Moving Object Detection and Mosaic Construction by Image Stitching

Jun-Wei Hsieh
Department of Electrical Engineering
YuanZe University, Taoyuan, Taiwan.
Tel:886-3-463-8800 Ext. 430 Fax:886-3-463-9355
shieh@saturn.yzu.edu.tw

Abstract

This paper proposes a novel edge-based stitching method to detect moving objects and construct mosaics from images. The method is a coarse-to-fine scheme which first estimates a good initialization of camera parameters with two complementary methods and then refines the solution through an optimization process. The two complementary methods are the edge alignment and correspondence-based approaches, respectively. Since these two methods are complementary to each other, the desired initial estimate can be obtained more robustly. After that, a Monte-Carlo style method is then proposed for integrating these two methods together. Then, an optimization process is applied to refine the above initial parameters. Since the found initialization is very close to the exact solution and only errors on feature positions are considered for minimization, the optimization process can be very quickly achieved. Experimental results are provided to verify the superiority of the proposed method.

Keywords: Image registration, image-based rendering, mosaics, moving object detection, and video retrieval.

1. Introduction

Image stitching is the process of recovering the existing camera motions between images and then compositing them together. This technique has been successfully applied to many different applications like video compression [1], video indexing [2]-[3], object tracking [9], or creation of virtual environments [5]-[8], [12]. For example, Shum and Szeliski [7]-[8] proposed methods to stitch a set of images together to form a panorama. In addition, Irani and Anandan [2] used this technique to represent and index different video contents. Moreover, Jin *et al.* [9] used this technique to compensate unwanted camera motions for extracting desired objects from video sequences. For most methods in this field, an affine camera model is used to approximate the possible motions between two consecutive frames. Then, the parameters of this model can be recovered from pair of images by two common methods, i.e., the correlation-based approach and the optimization-based one. For the first approach, the measure "correlation" can be calculated in frequency domain or spatial domain. For example, Kuglin and Hines [4] presented a phase-correlation method to estimate the displacement between two adjacent images in frequency domain. For the approaches in spatial domain, Sawhney and Ayer [1] proposed a feature matching approach to estimate the dominant and multiple camera motions. In addition, Zoghlami *et al.* [11] proposed a corner-based approach to build a set of correspondences for computing the transformation parameters from pair of images. However, the establishment of good correspondences is a challenging work when images have nonlinear intensity changes [17]. In order to avoid this problem, some researchers [6], [13] treated the stitching problem as a global optimization problem. For example, Szeliski [6] proposed a nonlinear minimization algorithm for automatically registering images by minimizing the discrepancy in intensities between images. In addition, Davis [13] proposed an optimization scheme to obtain a least-square solution by globally optimizing all pair-wise registrations. In comparison with the correlation-based method, the global optimization approach performs more robustly but will be trapped on a local minimum if the starting point is not properly initialized.

In this paper, we present an edge-based stitching technique to detect moving objects and construct mosaics from consecutive images. In general, the transformations between consecutive images can be described by a planar perspective motion model. Since the transformation is non-linear, the paper uses a coarse-to-fine approach to robustly and accurately recover the desired model parameters. Firstly, at the coarse stage, two complementary methods, i.e., the edge alignment and the correspondence-based approaches, are proposed to get respective initial estimates from images. Then, at the fine stage, the found initial estimate can be further refined through an optimization process. The edge alignment method finds possible image translations by checking the consistencies of edge positions between different images. It is simple and efficient without involving any optimization process or building any correspondences. In addition, the method has better capabilities to overcome large displacements and lighting variations between images. On the other hand, the correspondence-based method obtains desired model parameters from a set of correspondences by using a new feature extraction and a new correspondence building method. Especially, when building correspondences, a new measure is defined to measure the goodness of each match such that all false correspondences between features can be eliminated as well as possible. Compared with the edge alignment method, the correspondence-based approach can solve more general camera motion model but fails to work when images have large lighting changes. Therefore, due to the complementary property of the two methods, we can obtain the desired initial estimate more robustly. After that, a Monte-Carlo style method with gird partition is then proposed to integrate these methods together. The grid partition scheme can much enhance the accuracy of each try for deriving the correct parameters. Then, the found parameters are further refined through an optimization process. Sine the minimization is only applied to the positions of matching pairs, the optimization process can be performed very efficiently. From experimental results, the proposed method indeed achieves great improvements in terms of stitching accuracy, robustness, and stability.

