We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



2

High-Quality Seamless Panoramic Images

Jaechoon Chon, Jimmy Wang, Tom Slankard and John Ristevski
Earthmine Inc., Berkeley, USA

1. Introduction

Image mosaicing is the creation of a larger image by stitching together smaller images. Image mosaicing has many different applications, such as satellite imagery mosaics (Chon et al., 2010,) the creation of virtual reality environments (Szeliski, 1996; Chon et al., 2007; Brown and Lowe, 2007,) medical image mosaics (Chou et al., 1997,) and video compression (Irani et al., 1995; Irani and Anandan, 1998; Kumar et al., 1995; Lee et al., 1997; Teodosio and Bender, 1993).

Image mosaicing has four steps: 1) estimating relative pose among smaller images to project on a plane or specific defined surfaces, 2) projecting those images on the surface, 3) correcting photometric parameters among projecting images, and 4) blending overlapping images.

The first step has two categories: 2D plane images (For example, scanned maps in sections and orthoimages) based methods and perspective image based methods. Orthoimages are generated by correcting each image pixel from a perspective image with a help of a Digital Elevation Model (DEM,) produced by a stereo camera system, a laser/radar aerial system or obtained of photogrammetric works. To align the images with the DEM, we need to estimate absolute orientation with ground control points (GCP.) In the case of perspective images, we need to estimate relative pose using homography, affine transformation, colinear conditions, and coplanner condition with feature correspondences. (Hartley and Zisserman & Books, 2003; McGlone et al. & Books, 2004; Szeliski & Books, 2011)

The second step projects all images onto a specific defined surface, such as 2D plane, cylinder (Chen, 1995; Szeliski, 1996), sphere (Coorg and Teller, 2000; Szeliski and Shum, 1997), and multiple projection planes (Chon et al., 2007.) In the case of cylinder and sphere, we assume that images are captured by fixing the position of a camera and rotating it at the position. And all images are projecting onto a cylinder or a sphere surface.

In the third step, we have to balance photometric parameters among projected images because exposure affects the photometric parameters of each image. At the last step, we have to create one image from multiple overlapping images.

Ideally each sample (pixel) along a ray would have the same intensity in every image that it intersects, but in reality this is not the case. Even after gain compensation some image edges are still visible due to a number of un-modeled effects, such as vignetting (the decrease of intensity towards the edge of the image,) parallax effects due to unwanted motion of the optical center, registration errors due to an incorrect or approximate camera model, radial distortion, and so on.

Because of this, a good blending strategy is important (Brown and Lowe, 2007; Goldman, 2011; Kim and Pollefeys, 2008; Zomet et al., 2006.) In particular, applying seam-line detection algorithms before applying blending algorithms is an effective strategy when using image data that has parallax effects, because a seam-line detected by optimal path finding algorithms passes equal-depth pixels or similar colors pixels overlapping images.



a) Vehicle system



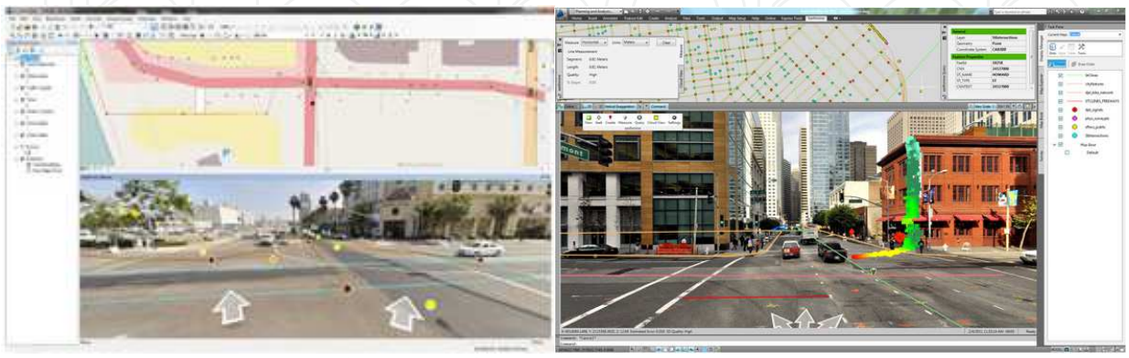
b) Quad cycle system



c) 4 stereo fish-eye images



d) Image and 3D range panoramas



e) Applications of ArcGIS and AutoCAD Map3D

Fig. 1. EARTHMINE Inc.'s 3D mobile mapping systems, 3D panorama generated from eight fish-eye images captured using the system, and its applications.

