

Image Mosaic Algorithm Based on Vertical Distance of Edges

Yongqi Wu Xizhi Hu Libiao Jiang Zemao Chen

Abstract This paper presents a new algorithm of automatic image mosaic, which generate mosaic image by comparing the vertical distance of edges in two neighboring images. This algorithm selects points to be matched by statistical features of the images before the actual matching stage, reducing the amount of computation, saving time for higher accuracy. Based on vertical distance of edges, the matching stage of this algorithm is able to locate the optimal joint, which bears high consistency in two neighboring images. To stitch two images on the optimal joint generates a stitching line that is less noticeable. This feature makes it unnecessary to smooth large area to merge two images, leaving more untouched details in the mosaic image. The algorithm also computes the vertical displacement of the two images, so it is able to deal with images shot by unfixed cameras.

Keywords: image processing · image mosaic · edges · distance

Yongqi Wu (✉)

School of Mechanical and Automotive Engineering, South China University of Technology,
Guangzhou, R.P China
e-mail: 103075390@qq.com

Yongqi Wu

School of Mechanical and Automotive Engineering, South China University of Technology,
Guangzhou, R.P China

Xizhi Hu

School of Mechanical and Automotive Engineering, South China University of Technology,
Guangzhou, R.P China

Libiao Jiang

School of Mechanical and Automotive Engineering, South China University of Technology,
Guangzhou, R.P China

Zemao Chen

School of Mechanical and Automotive Engineering, South China University of Technology,
Guangzhou, R.P China

1 Introduction

Panoramic image technique is an essential part of image based rendering and broadly applied to the fields of virtual reality, video compression and transmission, medical equipment, etc[1]. Panoramic image is the image resemble to one's vision from a fixed point to the whole space around, showing all one can see around from the fixed point. A simple form of panoramic image is the image of a same horizontal plane around the visual point. Panoramic image can be obtained with special instruments, but these instruments are quite expensive[2]. A more accessible alternate is image mosaic with computers, as in figure 1. Image mosaic is a basic technique of image based rendering. Generally it pieces together images with overlapping area obtained by normal cameras to generate a larger image or a panoramic image. The process is accomplished by computers with image mosaic algorithm. Generally the overlapping area is roughly known, and the process can be restricted in this area.

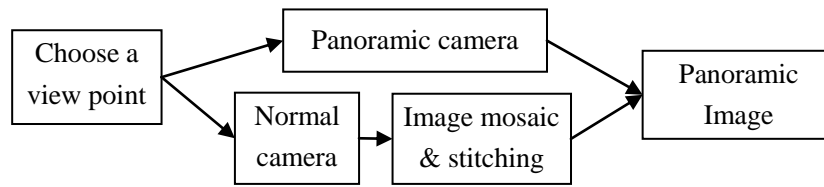


Fig. 1 Steps to obtain a panoramic image

A common method of image mosaic is to find in the overlapping area the pair of points, which matches best according to similarity criteria. This paper presents a new method to compute the positional correspondence of two images with roughly the same horizontal plane and generate a larger image. Firstly, edges in the images are extracted with the Canny Algorithm, and then every column of pixels is analyzed statistically in terms of edge pixel distribution. Columns that are likely to match are further judged by comparing the vertical distance of edge pixels. Neighboring columns of the matching column are also examined, and the pair of columns with the minimum total difference is determined to be the joint column in the larger image to be generated.

2 Previous Works

There are loads of algorithms of image mosaic, most of which fall into two categories[2]. The first kind is based on area. Algorithms of this kind take the

neighborhood of a pixel as a template, and search on the other image for a pixel whose neighborhood is similar to the template, in order to determine the overlapping area of the two image. Usually a function of similarity is used as the criteria in searching. Fu and Chen[1] describes an algorism based on area searching. Their method extracts vertical edges in the images and searches overlapping area by vertical edges, which can be inaccurate if there is a minor angle difference between two images. Zang[2] presents an algorism which uses ring templates in area searching. This algorism directly compares grey levels of every pixel in the templates, which may cause considerable amount of computation. It is also unlikely to work satisfactorily in occasions with minor angle differences. In addition, based on area searching, these algorisms are usually slow, as screening and comparing areas of pixels usually implies large amount of computation.

Methods of the second kind are based on features of the images. Such methods do not compare the grey level of pixels by area directly, but extract features of the image pixels and compare these features. Such features can be specific scene in the image, difference of grey level in two columns of pixels, characteristic line segments etc. Methods based on features are generally faster, as they abstract the information and avoid searching directly the similar areas of two images. Cao *et al.* [3] presents an algorism based on Canny Algorism. This algorism extracts edges and searches the pixel of max gradient on every column of the images, by which it determines which column in one image corresponds to that in another. Only concerning the max gradient pixels, this algorism is unlikely to work satisfactorily on complex images with multiple edges of similar gradients in every column. Li *et al.* [4] describes a corner point based detection algorism, which is a fast method but it is largely dependent of the presence of corner points.