The rest of the paper is organized as follows. In the next section, we will present an affine model to approximate the motions of a video camera. Then, details of the proposed method for recovering the parameters of this model are described in Section III. Section IV reports the experimental results. Finally, conclusions will be presented in Section V.

2. Camera Motion Model

Assume that input images are captured by a video camera. Then, the relationship between two adjacent images can be described by an affine camera model:

$$x' = \frac{m_0 x + m_1 y + m_3}{m_6 x + m_7 y + 1} \quad \text{and} \quad y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}, \quad (1)$$

where (x, y) and (x', y') are a pair of pixels in the two adjacent images I_0 and I_1, and $M = (m_0, m_1, ..., m_7)$ the motion parameters. The M can be obtained by minimizing the error function $E(M)$ as follows:

$$E(M) = \sum_i [I_1(x_i', y_i') - I_0(x_i, y_i)]^2 = \sum_i e_i^2, \quad (2)$$

where $e_i = I_1(x_i', y_i') - I_0(x_i, y_i)$. Then, Szeliski[7] gave the solution M with the iterative form:

$$M^T \leftarrow M^T + \Delta M^T, \quad (3)$$

where $\Delta M^T = (A + l I)^{-1} B$, $A = [a_{kn}]$, $B = [b_k]$, $a_{kn} = \sum_i \frac{\partial e_i}{\partial m_k} \frac{\partial e_i}{\partial m_n}$, $b_k = \sum_i e_i \frac{\partial e_i}{\partial m_k}$, and l is a coefficient obtained by the Levenber-Marquardt method [18]. The method works well if the initial value to the correct M is close enough. However, it suffers from low convergence and gets trapped in local minimum if the initialization is not proper, especially when images have large displacements. It is noticed that Eq.(3) tries to find desired solutions by minimizing intensity errors of all pixels between images. When the number of iterations increases, the calculation of intensity errors will become very time-consuming. Therefore, in what follows, we will propose a fast edge-based algorithm for tacking all the above problems.

3. Fast Algorithm for Mosaic Construction and Object Detection

In this paper, a coarse-to-fine approach is proposed to guide the optimization process. First, an edge-based approach is proposed to find a good initial estimate and then the initial result is refined through an optimization process. The initial estimate is got from two complementary methods, i.e., the edge alignment and the correspondence-based approaches. Since the two methods are complementary to each other, the robustness of the whole process of parameter estimation can be much enhanced. After that, a Monte Carlo style method is then used to integrate the above solutions together. For accuracy consideration, the found parameters can be further refined with an optimization process, which minimizes errors only on the coordinates of feature points. Sine the number of feature points is much smaller than the whole image, the optimization process can be performed extremely efficiently. The overall flowchart of the proposed approach is described in Fig. 1. In what follows, details of each proposed algorithm are described.

3.1 Translation Estimation Using Edge Alignment

As described in Fig. 1, for the purpose of robustness, we propose two strategies to find different initializations fed into an optimization process for deriving the correct model parameters. In this section, the edge alignment method is first proposed to estimate desired model translations from images. Let $g_x(p)$ be the gradient of a pixel p in the x direction of an image I, i.e.,

$$g_x(p(i,j)) = |I(p(i+1,j)) - I(p(i-1,j))|,$$

where $I(p)$ is the intensity of p. In addition, let $S_g(i)$ denote the sum of $g_x(p)$ obtained by accumulating $g_x(p)$ along pixels in the ith column, i.e.,

$$S_g(i) = \frac{1}{H}\sum_j |I(p(i+1,j)) - I(p(i-1,j))|,$$

where H is the height of I. If $S_g(i)$ is larger than a threshold, i.e., 15, the ith column is considered to have a vertical edge. After checking all pixels of input images column by column, a set of positions of vertical edges can be found.