In this chapter, we introduce luminance balancing using gamma correction, Dijkstra's algorithm for seam-line detection, our proposed fast color blending algorithm on seam-lines, and color balance and saturation implemented on EARTHMINE Inc.'s mobile mapping system (MMS) that generates 3D panoramic images as shown in Fig 1.

2. Color balance

A major challenge in merging multiple shots into a single panorama is that each individual image experiences its own exposure setting, even if each camera (when there are multiple cameras) is color calibrated. A common scenario that may cause such discrepancy is when the shots are taken, some cameras face towards the light source (e.g. the sun), and others face away from the light source. Due to the different exposure to the physical light source, some images will be brighter than others. Merging these images together inevitably causes a visible seam at the merging points.

To overcome this problem, we describe an algorithm that detects the differences in the luminance and chrominance in each image and attempts to equalize such differences. First, we assume that there exist overlapping regions between neighboring images. For the set of images, we assume there exist a single correction factor per channel (one luminance channel that governs image intensity and two chrominance channels that governs the color of the image) per image such that the luminance/chrominance of the overlapping areas match up.

The ideal approach would be to find corresponding pixels between the overlapping areas and find the optimal correction factors using these correspondences. Specifically, we can write the optimization process in the following form

$$\arg\min\left[\sum_p \left(\gamma_n I_{np} - \gamma_m I_{mp}\right)^2\right] \tag{2.1}$$

where I_{np} and I_{mp} are the p^{th} corresponding channel value for image I_n and I_m respectively; γ_n and γ_m are the correction factor for image n and m. A simple illustration is shown in figure 2.1.



Fig. 2.1. An illustration of a subset of the correspondences between two images in the overlapped region.

However, finding pixel correspondences between pairs of images is non-trivial and most currently available algorithms such as SIFT (Lowe, 2004) are very computationally expensive. For applications that have limited processing power (e.g. image stitching on mobile platforms) or real time processing requirements, such approach is inadequate.

Brown and Lowe (2007) and Xiong and Pulli (2010) pointed out that because luminance/chrominance is a global entity, instead of using pixel correspondences, we can approximate them using the mean luminance / chrominance value between the overlapping region. Eq(2.1) thus becomes

$$\arg\min\left[\left(\gamma_n E[I_n]-\gamma_m E[I_m]\right)^2\right] \tag{2.2}$$

where E[In] and E[Im] denotes the average luminance / chrominance values of the overlapped image regions from image n and m respectively.

Note that there is a trivial solution to Eq(2.2), where when the γs are set to 0, the objective function is at its minimum. To prevent this, we add a regularization term in the form of 1 - γ, such that this term is non-zero when γ is 0. In addition, this term prevents the luminance / chrominance from changing too much, which is often desired. The objective function thus becomes

$$\arg\min_{\gamma_n,\gamma_m}\left[\omega_d\left(\gamma_n E[I_n]-\gamma_m E[I_m]\right)^2+\omega_p\left(1-\gamma_n\right)^2+\omega_p\left(1-\gamma_m\right)^2\right] \tag{2.3}$$

where w_d and w_p are weights for the two terms.

The human visual system perceives brightness according to the Weber-Fechner law where the perceived brightness is not a linear function of the physical brightness but rather follows a log function (red line in figure 2.1; blue line indicates a linear response).

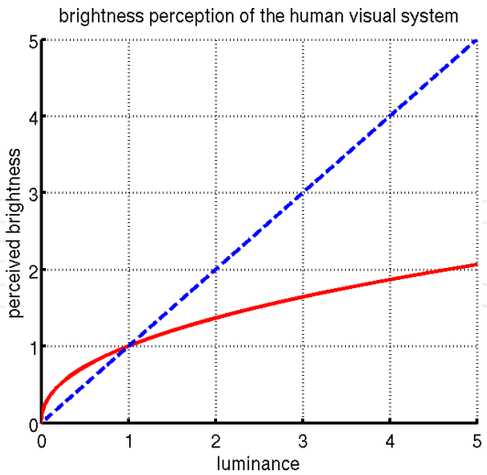


Fig. 2.2. An illustration of Weber-Fechner law of brightness perception.

To account for this effect, we linearize the luminance channel with a power function such that the brightness perception curve is linear (some cameras have this linearization in the post-processing step. If so this step can be omitted). The objective function for the luminance channel becomes

$$\arg\min_{\gamma_n,\gamma_m}\left[\omega_d\left(\gamma_n\frac{1}{P}\sum_P\left(I_{np}\right)^{2.2}-\gamma_m\frac{1}{P}\sum_P\left(I_{mp}\right)^{2.2}\right)^2+\omega_p\left(1-\gamma_n\right)^2+\omega_p\left(1-\gamma_m\right)^2\right] \quad (2.4)$$

where P is the number of overlapping pixels.

