Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Automatic Image Mosaic System Using Image Feature Detection and Taylor Series

Soo-Hyun CHO , Yun-Koo CHUNG, Jae Yeon LEE

161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350 KOREA
Electronics and Telecommunications Research Institute
Jsh62860@etri.re.kr

Abstract. Image mosaicing has been collecting considerable attention in the field of computer vision and photogrammetry. Unlike previous methods using a tripod, we have developed which can handle images taken with a hand-held camera to accurately construct a panoramic image. This paper proposes the automatic image mosaic system implementation, which it sees to use feature detection of the image, which is extracted a feature point adjustment from continuous two images hour automatically from pixel price of image and in order to accomplish. Feature based image mosaics require the integration of corner detection, corner matching, motion parameters estimation, and image stitching. We used Taylor series for parameter and perspective transform estimation.

1. Introduction

In the last few years the interest in mosaicing has grown in the vision community because of its many applications. The automatic construction of large and high-resolution image mosaics is an active area of research in the fields of photogrammetry, computer vision, image processing, medical image, real rendering, robot -vision and computer graphics. Image mosaics involve aligning a sequence of image into a larger image and are an important issue in many virtual reality problems. Mosaicing is a common and popular method of effectively increasing the field of view of a camera, by allowing several views of a scene to be combined into single view. The traditional approach, which uses correlation intensity based [1][2] image registration, suffers from computation inefficiency and is sensitive to variations in image intensity. To improve the efficiency of image mosaics, we used a feature-based approach. Two images belonging to a planar scene are related by an affine transformation and perspective transformation using Taylor series. One of the images is used as the reference image, and the second image is aligned with the reference image. To find the coordinate transformation between the two images, we first conduct corner detection to find the corners in these two images. Next, we perform a corner matching process to find the corresponding corner points between these two images. We used SSD (Sum of Squared Difference) method for corner matching. After initial matching, in order to remove mismatches corresponding point, which does a filtering processing from each image. We can estimate the transformation parameters using the Taylor series. System flowchart is shown in Fig 1. In the next section, we give implementation and algorithm of the system and experimental and results. Finally, conclusions are given in section 4

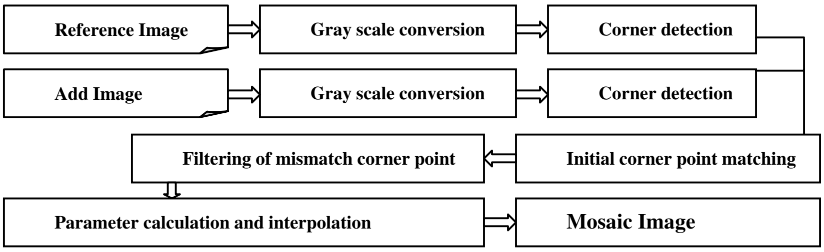
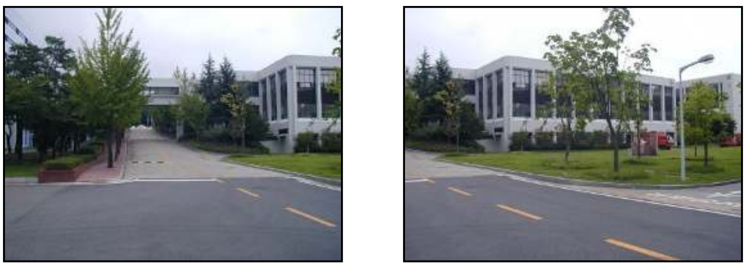


Fig. 1. Flowchart of Mosaic create system

2. System Implementation and Algorithm

2-1 Image Acquisition

In this paper, we used hand-held camera. The set may consist of two images taken of scene at different times, from different viewpoints. It is possible that all movements of camera, (i.e. pan, tilt, rotation, translation, scale, shear). Outside scene the image which it requires from the of course inside scene is possible. First, if image involves more than one band, say RGB, it will be converted to gray-scale image eq. (1). That is *I'()* is formulate as follow. The reference image and add image of original images are shown in Fig.2



