Fast Stitching Algorithm for Moving Object Detection and Mosaic Construction

Jun-Wei Hsieh
Department of Electrical Engineering
YuanZe University, Taoyuan, Taiwan, R.O.C.
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.

1. Introduction

Image stitching is the process of recovering the existing camera motion parameters between images and then compositing them together. This technique has been successfully applied to many different applications like video compression [1], video indexing [2], or creation of virtual environments [3]. For example, Shum and Szeliski [3] proposed a method to stitch a set of images together for constructing a panorama. In addition, Irani and Anandan [2] used this technique to represent and index different video contents. For most methods in this field, an affine camera model is commonly used to approximate possible motions between two consecutive frames. Then, this model can be recovered by two common methods, i.e., the correlation-based approach and the optimization-based one. For example, Kuglin and Hines [4] presented a phase-correlation method to estimate the displacement between two adjacent images in frequency domain. In addition, Zoghlami *et al.* [5] proposed a corner-based approach to build a set of correspondences for computing possible transformation parameters from pair of images. However, the establishment of good correspondences is a challenging work when images have nonlinear intensity changes [3]. In order to avoid this problem, Szeliski [3] proposed a nonlinear minimization algorithm for automatically registering images by minimizing the discrepancy in intensities between images. 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 motion model between consecutive images is non-linear. This paper uses a coarse-to-fine approach to robustly and accurately recover this nonlinear model. In the coarse stage, two complementary methods, i.e., the edge alignment and the correspondence-based approaches, are first proposed to get respective initial estimates from images. Then, in the refined stage, the found initial estimates are 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, efficient, and has good capabilities in overcoming large displacements and light 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. Due to the complementary property of the two methods, we can obtain the desired initial estimates 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 is performed very efficiently. From experimental results, the proposed method indeed achieves great improvements in terms of stitching accuracy, robustness, and stability.

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:

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

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

0-7803-7965-9/03/$17.00 ©2003 IEEE

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

where $e_i=I_1(x_i^{'},y_i^{'})-I_0(x_i,y_i)$. Then, Szeliski[3] gave the solution M with the iterative form:

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

where $\Delta M^T=(A+\lambda I)^{-1}B$, $A=[a_{kn}]$, $B=[b_k]$, $a_{kn}=\sum_i \frac{\partial e_i}{\partial m_k}\frac{\partial e_i}{\partial m_n}$, $b_k=\sum_i e_i\frac{\partial e_i}{\partial m_k}$, and λ is a coefficient. The method works well only if the initial value is very close to the correct M. If two images have large displacements and lighting changes, it will suffer from low convergence and get trapped in local minimum.

3. Fast Algorithm for Camera Compensation and Mosaic Construction

In this paper, a coarse-to-fine approach is proposed for solving Eq.(1). In this approach, a good initial estimate is first found with edges and then 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 parameter estimation can be much enhanced. The former has better capabilities to overcome large displacements and light variations between images. The latter can solve more general camera motion model but fails to work when images have large lighting changes. After that, a Monte Carlo style method is then applied to integrate the above solutions together. After integration, the found parameters are further refined with an optimization process, which minimizes errors only on the coordinates of feature points. Since only feature points are involved, the optimization process can be performed extremely efficiently. The overall flowchart is described in Fig. 1.

3.1 Translation Estimation Using Edge Alignment

In this section, the edge alignment method is first proposed to estimate desired translations from pair of images. Assume I_a and I_b are two images prepared to be stitched (see Fig. 2). Through a vertical edge detector (very simple), 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 I_a and I_b come from the same 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. However, in practice, due to noise, some edges will be lost and lead to that the relations between P_a^v and P_b^v are no longer one-to-one. For this problem, we 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

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

where N_b^v is the number of elements in P_b^v. Let T_d be a threshold, i.e., 4. Given a number k, we can determine the number N_p^v of elements in P_a^v whose $d_v(i,k)$ is less than T_d. With these N_p^v elements, the average value E_k^v of $d_v(i,k)$ can also be calculated. E_k^v is a good index to measure the goodness of k whether it is a suitable translation solution. That is, 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 T_p and T_e are 5 and 2, respectively. Let W_b denote the width of I_b. Through examining different k for all $|k|<W_b$, the set S_x can be obtained. Similarly, the set S_y of possible vertical translations can be determined by taking advantages of horizontal edges. 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.

3.2 Parameter Estimation by Feature Matching

In what follows, details of the correspondence-based approach are described.

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 σ in the *x* and *y* directions be then as $I_x^\sigma(x,y)=I*G_x^\sigma(x,y)$ and $I_y^\sigma(x,y)=I*G_y^\sigma(x,y)$, where G_x^σ and G_y^σ denote the first partial derivatives of a 2D Gaussian smoothing function $G^\sigma(x,y)$ in the *x* and *y* directions, respectively, where σ is a standard deviation. Then, the modulus of I_x^σ and I_y^σ is defined as:

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

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)* is a local maxima of $|\nabla I^\sigma(x,y)|_{\sigma=2}$;

C2: $|\nabla I^\sigma(x,y)|_{\sigma=2} = \max_{(x',y')\in N_p}\{|\nabla I^\sigma(x',y')|_{\sigma=2}\}$,

where N_p is a the neighborhood of *P(x,y)*.