We used a correction factor of 2.2 according to common practice (Poynton 2002). Xiong and Pulli (2009) also used a very similar objective. The final objective over all the images is thus

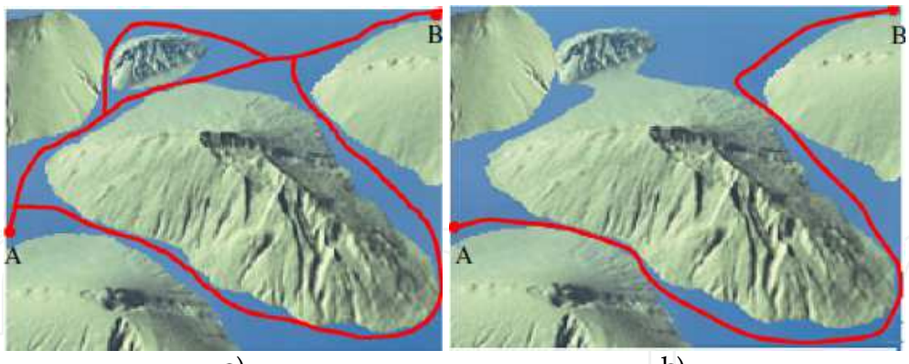
$$\arg\min_{\gamma_1,\cdots,\gamma_\kappa}\left[\sum_{k=0}^{\kappa}\left(\omega_d\left(\gamma_k E[I_k^l]-\gamma_{k-1}E[I_{k-1}^l]\right)^2+\omega_p\left(1-\gamma_k\right)^2\right)\right] \quad (2.5)$$

In the above equation, we assume that there are only two overlapping images (left and right) for a given image. While this is often the typical scenario, it is straightforward to generalize to any arbitrary amount of images. This objective function thus can be optimized with any of-the-shelf optimization package.

3. Seam line detection

To create a seamless panorama from a multiple stereo fish-eye camera system, with a system like the EARTHMINE Inc, is necessary an optimal seam-line detection algorithm (see Figs. 1(c) and 1(d).)

Dijkstra's algorithm to find an optimal path within a cost space (Bellman, 1957; Dijkstra, 1959.) A seam-line can be detect by using the Dijkstra's algorithm with a cost map built by a cross correlation between two overlapping images (Milgram, 1975, 1977; Davis, 1998; Efros and Freeman, 2001.)



a) b)

Fig. 3.1. Chon et al. (2010) proposed seam-line algorithm explained with the waterway on the terrain structure. a) High water level creates many possible paths between two points A and B. b) Lowered water level makes the maximum cost smaller.

Chon et al. (2010) proposed a novel algorithm for selecting a seam-line that at first identifies a subarea within two overlapping images such that in the subarea, at least a seam line exists and the maximal mismatch score in this subarea is minimized. Let the degree of mismatching of pixels between two images on the overlapped region form a cost field, which can be expressed in a 3D plot. If a threshold is set with all cost values under the

threshold filled with ``water'', then the 3D plot will look like the ones in Fig. 4, where Fig. 3.1(a) uses a larger threshold. The plots show some ``water ways'' between points A and B, and the threshold defines the water level. Fig. 3.1(b) shows that the allowed path is not near the shortest one. The technique then further applies Dijkstra's algorithm to find an optimal path within the restricted subspace. In this optimization phase, a cost conversion is applied to make a higher cost (the mismatch score) larger. This enables the search to find a possibly longer seam-line with less highly mismatched pairs.

Kerscher (2001) proposed a method called the "twin snake algorithm" to detect seam-lines. The algorithm starts with two initial vertical lines as control points of the snakes on the overlapping images. The snakes have two energy terms which are internal and external energies in general (Kass et al., 1987; Leymarie and Levine, 1993; Tiilikainen, 2007; Williams and Shah, 1990). The sum of the mismatching values on those lines and the relationship between neighbor control points are called *internal energy* and *external energy*, respectively. The twin snake algorithm builds an energy function E^*_{snake} from four terms: the internal term E_{int}, the photometric term E_{pho}, and the external force E_{ext}. The energy is calculated for each vertex $v(s)$ and integrated over the whole length of the snake:

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(v(s))ds = \int_0^1 \left(E_{\text{int}}(v(s)) + E_{\text{pho}}(v(s)) + E_{\text{ext}}(v(s)) \right) ds \tag{3.1}$$

The internal energy E_{int} tries to preserve a smooth shape for the curve. Photometric energy E_{pho} usually evaluates edge strength or similar measures in the examined image and tries to pull the snake to salient image features. External energy E_{ext} can be introduced by user interaction and are responsible for globally controlling and guiding the snake evolution (Kerscher, 2001.) The curve with minimum energy as shown in Fig. 3.2 is determined to be the optimal seam line.

