new positions located in $\hat{f}_1(\hat{x}, \hat{y})$, where $\hat{x} = x\cos\overline{\theta} + y\sin\overline{\theta}$, $\hat{y} = -x\sin\overline{\theta} + y\cos\overline{\theta}$. If an edge point in f_1 is rotated to a new position in $\hat{f}_1$, the angle θ is also adjusted to suit the coordinates of $\hat{f}_1$, i.e., $\hat{\theta} \rightarrow \theta + \overline{\theta}$. Here, $\overline{\theta}$ is the orientation difference between f_1 and f_2. Let $E(u)$ denote the set of edge points within a $(2M_s + 1)^2$ window of an image f, where u is the window's center. Given a feature point $\hat{p}_i$ in $\hat{f}_1(x, y)$, the matching problem is used to find a proper point $\tilde{q}$ in $E(q_k)$ for every $q_k \in FP_{f_2}$ such that the pair $\left\{ \hat{p}_i \leftrightarrow \tilde{q} \right\}$ becomes a matching pair. A pair $\left\{ \hat{p}_i \leftrightarrow \tilde{q} \right\}$ is qualified as a matching pair if two conditions are satisfied:

Condition 1: $C_{\hat{f}_1 f_2}(\hat{p}_i; \tilde{q}) = \max_{q_k \in FP_{f_2}} \max_{\tilde{q}_n \in E(q_k)} C_{\hat{f}_1 f_2}(\hat{p}_i; \tilde{q}_n)$,

and

Condition 2: $C_{\hat{f}_1 f_2}(\hat{p}_i; \tilde{q}) > T_c$ where $T_c = 0.75$.

Condition 1 enforces finding an edge point $\tilde{q} \in E(q_k)$ and $q_k \in FP_{f_2}$ such that the measure $C_{\hat{f}_1,f_2}$ is maximized. As for Condition 2, it forces the value of $C_{\hat{f}_1,f_2}$ of a matching pair to be larger than a threshold (0.75 in this case). Further, by introducing another constraint, the orientation criterion, the searching speed can be even faster. As we know, $\hat{f}_1(x, y)$ is obtained by rotating $f_1(x, y)$ with an angle $\overline{\theta}$; therefore, the orientation difference between $\hat{f}_1(x, y)$ and another image $f_2(x, y)$ is very small. Hence, it is reasonable to introduce another constraint which forces the orientation difference between $\hat{f}_1(x, y)$ and $f_2(x, y)$ to be less than 5°, i.e., $|A(\hat{p}_i) - A(\tilde{q})| < 5^\circ$. Adding this criterion and using it together with the previous two conditions will significantly speed up the search time. In real implementation, the orientation constraint will be tested first. If the constraint is not satisfied, it is not necessary to test Condition 1 and Condition 2. In this way, only a few pairs are needed to calculate the cross–correlation measure $C_{\hat{f}_1,f_2}$, which is considered a time bottleneck of the whole process.

E. Eliminating the False Matching Pairs

A new method is proposed to eliminate the incorrectly matched pairs. Li *et al.*[12] proposed an iterative scheme that could not remove false pairs completely and efficiently. Here, we present a non–iterative scheme based on the idea that the distance between two points in the same image will be preserved when it undergoes a rigid transform. Let $MP = \{p_i \Leftrightarrow q_i\}_{i=1,2,..,N_m}$ be a set of matching pairs, where N_m represents the number of elements in *MP*, $p_i = (p_x^i, p_y^i)^t$ is a point in $\hat{f}_1(x, y)$, and $q_i = (q_x^i, q_y^i)^t$ is a point in $f_2(x, y)$. If all the matching pairs in *MP* are correct, then the following equation should hold, i.e.,

$$p_i = s\ q_i + T, \text{ for } i=1, 2, ..., N_m, \tag{12}$$

where s and T are, respectively, a scalar and a translation vector. Since the orientation difference between $\hat{f}_1(x, y)$ and $f_2(x, y)$ is very small, the rotation matrix is thus not part of Eq. (12). Let $\{p_i \Leftrightarrow q_i\}$ and $\{p_j \Leftrightarrow q_j\}$ be two correct matching pairs in *MP*. The scale s between $\hat{f}_1(x, y)$ and $f_2(x, y)$ can be estimated as follows:

$$s = d_2/d_1,$$

where $d_1 = \sqrt{(p^{\,i}_x - p^{\,j}_x)^2 + (p^{\,i}_y - p^{\,j}_y)^2}$ and $d_2 = \sqrt{(q^{\,i}_x - q^{\,j}_x)^2 + (q^{\,i}_y - q^{\,j}_y)^2}$. Once the scale s is known, the translation T^i between p_i and q_i can be calculated by using Eq. (12). Furthermore, the translation T^j between p_j and q_j can be obtained accordingly. Since the pairs $\{p_i \Leftrightarrow q_i\}$ and $\{p_j \Leftrightarrow q_j\}$ are correct matching pairs, the difference between T^i and T^j should be very small. Therefore, by checking the distance between T^i and T^j, we can decide whether $\{p_i \Leftrightarrow q_i\}$ and $\{p_j \Leftrightarrow q_j\}$ are consistent or not.

Next, we propose a non–iterative method based on the aforementioned consistency test to eliminate those mistakenly matched pairs. Let $S(i)$ denote a counter of the number of times the $\{p_i \Leftrightarrow q_i\}$ pair is consistent with other matching pairs. Assume that $\{p_j \Leftrightarrow q_j\}$ is a matching pair to be checked. $\{p_j \Leftrightarrow q_j\}$ is considered to be consistent with $\{p_i \Leftrightarrow q_i\}$ if and only if the distance between their translation vectors, T^i and T^j, is less than a threshold (which is set to 5 in this paper). If the two pairs are consistent, $S(i)$ is increased by 1. The process proceeds until all the matching pairs are compared. Since there are N_m elements in MP, the total number of consistency tests will be $N_m(N_m - 1)/2$. After the consistency test, a counter value $S(i)$ will be associated with every matching pair $\{p_i \Leftrightarrow q_i\}$, for i=1, 2, ..., N_m. Since two matching pairs can uniquely determine a set of registration parameters, the value 2 can be used as a threshold to determine whether the pair $\{p_i \Leftrightarrow q_i\}$ can be accepted. However, if $\{p_i \Leftrightarrow q_i\}$ is said to be a correct match, to compensate for inaccuracies, we require that the value of $S(i)$ be larger than 2. Therefore, if the value of $S(i)$ is less than or equal to 2, then its corresponding matching pair is considered mismatched and should be eliminated.

IV. Deriving the Correct Transformations

After eliminating all the false matching pairs, a set of correct matching pairs is left. Assume that this matching set is $\{u_i \Leftrightarrow v_i\}_{i=1,2,..,N_c}$, where N_c is the total number of correct matching pairs. In general, the 2–D point sets $\{u_i\}$ and $\{v_i\}$ should satisfy the following relation:

$$v_i = s\ R\ u_i + T, \text{ for } i=1, 2, ..., N_c, \tag{13}$$

where s is a scalar, $R = \begin{pmatrix} \cos\hat{\theta} & \sin\hat{\theta} \\ -\sin\hat{\theta} & \cos\hat{\theta} \end{pmatrix}$ represents a rotation matrix , $T=(t_x, t_y)^t$ is a translation vector, $u_i = (u^{\,i}_x, u^{\,i}_y)^t$ is a point in $\hat{f}_1(x, y)$, $v_i = (v^{\,i}_x, v^{\,i}_y)^t$ represents a point in $f_2(x, y)$, and $\hat{\theta}$ is the orientation

difference between $\hat{f}_1(x, y)$ and $f_2(x, y)$. In Section III.C, the initial orientation angle $\overline{\theta}$ between $f_1(x, y)$ and $f_2(x, y)$ has been estimated by using the histogram $\overline{H}(\theta)$. However, $\overline{\theta}$ can only be considered a rough guess. In what follows, we shall take advantage of the set of correct matching pairs, $\{u_i \Leftrightarrow v_i\}_{i=1,2,..,N_c}$, to fine tune the previous result. Basically, the fine tuning, $\hat{\theta}$, is conducted to derive the orientation difference between $\hat{f}_1$ and f_2. If $\hat{\theta}$ can be derived, a more accurate rotation angle, $\theta = \hat{\theta} + \overline{\theta}$, between f_1 and f_2 can be obtained. Next, a method which can be used to derive s, $\hat{\theta}$, and T based on the correct matching pairs is presented.

In order to derive s, R, and T based on the set of correct matching pairs $\{u_i \Leftrightarrow v_i\}_{i=1,2,..,N_c}$, we will introduce an error function as follows:

$$\Phi = \sum_{i=1}^{N_c} \| s\ R\ u_i + T - v_i \|^2.$$

By minimizing Φ, a set of optimal solutions can be derived. In [20], Umeyama proposed a good approach to solving the above problem. Here, we will simply follow his methodology. From $\frac{\partial \Phi}{\partial T} = 0$, one can obtain

$$T = \overline{v} - s\ R\ \overline{u}, \tag{14}$$

where $\overline{u} = \frac{1}{N_c}\sum_{i=1}^{N_c} u_i$ and $\overline{v} = \frac{1}{N_c}\sum_{i=1}^{N_c} v_i$. Substituting Eq. (14) into Eq. (13) and from $\frac{\partial \Phi}{\partial s} = 0$, we have

$$s = \sum_{i=1}^{N_c} \tilde{v}_i^t\ R\ \tilde{u}_i \Big/ \sum_{i=1}^{N_c} \tilde{u}_i^t\ \tilde{u}_i, \tag{15}$$

where $\tilde{u}_i = u_i - \overline{u}$ and $\tilde{v}_i = v_i - \overline{v}$. Substituting Eq. (14) and (15) into Eq. (13) and reorganizing the content of Φ, we obtain

$$\Phi = \sum_{i=1}^{N_c} \tilde{u}_i^t\ \tilde{v}_i - \left[\sum_{i=1}^{N_c} \tilde{v}_i^t\ R\ \tilde{u}_i\right]^2 \Big/ \sum_{i=1}^{N_c} \tilde{v}_i^t\ \tilde{v}_i.$$

Here, minimizing Φ can be converted into maximizing the term

$$\Phi' = \left[\sum_{i=1}^{N_c} \tilde{v}_i^t\ R\ \tilde{u}_i\right]^2. \tag{16}$$

Now, the problem at hand is to solve R. In [8], Arun *et al*. proposed a singular value decomposition (SVD) method to solve R. The procedure is illustrated as follows.