3 Algorithm

Our goal in this work is to determine the positional correspondence of the partial images in order to compose a panoramic image. These images are shot from a particular point, on a same horizontal plane and each with a particular angle difference to cover the whole scene around. Every two images have overlapping area, according to which accurate positional correspondence is detected. Although it is preferable if the images are on the same horizontal plane, on some occasion such as handheld work, the horizontal planes in different images can not be exactly the same. There may be a small difference in vertical position and rotational angle. The ability to deal with such minor difference between two images is also considered. The algorism presented in this paper is based on features (edges) analysis, innovatively utilizing the vertical distance of edges to determine the positional correspondence of two

images. It's able to detect the vertical offset of two images tolerable to minor rotational angles.

To generate a panoramic image is actually to find out the positional correspondence of every pair of neighboring images, so it's sufficient to only discuss the mosaic of two images. 20% of the images' area needs to be overlapping, so that the information is adequate for image mosaic. For a pair of neighbor images A and B, to find out the positional correspondence of A and B is equivalent to determine which point (pixel) in the overlapping area of B corresponds to that in A. Then a larger image composed of A and B can be generated according to this information.

3.1 Edge Detection and Column Selecting

The goal of this first stage is to detect edges and select columns of pixels for further matching, so that the matching stage would not have to compare every pair of columns in the overlapping area. Canny edge detector [5] is used to extract edge information from the original images. Vertical positions (in amount of pixels) of the edge pixels in every column are recorded, and mean and variance of such positions μ_j and δ_j are calculated (j for the jth column). As the mean value stands for the average vertical positions of the edge pixels, and the variance stands for the degree of dispersion, neighboring columns with large difference in these values means that the image change relatively greater in this area. Such columns are selected for further matching, because area with larger difference contains more information, which leads to more accurate results. The difference in question is measured by the increment between the two columns on either sides of the column:

$$\mu'_j = \frac{\mu_{j+1} - \mu_{j-1}}{2} \quad (1)$$

$$\delta'_j = \frac{\delta_{j+1} - \delta_{j-1}}{2} \quad (2)$$

Because of there exists certain disturbance in the images, using the increment between every other column yields results that are more stable. Experiments suggest that the threshold value should be about 1% of the height of the image. Columns with values larger than the threshold are selected. The selected columns are divided into 2 groups: the positive increment group and the negative increment group, depending on the sign of the increments. The positive group in image A will be matched with that in image B, and the same to the negative group. Selecting and dividing groups in this manner can reduce the work in the matching step. For example, a column where the average vertical position of edge pixels is rising will be matched to columns where the position is also rising. Accuracy is also enhanced, because around such

columns edges are more likely complicated and changing. Information for matching is more sufficient in these areas; mismatch caused by insufficient edge pixels is avoided (for example, 2 neighboring edge pixels isolated in an area will perfectly match that in another image if not ruled out.)

3.2 Matching

The matching stage is divided into two steps. First, the neighborhood of the selected columns is compared by the sequence of variance of pixel positions. For every pair of selected columns (for example, the m th column in image A and the n th column in image B), the sequence of variance is $S_m^A(i)$ and $S_n^B(i)$. $S_m^A(i)$ means the variance of the i th column in the neighborhood of the m th column in image A. The width of the neighborhood is 10% of the width of the image. The selected column can be in any position in the neighborhood, but the position must be the same in the two neighborhoods being compared. The similarity is measure by:

$$Dif(S_m^A, S_n^B) = \sum_{i=1}^L |S_m^A(i) - S_n^B(i)| \quad (3)$$

which is smaller if the similarity is greater.

Every pair of compared column has a *Dif* value. The pairs of columns are divided into groups by the standard of the images' position correspondence they indicate. For example, the pair (m, n), which means the m th column in image A and the n th column in image B, and the pair (m', n') are divided into a group if $(m-n)-(m'-n')$ is smaller than a threshold (5% of the image width). Only the pair of columns with the smallest *Dif* value in every group will be compared in the next step, because all pairs in a group indicates similar results. Take this smallest *Dif* value as the *Dif* value of the group, and the horizontal position (m, n) of the columns with the smallest *Dif* value as the position of the group. Arrange the groups by *Dif* values and select the first 20 groups (more for higher accuracy, but 20 is adequate as tests indicates) with the smallest *Dif* values into the second step.

