PANORAMA STITCHING

[1] *Shao-Yeh Huang* (黃紹曄) Chiou-Shann Fuh (傅楸善)

[1] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan,
E-mail: b04902074@ntu.edu.tw fuh@csie.ntu.edu.tw

ABSRACT

In this paper, an automatic panoramic image stitching algorithms is presented. The panoramic image will be generated by several processes, including the boundary alignment and blending. First, automatically detecting the overlapping area by the Scale-Invariant Feature Transform (SIFT) algorithms, matching the feature points by Random Sample Consensus (RANSAC) and stitching the images with inverse wrapping to handle different focal lengths. Finally, aligning the edge with the blending, this is achieved by repeated experiments and better solution.

Keywords*: automatic panoramic image, image stitching, image blending.*

1. INTORDUCTION

Panoramic image stitching is an extensive research for a long time [1, 2]. The basic solution is followed by the SIFT feature detecting [3] and image matching by RANSAC [4]. These are the preprocessing of the image stitching and warping the images by inverse warping [5] and there are many researches about this, acquiring an automatic and more robust result [6, 7], enhancing the performance the panoramic images. Due to the mature research about this field, the blending algorithms [8] increase the quality of the panoramic images.

In this paper, we will follow the general process of the image stitching and modify the algorithms to fit our experiments. We aim to provide different kinds of blending techniques for users to choose and generate a smoother and more natural panorama image.

2. APPROACH

The approach to solve the panorama stitching involves a sequence of processes. The preprocessing contains the feature detection and feature matching. After preprocessing, it needs to align the two photographs together. This may conduct some transformation by image warping. Moreover, at the edges of the boundary area smoother and more natural.

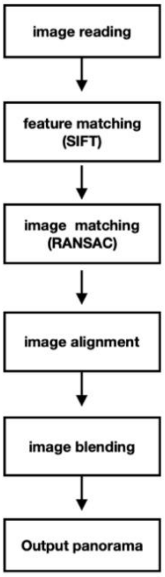

Figure 1 The procedure of the panorama stitching.

3. FEATURE MATCHING

The first step in panorama stitching is feature matching. We use SIFT [9] to match the feature points in the photographs. SIFT features are located at scale-space of a difference of Gaussian (DOG) function. At each feature location, a characteristic scale and orientation is established. This gives a similarity-invariant frame in which to make measurements. By accumulating gradients in local gradients in orientation histograms, the edges can shift slightly without altering the descriptor vector. Between two photographs, there usually has small pixel movement. Therefore, the spatial accumulation is important to shift invariance and helpful to solve this movement between two continuous photographs.

Since SIFT features are invariant under rotation and scale changes, the system can handle photographs with varying orientation and zooming. This is superior to the traditional feature matching techniques such as the Harris corner detection algorithm [10]. Since the Harris corner are not invariant to changes in scale.

The detail of SIFT is computing the DOG by constructed the Gaussian Pyramid which is the different scale-space kernel of the Gaussian function. Therefore, the scale space of an image is defined as a function, $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where $*$ is the convolution operation in x and y, and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} e^{-(x^2+y^2)/2\sigma^2}.$$

Therefore, we can derived $D(x, y, \sigma)$ by computed the difference of two nearby scales separated by a constant multiplicative factor k :