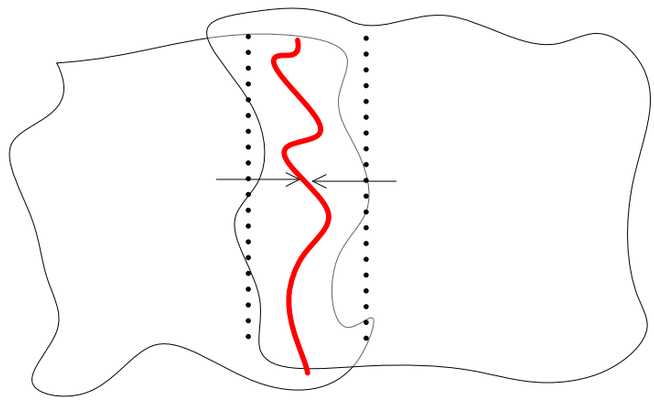


Fig. 3.2. Initial vertical lines as control points (dotted line) and a detected seam line (curved line).

This algorithm cannot completely overcome the local minima problem, and it requires a high computation load. Even though Chon et al. (2010) avoid the local minima problem, it requires a high computation cost as well because of finding the best threshold.

3.1 Cost space

We adopts Dijkstra's algorithm to find an optimal seam-line on a cost space built by using the normalized cross correlation (*NCC*) between two overlapping images for pixel (*u,v*) (Lewis, 1995). The *NCC* has been widely using image matching for stereo matching, feature tracking, etc.

$$NCC(u,v) = \frac{\sum_{i=u-w/2}^{u+w/2}\sum_{j=v-h/2}^{v+h/2}\left(I(i,j)-I_{uv}\right)\left(I'(i,j)-I'_{uv}\right)}{\sqrt{\sum_{i=u-w/2}^{u+w/2}\sum_{j=v-h/2}^{v+h/2}\left(I(i,j)-I_{uv}\right)^2\sum_{i=u-w/2}^{u+w/2}\sum_{j=v-h/2}^{v+h/2}\left(I'(i,j)-I'_{uv}\right)^2}} \tag{3.2}$$

where I_{uv} and I'_{uv} are averages of each image in a 5×5 window.

The *NCC* between two images for pixel (u; v) is computed using the 5×5 subimages as in Eq. (3.2). Note that *NCC* has a range of [-1.0, 1.0]. A cost (degree of mismatch) at pixel *(u, v)*, *cost(u, v)*, is defined as

$$\cos t(u,v) = \{1.0 - NCC(u,v)\} / 2.0. \tag{3.3}$$

The cost value approaches 0.0 for similar pixel points and 1.0 for dissimilar pixels.

3.2 Dijkstra's algorithm

Dijkstra's algorithm is a global optimization technique that determines the optimal path on the cost space by taking the local minimum operation at each node. To apply to optimal seam line searching, each pixel in an overlapping area is associated with a node, which has 8 neighboring nodes, with four in diagonal directions.

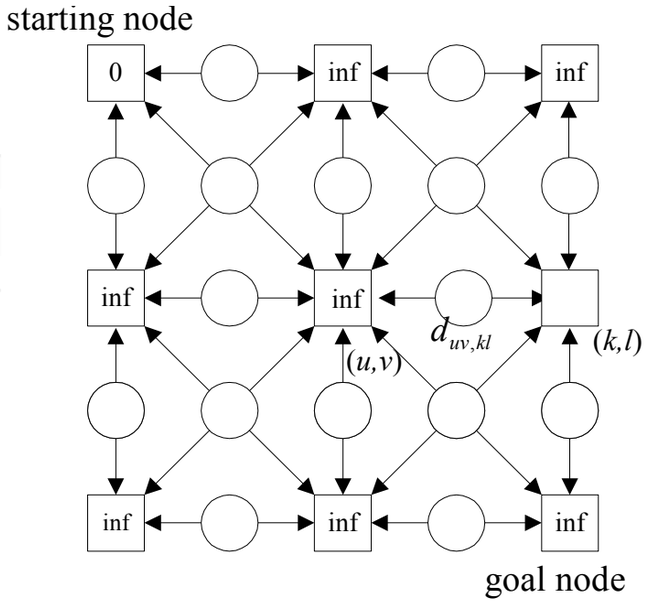


Fig. 3.3. The cost $d_{uv,kl}$ between two nodes (*u,v*) and (*k,l*) built by using *NCC*.

