

An Image Mosaic Algorithm for Spherical Panorama

Fan Tang^{1, a}, BoTao Zhang^{2, b}, HengZhu Liu^{3, c}

^{1, 2, 3}National University of Defense Technology, Deya Road No.109, ChangSha 410073, China

^a12tf@sina.com, ^b357361589@qq.com

Keywords: image mosaic; panoramic image; optical flow; image registration; image blending

Abstract. Visual reality is one of the hot topics in these years. Panoramic photography is one of the VR technique. Image mosaic algorithms is the main technology to stitch the little scale images to a large scale panoramic image. In this paper, an improved image mosaic method is presented. Because of using fish-eye images, spherical panorama model was chosen to describe the alignment between a pair of images. A feature-based method is devised to estimate the parameters of the registration model. In order to reduce registration errors, optical flow was applied to local alignment. Before blending the images, a brightness adjustment algorithm base on sample point will be introduced.

Introduction

Panorama photography is a popular way to get large scale photos. Image mosaic [1] is a solution to align and stitch little scale images to large scale image. Image mosaic usually divided into two steps [2]: image registration and image stitching.

The aim of image registration is to register the image in the same coordinate system according to the geometric model. It can discover the correspondence relationships among images with varying degrees of overlap. Image stitching is to stitch the registered images to a large image. Image stitching algorithms take the alignment estimates produced by such registration algorithms and blend the images in a seamless manner.

If the registration is not accurate, the ghosting would occur on the blended image. Another encountered problem is exposure difference. Because of the exposure difference, a boundary between light and shade would appear on the blended area. That will make the blended image look like artifacts. Ghosting and exposure differences are the main problem of all image mosaic algorithm to solve.

Optical flow has become one of the most widely used techniques to image align since its first use. Application of it range from tracking [3] and layered motion estimation, to mosaic construction [4], medical image registration, and face coding. Numerous algorithms have been proposed and a variety of extensions have been made to the original formulation. In this paper, we will use a extended optical flow algorithm to align image and then use the aligned image to blend. This method can reduce ghosting and exposure difference obviously.

Optical Flow

The Principle of Optical Flow algorithm. Optical flow is the description of motion information of brightness patterns in an image, which can assess motion between two frames (or a sequence of frames) without any other prior knowledge about the content of those frames. In 1981, Lucas and Kanade proposed an algorithm [5] to compute dense optical flow. The method is based on three assumptions as below:

- Brightness constancy. A pixel from the image of an object in the scene does not change in appearance as it (possibly) moves from frame to frame. Let the image brightness at the point (x, y) in the image plane at time t be denoted by $I(x, y, t)$, so we have:

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \quad (1)$$
- Temporal persistence or “small movements”. The image motion of a surface patch changes slowly in time. In practice, this means the brightness of a particular point in the image plane is

constant to the motion of the brightness pattern, so that $dI / dt = 0$. Using the chain rule for differentiation we see that,

$$I_x u + I_y v + I_t = 0, u = dx / dt, v = dy / dt. \quad (2)$$

- **Spatial coherence.** Neighboring points in a scene belong to the same surface, have similar motion, and project to nearby points on the image plane. Now we can easily solve for the motion of the central pixel by using the surround pixels to set up a system of equations. To solve for this system, a least-squares minimization of the equation was set up, whereby it is solved in standard form as $[u \ v]^T = G^{-1}b$, matrix G and b defined as :

$$G = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}, b = \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}. \quad (3)$$

Pyramid Lucas-Kanade Optical flow. Lucas-Kanade optical flow has become one of the most widely used techniques in computer vision, but in fact it does not work very well for exactly this reason: a large window is needed to catch large motions, but a large window too often breaks the coherent motion assumption! To circumvent this problem, an intuitive way is to reduce the image size. We can track first over larger spatial scales using an image pyramid [6] and then refine the initial motion velocity assumptions by working our way down the levels of the image pyramid until we arrive at the raw image pixels.

Let I and J be two 2D gray-scaled images. The two quantities $I(x, y)$ and $J(x, y)$ are then the grayscale value of the two images at the location $\mathbf{x} = (x, y)$, where x and y are the two pixel coordinates of a generic image point \mathbf{x} . Consider an image point $\mathbf{u} = (u_x, u_y)$ on the image I . The goal of pyramid Lucas-Kanade optical flow is to find the location $\mathbf{v} = \mathbf{u} + \mathbf{d} = (u_x + d_x, u_y + d_y)$ on the image J subject to $I(\mathbf{u})$ and $J(\mathbf{v})$ are “similar”. The vector $\mathbf{d} = (d_x, d_y)$ is the optical flow at \mathbf{u} . The word “similar” means that the vector \mathbf{d} minimizes the residual function defined as follow:

$$\varepsilon(\mathbf{d}) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x, y) - J(x + d_x, y + d_y))^2. \quad (4)$$

Let ω_x and ω_y two integers. The similarity function is measured on a image neighborhood of size $(2\omega_x + 1) * (2\omega_y + 1)$.