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma).$$

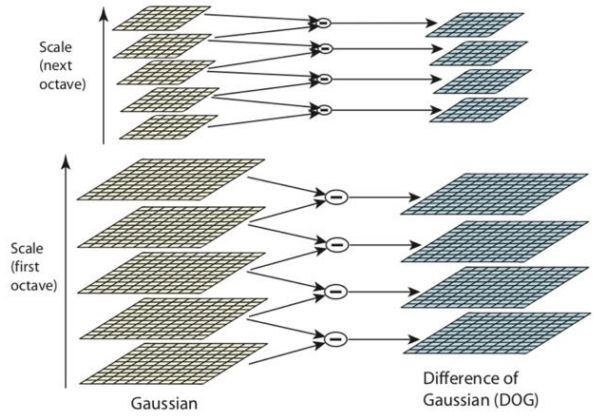

Figure 2 The Gaussian Pyramid and generating the difference of Gaussian

After derived the $D(x, y, \sigma)$, we need to detect the local maxima and minima of $D(x, y, \sigma)$. Each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below. However, we may find too many feature points in various scale space and this is costly. Therefore, we determine the location of the key point and fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast or are poorly localized along an edge. The Taylor expansion of the scale-space function, $D(x, y, \sigma)$, can help to shift and the origin is at the sample point: more precisely and boosting the quality of panorama.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial D}{\partial x^2} x$$

Although the Taylor expansion can acquire more accurate key point localization, it is not sufficient to reject key point with low contrast. The DOG will have a strong response along edges, even if the location along the edge is poorly determined. A poorly defined peak in the DOG will have a large principal curvature across the edge but a small one in the perpendicular direction. The principal curvatures can be computed from a 2x2 Hessian matrix, H, computed at the location and scale of the key point:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigenvalues of H are proportional to the principal curvatures of D. Let α be the eigenvalue with the largest magnitude and β be the smaller one. Then, we can compute the sum of the eigenvalues from the trace of H and their product from the determinant:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$Det(H) = D_{xx} D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let γ be the ratio between the largest magnitude eigenvalue and the smaller one, so that $\alpha = \gamma\beta$. Then,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma\alpha + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma}$$

When γ is bigger, there will be a higher absolute difference between α and β, then we can discard that extreme point.

Figure 3 The original photos before the SIFT feature points detection.

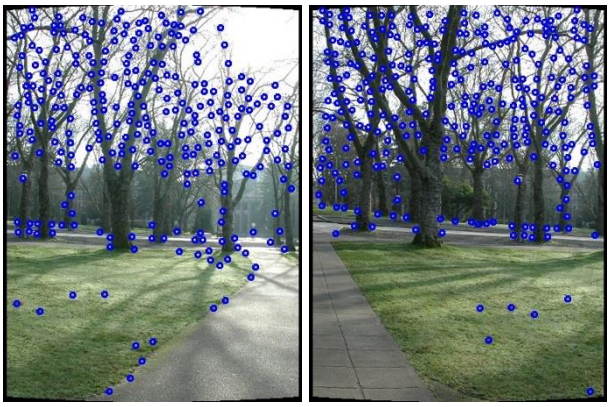

Figure 4 The photos with detected feature points by SIFT detection.

4. IMAGE MATCHING

At this stage, we are going to find all matching images and classifying and connecting the photographs sets to become panoramas. Since each photograph could match with the others, the brutal-force solution is matching every possible solution but the computation cost will be very high. Due to image geometry, it is not necessary to try every possible solution, only to match the photographs in a small number of neighbors.

The ideas to solve this problem is using the RANSAC algorithm and probabilities model function. For each pair of potentially matching images, there are the inliers with geometrically consistent and the outliners overlap but not consistent. The verification can be achieved by comparing the probabilities of the sets of inliers and outliers which generating the correct matching or the false matching.

For a given image we denote the total number of features in the area of overlap n_f and the number of inliers n_i . The event that this image match correctly/incorrectly is resented by the binary variable $m \epsilon \{0,1\}$. The event that the i^{th} feature match $f^{(i)} \epsilon \{0,1\}$ is an inlier/outlier is assumed to be independent Bernoulli, so that the total number of inliers is Binomial

$$p\left(f^{(1:n_f)}\middle| m=1\right) = B(n_i; n_f; p_1)$$
$$p\left(f^{(1:n_f)}\middle| m=0\right) = B(n_i; n_f; p_0)$$

where p_1 is the probability of a feature is an inlier given a correct image match, and p_0 is the probability of a feature is an inlier given a false image match. The set of feature match variables $\{f^{(i)}, i = 1, 2, \dots, n_f\}$ is denoted $f^{(1:n_f)}$. The number of inliers $n_i = \sum_{i=1}^{n_f} f^{(i)}$ and *B(.)* is the Binomial