Assume I_a and I_b are two images prepared to be stitched and shown in Fig. 2 (a) and (b), respectively. Through the above vertical edge detector, the positions of vertical edges in I_a and I_b can be obtained as P_a^v = (100, 115, 180, 200, 310, 325, 360, 390, 470) and P_b^v = (20, 35, 100, 120, 230, 245, 280, 310, 390), respectively. If the images I_a and I_b come from the same static scene, there should exist an offset d_x such that $P_a^v(i) = P_b^v(j) + d_x$ and the corresponding relation between i and j is one-to-one. Then, the offset d_x is the desired translation solution between I_a and I_b in the x direction, i.e., d_x = 80. Based on this idea, in what follows, a novel method will be proposed to estimate desired translation parameters from images without building any correspondences or involving any optimization processes.

Before describing the proposed method, we shall know in practice due to noise, some edges will be lost or undetected. The lost or undetected edges will lead to that the relations between P_a^v and P_b^v are no longer one-to-one. For this problem, this paper defines a distance function $d_v(i,k)$ to measure the distance of a position $P_a^v(i)$ to the translation solution k as

$$d_v(i,k) = \min_{1\le j\le N_b^v} |P_a^v(i) - k - P_b^v(j)|, \quad (4)$$

where N_b^v is the number of elements in P_b^v. Let T_d denote a threshold and set to be 4. Given a number k, we want to determine the number N_p^v of elements in P_a^v whose $d_v(i,k)$ is less than T_d. In addition, we denote the average value of $d_v(i,k)$ for these N_p^v elements as E_k^v, which can be used as an index to measure the goodness of k to see whether it is a suitable translation solution. If E_k^v is smaller enough and N_p^v is larger enough, the position k will be a good horizontal translation. More precisely, if $E_k^v \le T_e$ and $N_p^v \ge T_p$, the k is collected as an element of the set S_x of possible horizontal translations, where the two thresholds T_p and T_e are set to be 5 and 2, respectively. Let W_b denote the width of the input image I_b. Through examining different k for all $|k| < W_b$, the set S_x can be obtained.

On the other hand, let P_a^h and P_b^h denote as the sets of horizontal edge positions in I_a and I_b, respectively. With P_a^h and P_b^h, we can define a distance function d_h as follows:

$$d_h(i,k) = \min_{1\le j\le N_b^h} |P_a^h(i) - k - P_b^h(j)|, \quad (5)$$

where N_b^h is the number of elements in P_b^h. Let H_b denote the height of the input image I_b. According to d_h, with the similar method to obtain S_x, by examining different k for all $|k| < H_b$, the set S_y of possible vertical translations can be obtained. With S_x and S_y, the set S_{xy} of possible translations can be obtained as: $S_{xy} = \{(x,y) \mid x \in S_x, y \in S_y\}$. The best translation can be then determined from S_{xy} through a correlation technique. Details of the whole algorithm are summarized as follows.

Edge-based Translation Estimation Algorithm:

I_a and I_b: two adjacent images prepared to be stitched.

S1: Apply a vertical edge detector to find the sets P_a^v and P_b^v of vertical edge positions from I_a and I_b, respectively.

S2: Determine the set S_x of possible horizontal translations from P_a^v and P_b^v based on $d_v(i,k)$ (see Eq.(4)).

S3: Apply a horizontal edge detector to find the sets P_a^h and P_b^h of horizontal edges from I_a and I_b, respectively.

S4: Determine the set S_y of possible vertical translations from P_a^h and P_b^h based on $d_h(i,k)$ (see Eq.(5)).

S5: Let S_{xy} denote the set of possible translations, i.e., $S_{xy} = \{(x,y) \mid x \in S_x, y \in S_y\}$.

S6: Determine the best solution (t_x, t_y) from S_{xy} through a correlation technique.

3.2 Motion Parameter Estimation by Feature Matching

As described in Fig. 1, two strategies are used to find respective initial estimates of camera parameters for further optimization process. In this section, details of the correspondence-based method are described. In Section 3.2.1, we will propose a new method to extract a set of useful feature points from images based on edges. Then, details of building correspondences between features are described in Section 3.2.2. However, due to noise, many false matches will also be generated. In Section 3.2.3, a new scheme is proposed to eliminate all impossible false matches.