Step 1: Calculate the 2×2 matrix $H = \sum_{i=1}^{N_c} \tilde{u}_i \ \tilde{v}_i^t$.

Step 2: Find the SVD of H, i.e., $H = U\Lambda V^t$.

Step 3: $R = VU^t$. (17)

The whole procedure for estimating the registration parameters can be summarized as follows. First, the rotation matrix R is found by solving Eq. (17). Then, the fine tuning, $\hat{\theta}$, between $\hat{f}_1(x, y)$ and $f_2(x, y)$ can be estimated from R. Next, by solving Eq. (15), the scale s can be obtained. Furthermore, by solving Eq. (14), the translation vector T is obtained. Based on these parameters, accurate registration between images can be achieved.

V. The Matching Algorithm

In Sections III and IV, we have described how a number of correct matching pairs between two images is obtained and how they are used to derive the registration parameters s, R, and T. Based on these parameters, accurate registration can be achieved. However, if accuracy is a major concern in a system, the current status may not satisfy the requirement. In what follows, we propose using an iterative scheme to refine the registration results. The proposed scheme is a two–stage algorithm. The purpose of the first stage is to obtain the initial values of the registration parameters. The method of achieving this goal has been described in Sections III and IV. In the second stage, an iterative procedure is proposed to repeatedly refine the registration parameters. In what follows, we shall describe the details of the second stage as well as the complete algorithm.

In the first step of the second stage, with the assistance of the initial values of s, R, and T, the input image $f_1(x, y)$ is transformed to $\hat{f}_1(\hat{x}, \hat{y})$, and the relation between $(x, y)^t$ and $(\hat{x}, \hat{y})^t$ is as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{s} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \hat{x} - t_x \\ \hat{y} - t_y \end{pmatrix}, \tag{18}$$

where $\theta = \overline{\theta} + \hat{\theta}$, $\overline{\theta}$ is obtained by applying the method described in Section III.C, and $\hat{\theta}$ is obtained from Eq. (17) . Let $M_p^c = \left\{ u_i \Leftrightarrow v_i \right\}_{i=1,2,..,N_c}$ denote the set of matching pairs obtained in the previous stage, where N_c is the number of elements in M_p^c, u_i is a point in f_1, and v_i is a point in f_2. For each point v_i in f_2, our goal is to find its exact corresponding point $\hat{u}_i$ in $\hat{f}_1$ such that the refined corrections for the

parameters *s, R*, and *T* can be obtained. It is known that $\hat{f}_1$ is obtained by transforming f_1 with the initial values of parameters *s, R*, and *T*; therefore, the location of $\hat{u}_i$ in $\hat{f}_1$ should have a coordinate very similar to v_i in f_2. Hence, for each v_i in f_2, its corresponding point $\hat{u}_i$ in $\hat{f}_1$ can be found by searching the pixels within a neighborhood centered at the coordinate v_i in $\hat{f}_1$ such that the measure $C_{\hat{f}_1 f_2}$ is maximized and has a value larger than a threshold, that is

$$C_{\hat{f}_1 f_2}(\hat{u}_i, v_i) = \max_{u \in N_{v_i}} C_{\hat{f}_1 f_2}(u, v_i), \text{ and } C_{\hat{f}_1 f_2}(\hat{u}_i, v_i) \geq 0.75, \tag{19}$$

where N_{v_i} is the neighborhood centered at the coordinate v_i. In most cases, N_c is small (< 10). If it is larger, then the set M_p^c can be re–sampled so that it will be controlled properly. In general, the use of only a few feature points is sufficient to derive *R, T,* and *s* with high accuracy. Therefore, the refining process can be performed very quickly. Let $\hat{M}_p^c = \{\hat{u}_i \Leftrightarrow v_i\}_{i=1,2,..,\hat{N}_c}$ denote the set of matching pairs obtained from the refining process, where $\hat{N}_c$ is the number of elements in $\hat{M}_p^c$. From the set $\hat{M}_p^c$, the refined corrections $\hat{s}, \hat{\theta}, \hat{t}_x$, and $\hat{t}_y$ for the parameters s, θ, t_x, and t_y can be obtained by Eq. (14), (15), and (17), respectively. With $\hat{s}$, $\hat{\theta}$, $\hat{t}_x$ and $\hat{t}_y$, more accurate values of s, θ, t_x and t_y can be derived. Since

$$\begin{aligned} v_i &= \hat{s}R(\hat{\theta})\hat{u}_i + \hat{T} \\ &= \hat{s}R(\hat{\theta})[sR(\theta)u_i + T] + \hat{T} \\ &= s\hat{s}R(\theta + \hat{\theta})u_i + [\hat{s}R(\hat{\theta})T + \hat{T}], \end{aligned}$$

the values of s, θ, t_x and t_y can be corrected based on the following equation [3]:

$$(s, \theta, t_x, t_y)^t \Leftarrow (s\hat{s}, \theta + \hat{\theta}, \hat{s}\cos\hat{\theta}t_x + \hat{s}\sin\hat{\theta}t_y + \hat{t}_x, -\hat{s}\sin\hat{\theta}t_x + \hat{s}\cos\hat{\theta}t_y + \hat{t}_y)^t. \tag{20}$$

Using the results of (20) as a new set of initial values in Eq. (18) and applying the refining process iteratively, all the registration parameters can be updated continuously until satisfactory accuracy is achieved. In our study, we set the number of iterations to 3 and obtain very superb results. The proposed two–stage algorithm is illustrated as follows:

Stage A: A.1. Extract feature points from the input images f_1 and f_2.

A.2. Estimate the edge orientation using Eq. (9), and calculate the angle histogram $\overline{H}(\theta)$ with the measure $\overline{C}_{f_1 f_2}$.

A.3. From the angle histogram $\overline{H}(\theta)$, estimate the rotation difference $\overline{\theta}$.

A.4. Rotate $f_1(x, y)$ to $\hat{f}_1(x, y)$ with the angle $\overline{\theta}$.

A.5. Based on $\hat{f}_1(x, y)$ and $f_2(x, y)$, perform an initial matching to obtain the initial registration parameters $s, \hat{\theta}, t_x$ and t_y.

A.6. Set I=0 and $\theta = \overline{\theta} + \hat{\theta}$;

Stage B: B.1. Apply an affine transformation with the parameters (s, θ, t_x, t_y) to $f_1(x, y)$ and obtain $\hat{f}_1(x, y)$.

B.2. Based on $\hat{f}_1(x, y)$ and $f_2(x, y)$, perform a refined matching to obtain the matching set $\hat{M}_p^c$.

B.3. From the set $\hat{M}_p^c$, obtain the refined corrections $(\hat{s}, \hat{\theta}, \hat{t}_x, \hat{t}_y)$ for the parameters (s, θ, t_x, t_y) by using the method described in Section IV;

B.4. Update (s, θ, t_x, t_y) with the corrections $(\hat{s}, \hat{\theta}, \hat{t}_x, \hat{t}_y)$ by using Eq. (20).

B.5. I=I+1;
If $I \leq 2$ then goto Step B.1;
else stop.

VI. Experimental Results

In the experiments, a number of synthetic and real images were adopted as test images. Among them, the synthetic images were used to verify whether the proposed theory is accurate and robust. On the other hand, the real images were used to examine how well this algorithm works. All the synthetic images were of size 420×420. The synthetic images included Pentagon images, texture images, building images, and mountain images. For each synthetic image pair, one image was generated with a synthetic camera motion from the other image. The parameters for each synthetic camera motion are listed in Table 1. As to the real images, all the images were of size 512×512. They were the aerial images, rock images, texture images, and building images. The matching window parameter M used to calculate C_{f_1,f_2} was set to

9. In order to obtain the initial angle more efficiently, M was set to 7 in the calculation of $\overline{C}_{f_1,f_2}$. The neighborhood parameter M_e used to select a number of edge points for line–fitting was set to 6. The search space parameter M_s for obtaining matching pairs was set to 5. In what follows, we will describe the experimental results in more detail.