Let the node at which we are starting be called the initial node. Let (u,v) specify a node and (k,l) be a neighboring node of node (u,v). NBR(u,v) indicates the set of neighboring nodes of node (u,v). Let the cost $d_{uv,kl}$ be a path cost between two nodes (u,v) and (k,l). Let the global minimum cost $Di(u,v)$ be a accumulated cost from the starting node to (u,v). Dijkstra's algorithm will assign some initial costs to avoid some areas and will try to improve them step by step.

Assign to every node a tentative cost Di: set it to zero for the starting node and to infinity for all other nodes.

1. Mark all nodes as unvisited. Set the starting node as current. For current node (u,v), consider all its unvisited neighbors and calculate their tentative cost Di.

$$Di(u,v) = \min\left\{d_{uv,kl} + D_i(k,l); (k,l) \in NBR(u,v)\right\} \tag{3.4}$$

   For example, if current node (u,v) has cost of 6, and a path cost $d_{uv,kl}$ is 2, the cost to node (k,l) through node (u,v) will be 6+2=8. If this cost is less than the previously recorded cost, overwrite the cost. All unvisited neighbors are added to an unvisited set.
2. When we are done considering all neighbors of the current node, mark it as visited. A visited node will not be checked ever again; its cost recorded now is final and minimal.
3. The next current node will be the node with the lowest cost in the unvisited set.
4. If all nodes have been visited, finish and then find the path with the minimum cost between the starting and goal nodes.

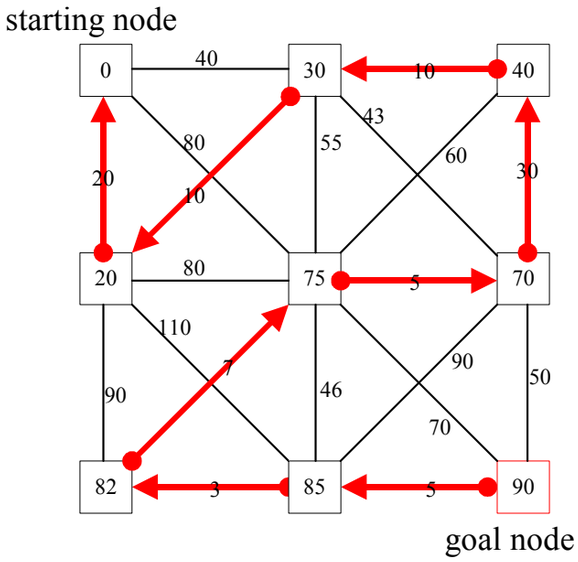


Fig. 3.4. Finding the path with the minimum cost from the goal to staring nodes.

Fig. 3.5 shows four seam-lines detected on four overlapping images using Dijkstra's algorithm on the cost spaces. Each overlapping image is generated from fish-eye images as shown in Fig. 1(c) using a sphere projection method (Coorg and Teller, 2000; Genner, 2006; Kim et al., 2004; Szeliski and Shum, 1997; Yakimovsky and Cunningham, 1978.) The size of each overlapping image is about 30 % of a panoramic image.

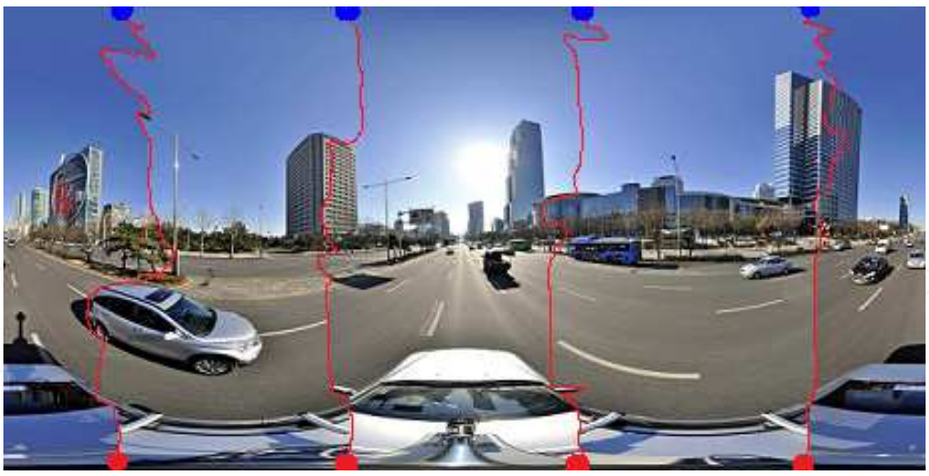


Fig. 3.5. Four seam-lines detected on four overlapping images in a panorama; Blue and red points are starting and goal nodes, respectively.