3.2.1 Feature Extraction

In this section, we will use several edge operators to extract a set of useful feature points. First of all, let the gradients of an image *I(x,y)* at scale s in the x and y directions be denoted as $I_x^s(x,y) = I * G_x^s(x,y)$ and $I_y^s(x,y) = I * G_y^s(x,y)$, where G_x^s and G_y^s are the first partial derivatives of a 2D Gaussian smoothing function $G^s(x,y)$ in the x and y directions, respectively, where s is a standard deviation. Then, the modulus of I_x^s and I_y^s is defined as:

$$|\nabla I^s(x,y)| = \sqrt{|I_x^s(x,y)|^2 + |I_y^s(x,y)|^2}\ .$$

Since we are interested in some specific feature points for image stitching, additional constraints have to be introduced. In what follows, two conditions adopted here for judging whether a point *P(x,y)* is a feature point or not are summarized as follows:

C1: *P(x,y)* must be an edge point of the image *I(x,y)*. This means that *P(x,y)* is a local maxima of $M_2I(x,y)$, and $M_2I(x,y)$ > a threshold;

C2: $|\nabla I^s(x,y)|_{s=2} = \max_{(x',y')\in N_p} \{|\nabla I^s(x',y')|_{s=2}\}$, where N_p is a the neighborhood of *P(x,y)* within an 13×13 window.

3.2.2 Correspondence Establishment

In the previous section, we have described how the feature points between $I_a(x,y)$ and $I_b(x,y)$ are derived. Now, we are ready to find the matching pairs between I_a and I_b. Let $FP_{I_a} = \{p_i = (p_x^i, p_y^i)\}$ and $FP_{I_b} = \{q_i = (q_x^i, q_y^i)\}$ be two sets of feature points extracted from two images I_a and I_b, respectively. In addition, N_{I_a} and N_{I_b} represent the number of elements in FP_{I_a} and FP_{I_b}, respectively. The similarity between two feature points p and q is measured by their normalized cross-correlation denoted as $C(p,q)$. For each point p_i in FP_{I_a}, find the maximum peak of the similarity measure as its best matching point q in another image I_b. Then, a pair $\{p_i \Leftrightarrow q_i\}$ is qualified as a matching pair if two conditions are satisfied:

$$C(p_i,q_i) = \max_{q_k \in FP_{I_b}} C(p_i,q_k) \text{ and } C(p_i,q_i) \geq T_c, \quad (6)$$

where $T_c = 0.75$. The first condition enforces to find a feature point $q_k \in FP_{I_b}$ such that the measure C_{I_a,I_b} is maximized. As for Condition 2, it forces the value C_{I_a,I_b} of a marching pair to be larger than a threshold (0.75 in this case).

3.2.3 Eliminating False Matches

In the previous section, through matching, a set of matching pairs has been extracted. However, if the relative geometries of features are considered, the matching results can be refined more accurately. Let $MP_{I_a,I_b} = \{p_i \Leftrightarrow q_i\}_{i=1,2...}$ be the set of matching pairs, where p_i is an element in FP_{I_a} and q_i another element in FP_{I_b}. Let $Ne_{I_a}(p_i)$ and $Ne_{I_b}(q_i)$ be the neighbors of p_i and q_i within a disc of radius *R*, respectively. Assume that $NP_{p_iq_j} = \{n_k^1 \Leftrightarrow n_k^2\}_{k=1,2....}$ is the set of matching pairs, where $n_k^1 \in Ne_{I_a}(p_i)$, $n_k^2 \in Ne_{I_b}(q_j)$, and all elements of $NP_{p_iq_j}$ belong to MP_{I_a,I_b}. The proposed method is based on a concept that if $\{p_i \Leftrightarrow q_i\}$ and $\{p_j \Leftrightarrow q_j\}$ are two good matches, the relation between p_i and p_j should be similar to the one between q_i and q_j. Based on this assumption, we can measure the goodness of a matching pair $\{p_i \Leftrightarrow q_i\}$ according to how many matches $\{n_k^1 \Leftrightarrow n_k^2\}$ in $NP_{p_i q_i}$ whose distance $d(p_i, n_k^1)$ is similar to the distance $d(q_j, n_k^2)$, where $d(u_i, u_j) = \| u_i - u_j \|$, the Euclidean distance between two points u_i and u_j. Then, the measure of goodness for a match $\{p_i \Leftrightarrow q_i\}$ can be defined as:

$$G_{I_a I_b}(i) = \sum_{\{n_k^1 \Leftrightarrow n_k^2\} \in NP_{p_i q_i}} \frac{C(n_k^1, n_k^2) r(i,k)}{1 + dist(i,k)},$$