Fig.5 shows a series of images used to generate synthetic images for registration. Each image in Fig. 5 generates two different synthetic image pairs with two different camera motions. Therefore, there are eight synthetic image pairs for testing. All the synthetic camera motions are listed in Table 1. Columns 2, 4, 6, and 8 show different registration parameters used to generate various synthetic image pairs. The estimated registration parameters (s, t_x, t_y, θ) are, respectively, shown in columns 3, 5, 7, and 9. The accuracy of our algorithm can be verified by comparing the true registration parameters and the estimated ones. Even though various kinds of images are handled, our algorithm still produces very accurate registration results. In order to demonstrate the power of our method for estimating the initial orientation difference between images, the results of Zheng–Chellappa's method [3] are also listed in Table 1 for comparison. In Table 1, θ_1 denotes the initial rotation angle estimated from the angle histogram $\overline{H}(\theta)$, and θ_2 denotes the initial rotation angle obtained from Zheng–Chellappa's method. From Table 1, it is easy to find that the rotation angle, θ_1, estimated from $\overline{H}(\theta)$ is very close to the true rotation angle. Even though there are significant scene changes between images, our estimation method still produces very accurate results. As to Zheng–Chellappa's approach, it works well for most cases. However, the success of their method is based on the assumption that the images are taken at a stationary illumination source. If the images have significant scene changes, the illumination conditions will be changed, and the approach might fail. Fig. 6 shows two examples where Zheng–Chellappa's method cannot work well. (a) and (b) are the building image pair. (c) and (d) are the mountain image pair. Their true rotation parameters, θ's, are 75.0° and -75.0°, respectively. By applying Zheng–Chellappa's method, the estimations, θ_2's, for the initial rotation angles are 100.6° and -48.2°, respectively. Clearly, in both cases, Zheng–Chellappa's method produced poor estimations for the initial rotation angles.

An important application of image registration is in automatic aerial image analysis. Fig. 7 shows an example of aerial images of a rural area of Taiwan. (a) and (b) show the input image pair, and (c) shows the result after applying our registration algorithm. The two images in (a) and (b) have significant changes in rotation, translation, and scaling. It is notable that the intensity differences between (a) and (b) are also large. In this experiment, the estimated transform parameters were s= 1.061, θ=–41.205 $^\circ$, t_x= –141.444,

and t_y=–239.9. The correctness of the registration result can be verified by checking the continuities of the ridges between fields or mountains in Fig. 7(c). The feature points obtained by applying wavelet transforms are also shown in Fig. 7(a) and (b). The number of feature points detected in each image is roughly 100 points. This number is controlled by the size of the neighborhood N_p (see Eq. (3)). The larger the size of N_p is, the fewer feature points are detected. For most cases, the radius of neighborhood N_p is chosen to be 15. The white symbols, ''X'', indicate the positions where feature points are located. Fig. 8 and 9 show two other experiments on aerial images of Taiwan. In Fig. 8, the estimated transform parameters are s= 0.963, θ=–6.052 $^\circ$, t_x= –271.135, and t_y=134.781; and in Fig. 9, the estimated transform parameters are s= 0.967211, θ=–1.08 $^\circ$, t_x= –325.310, and t_y=–63.532. In both cases, our method produces very accurate registration results. On the other hand, in order to demonstrate how the registration results are improved by the iterative processes, Table 2 lists the results of registration with the above three aerial images.

In order to demonstrate the power of our approach, we also apply our algorithm to some images which have complicated textures. In general, the features in a complicated texture image are difficult to extract. Fig. 10 shows an example of images with complicated rock textures. (a) and (b) are the input images, and (c) is the mosaic of (a) and (b). The estimated transform parameters are s= 1.04161, θ=–83.054 $^\circ$, t_x= –65.089, and t_y=–277.678. Despite the fact that these images containing complicated textures, our algorithm still produces accurate registration results. Fig. 11 shows another example of images with complicated grass and sand textures. The estimated transform parameters are s= 0.947988, θ=–27.218 $^\circ$, t_x= –345.254, and t_y=–71.719.

Stereo vision and motion estimation are two other important applications in image registration. In both applications, the camera motion between consecutive frames is usually assumed to be small and smooth. If the camera motion is significant, it is difficult to track features and obtain their information. Therefore, consecutive images must be registered in advance. Previously, such an initial matching or registration is usually performed by manual methods. However, our registration method can perform such an initial matching automatically. Fig. 12 gives an example of such an application. (a) and (b) are the input images. (c) is the mosaic of (a) and (b). The estimated transform parameters are s= 0.998617, θ=22.849 $^\circ$, t_x= –117.748, and t_y=43.7386. With the help of the transform parameters, the camera motion can be compensated for well. In this way, the disparities of features between images can be easily found. Using these disparities, further surface recovery can be performed.

VII. Conclusions

In this paper, we have proposed a new edge–based approach to efficiently deal with the image registration problem. The proposed method applies the wavelet transform technique to extract feature points from a partially overlapping image pair. By defining a similarity measure metric, the two sets of feature points can be compared, and the correspondences between the feature points can be established. Once the set of correctly matched feature point pairs between two images are found, the registration parameters can be derived accordingly. The proposed method can tolerate approximately 10% scaling variation and does not have to restrict the position and orientation of the input images. Compared with conventional algorithms, the proposed scheme is a great improvement in the sense of efficiency as well as reliability for the image registration problem.

REFERENCES

[1] L. G. Brown, ‘‘A survey of image registration techniques,’’ *ACM Computer Surveys*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[2] A. D. Ventura, A. Pampini, and R. Schettini, ‘‘Image Registration by recognition of corresponding structures,’’ *IEEE Trans. Geosci. Remote Sensing*, vol. 28, no. 3, pp. 305–313, May 1990.

[3] Q. Zheng and R. Chellappa, ‘‘A computational vision approach to image registration,’’ *IEEE Trans. Image Processing*, vol. 2, no. 3, pp. 311–326, July 1993.

[4] S. T. Barnard and M. A. Fischler, ‘‘Computational Stereo,’’ *ACM Computer Surveys*, vol. 14, no. 4, pp. 553–572, Dec. 1982.

[5] H. J. Lee and W. L. Lei, ‘‘Region matching and depth finding for 3–D objects stereo aerial photographs,’’ *Pattern Recognition*, vol. 23, no. 1/2, pp. 81–94, 1990.

[6] Y. C. Hsieh, D. M. Mckeown, Jr., and F. P. Perlant, ‘‘Performance evaluation of scene registration and stereo matching for cartographic feature extraction,’’ *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 214–238, Feb. 1992.

[7] Q. Zheng and R. Chellappa, ‘‘Automatic feature point extraction and tracking in image sequences for arbitrary camera motion,’’ *I. J. Computer Vision*, vol. 15, pp. 31–76, 1995.

[8] K. S. Arun, T. S. Huang, and S. D. Blostein, ‘‘Least–square fitting of two 3–D point sets,’’ *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–9, no. 5, pp. 698–700, Sep. 1987.

[9] J. J. Rodriguez and J. R. Aggarwal, ‘‘Matching aerial images to3–D terrain maps,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. 12, no. 12, pp. 1138–1149, Dec. 1990.

[10] A. Goshtasby, ‘‘Image registration by local approximation methods,’’ *Image Vision Comput*., vol. 6, no. 4, pp. 255–261, Nov. 1988.

[11] B. S. Manjunath, R. Chellappa, and C. Malsburg, ‘‘A feature based approach to face recognition,’’ *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Champaign, Illinois, pp. 373–378, 1992

[12] Hui Li, B. S. Manjunath, and S. K. Mitra, ‘‘A contour–based approach to multisensor image registration,’’ *IEEE Trans. Image Processing*, vol. 4, no. 3, pp. 320–334, March 1995.

[13] E. De Castro and C. Morandi, ‘‘Registration of translated and rotated image using finite Fourier transform,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. 9, no. 5, pp. 700–703, Sep. 1987.

[14] T. Schenk, J. C. Li and C. Toth, ‘‘Towards an autonomous system for orienting digital stereo pairs,’’ *Photogrammetric Eng. Remote Sensing*, vol. 57, no. 8, pp. 1057–1064, Aug. 1991.

[15] A. Goshtasby, ‘‘Template Matching in Rotated Images,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. 7, no. 3, pp. 338–344, May 1985.

[16] F. P. Perlant and D. M. Mckeown, ‘‘Scene registration in aerial image analysis survey,’’ *Photogrammetric Eng. Remote Sensing*, vol. 56, no. 4, pp. 481–493, Apr. 1990.

[17] G. C. Stochman, S. Kopstein, and S. Benett, ‘‘Matching images to models for registration and object detection via clustering,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. 4, no. 3, pp. 229–241, 1982.

[18] S. Alliney and C. Morandi, ‘‘Digital image registration using projections,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI–8, no. 2, pp. 222–233, March 1986.

[19] J. F. Canny, ‘‘A computational approach to edge detection,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI–8, no. 6, pp. 679–697, Nov. 1986.

[20] S. Umeyama, ‘‘Least–squares estimation of transformation parameters between two point patterns,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI–13, no. 4, pp. 376–380, April 1991.

[21] S. Mallat and S. Zhong, ‘‘Characterization of signals from multi–scale edges,’’ *IEEE Trans. Pattern Anal. Machine Intell*., vol. PAMI–14, no. 7, pp. 710–732, July 1992.

[22] J. S. Lee, Y. N. Sun, C. H. Chen and C. T. Tsai, ‘‘Wavelet based corner detection,’’ *Pattern Recognition*, vol. 26, no. 6, pp. 853–865, 1993.

[23] T. Chang and C. C. Jay Kuo, ‘‘Texture analysis and classification with tree–structured wavelet transform,’’ *IEEE Trans. Image Processing.*, vol. 2, no. 4, pp. 429–441, October 1993.

[24] Y. Sheng, D. Roberge, Harold Szu, and T. Lu, ‘‘Optical wavelet matched filters for shift–invariant pattern recognition,’’ *Optics Letters*, vol. 18, no. 4, pp. 299–301, Feb. 1993.

[25] J. Zhou, X. Fand, and B. K. Ghosh, ‘‘Image segmentation based on multiresolution filtering,’’ in *Proc. IEEE Conf. Image Processing*, Austin, Texas, vol. 3, pp. 483–487, Nov. 1994.