4. Fast color blending on seam-lines

Even though color transitions among all overlapping images are smoothed, the color matching only provides an approximate match. Because color differences among the images corrected by using the color matching method are not enough, Xiong and Pulli (2010) proposed an effective blending method with fast processing speed and high blending quality.

The method calculates the color differences between each p and all points $D(p)$ on each of a seam lines m_c and then interpolate $D(p)$ at a pixel q of the blending image (see Fig. 4.1(a)). Finally, the color value $C(q)$ at pixel q are changed as $C(q) = C(q) + D(p)$.

$$D(q) = \sum_{i=1}^{n} w_i(q) D(p_i) \tag{4.1}$$

Where i is an index of the seam points consisting of a seam-line m_c and weights $w_i(q) = \dfrac{1/\|p_i - q\|}{\sum_{j=1}^{n} 1/\|p_j - q\|}$ are the inverse coordinate distance to the boundary pixels.

Additionally, they proposed the creation of a color difference distribution process to enforce color consistency for 360-degree panorama. They attenuate the color of the pixel q in the blending area on the current scan line with

$$C(q) = C(q) + \left(1 - \frac{x}{x_b}\right) D(q) \tag{4.2}$$

where x and x_b are the horizontal distances between q and the seam point p of the seam line m_s and between p and the end of the blending area, respectively, in Fig. 4.1(b).

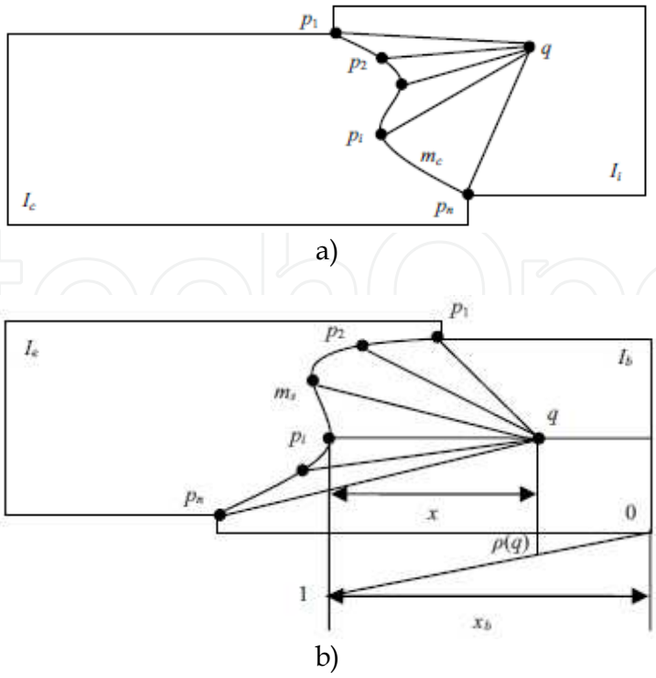


Fig. 4.1. Image blending on a seam line and color difference distribution for 360-degree panoramic images (Xing and Pulli, 2010).

4.1 The proposed method

If overlapping images are well aligned, Xiong and Pulli (2010) proposed method works perfectly. However, poor image alignment due to parallax, registration errors, and radial distortions can lead to poor blending results in general.

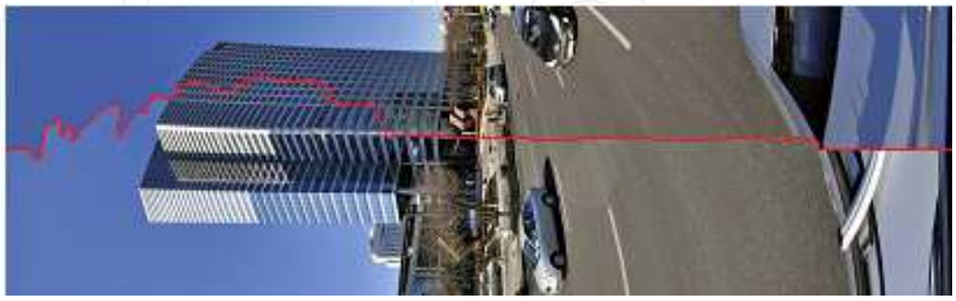
To cope with this problem, we propose filtered color differences using multiple major color differences detected on a seam-line. When the seam points are well aligned, the difference between the neighboring points will be smoothly changed. Because seam points with misalignment cause sudden changes, we simply filter out these seam points and keep good aligned seam points. And we then replace the misaligned seam points with interpolated color differences using the ends of two neighbor seam points on a seam-line. We detect smoothly changed color differences using a median filter and the standard deviation of the changes.