$$B(x; n, p) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}.$$

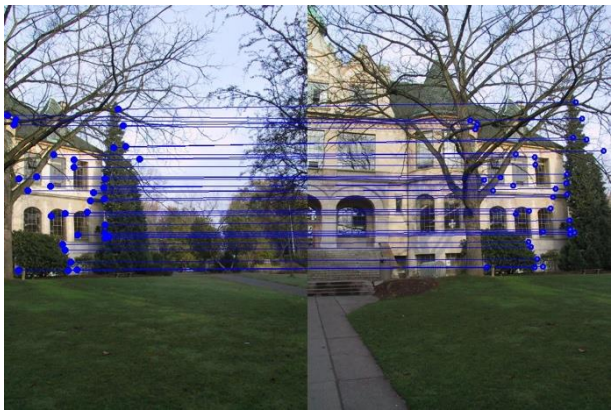

Figure 5 The photos after the RANSAC and matching points draw by the horizontal line.

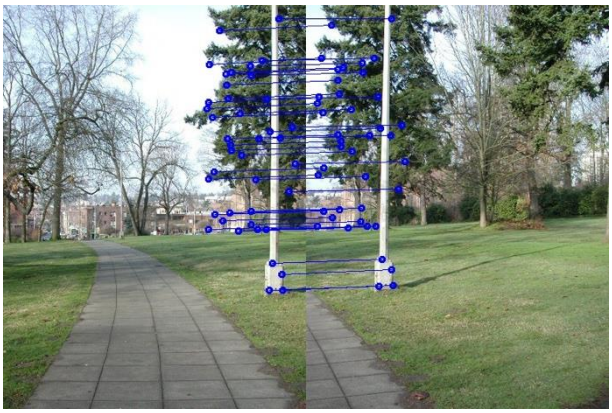

Figure 6 The photos after the RANSAC and matching points draw by the horizontal line.

5. CYLINDRICAL PROJECTION

After determining the sequence of the photograph, we project the photograph into the cylindrical surface. Each photograph may have the different focal lengths and shifting during shooting. The cylindrical projection can make each photograph have same camera parameters and the uniform coordinate will make alignment more precisely and boosting the quality of panorama.

Assume for now that the camera is in its canonical position, i.e., its rotation matrix is the identity, $R = I$, so that the optic axis is aligned with the z axis and the y axis is aligned vertically. The 3D ray corresponding to an (x, y) pixel is therefore (x, y, f). We wish to project this image onto a cylindrical surface of unit radius. Points on this surface are parameterized by an angle θ and a height h, with the 3D cylindrical coordinates corresponding to (θ, h), given by

$$(\sin\theta\,, h,\, \cos\theta) \propto (x, y, f)$$

as shown in Figure 5. From this correspondence, we can compute the formula for the warped or mapped coordinates more precisely and boosting the quality of panorama.

$$x' = s\theta = s\, tan^{-1}\frac{x}{f}$$
$$y' = sh = s\frac{y}{\sqrt{x^2 + f^2}}$$

where *s* is an arbitrary scaling factor (sometimes called the radius of the of the cylinder) that can be set to $s = f$ to minimize the distortion (scaling) near the center of the image.

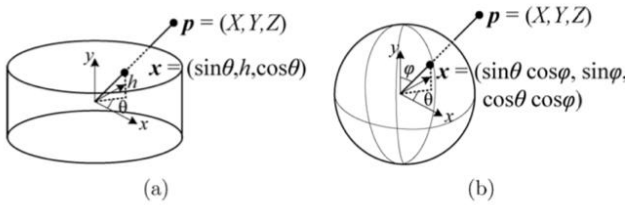

Figure 7 Projection from 3D to cylindrical and spherical coordinates.