where $dist(i,k) = [d(p_i, n_k^1) + d(q_i, n_k^2)]/2$, $C(n_k^1, n_k^2)$ the correlation measure between n_k^1 and n_k^2, $r(i,k) = e^{-m(i,k)/T_1}$, with a threshold T_1, and $u(i,k) = |d(p_i, n_k^1) - d(q_i, n_k^2)| / dist(i,k)$. The contribution of a pair $\{n_k^1 \Leftrightarrow n_k^2\}$ in $NP_{p_i q_i}$ monotonically decreases based on the value of *dist(i, k)*.

After calculating the goodness of each pair $\{p_i \Leftrightarrow q_i\}$ in MP_{I_a, I_b}, we can obtain their relative goodness $G_{I_a I_b}(i)$ for further eliminating false matches. Assume $\bar{G}$ is the average value of $G_{I_a I_b}(i)$ for all matching pairs. If the value of $G_{I_a I_b}(i)$ is less than $0.75\,\bar{G}$, the matching pair $\{p_i \Leftrightarrow q_i\}$ is eliminated.

3.3 Motion Parameter Estimation Using Monte Carlo Method

In this section, a Monte-Carlo-style method is proposed for integrating these methods together for further optimization process. The spirit of the Monte Carlo method is to use many tries to find (or hit) the wanted correct solution. Assume each try can generate a solution and the probability to find a correct solution for each try is *r*. After *k* tries, the probability of continuous failure to find a correct solution is $s = (1-r)^k$. Clearly, even though *r* is very small, after hundreds or thousands of tries, *s* will tend very closely to zero. In other words, if we define a try as a random selection of four matching pairs, each try will generate a solution. Then, it is expected that a correct solution *M* will be obtained after hundreds or thousands of tries.

As we knew, for each try, four matching pairs will be selected to obtain a possible solution. If MP_r has N_r elements and N_c ones are correct, the probability to select four correct pairs for each try will be $\frac{N_c(N_c-1)(N_c-2)(N_c-3)}{N_r(N_r-1)(N_r-2)(N_r-3)}$. In what follows, a method is proposed to improve the probability for each try to find a correct solution by separating images into grids. Assume all the correct and false matching pairs distribute very randomly. Then, if the input images are segmented into several grids, in each grid the probability to select a correct matching pair is still N_c / N_r. Therefore, we can select four different girds first and then get one matching pair from each grid. With this method, the probability to select four correct matching pairs will become N_c^4 / N_r^4. Clearly, $\frac{N_c(N_c-1)(N_c-2)(N_c-3)}{N_r(N_r-1)(N_r-2)(N_r-3)} < N_c^4 / N_r^4$ if $N_c < N_r$. Thus, the suggested method can better enhance the hit rate of finding four correct matching pairs to derive desired parameters.

On the other hand, since the Monte Carlo method uses lots of tries to find final desired solutions, we should propose a verification process to determine which try is the best. Assume $M^i = (m_0^i, m_1^i, ..., m_7^i)$ is the solution got from the *i*th try. The verification process can be achieved by comparing how many matching pairs in MP_r are consistent to M^i. Let $\{p \leftrightarrow q\}$ be a matching pair and the consistent error $e(p,q,M^i)$ of this pair to M^i be:

$$e(p,q,M^i) = \sqrt{\left(q^x - \frac{m_0^i p^x + m_1^i p^y + m_2^i}{m_6^i p^x + m_7^i p^y + 1}\right)^2 + \left(q^y - \frac{m_3^i p^x + m_4^i p^y + m_5^i}{m_6^i p^x + m_7^i p^y + 1}\right)^2}. \quad (7)$$

For each matching pair $\{p_k \leftrightarrow q_k\}$ in MP_r, if $e(p_k, q_k, M^i) < T_e$, the pair $\{p_k \leftrightarrow q_k\}$ is said to be consistent to M^i, where T_e is a threshold set to 6 for the consistency check. Based on Eq.(7), a counter $c(M^i)$ is used to record how many matching pairs in MP_r which are consistent to M^i. After several tries, the best solution $\bar{M}$ can be obtained as follows:

$$\bar{M} = \arg\max_{M^i} c(M^i). \quad (8)$$

When initialization (*i*=0), M^0 is got from the edge alignment approach (see Section 3.1).