4.2 Detection of good aligned seam points

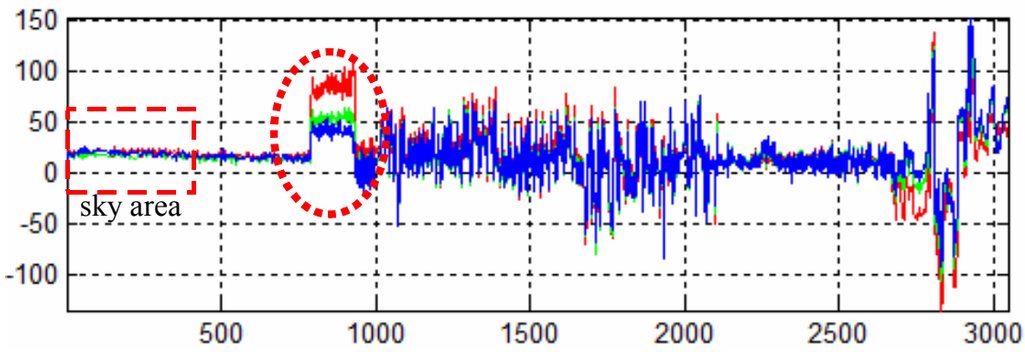
Fig. 4.2(a) shows an overlapping image that is located at the rightmost side of the panoramic image shown in Fig. 3.5. Fig. 4.2(b) shows a graph of color differences corresponding to the seam points of the seam line shown in Fig 4.2(a). If the seam points are well aligned, the differences will be smooth like the differences for seam points in the sky area.

Before applying Eq (4.2), we have to remove the color differences of poorly aligned seam points. To detect smooth changing parts, we use a threshold based on the standard deviation of the color differences. The graph in Fig 4.2(c) depicts the standard deviations of the color differences for each channel. The standard deviation of poorly aligned seam points is higher than that of the sky and road.

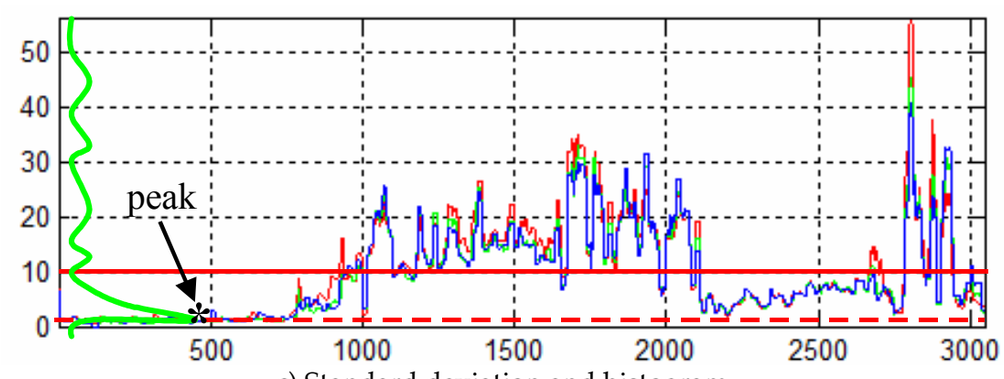
To evaluate whether seam points are poorly aligned or well aligned, we first build a histogram that accumulates the standard deviation in the horizontal direction, shown in Fig 4.2(c) as the thick green curve, and then find the peak point '*' in the thick green. A dotted red horizontal line in Fig. 4.2(c) chosen by the peak point is a reference standard deviation. We add a margin into the reference standard deviation to calculate a threshold, a red horizontal line in Fig. 4.2(c).



a) A seam-line



b) Color differences



c) Standard deviation and histogram

Fig. 4.2. The color differences and standard deviation of seam points; '*' is the most peak point in a histogram and the dotted and solid horizontal lines are the reference standard deviation and threshold, respectively, in c).

We cannot remove the color differences in a dotted circle in Fig. 4.2(b) using only this single threshold, because the standard deviation of the color differences of seam points in the circle

is low like the reference standard deviation. To remove the color differences of those seam points on the seam-line, we use the same process to build the second histogram using the color differences of the remaining seam points, which are determined by using the first histogram built by using the standard deviation.

After detecting a peak point on the second histogram as a reference of color difference, two thresholds are calculated by adding and subtracting a margin. In our experiments, the margin is set as 15. If a single reference is only applied, bad blending will occur as shown in Fig 4.3.