[26] J. W. Hsieh, H. Y. Mark Liao, M. T. Ko, and K. C. Fan, ‘‘Wavelet–based shape from shading,’’ *Graphical Models and Image Processing,* vol. 57, no. 4, pp. 343–362, July 1995.

[27] S. Mallat, ‘‘Multifrequency channel decompositions of images and wavelet models,’’ *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP–37, no. 12, pp. 2091–2110, Dec. 1989.

[28] S. Mallat, ‘‘A theory for multiresolution signal decomposition: The wavelet representation,’’ *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–11, no. 12, pp. 674–693, July 1989.

[29] S. Mallat, ‘‘Multiresolution approximation and wavelet orthonormal bases of $L(R^2)$,’’ *Trans. Amer. Math. Soc.*, vol. 3–15, pp. 69–87, Sept. 1989.

[30] I. Daubechies, ‘‘The wavelet transform, time–frequency localization and signal analysis,’’ *IEEE Trans. Information Theory.*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.

[31] Y. Xu, J. B. Weaver, D. M. Healy, and J. Lu, ‘‘Wavelet transform domain filters: a spatially selective noise filtration technique,’’ *IEEE Trans. Image Processing*, vol. 3, no. 6, pp. 747–758, Nov. 1995.

[32] William H. Press, *et al.*, *Numerical Recipes in C: the Art of Scientific Computing,* Cambridge University Press, 2nd ed., pp. 661–662, 1992.

[33] J. W. Hsieh, H. Y. Liao, and K. C. Fan, ‘‘A new approach for edge detection using wavelet transforms,’’ *Asian Conference on Computer Vision*, Osaka, Nov. 23–25, pp. 520–525, 1993.

Caption of Tables

Table 1: The parameters of each synthetic camera motion and the registration results after applying our automatic registration algorithm.

Table 2: Three examples showing how registration parameters are improved by iterative processes. Here, the true registration parameters are not listed due to the true camera motions are unknown. Readers can examine the final results from Figs. 7(c), 8(c) and 9(c).

Captions of Figures

Figure 1: Edge correlation of $R_2(1,x)$ on a 1–D signal function $f(x)$. (a) The original signal function $f(x)$. (b)–(d) The results of the wavelet transform of $f(x)$ for j=1, 2, 3. (e) The result of $R_2(1,x)$.

Figure 2: An example illustrating some cases violating the connection constraint and the direction consistency constraint within the neightborhood of an edge point p.

Figure 3: An example illustrating how the angle of an edge line is extended to a range from 0° to 360° by using the extremes of $W^1_{2^2}f$ and $W^2_{2^2}f$. The edge angle in (a) is θ, but the one in (b) is $\theta + 180^\circ$ due to different signs of the extremes of $W^1_{2^2}f$ and $W^2_{2^2}f$.

Figure 4: Using the angle histogram to estimate the orientation difference between two partially overlapping Pentagon images. (a) The original Pentagon image. (b) The Pentagon image rotated with 90°. (c) The angle histogram $H(\theta)$. (d) The modified angle histogram $\overline{H}(\theta)$. From $\overline{H}(\theta)$, the rotation angle θ is located at 88°.

Figure 5: A set of images used to generate synthetic images for registration. (a) Pentagon image. (b) Texture image. (c) Building image. (d) Mountain image.

Figure 6: Two examples with significant scene changes. (a) and (b) are the building image pair. (c) and (d) are the mountain image pair. For both image pairs, poor estimation for the initial rotation angles are produced by adopting Zheng–Chellapa's method.

Figure 7: (a), (b) Two aerial images of rural areas of Taiwan; each small white *St*. Andrew cross indicates an extracted feature point. (c) The registration result of (a) and (b). The estimated registration parameters are s= 1.061, θ= –41.205 $^\circ$, t_x= –141.444, and t_y=–239.9.

Figure 8: (a), (b) Two aerial images of urban areas of Taiwan. (c) The registration result of (a) and (b). The estimated registration parameters are s= 0.963, θ=–6.052 $^\circ$, t_x= –271.135, and t_y=134.781.

Figure 9: (a), (b) Two additional aerial images of urban areas of Taiwan. (c) The registration result of (a) and (b). The estimated registration parameters are s= 0.967211, θ=–1.08 $^\circ$, t_x= –325.310, and t_y=–63.532.

Figure 10: Registration of rock images. (a), (b) Two rock images. (c) The registration result of (a) and (b). The estimated registration parameters are s= 1.04161, θ=–83.054 $^\circ$, t_x= –65.089, and t_y=–277.678.

Figure 11: Registration of texture images with grass and sand. (a), (b) Two grass–sand images. (c) Mosaic of (a) and (b). The estimated registration parameters are s= 0.947988, θ=–27.218 $^\circ$, t_x= –345.254, and t_y=–71.719.

Figure 12: Estimation of camera motion. (a), (b) Two building images. (c) Mosaic of (a) and (b). The estimated transform parameters are s= 0.998617, θ=22.849 $^\circ$, t_x= –117.748, and t_y=43.7386.

List of Table

Test Images	Scale: s		Translation: t_x		Translation: t_y		Rotation: θ		θ_1	θ_2
	True	Estimated	True	Estimated	True	Estimated	True	Estimated		
Pentagon Images (1)	0.9	0.9005	10.0	9.9865	10.0	10.101	30.0	30.08	27	66.2
Pentagon Images (2)	1.1	1.0997	–20.0	–19.967	–20.0	–19.972	–45.0	–44.98	–45	–42.3
Texture Images (1)	0.95	0.9509	5	4.984	30.0	29.937	80.0	79.948	82	87.03
Texture Images (2)	1.10	1.1019	70	69.882	–10.0	–10.029	–75.0	–75.012	–76	–71.19
Building Images (1)	1.10	1.1015	75.0	74.994	75.0	75.10	–60.0	–60.001	–59	–70.84
Building Images (2)	0.90	0.8988	50.0	49.556	50.0	50.013	75.0	74.930	72	100.6
Mountain Images (1)	0.90	0.8993	–145	–145.06	110.0	109.974	85.0	84.977	83	76.4
Mountain Images (2)	1.10	1.0967	140	140.022	–65.0	–65.056	–75.0	–74.734	–72	–48.2

θ_1: the initial rotation angle obtained from the angle histogram $\overline{H}(\theta)$.

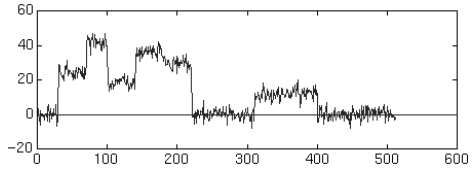
θ_2: the initial rotation angle obtained by Zheng–Chellappa's method.

Table 1. The parameters of each synthetic camera motion and the registration results after applying our automatic registration algorithm.

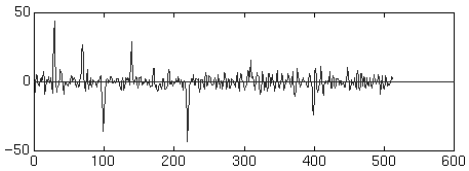
Test Images	Iteration: i	Scale: s	Translation: t_x	Translation: t_y	Rotation: θ
Aerial Images I (Fig. 7)	i=0	1.0485	–143.397	–236.235	–42
	i=1	1.062	–142.571	–238.811	–41.673
	i=2	1.0610	–141.444	–239.900	–41.205
Aerial Images II (Fig. 8)	i=0	0.9688	–268.047	135.901	–7
	i=1	0.9653	–272.056	134.572	–6.756
	i=2	0.9629	–271.135	134.781	–6.052
Aerial Images II (Fig. 9)	i=0	0.9832	328.424	–67.298	1
	i=1	0.9646	325.345	–65.256	–0.752
	i=2	0.9672	325.310	–63.532	–1.081

Table 2. Three examples showing how registration parameters are improved by iterative processes. Here, the true registration parameters are not listed due to the true camera motions are unknown. Readers can examine the final results from Figs. 7(c), 8(c) and 9(c).

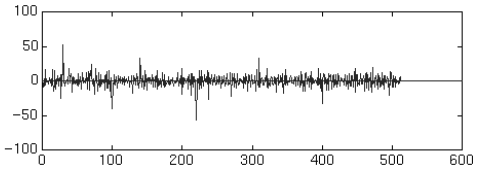
List of Figures



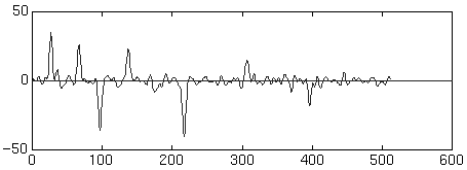
(a) A signal *f(x)*.



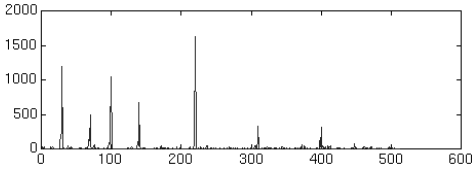
(c) $W_{2^2}f(x)$



(b) $W_{2^1}f(x)$



(d) $W_{2^3}f(x)$



(e) $R_2(1,x) = |W_{2^1}f(x)\ W_{2^2}f(x)|$

Figure 1. Edge correlation of $R_2(1,x)$ on a 1–D signal function $f(x)$. (a) The original signal function $f(x)$. (b)–(d) The results of the wavelet transform of $f(x)$ for j=1, 2, 3. (e) The result of $R_2(1,x)$.