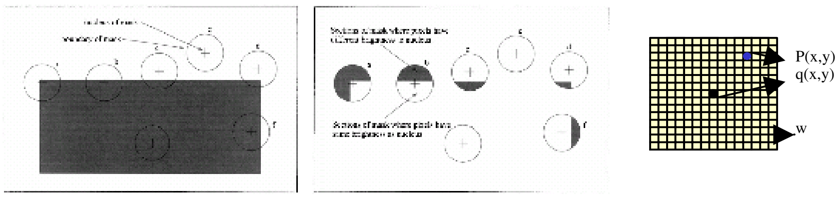
(a) Reference Image (b) Add Image

Fig. 2. Reference image and Add image of test image

$$I'(x,y) = 0.2999 \times I(x,y)_{red} + 0.587 \times I(x,y)_{green} + 0.114 \times I(x,y)_{blue} \tag{1}$$

2-2 Corner detection using SUSAN method

In this approach, we first detect corner points using the SUSAN(Smallest Univaule Segment Assimilating Nucleus) principle in the each two images [3]. The response is processed to output the set of corners. The mask is places at each in the image and, for each point, the brightness of each pixel within the mask is compared with that of the nucleus, i.e. the center point. The comparison uses the eq.(2) here , $\vec{r}_0$ is the position of nucleus, $\vec{r}$ position of an other point within the mask, $I(\vec{r}\ )$ the brightness of any pixel, t the brightness difference threshold and c the output. With figure 3, Considering 7×7 pixel region window (*w*) that point in the center and optional point *p(x, y)* in the image, from inside the w different one point *q(x, y)* it does. *I(p)* and *I(q)* shows intensity values each point from *p* and *q*. Calculate the number of pixels within the circular mask which have similar brightness to the nucleus. Input image is gray scale value, T the brightness difference threshold and G the Gaussian value eq(3)



(a) (b)

Fig. 3. SUSAN principle: (a) circular masks at different places on the image, (b) USAN show in white color

(a-1) reference image (a-2) add image

(b-1) reference image (b-2) add image

Fig. 4. The result of corner detection (a) image G=1500 T=25, (b) image G=1000,T=50

$$C(\vec{r},\vec{r}_0)=\begin{cases}1 if\ |I(\vec{r})-I(\vec{r}_0)|\le t\\ 0 if\ |I(\vec{r})-I(\vec{r}_0)|>t\end{cases} \quad (2)$$

$$S=\left(\sum e^{-\left(\frac{I(x,y)-I(x+dx,y+dy)}{T}\right)^6}\right)/G \quad (3)$$

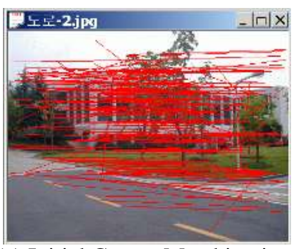
2-3 Initial Corner Point Matching

Feature matching is a key component in many computer vision applications, for example stereo- vision, motion tracking, and identification. Of all possible features, “corner” are the most widely used: there two-dimensional structure providing the most information about image motion. A number of correlation-based algorithms attempt to find points of interest on which to perform the correlation. To match the correlation corner points between the two images, we used SSD (Sun of Squared Difference) method [4]. We measure the similarity between the two-correlation windows using those detected corner points. Correlation scores are computed by comparing a fixed window in the second. SSD method is practical method, which produces reliable results with a minimum of computation time in comparison with the other method. Here I_1 is reference image, I_2 is add image, N is mask size.

$$G'=\sum_{i,j=-N/2}^{N/2}(I_1(x+i,y+j)-I_2(x+i+dx,y+j+dy))^2 \quad (4)$$

2-4 Filtering of Mismatch Corner Point

After initial corner point matching, we need filtering to delete a mismatch feature point. General method is vector filtering [4], the median flow filtering, rotational cross-correlation filtering method in other to removes of mismatch point. This method is, which feature points definite local region of overlapping region, we used similarity values of rotation angle and length. The results of initial corner matching image and after filtering matching image for the line are shown in 4