3.2.2 Correspondence Establishment

Let $FP_{I_a}=\{p_i\}_{i=1,\ldots,N_{I_a}}$ and $FP_{I_b}=\{q_i\}_{i=1,\ldots,N_{I_b}}$ be two sets of feature points obtained from adjacent images I_a and I_b, respectively. 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. In other words, a pair $\{p_i \Leftrightarrow q_i\}$ is qualified as a matching pair if $C(p_i, q_i) = \max_{q_k \in FP_{I_b}} C(p_i, q_k)$ and $C(p_i, q_i) \geq T_c$, where $T_c = 0.75$ and $C(p,q)$ is a cross correlation measure. Then, a set of matching pairs can be extracted. However, the set of matching pairs can be further refined if relative geometries of feature points are considered. In what follows, we will define a matching goodness measure and a relation constraint to refine the matching results.

Let $MP_{I_a, I_b} = \{p_i \Leftrightarrow q_i\}_{i=1,2...}$ be the set of matching pairs, where p_i and q_i are points in I_a and I_b, respectively. Let $Ne_{I_a}(p_i)$ and $Ne_{I_b}(q_i)$ denote the neighbors of p_i and q_i within a disc of radius R, respectively. Assume that $NP_{p_i q_j} = \{n_k^1 \Leftrightarrow n_k^2\}_{k=1,2,...}$ is the set of matching pairs, where $n_k^1 \in Ne_{I_a}(p_i)$ and $n_k^2 \in Ne_{I_b}(q_j)$. The proposed method works based on the 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. Then, we can measure the goodness of a matching pair $\{p_i \Leftrightarrow q_i\}$ according to how many matches $\{n_j^1 \Leftrightarrow n_j^2\}$ in $NP_{p_i q_i}$ whose distance $d(p_i, n_j^1)$ is similar to the distance $d(q_j, n_j^2)$, where $d(u_i, u_j)$ is the Euclidean distance between u_i and u_j. Then, the measure of goodness of $\{p_i \Leftrightarrow q_i\}$ can be defined as:

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

where $D(i,k) = [d(p_i, n_k^1) + d(q_i, n_k^2)]/2$, $r(i,k) = e^{-\mu(i,k)/T_1}$, with a threshold T_l, and $\mu(i,k) = |d(p_i, r_k^1) - d(q_i, r_k^2)| / D(i,k)$. Let $\bar{G}$ be the average value of all $G_{I_a I_b}(i)$. If $G_{I_a I_b}(i) < 0.75\,\bar{G}$, the pair $\{p_i \Leftrightarrow q_i\}$ is eliminated.

3.3 Estimation Using Monte Carlo Method

In this section, a Monte-Carlo-style method is proposed for integrating two above methods together for further optimization process. The spirit of the Monte Carlo method is to use many tries to find wanted correct solutions. If we define a try as a random selection of four matching pairs, each try will generate a solution by solving Eq.(1). Then, it can be expected that a correct solution M will be obtained after hundreds or thousands of tries. Assume all the correct and false matching pairs distribute very randomly. We segment input images into several grids. The strategy to choose each try is: randomly 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 much enhance for getting correct solutions (due to the limited paper space, the proof is ignored).

Assume $M^i = (m_0^i, m_1^i, ..., m_7^i)$ is the solution got from the ith try. To determine which try is the best, we define the consistent error of a matching pair $\{p \leftrightarrow q\}$ to M^i as follows:

$$e(p,q,M^i) = \sqrt{(q^x - \frac{m_0^i p^x + m_1^i p^y + m_2^i}{m_6^i p^x + m_7^i p^y + 1})^2 + (q^y - \frac{m_3^i p^x + m_4^i p^y + m_5^i}{m_6^i p^x + m_7^i p^y + 1})^2}. \quad (5)$$

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. Then, a counter $c(M^i)$ can be used to record how many matching pairs in MP_r which are consistent to M^i. The best solution $\bar{M}$ is then obtained according to the following equaiton:

$$\bar{M} = \operatorname*{argmax}_{M^i} c(M^i), \quad (6)$$

where M^0 is got from the edge alignment approach.

3.4 Parameter Refinement through Optimization

With the Monte Carlo method, the best estimate $\bar{M}$ can be found from MP_r. However, $\bar{M}$ can be further refined if additional optimization process is applied. Let $MP_{\bar{M}}$ be denoted as a new set of matching pairs: $MP_{\bar{M}} = \{p_k \Leftrightarrow q_k, k = 1,2,...,N_{\bar{M}}\}$, where $p_k \in FP_{I_a}$, $q_k \in FP_{I_b}$, and $e(p_k, q_k, \bar{M}) < T_e$. With $MP_{\bar{M}}$, we can define an error function as:

$$\Phi(\bar{M}) = \sum_{k=1}^{N_{\bar{M}}} e(p_k, q_k, \bar{M}), \quad (7)$$

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:

$$\bar{M}_{r+1}^T = \bar{M}_r^T + (A + \lambda)^{-1} B, \quad (8)$$

where $[A]_{ij} = \sum_{k=1}^{N_{\bar{M}}} \frac{\partial e_k}{\partial \bar{m}_i} \frac{\partial e_k}{\partial \bar{m}_j}$, $[B]_i = \sum_{k=1}^{N_{\bar{M}}} e_k \frac{\partial e_k}{\partial \bar{m}_i}$, and λ is a coefficient obtained by the Levenber-Marquardt method [6]. The above minimization process converges very quickly since only few feature positions are involved into minimization and the initial value is close to correct one.