3.4 Parameter Refinement through Optimization

With the Monte Carlo method, the best estimate

$\bar{M}$ can be found from MP_r. However, if an optimization process is applied, $\bar{M}$ can be further refined. In Section 3.2, two sets of feature points, i.e., FP_{I_a} and FP_{I_b}, have been extracted from the images I_a and I_b, respectively. For each point p_i in FP_{I_a} and q_j in FP_{I_b}, according to Eq.(7) and $\bar{M}$, if $e(p_i, q_j, \bar{M}) < T_e$, we denote $\{p_i \Leftrightarrow q_j\}$ as a new match. Then, after checking all elements in FP_{I_a} and FP_{I_b}, a new set $MP_{\bar{M}}$ of matching pairs can be obtained:

$$MP_{\bar{M}} = \{p_k \Leftrightarrow q_k, k = 1,2,...,N_{\bar{M}}\},$$

where $e(p_k, q_k, \bar{M}) < T_e$, $p_k \in FP_{I_a}$, and $q_k \in FP_{I_b}$. Then, we can define an error function as:

$$\Phi(\bar{M}) = \sum_{k=1}^{N_{\bar{M}}} e(p_k, q_k, \bar{M}), \quad (9)$$

where $\{p_k \Leftrightarrow q_k\}$ is an element in $MP_{\bar{M}}$. By calculating the gradient and Hessian matrix of Φ, $\bar{M}$ can be updated with the iterative form:

$$\bar{M}_{t+1}^T = \bar{M}_t^T + (A + l)^{-1} B, \quad (10)$$

where $[A]_{ij} = \sum_{k=1}^{N_{\bar{M}}} \frac{\partial e_k}{\partial \bar{m}_i} \frac{\partial e_k}{\partial \bar{m}_j}$, $[B]_i = \sum_{k=1}^{N_{\bar{M}}} e_k \frac{\partial e_k}{\partial \bar{m}_i}$, t is the iteration number, and l is a coefficient obtained by the Levenber-Marquardt method [18]. The above minimization process quickly converges since only the coordinates of feature positions are considered into minimization and the initial estimate of $\bar{M}$ is very close to the final solution.

4. Experimental Results

In order to analyze the performance of the proposed method, a series of real images were adopted as test images. Fig. 3 shows the result for mosaic construction when a series of panoramic images are used. In this case, before stitching, all the images are projected into a cylindrical map [5]. Then, only the translation parameters need to be estimated. Fig. 4 shows the case when images have larger intensity differences. (a) and (b) are the original images and (c) is the stitching result. The large lighting changes will lead to the instability of feature matching in the traditional matching techniques like block matching or phase correlation techniques. However, in this paper, the proposed edge alignment algorithm tries to find all possible translations by checking the consistence of edge positions instead of comparing the intensity similarity of images. Therefore, even though images have larger lighting changes, the proposed method still works well to find all desired camera parameters for stitching. Fig. 5 shows the case when images have some moving objects. The moving object will disturb the work of image stitching. However, the proposed method still successfully stitches them together. Fig. 6 shows the result when images have some rotation and skewing effects. In this case, the proposed Monte Carlo method still works well to find the correct camera parameters.

The proposed method also can be used in camera compensation for extracting moving objects from video sequence. Fig. 7 shows two frames got from a movie. In order to detect the moving object, a static background should be constructed. With the proposed method, the camera motion between Fig. 7(a) and Fig. 7(b) can be well found and compensated. Fig. 7(c) is the mosaic of Fig. 7(a) and (b). Then, the moving object can be detected by image differencing like Fig. 7(d). The detection result is very useful for various applications like intelligent transportation system, video indexing, video surveillance, and etc. Fig. 8 is another case when a moving car appears in the video sequence. From the experimental results, it is obvious that the proposed method is indeed an efficient, robust, and accurate method for image stitching.