The inverse of this mapping equation is given by

$$x = f\,tan\,\theta = f\,tan\frac{x'}{s}$$

$$y = h\sqrt{x^2 + f^2} = \frac{y'}{s} f\sqrt{1 + tan^2\,x'/s} = f\frac{y'}{s} sec\frac{x'}{s}$$

Images can also be projected onto a spherical surface, which is useful if the final panorama includes a full sphere or hemisphere of views, instead of just a cylindrical strip. In this case, the sphere is parameterized by two angles $(\theta, \emptyset)$, with 3D spherical coordinates given by

$$(sin\,\theta\,cos\emptyset, sin\emptyset,\ cos\,\theta cos\emptyset) \propto (x, y, f)$$

The correspondence between coordinates is now given by

$$x' = s\theta = s\tan^{-1}\frac{x}{f}$$

$$y' = s\emptyset = s\,tan^{-1}\frac{y}{\sqrt{x^2 + f^2}}$$

while the inverse is given by

$$x = f\tan\theta = f\tan\frac{x'}{s}$$

$$y = \sqrt{x^2 + f^2}tan\emptyset = tan\frac{y'}{s} f\sqrt{1 + \tan^2 x'/s}$$

$$= ftan\frac{y'}{s} sec\frac{x'}{s}.$$

Figure 8 The original photos before the inverse wrapping by cylindrical projection.

Figure 9 The photographs after the inverse wrapping by cylindrical projection.

5. BUNDLE ALIGNMENT

Given a set of geometrically consistent matches between the photographs, we need to solve the camera parameters jointly. This is an essential step when concatenating the pairwise photographs and the accumulated errors will decrease the quality of the panoramas. Therefore, when stitching photograph one by one, we need to add the bundle adjustment to it. The new photograph is initialized with the same rotation and focal length as the photograph to which it best matches.

After the adjustment of cylindrical projection, we acquire the same camera parameters. However, directly add the photograph one at a time will lead to the accumulated errors and the presence of gap or overlapping. [11] This can be fixed by stretching the alignment of all the photograph using a process called gap closing. [12] However, a better solution is to simultaneously align all the photograph together using a least squares framework to correctly distribute any mis-registration errors and this is called bundle alignment [13].

We formulate the problem of global alignment using a feature-based approach, since this results in a simpler system. Given a correspondence $u_i^k \leftrightarrow u_j^l$ (u_i^k denotes the position of the kth feature in image i), the residual is

$$r_{ij}^k = u_i^k - p_{ij}^k$$

where p_{ij}^k is the projection from image j to image i of the point corresponding to u_i^k

$$p_{ij}^k = K_i R_i R_j^T K_j^{-1} u_j^l$$

The error function is the sum over all images of the robust residual errors

$$e = \sum_{i=1}^{n} \sum_{j\epsilon\delta(i)} \sum_{k\epsilon\delta(i,j)} h(r_{ij}^k)$$

where n is the number of images, $\delta(i)$ is the set of images matching to image i, $\delta(i,j)$ is the set of feature matches between images i and j. We use a Huber robust error function.

$$h(x) = \begin{cases} |x|^2, & if\ |x| < \sigma \\ 2\sigma|x| - \sigma^2, & if\ |x| \geq \sigma \end{cases}$$

This error function combines the fast convergence properties of an L_2 norm optimization scheme for inliers (distance less than σ), with the robustness of an L_1 norm scheme for outliers (distance greater than σ). We use an outlier distance $\sigma = \infty$ during initialization and $\sigma = 2$ pixels for the final solution.

6. IMAGE BLENDING

Image blending is the final stage of panorama stitching. However, image blending is the most critical part in the procedure and deciding the quality of the output panorama. From the result of previous steps, we already determine the sequence of the photograph and the position to align them, but how to align the photograph is complicated and studied for years. Due to the limited ability of human eyes and the preference of the people is very diverse and inconsistent. However, there are some obvious flaws to be discovered by the users. For example, the ghost eliminating [14], solving the stitching errors [15], removing the color shifting [16] are frequently problems that may happen when aligning and blending.