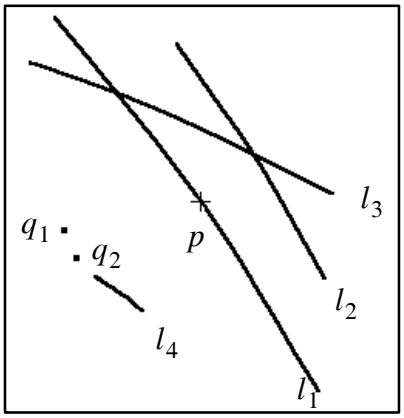
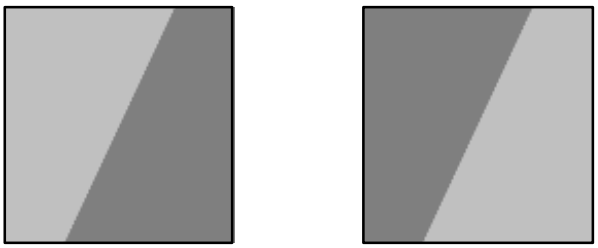


Figure 2. An example illustrating some cases violating the connection constraint and the direction consistency constraint within the neightborhood of an edge point p.



The extreme of $W^1_{2^2}f > 0$. The extreme of $W^2_{2^2}f > 0$.

(a) $\measuredangle = \theta$

The extreme of $W^1_{2^2}f < 0$. The extreme of $W^2_{2^2}f < 0$.

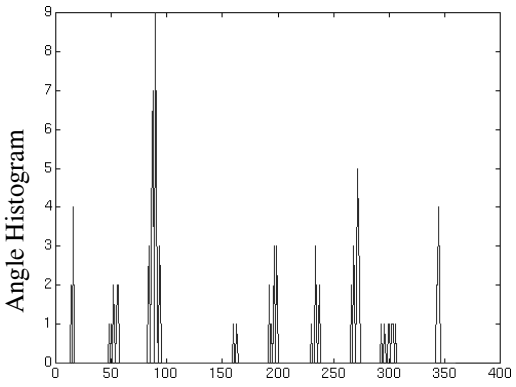
(b) $\measuredangle = \theta + 180^\circ$

Figure 3. An example illustrating how the angle of an edge line is extended to a range from 0° to 360° by using the extremes of $W^1_{2^2}f$ and $W^2_{2^2}f$. The edge angle in (a) is θ, but the one in (b) is $\theta + 180^\circ$ due to different signs of the extremes of $W^1_{2^2}f$ and $W^2_{2^2}f$.

(a)

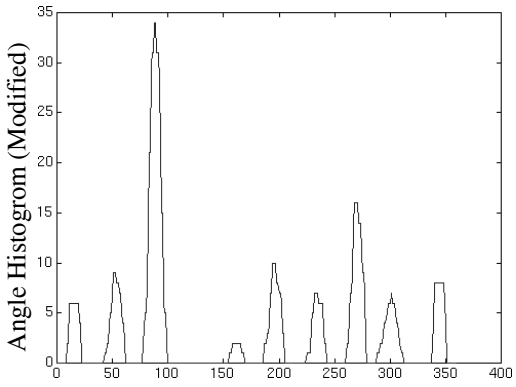
(b)

The Maximum Peak at 87°.



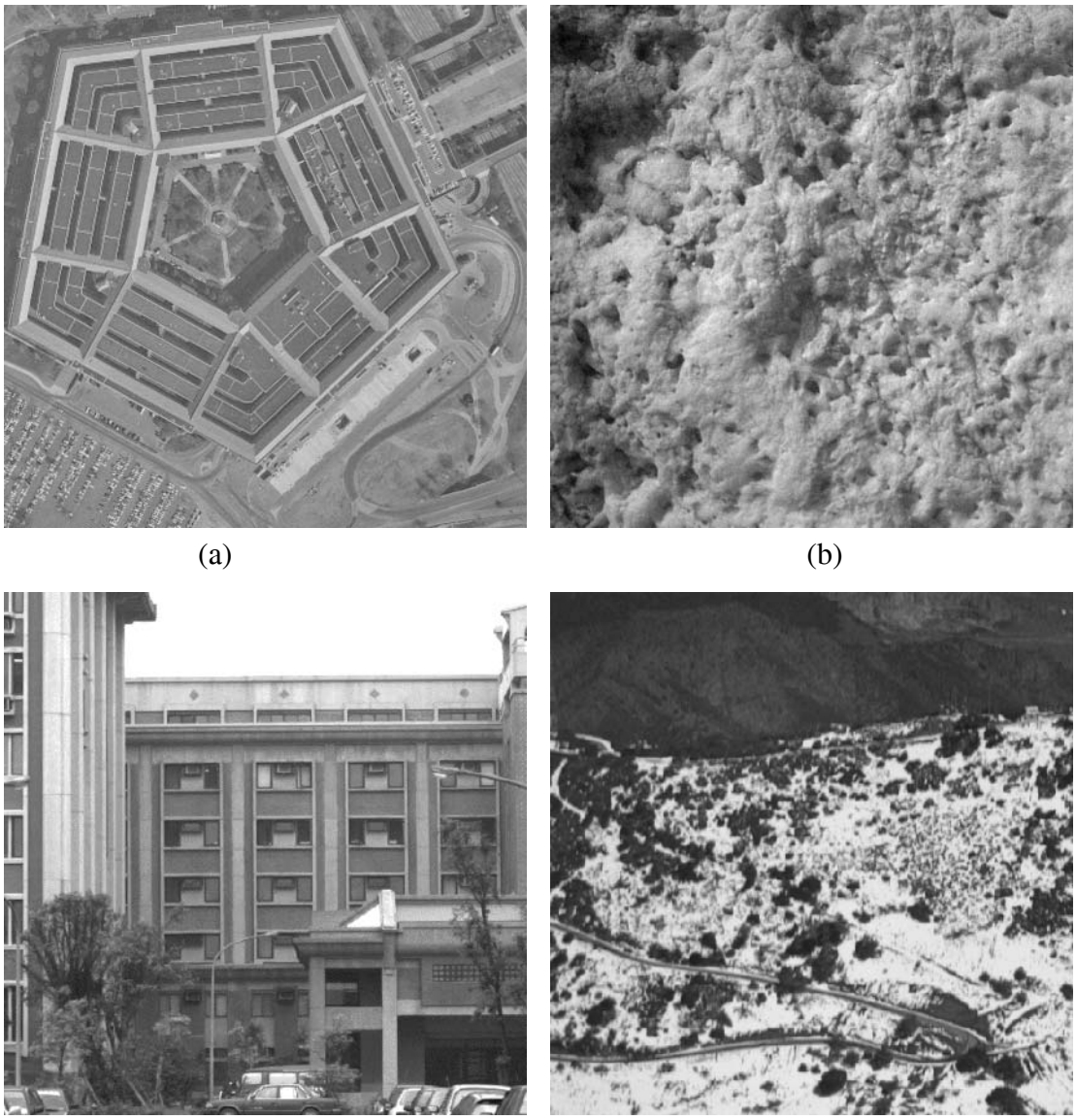
(c) $H(\theta)$

The Maximum Peak at 88°.



(d) $\overline{H}(\theta)$

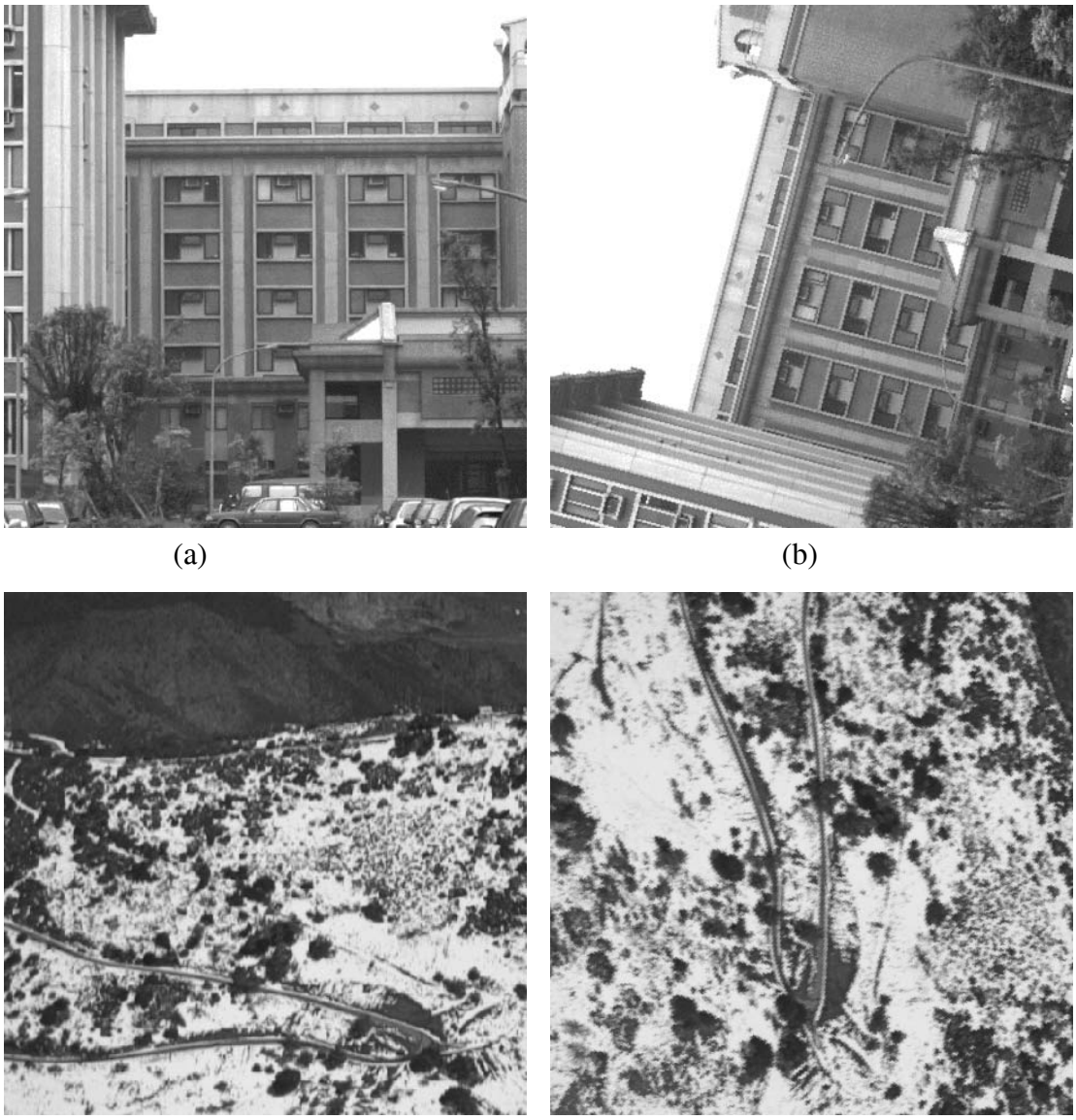
Figure 4. Using the angle histogram to estimate the orientation difference between two partially overlapping Pentagon images. (a) The original Pentagon image. (b) The Pentagon image rotated with 90°. (c) The angle histogram $H(\theta)$. (d) The modified angle histogram $\overline{H}(\theta)$. From $\overline{H}(\theta)$, the rotation angle θ is located at 88°.