5. Conclusions

In this paper, we have proposed an edge-based method for stitching series of images from a video camera. In this approach, for robustness consideration, the initial estimate is estimated from two different schemes, i.e., the edge alignment approach and the correspondence-based one. Since the two methods are complementary to each other, much robustness can be gained during the parameter estimation process. To integrate these two methods together, a Monte-Carlo style method is proposed to find the best motion parameters. Then, the solution is refined through an optimization process. The contributions of this paper can be summarized as follows:

(a) This paper proposed an edge alignment scheme for estimating translation parameters using edges. The method has better capabilities to overcome the problems of large displacements and lighting changes between images.

(b) A new feature extraction scheme was proposed to extract a set of useful features.

(c) When building correspondences, a new scheme was proposed to eliminate many false matches by judging the goodness of a matching pair. Through the judgment, a set of desired correspondences can be obtained more reliably.

(d) A grid partition scheme was proposed to enhance the hit rate of obtaining four correct matching pairs. Then, the correct parameters can be found with less tries.

(e) An efficient optimization process was proposed for refining the estimated parameters more accurately. Since only the errors on feature positions are considered, the minimization process can be performed extremely efficiently.

Experimental results have shown our method is superior in terms of stitching accuracy, robustness, and stability.

Acknowledges

This work was supported in part by National Science Council of Taiwan under Grant NSC91-2213-E-150-019, Taiwan.

References

[1] H. Sawhney and S. Ayer, "Compact representation of video through dominant and multiple motion estimation," *IEEE trans. Pattern Anal. Machine Intell.*, vol. 18, 814-830, Aug. 1997.

[2] M. Irani and P. Anandan, "Video indexing based on mosaic representation," *Proc. IEEE*, vol.86, pp. 905-921, May, 1998.

[3] M. Bonnet, "Mosaic representation for video shot description," *Proc. MPEG-7 Evaluation Ad Hoc Meeting*, pp. 636, Feb. 1999.

[4] C. Kuglin and D. Hines, "The Phase Correlation Image Alignment Method," *Proc. of the IEEE Int. Con. on Cybernetics and Society*, pp.163-165, 1975.

[5] S. Chen, "Quicktime VR-an image-based approach to virtual environment navigation," *Proc. SIGGRAPH '95*, pp.29-38, 1995.

[6] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Computer Graph and Application*, vol. 16, pp. 22-30, March 1996.

[7] H. Y. Shum and R. Szeliski, "Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment,"*International Journal of Computer Vision*, vol. 36, no. 2, pp. 101-130, 2000.

[8] R. Szeliski and H. Y. Shum, "Creating full view panoramic image moaics and environment maps," *Proc. Computer Graphics Annu. Conf. Series*, pp. 251-259, 1997.

[9] J. S. Jin, Z. Zhu and G. Xu, "A stable vision system for moving vehicles", *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, No. 1, pp.32-39, 2000.

[10] H. Nicolas, "New methods for dynamic mosaicking," *IEEE trans. Image Processing*, vol. 10, no. 8, pp. 1239-1251, Aug. 2001.

[11] I. Zoghlami, O. Faugera, and R. Deriche, "Using geometric corners to build a 2D mosaic from a set of images," *Proc. Conf. Computer Vision and Pattern Recognition*, Puerto Rico, pp.420-425, 1997.

[12] C.T. Hsu et al., "Feature-based video mosaic," *Proceedings of ICIP 2000*, vol.2, pp.887-890, Vancouver, Canada, Sep. 2000.

[13] J. Davis, "Mosaics of scenes with moving objects," *IEEE Proc. CVPR* 1998.

[14] C. Guestrin, F. Cozman, and E. Krotkov, "Fast Software Image Stabilization with Color Registration," *In Proceedings of Intelligent Robots and Systems Conference*, pp.19-24, Victoria, Canada, October 1998.

[15] S. B. Kang, "A survey of image-based rendering techniques," Technical Report 97/4, Digital Equipment Corporation, Cambridge Research Lab.

[16] J. W. Hsieh, H. Y. Mark Liao, K. C. Fan, M. T. Ko, and Y. P. Hung, "Image Registration Using a New Edge-based Approach," *Computer Vision and Image Understanding*, 67, 112-130, 1997.

[17] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis and Machine Vision, London, U. K.: Chapman & Hall, 1993.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in C: the Art of Scientific Computing, Cambridge University Press.