(a) Initial Corner Matching image

(b) After Corner Matching Image

Fig. 5. Example test results of corner matching image

2-5 Parameter Calculations

Using homogeneous coordinates, 2D planar projective transformation plus affine transformation method employs the following equations:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad \mathbf{(5)}$$

(x, y)=image coordinates, (X,Y,Z)=world coordinates

Geometric correspondence is achieved by determining the mapping function that governs the relationship of all points among a pair of images. There are several common mapping function models in image registration. The general form for the mapping function induced by the deformation is [*x, y*]=[*X(u, v), Y(u, v)*] where [*u, v*] and [*x, y*] denote corresponding pixels in I_1 and I_2, respectively, and *X* and *Y*, are arbitrary mapping function that uniquely specify the spatial transformation. In registering I_1 and I_2, we shall be interested in recovering the inverse mapping function *U* and *V* that transform I_2 back into I_1 [*u, v*]=[*U(x, y), V(x, y)*] In this section, we extend the results of affine registration and Biquadratic using a Taylor series. They include (1) 3-parameter rigid transformation (translation), (2) 6-parameter affine transformation (translation, scale, shear) using first-order Taylor series, (3) 12-parameter Biquadratic transformation (translation, rotation, full motion) using second-order Taylor series. We leverage the robust affine registration algorithm to handle the more perspective registration problem. A local affine approximation is suggested by expanding the perspective transformation function about a point using a first – order Taylor series. In general, Taylor series equation:

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 \cdots \\ &+ \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n \end{aligned} \quad \mathbf{(6)}$$

The approximation holds in a small neighborhood about point (x_0, y_0). As a result, affine transformation equations: (7)

$$\begin{aligned} u &= Ax + By + C \\ v &= Dx + Ey + F \end{aligned} \quad \mathbf{(7)}$$

The affine transformation with first polynomial, it is possibility of the image mosaic create according to translation motion. But in the case of rotation, pan and tilt motion, it is difficult to expect good results. It is possible that all movements of camera, (i.e. pan, tilt, rotation, translation, scale, shear), in order to mosaic image creation. The traditional method uses perspective transformation using 8-parameters. A weak point of the method, the calculation process being complicated and the error scope is big. In order to overcome the method's defects, this paper uses second – order Taylor series possible full motion mosaic image create. The method is Biquadratic using 12-parameter. The equation (8,9) is Biquadratic (second polynomial) using second-order Taylor series.

$$\begin{aligned} u &= U(x,y) \\ &= U(x_0,y_0) + \frac{\partial U(x_0,y_0)}{\partial x}(x-x_0) + \frac{\partial U(x_0,y_0)}{\partial y}(y-y_0) \\ &+ \frac{\partial U^2(x_0,y_0)}{\partial x^2}(x-x_0) + \frac{\partial U^2(x_0,y_0)}{\partial y^2}(y-y_0) + \frac{\partial U}{\partial x}\frac{\partial U}{\partial y}(x-x_0)(y-y_0) \\ &= Ax + By + C + Dx^2 + Ey^2 + Fxy \end{aligned} \quad \mathbf{(8)}$$

$$\begin{aligned} v &= V(x,y) \\ &= V(x_0,y_0) + \frac{\partial V(x_0,y_0)}{\partial x}(x-x_0) + \frac{\partial V(x_0,y_0)}{\partial y}(y-y_0) \\ &+ \frac{\partial V^2(x_0,y_0)}{\partial x^2}(x-x_0) + \frac{\partial V^2(x_0,y_0)}{\partial y^2}(y-y_0) + \frac{\partial V}{\partial x}\frac{\partial V}{\partial y}(x-x_0)(y-y_0) \\ &= Gx + Hy + I + Jx^2 + Ky^2 + Lxy \end{aligned} \quad \mathbf{(9)}$$

Compared with Affine transformation and Biquadratic, Affine transformation is possible slow-moving camera movement and translation motion be unchanged viewpoint of users. But optical rolling motion of the camera to free motion it cannot obtain good result. Biquadratic algorithm can obtain good result free motions of camera, rotation, rolling, zoom in and zoom out. For example affine transformation, given the four corners of a tile in observed image I_2 and their correspondences on reference image I_1, we may solve for the best affine fit by using the least squares approach. We may relate these correspondence in the form U=WA. Eq (10):

$$\begin{bmatrix} u_1 \\ \vdots \\ u_4 \\ v_1 \\ \vdots \\ v_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_4 & y_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} \quad \mathbf{(10)}$$

The pseudoinverse solution $A=(W^TW)^{-1}W^TU$ is computed to solve for the six affine coefficients. After finding the transformation function for each successive image pair, we compute the transformation function of each image relative to the base image based on the associative of the matrix multiplication. Then, we can combine all of the image frames into one big image.