(a) (b)

(c) (d)

Figure 5. A set of images used to generate synthetic images for registration. (a) Pentagon image. (b) Texture image. (c) Building image. (d) Mountain image.

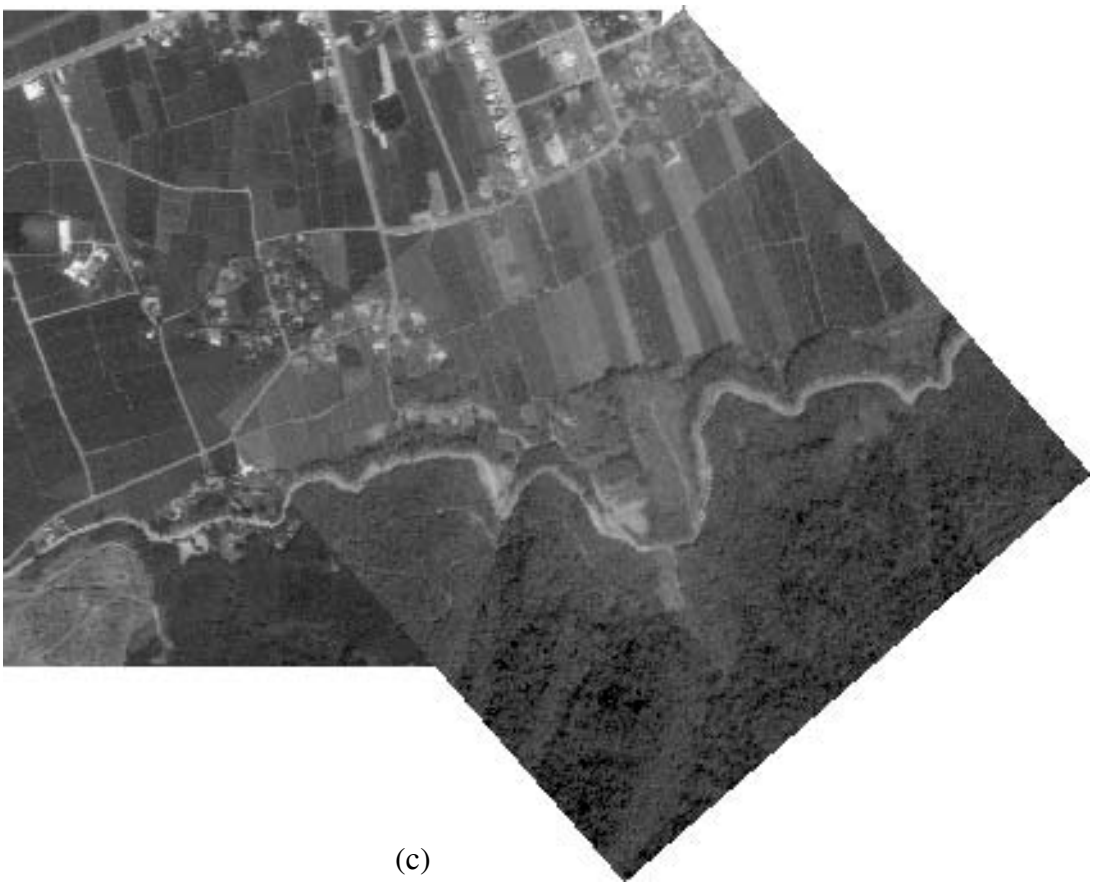


(a) (b)

(c) (d)

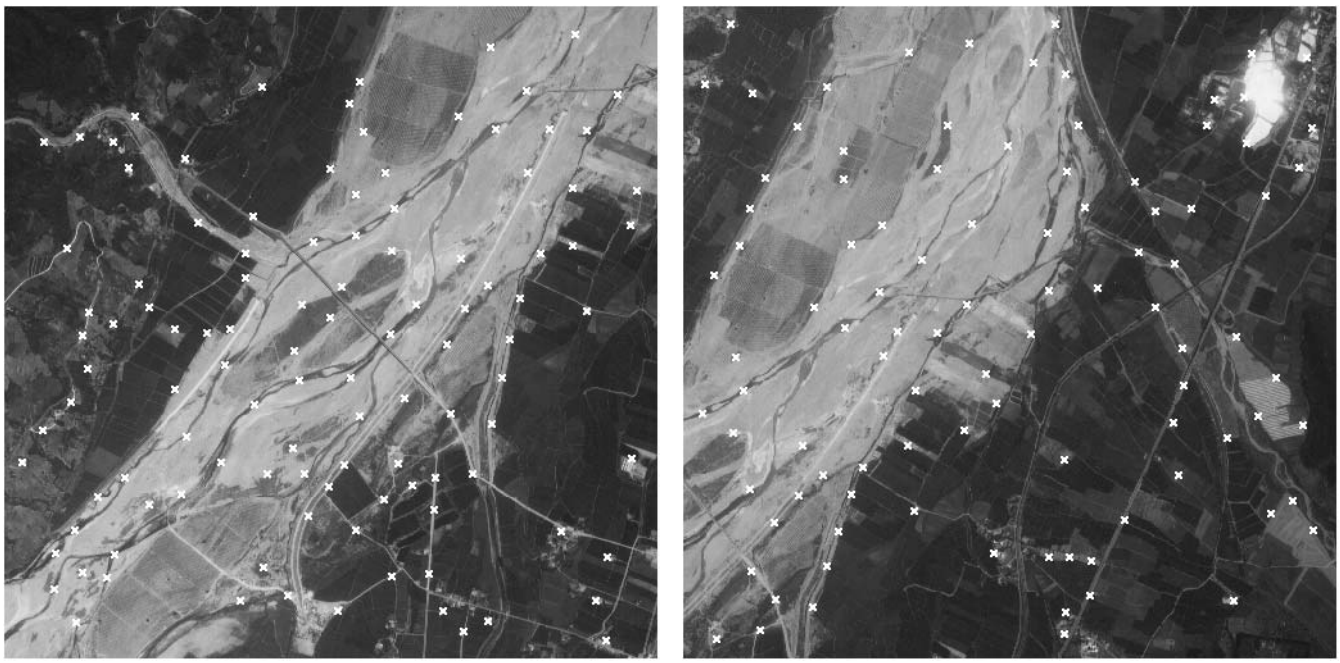
Figure 6. Two examples with significant scene changes. (a) and (b) are the building image pair. (c) and (d) are the mountain image pair. For both image pairs, poor estimation for the initial rotation angles are produced by adopting Zheng–Chellapa's method.