4. Experimental Results

Fig. 3 shows the result for mosaic construction when a series of panoramic images are used. Fig. 4 shows the case when images have larger intensity differences.

The large lighting changes will lead to the instability of feature matching for most traditional correlation techniques. However, the proposed edge alignment algorithm still works well to find desired parameters. Fig. 5 shows the case when images have moving objects. The moving object will disturb the work of image stitching and analysis. Fig. 6 shows the result when images have some rotation and skewing effects. In addition to mosaic constructions, the proposed method also can be used to extract moving objects from video sequence. Fig. 7 shows two frames got from amovie. (c) is the mosaic of (a) and (b). (d) is the detected moving object. From all experimental results, it has been proved that the proposed method is indeed an efficient, robust, and accurate method for image stitching.

References

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

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

[3] 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.

[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] 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.

[6] 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.

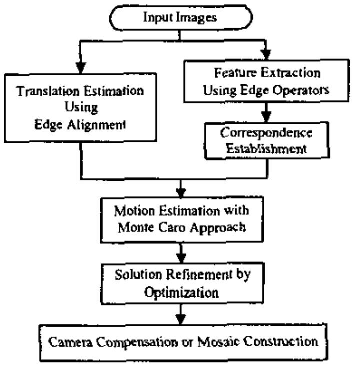
Input Images
Translation Estimation Using Edge Alignment
Feature Extraction Using Edge Operators
Correspondence Establishment
Motion Estimation with Monte Caro Approach
Solution Refinement by Optimization
Camera Compensation or Mosaic Construction

Fig. 1: Flowchart of the proposed method.

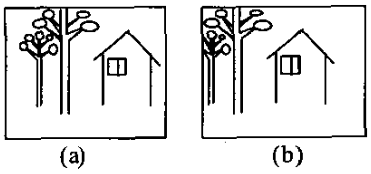

Fig. 2 Edge results of two images.

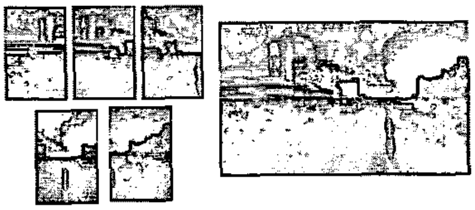

Fig. 3: Stitching result of a series of panoramic images.

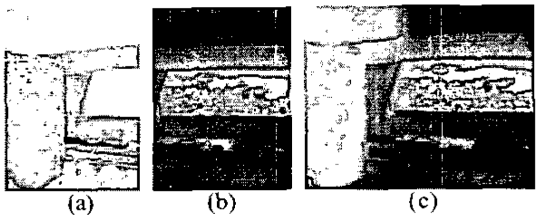

Fig. 4: Stitching result of two images with larger lighting changes. (c) is the stitching result of (a) and (b)

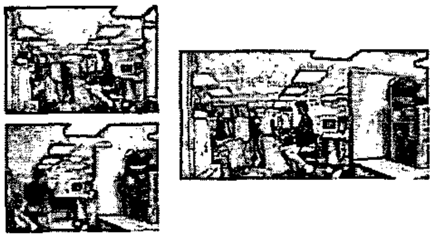

Fig. 5: Stitching result when images have moving objects.

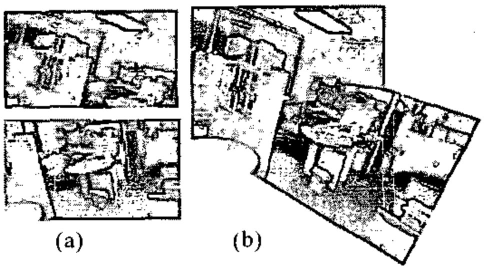

Fig. 6: Stitching result when the camera has rotation changes.

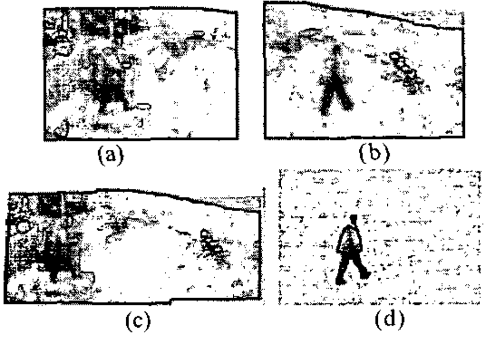

Fig. 7: Mosaic construction and object detection. (c) is the mosaic result of (a) and (b). (d) is the object detection result by image differencing.