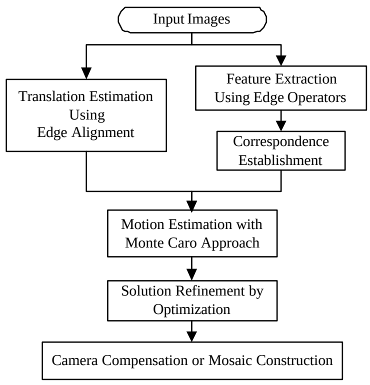
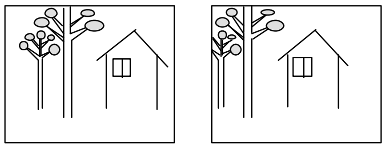


Fig. 1: Flowchart of the proposed method.



(a) (b)

Fig. 2 Edge results of two images.

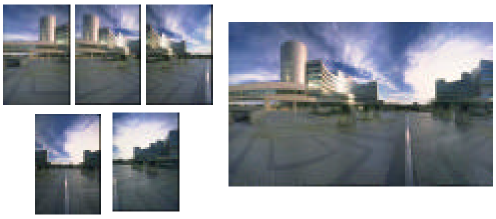


Fig. 3: Stitching result of a series of panoramic images.

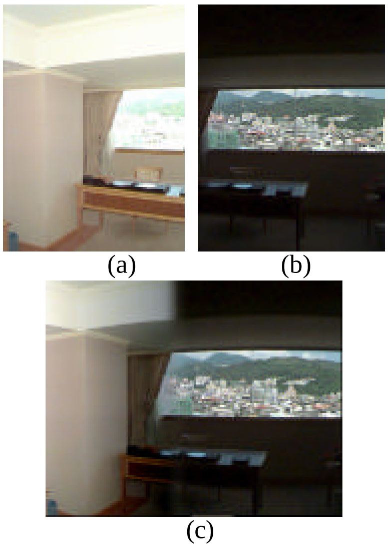


Fig. 4: Stitching result of two images with larger lighting changes. (c) is the stitching result of (a) and (b)

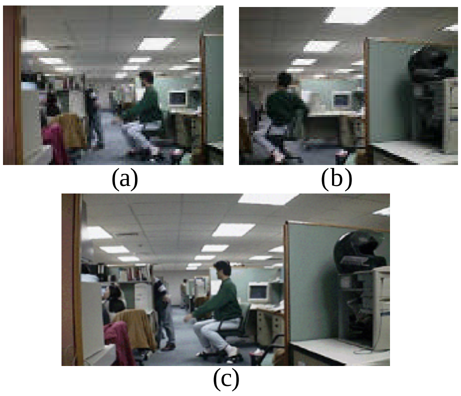


Fig. 5: Stitching result when images have moving objects. (a) and (b) Original Images. (c) Stitching result.

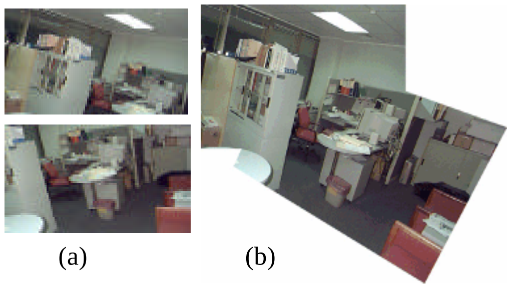


Fig. 6: Stitching result when the camera has rotation changes.

Fig. 7: Mosaic construction and object detection. (c) is the mosaic result of (a) and (b). (d) is the object detection result by image differencing.

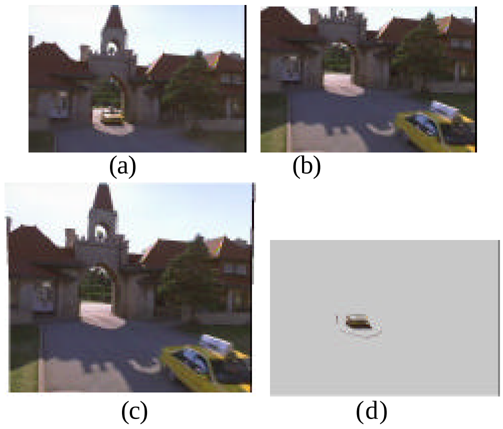


Fig. 8: Mosaics and object detection when images have a moving object. c) is the mosaic result of (a) and (b). (d) is the object detection result by image differencing.