(a) (b)



(c)

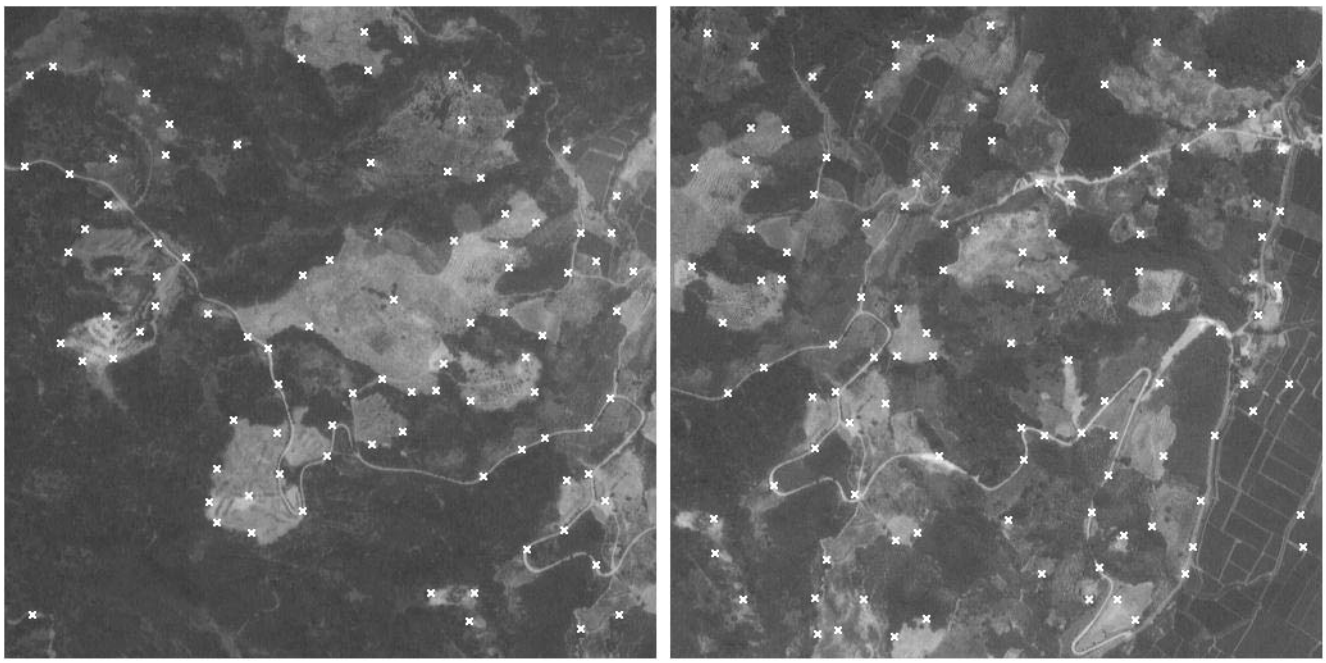
Figure 7. (a), (b) Two aerial images of rural areas of Taiwan; each small white *St*. Andrew cross indicates an extracted feature point. (c) The registration result of (a) and (b). The estimated registration parameters are s= 1.061, θ= –41.205 $^{\circ}$, t_x= –141.444, and t_y=–239.9.



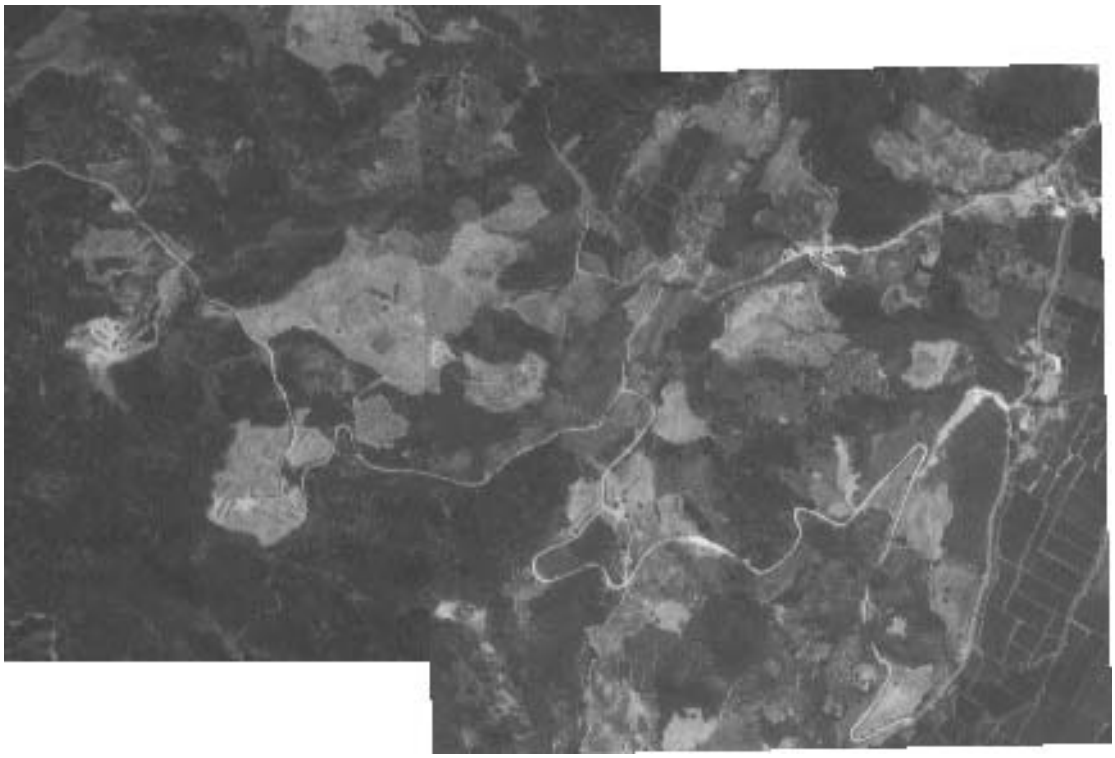
(a) (b)

(C)

Figure 8. (a), (b) Two aerial images of urban areas of Taiwan. (c) The registration result of (a) and (b). The estimated registration parameters are s= 0.963, θ=–6.052 $^{\circ}$, t_x= –271.135, and t_y=134.781.

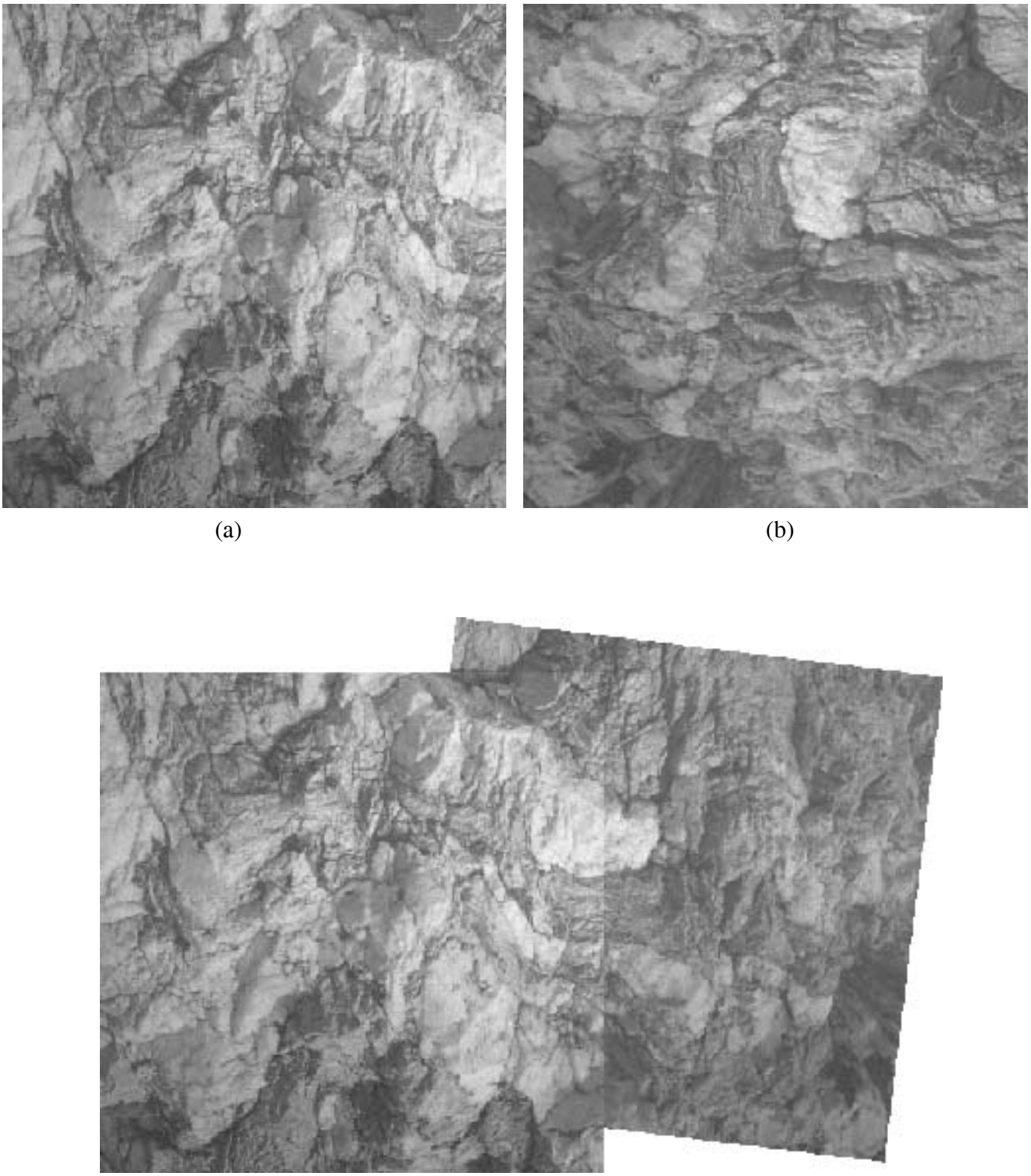


(a) (b)



(c)