2-6 Interpolation

Interpolation process can be classified into four methods (i.e. Nearest Neighbor interpolation, Bilinear interpolation, Cubic Convolution, B – Spline interpolation). In this approach, we used bilinear interpolation. A critical portion of warping images is interpolation for the resulting pixel values. Without a decent facility for interpolation, precise movement in the error minimization technique would not be possible. The process of bilinear interpolation requires four neighboring values to the coordinate at which we need to interpolation. The interpolation is performed in a separable manner as illustrated by the diagram below. First, the interpolation occurs in the X direction, followed by an interpolation in the Y direction.

3. Experimental Results

To evaluate the performance of our proposed algorithm, we implemented in visual C++ language. The test images we used were obtained in the outside scene and inside using hand held camera for minimum time interval. Since only 2D motion parameters were estimated, the test images were constrained to or close to planar image sequences. Each color image has 640×480, 320×240 pixels, and image format has JEPG, BMP. In general, feature based correlation is sensitive to rotation motion. In order to feature matching, we need feature matching point more than 40 point. Using our proposed approach, part of matching took 60% of total processing time. Processing time results of two images for different processing are shown in Table.1. Total processing time is different each image, (i.e. image size, natural scene, inside scene, difficult scene…). Fig. 5. Shows the result image using the affine transformation (6-parameter) and Fig. 6. Shows the result image using Biquadratic (12-paramater).. In Fig. 7, shows the results of test image. The display is shown in Fig. 8

Fig. 6. The result image using affine transformation. **Fig. 7.** result image using Biquadratic

(Unit: seconds)

Image size	Feature extraction	Initial matching	Filtering	Total time
320*240	1	1.30	1	4
640*480	2	7	4	16

Table 1. The result of processing time

4. Conclusions

We have presented a distributed image mosaic system that can quickly, hand held camera full motion and automatically align a sequence of images to create a larger image. A further direction of this research will be effective algorithm development for shorten the corner matching process time and blending processing for natural boundary region.

(a) reference image

(b) add image

(c) result image of using affine transformation

(d) result image of using Biquadratic

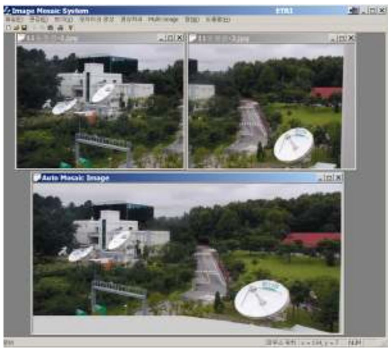


Fig. 8. Results of test images

Fig. 9. The interface of image mosaic create system

Reference

1 Richard Szeliski, " Image Mosaicing for tele Reality applications", Cambridge Research Lab Technical Report, May 1984

2 Richard Szeliski,HeungYeung Shum"Creating Full View Panoramic image Mosaic and Environment maps", SIGRAPH 87, pp 251~258

3 S.M Smith and J,M Brady, "Susan a new Approach to low level image processing", Technical Report TR95SMSIc, 1995

4 P.Smithm, D Sinclaif, R. Cipolla and K. Wood, "Effective Corner matching", British machine vision conference, Vol 2 p 545~556, September 1998

5 Richard Szeliski, "Video mosaics for virtual Environments", IEEE Computer Graphics and Application, 1966, pp 22～30.

6 Naoki CHIBA, Hiroshi KANO, Michihiko MINOH and Masashi YASUDA "Feature –based image mosaicng", IEICE, Japan D-II Vol, J82 No 10 pp 1589~1999 1999.10