Now let us summarize the process of the entire algorithm. First of all, we should build the pyramid representations of I and J : $\{I^L\}$ and $\{J^L\}$, $L = 0, \dots, n$ and initialization of pyramidal guess $\mathbf{g}^n = (0, 0)$. The relationship between neighboring level is

$$\mathbf{g}^{L-1} = 2(\mathbf{g}^L + \mathbf{d}^L), \quad (5)$$

in which \mathbf{d}^L means optical flow gain at pyramid level L . The gain \mathbf{d}^L can be computed by an iterative way. According the equations as follow:

$$I_x = \frac{I^L(x+1, y) - I^L(x-1, y)}{2} \quad (6)$$

and

$$I_y = \frac{I^L(x, y+1) - I^L(x, y-1)}{2}, \quad (7)$$

we get horizontal gradient I_x and vertical gradient I_y . The image difference I_t can be expressed as:

$$I_t = I^L(x, y) - J^L(x + g_x^L + v_x^{k-1}, y + g_y^L + v_y^{k-1}) \quad (8)$$

According to equation (3), we get $\mathbf{v}^k = G^{-1} b_k$. The guess for next iteration is $\mathbf{v}^k = \mathbf{v}^{k-1} + \mathbf{v}^k$, initialization of iteration $\mathbf{v}^0 = (0, 0)$. Until the iteration reach the iterative limit K or $\mathbf{v}^k < accuracy\ threshold$, we terminate the iteration. The optical flow gain $\mathbf{d}^L = \mathbf{v}^k$. After computing from top level to bottom level of the pyramid, we get the optical flow:

$$\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0. \quad (9)$$

Observe that this solution may be expressed in the following extended form:

$$\mathbf{d} = \sum_{L=0}^n 2^L \mathbf{d}^L. \quad (10)$$

The clear advantage of a pyramidal implementation is that each residual optical flow vector \mathbf{d}^L can be kept very small while computing a large overall pixel displacement vector \mathbf{d} . Assuming that each elementary optical flow computation step can handle pixel motions up to d_{\max} , then the overall pixel motion that the pyramidal implementation can handle becomes $d_{\max_final} = (2^{L+1} - 1) d_{\max}$. This enables large pixel motions, while keeping the size of the integration window relatively small.

Image Mosaic

Image Registration. Before blending images, it is necessary to align the images by confirming the position between images. We need to establish the mathematical relationships that map pixel coordinates from one image to another. For standard lens images, the mathematical model is usually established by affine transformation, and the least squares method is used to optimize the parameters. Fish-eye image alignment is relatively complex, because it is not two-dimensional image essentially. Before aligning images, fish-eye image should be corrected according to its projection model [7].

After correction, we can establish the mathematical relationships that map pixel coordinates from one image to another. The approach is to first extract distinctive features from each image, to match these features to establish a global correspondence, and to then estimate the geometric transformation between the images. David Lowe's Scale Invariant Feature Transform (SIFT) [8] is a popular method to find and match the keypoints. RANSAC [9] was used to erase the error matched keypoints. Then we project the matched pair points to the spherical surface according to the fisheye lens projection model. The relationship between the matched points $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ can be defined as:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}. \quad (11)$$

The matrix \mathbf{H} with nine parameters is the relationship of the geometric transformation. This is a mini value optimization problem. Levenberg-Marquardt algorithm [10] have a perfect effect to solve this problem.

In fact, there are still small differences between two images after alignment, which derived from lens radial distortion, 3D parallax (failure to rotate the camera around its optical center), small scene motions such as waving tree branches and large-scale scene motion such as people moving in and out pictures. These differences will lead to a problem that the "ghosting" appeared on the merged image. To deal with this situation, we should reduce the difference between two images. Using the optical flow information to remap two images is a feasible way. Three assumptions must be satisfied before computing optical flow. The small differences after image alignment is fit to the second assumption: temporal persistence.

Image Brightness Adjustment. Due to camera auto exposure and white balance affected by ambient light, there exist a difference in brightness and color between the images. This is contrary to the brightness constancy, so we should adjust the brightness of the images.

Sample point [11] is one of the way to adjust image brightness. The three image channels RGB are handled separately, and the difference in brightness and color is approximated by a linear equation:

$$\begin{bmatrix} R_2 \\ G_2 \\ B_2 \end{bmatrix} = \begin{bmatrix} a_r & 0 & 0 \\ 0 & a_g & 0 \\ 0 & 0 & a_b \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} + \begin{bmatrix} b_r \\ b_g \\ b_b \end{bmatrix}. \quad (12)$$

where a and b are linear transformation parameter. The sample points are chosen from the SIFT match previous step and its neighbor points. MSAC [12] was used to refine the sample points. Regression analysis are used to fetch the linear transform parameters of the three channels. The detail of the algorithm can find in [11].