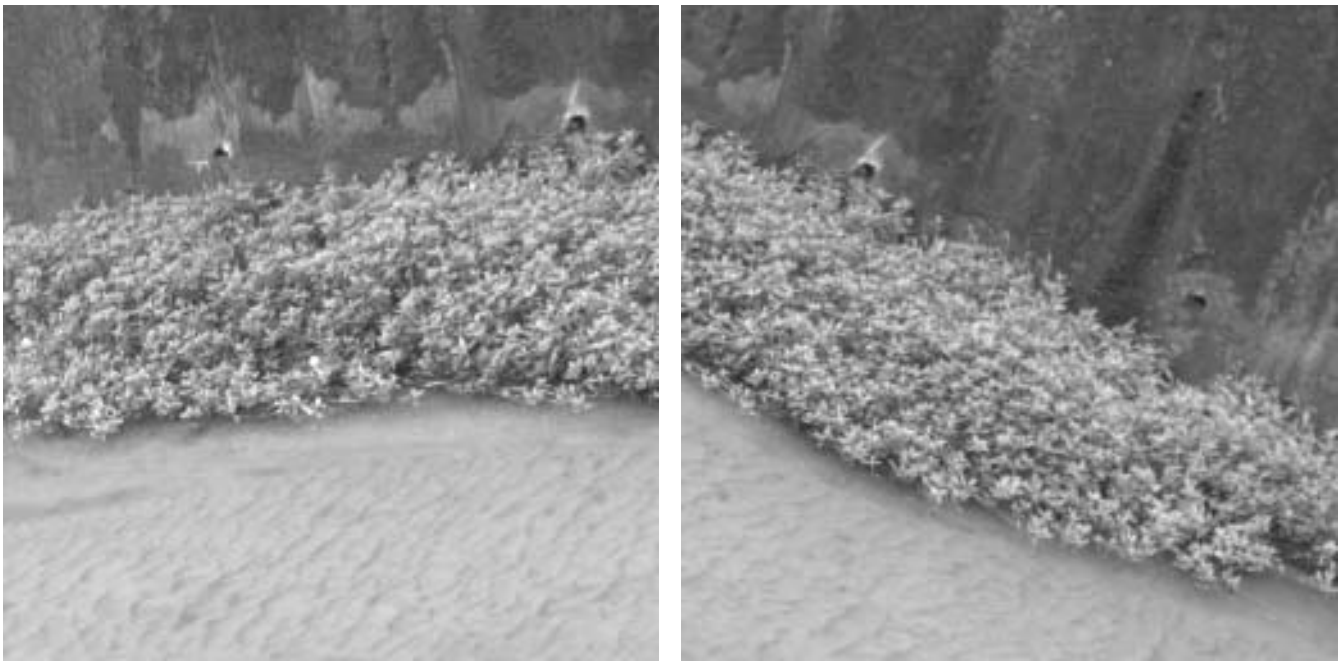
Figure 9. (a), (b) Two additional aerial images of urban areas of Taiwan. (c) The registration result of (a) and (b). The estimated registration parameters are s= 0.967211, θ=–1.08 $^{\circ}$, t_x= –325.310, and t_y=–63.532.



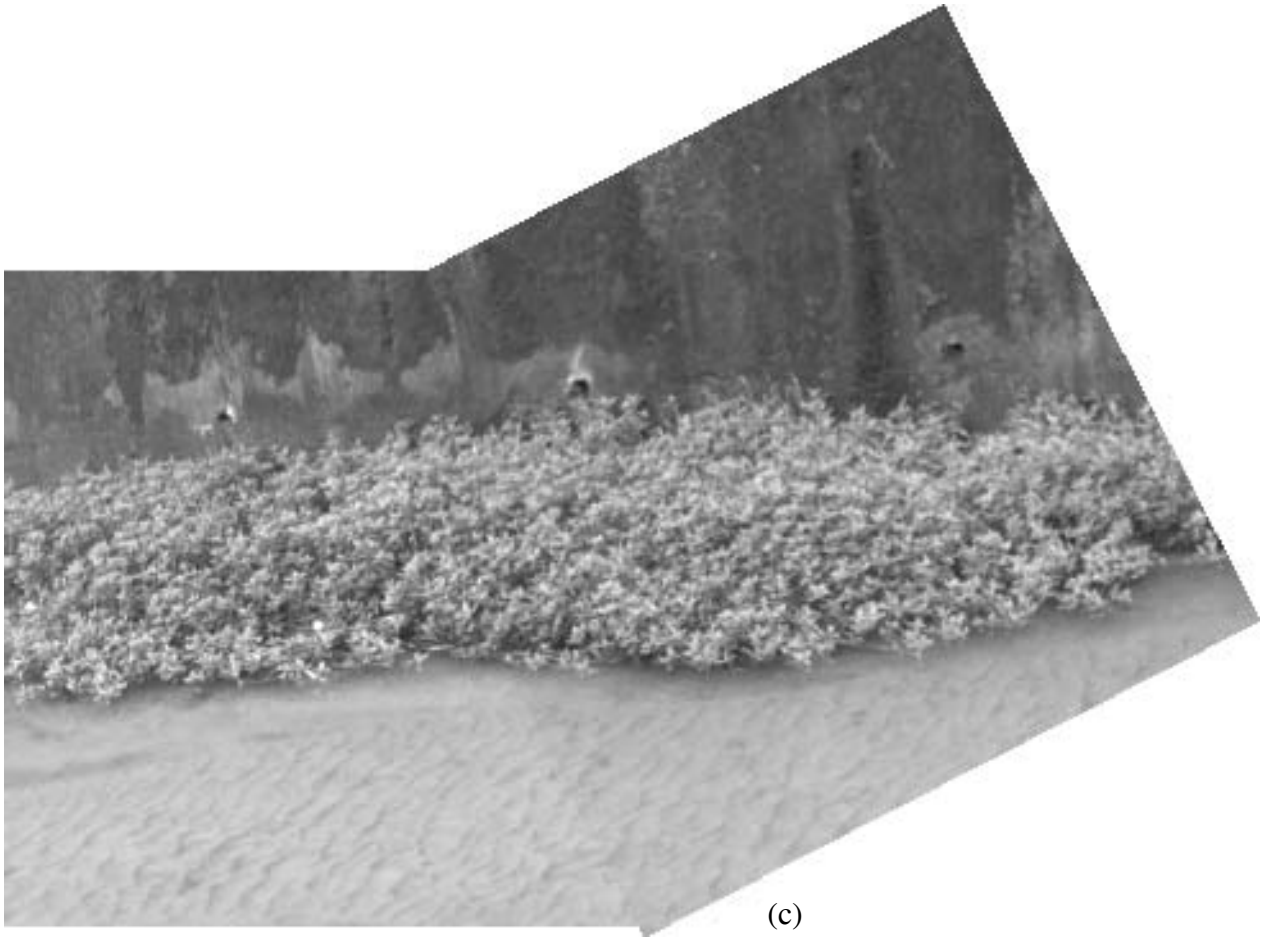
(a) (b)

(c)

Figure 10. Registration of rock images. (a), (b) Two rock images. (c) The registration result of (a) and (b). The estimated registration parameters are s= 1.04161, θ=–83.054 $^{\circ}$, t_x= –65.089, and t_y=–277.678.



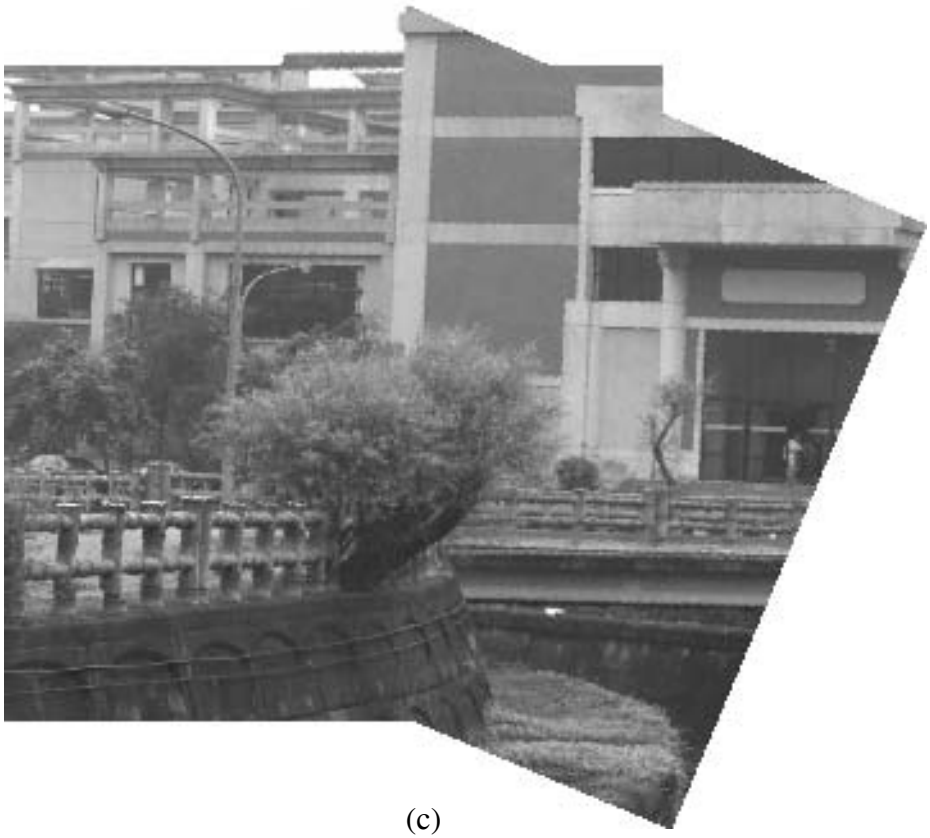
(a) (b)



(c)

Figure 11. Registration of texture images with grass and sand. (a), (b) Two grass–sand images. (c) Mosaic of (a) and (b). The estimated registration parameters are s= 0.947988, θ=–27.218 $^{\circ}$, t_x= –345.254, and t_y=–71.719.

(a) (b)



(c)

Figure 12. Estimation of camera motion. (a), (b) Two building images. (c) Mosaic of (a) and (b). The estimated transform parameters are s= 0.998617, θ=22.849 $^{\circ}$, t_x= –117.748, and t_y=43.7386.