The second step is the final matching by vertical distance sequence of edge pixels, shown in figure 2. A value in the sequence represents the distance from this pixel to the nearest edge pixel.

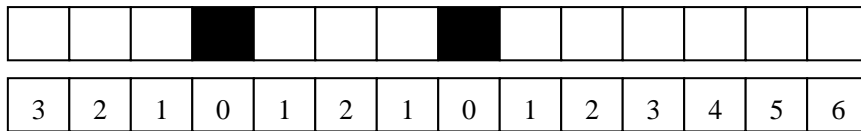


Fig.2 Vertical distance sequence of edge pixels

The matching method is as follows. In the neighborhoods of a selected

group's position (m, n), which is the mth column in image A and the nth column is image B, take 10 (more for higher accuracy, but 10 is generally enough) equally spaced columns each for matching. The width of the neighborhood is 10% of the image width and usually the position of the column (m or n) should be in the middle of the neighborhood, but if the position is near one end of the image, the neighborhood can be 10% of the image on the end. These two groups of 10 columns are matched first-to-firstly, second-to-secondly and so on. Offset one of a pair of columns from $-h/2$ to $h/2$ (h for height of the image) and calculate the difference of their vertical distance sequences:

$$Dist_{m,n}^j = \min_{offset \in (-h/2, h/2)} \left[\sum_{i+offset \in (0, h)} |D_m^j(i+offset) - D_n^j(i)| / (h-offset) \right] \quad (4)$$

where $D_m^j(i)$ is the distance sequences of the jth column in the 10 columns. Sum up the 10 pairs' difference of vertical distance sequences:

$$GroupD_{m,n} = \sum_{j=1}^{10} Dist_{m,n}^j \quad (5)$$

which is the measurement of similarity of the group. The group with the smallest $GroupD$ is the best matching group. In the best matching group, the pair of columns with the smallest $Dist$ value is the best matching columns, and the offset value which leads to the smallest $Dist$ value is the vertical offset of the images in this pair of columns. The optimized mosaic result can be reached by taking this pair of columns as the joint of two images. As the columns with smaller $Dist$ value bears higher consistency, jointing two images on this column is the less noticeable and the smoothing step discussed in next section can be restricted in a small scale.

3.3 Stitching

The optimized joint column has been found, but as the lighting condition could be different and the images may not be of accurately the same horizontal plane, jointing two images on the column without any processing will produce an apparent stitching line. In order to render a better stitched image, smoothing is needed near the joint column. The most common method will meet the need:

$$P_{tran} = (1-\omega)P_1 + \omega P_2 \quad (6)$$

where P_{tran} is the grey level of the stitching area, P_1, P_2 is the grey levels of the pixels in the origin images, ω is the weighting coefficient, which gradually change from 0 to 1 within the stitching area. Middle of the stitching area is the

joint column, and the horizontal width of the stitching area is 2% to 5% of the image width, which is enough as the optimized joint (the least apparent column to joint the images) is determined, and the smoothed area should be kept to the least in order to retain original image details.

4 Result

Some of the test results are shown in figure 3. More tests indicate that the algorism presented in this paper works satisfactorily in various environments, including where origin images have minor rotational angle and horizontal plane difference. Unlike common image mosaic algorithms, this algorism compared two images with vertical distance sequence of edge pixels, searching not only the positional correspondence of the images, but also the optimized joint column which contains the least details in the image. This feature allows less smoothing near the joint, avoiding obscurity and ghost image caused by smoothing large area. However, as there's a shooting angle difference between two images, results for close shots are less satisfactory (the ceiling in the second result in figure 3). This problem can be solved by involving cylindrical projection, but it's outside the scope of this paper.



Fig.3 Test results

References

1. FU H, CHEN Z (2004) Panoramic Image Stitching Based on Vertical Edge Sharping (in Chinese). Computer Engineering, 2004, 30(4): 132-134
2. ZANG Y (2007) Research on Mosaics Algorithm Based on Panoramic Images (in Chinese). Microcomputer Information, 2007, 23(6-3): 311-312
3. CAO J, FENG J, SU Z (2003) A Panoramic Image Mosaic Algorithm (in Chinese). Journal of Dalian University of Technology, 2003, 43(S1): 180-182
4. LI Y, WANG X, DUAN J (2010) Overlapping Image Mosaic Method Based on Feature Points Detection (in Chinese). 2010, 28(6): 596-600
5. CANNY J. A Computational Approach to Edge Detection(1986). Pattern Analysis and Machine Intelligence, 1986, 8(6):679-698.
6. MCMILLAN L, BISHOP G. Plenoptic Modeling: An Image-Based Rendering System. Computer Graphics, SIGGRAPH'95, 1995,8:39-46.