Ideally each pixel along a ray would have the same intensity in every image that it intersects, but in real world is not easy. There are a number of reasons for this. For example, the change in exposure time, the vignetting effect which is the intensity decreases towards the edge of the image, the parallax effects due to unwanted motion of the optical center, and any mis-registration errors due to mis-modelling of the camera, radial distortion etc. Because of this a good blending strategy is important.

In order to combine information from multiple images we assign a weight function to each image:

$$w(x,y) = w(x)w(y)$$

where $w(x)$ varies linearly from 1 at the center of the image to 0 at the edge. A simple approach to blending is to perform a weighted sum of the image intensities along each ray using these weight functions. However, this can cause blurring of high frequency detail if there are small registration errors. To prevent these, we have applied the multi-band blending strategy developed by Burt and Adelson [17]. The idea behind multi-band blending is to blend low frequencies over a large spatial range, and high frequencies over a short range. This can be performed over multiple frequency bands using a Laplacian Pyramid in Figure 10.

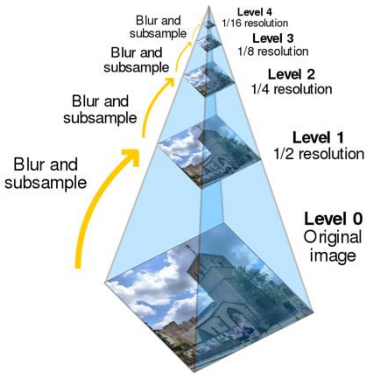

Figure 10 The procedure of Laplacian pyramid.

In our implementation, we have used a simple 2 band scheme. A low-pass image is formed with spatial frequencies of wavelength greater than 2 pixels relative to the rendered image, and a high-pass image with spatial frequencies less than 2 pixels. We then blend the low-frequency information using a linear weighted sum, and select the high-frequency information from the image with the maximum weight.

Whilst it would be desirable to use more frequency bands in the blending scheme, an open problem is to design suitable spline functions for arbitrarily overlapping images.

7. EXPERIMENT STEPS

The experiment needs some image database. There are many open databases on the Internet. As a result, we can evaluate the panorama by these open data. We download the famous photographs sample on the Internet in Figure 11. At the beginning, we need to do the feature detection and feature matching in Figure 12. The SIFT algorithm is a robust algorithm to find the feature points. After detecting the key points in photograph, the inlier technique and RANSAC algorithm can match the photographs excellently in Figure 13. Finishing the above processes, we start to align the photographs together. The alignment usually involves the image warping and image blending. The backward image warping algorithm can transform the better result in Figure 14. However, the image blending is another challenge. Since the two photographs may have the shaking and lighting problems. We put the effort on blending to make the result more obvious and delightful. Therefore, our panorama has a better result and smooth boundary.

Figure 11 The original photographs without any processing.

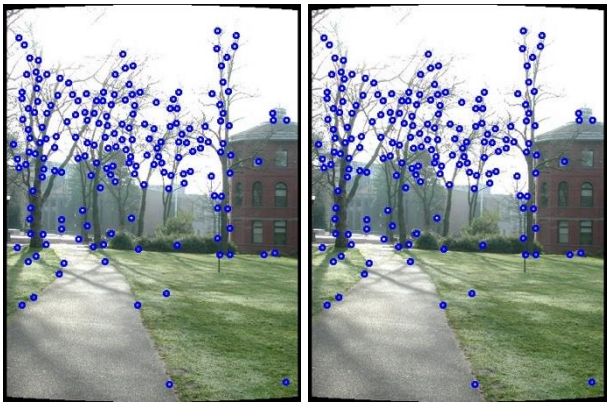

Figure 12 The photographs with detected feature points by SIFT detection.

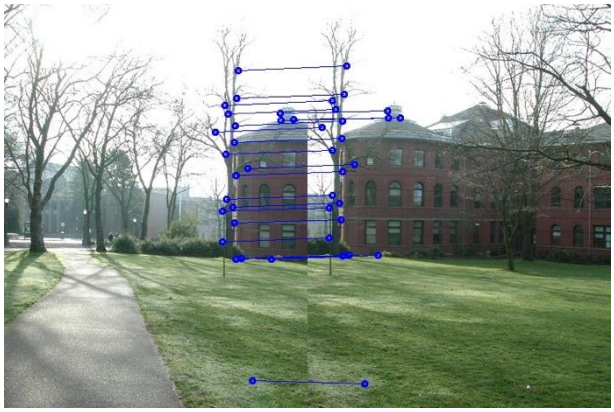

Figure 13 The photographs after the RANSAC and matching points draw by the horizontal line.