Local Alignment by Using Optical Flow. The left image of the overlap region is denoted as L and the right one is denoted by R . The optical flow $D_1(dx^1, dy^1)$ of the left image L to the right image R is obtained by the pyramid LK algorithm. We also compute the optical flow $D_2(dx^2, dy^2)$ of the right

image R to the left image L . An image point $\mathbf{u} = (u_x, u_y)$ on the image L and an image point $\mathbf{v} = (v_x, v_y)$ on the image R satisfy these equations $\mathbf{v} = \mathbf{u} + \mathbf{d} = (u_x + d_x^1, u_y + d_y^1)$ and $\mathbf{u} = \mathbf{v} + \mathbf{d} = (v_x + d_x^2, v_y + d_y^2)$. Let *INTER* be the interpolation mode, and bicubic interpolation method has a smoother image edge. Let w be the width of image L . Let $x = x/w$, and x be the pixel at horizontal coordinate. The function $remap(p, mode)$ means that the value at the point p is interpolated by the interpolation mode. Here we use bicubic interpolation mode. The remap of two image can be defined as:

$$R_{remap}(x, y) = remap(R(x + d_x^1, y + d_y^1), INTER). \quad (13)$$

and

$$L_{remap}(x, y) = remap(L(x + (1 -)d_x^1, y + (1 -)d_y^1), INTER). \quad (14)$$

Image Blending. After local alignment, we still need to blend the images to compensate for exposure differences and other mis-alignment. The linear blending method can be used to accomplish this. Let $x = x/w$, and x be the pixel at horizontal coordinate. The blended image can be compute according this equation:

$$I_{blended}(x, y) = (1 -)L_{remap}(x, y) + R_{remap}(x, y). \quad (15)$$

Experiment and Result

Our experiment used a panorama camera with four wide angel lens surrounded horizontally. We take four picture at the same time, so we think the assumption spatial coherence is satisfied. After distortion correction of the four images and cut the black region, we get four corrected images shows in Fig.1. There are still little differences between them, it can be thought as small movement. The assumption temporal persistence is satisfied.



Figure 1. Four images after distortion correction and cut out the black region.

Because of the sunlight, we can see that visible exposure difference appeared between the four images. We use the sample point method to adjust the brightness of the four images. The result shows in Fig.2. The brightness of four images just have a little difference, the assumption brightness constancy is satisfied.



Figure 2. The four images after brightness adjustment.

Since the three assumptions are all satisfied, we use the optical flow to remap the images, and blend them. The results of the image mosaic has been shown in Fig.3. Compared with other image mosaic method, this method can reduce ghosting and exposure differences obviously.



Figure 3. The result of our image mosaic algorithm. Ghosting has been reduced.

Conclusions

In this paper, we introduced a method to get spherical panorama by using fish-eye lens. This method have a nice effect to deal with exposure difference and ghosting. But it is out of work to deal with large motion in the overlap region. The problem of parallax has also not been adequately solved. This method is effective for small amounts of parallax.

References

- [1] H.Y.Shum and R.Szeliski. "Panoramic Image Mosaics[R]". MSR-TR-97-23.
- [2] R.Szeliski. "Image alignment and stitching: a tutorial". Foundations & Trends in Computer Graphics & Vision, 2(1): 1-104, 2006
- [3] M.Black and A.Jepson. "Eigen-tracking:Robust matching and tracking of articulated objects using a view-based representation". International Journal of Computer Vision, 36(2):101-130, 1998.
- [4] H.Y.Shum and R.Szeliski. "Construction of panoramic image mosaics with global and local alignment". International Journal of Computer Vision, 16(1):63-84, 2000.
- [5] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". In Proceedings of the International Joint Conference on Artificial Intelligence, pages 674.679, 1981
- [6] J.Y.Bouguet. "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm". Technical Report, Intel Microprocessor Research Labs, 1999
- [7] J Kanala, SS Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". IEEE Transaction on Pattern Analysis & Machine Intelligence, 28(8):1335-1340, 2006
- [8] D.G.Lowe, "Distinctive image features from scale-invariant keypoints". International Journal of Computer Vision, vol.60, no.2, pp.91-110, November 2004
- [9] M.A.Fischler and R.C.Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". Communications of the ACM , vol.24, no.6, pp.381-395, June 1981

- [10] J.J.More. "The Levenberg-Marquardt algorithm: Implement and theory". Springer Berlin Heidelberg, 630:105-116, 1978
- [11] ZHAO Hui, CHEN Hui, YU Hong. "Image Mosaic Based on Automatic Brightness and White Blance Ajustment". Sciencepaper Online, 2005.12,"In Chinese".
- [12] David P.C. "Image Mosaicing and Super-resolution[D]". Oxford University 2001