Figure 14 The photographs after the inverse wrapping by cylindrical projection.

Figure 15 The result of stitching photographs.

8. RESULT

We follow the procedure of the image stitching and use the robust algorithm currently: modifying the algorithms and fitting our experiment and programming. After the sequence of the experiment, we stitch the photographs together, and the boundary of each photograph is smooth in Figure 16. Moreover, our results of the panorama stitching will be compared with some artifacts on the Internet in Figures 17 and 18. Since we try to find a more robust blending technique and do the repeatable experiments. We hope our result may defeat them, especially on the image edges. Although, human preference is very different, our photograph does not have apparently ghost effect or the color shift even the stitching errors.

9. FUTURE WORKS

We acquire a robust panorama stitching method from the several times trial and errors and finally obtain a delightful photograph. The program outputs the panorama from two or three photographs only need in

Figure 15 The panorama we took in real world.

Figure 16 The panorama from open dataset (1).

Figure 17 The panorama from open dataset (2).

seconds, however, the whole procedure takes much time when stitching a sequence of photos. Since the SIFT and RANSAC process take times, we need to reduce the more time in these processes or use other algorithms. Therefore, we could accelerate the program and get a panorama faster.

REFERENCES

[1] R. Szeliski, "Image Alignment and Stitching: A Tutorial," Technical Report MSR-TR-2004-92, Microsoft Research, 2004.

[2] D. Milgram, "Computer Methods for Creating Photomosaics," *IEEE Transactions on Computers*, C-24(11):1113–1119, 1975.

[3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

[4] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image analysis and automated cartography." *Communications of the ACM*, 24:381–395, 1981.

[5] He, K., Chang, H, and J. Sun, "Rectangling Panoramic Images via Warping." ACM Transactions on Graphics – SIGGRAPH 2013 Conference, 2013

[6] Murtadha Alomran and Douglas Chai, "Feature-Based Panoramic Image Stitching." 14th International Conference on Control, Automation, Robotics & Vision, 2016.

[7] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features." International Journal of Computer Vision, 2007.

[8] M. Brown and D. Lowe, "Recognising panoramas." In *Proceedings of the 9th International Conference on Computer Vision (ICCV03)*, Vol 2, pp. 1218–1225, Nice, 2003.

[9] M. Brown and D. Lowe, "Object Recognition from Local Scale-Invariant Features" International Conference on Computer Vision, 1999.

[10] Harris, C., Stephens, M., "A combined corner and edge detector." In: Proceedings of the Alvey Vision Conference, pp. 147–151 ,1988.

[11] H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion correction." IEEE Transactions on Pattern

Analysis and Machine Intelligence, vol. 21, no. 3, pp. 235–243,1999

[12] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and texture-mapped models." in Computer Graphics, pp. 251–258, 1997.

[14] Uyttendaele, M., Eden, A., and Szeliski, R. "Eliminating ghosting and exposure artifacts in image mosaics." in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR01)*, vol. 2, pp. 509–516, 2001.

[15] Herrmann C, Wang C, Bowen R S, Keyder E, Krainin M, Liu C, Zabih R. "Robust Image Stitching with Multiple Registrations." in Computer Vision – ECCV. Springer International Publishing, pp.53–69, 2018.

[16] Sadeghi, M., Heirati, S., Gheissari, N. "Poisson local color correction for image stitching." In International Conference on Computer Vision theory and Applications, pp. 275–282, 2008.

[17] P. J. Burt and E. H. Adelson. "A multiresolution spline with application to image mosaics." in *ACM Transactions on Graphics*, 2(4):217–236, 1983.