Fig. 4.3. Our extended algorithm with a single peak point

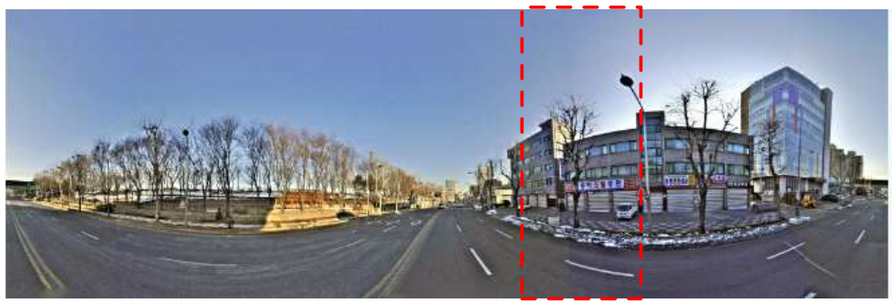
To suppress this problem, we choose multiple peak points in the histogram. If a peak point is including over 20% of the number of all seam points on the seam-line, the peak point becomes one of the references of color difference. Figs. 4.4(a) and 4.4(b) show color blending results by using one and two peak points chosen by our algorithm with each second histogram, respectively.

As the final step, we have to replace the color differences of seam points detected as poorly aligned seam points during the previous step. A line or curve is built by using both two ends of neighbor seam points from well-aligned seam points.

Figs. 4.5(a) and 4.5(b) show panoramic images created by using method of Xiong and Pulli (2010) and our extended method, respectively. When comparing the area in the dotted circle in Fig. 4.5(a), our extended method made a better result then that by Xiong and Pulli's method. However, our extended method did not perform as well within the dotted square in Fig. 4.5(b). This is originated from big differences among the values of the red, green, and blue channels of one or two end points of the remained seam points as reference color differences. If we change color differences into gray differences, this phenomenon will be disappear. However, subtle color differences among RGB channels could not recovered. We apply a median filter with a couple of seam points around the two end seam points to remove the incorrect differences. Fig. 4.6(a) shows an image including the phenomenon that is an overlapping image in the dotted square in Fig. 4.5(b) and a graph of color differences. Figs. 4.6(b) and 4.6(c) show a graph of filtered color differences and a final result using the filtered color differences, respectively.



a) A single peak point



b) Two peak points

Fig. 4.4. A panorama re-created by our extended seam-line blending algorithm with multiple peak points chosen in a histogram of color differences



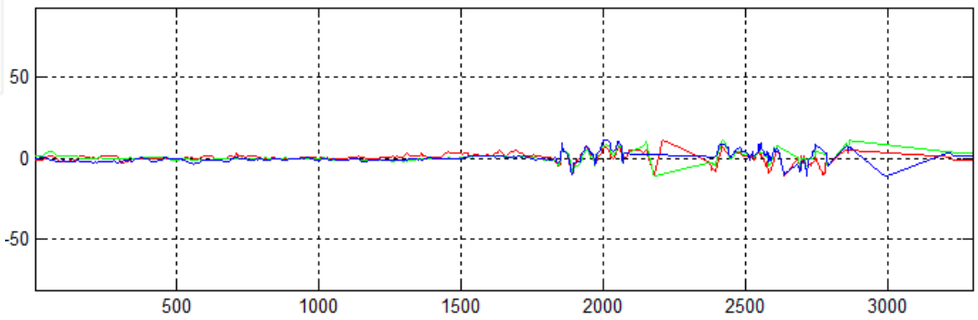
a)



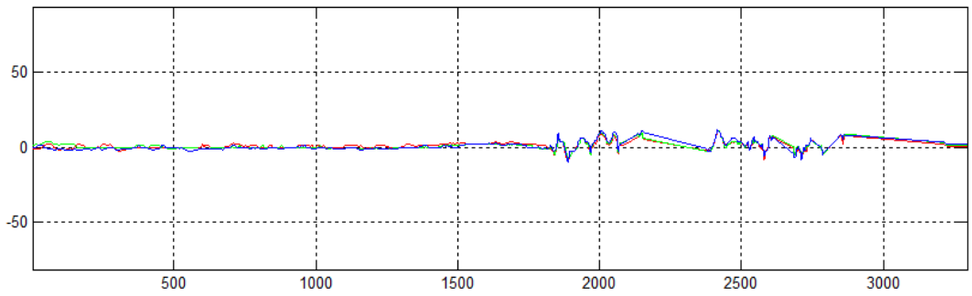
b)

Fig. 4.5. Comparison between a) Xiong and Pulli (2010) method and b) The extended method.





a) Linear interpolated color differences



b) Color differences by median filter



c) Final panoramic image

Fig. 4.6. Median filter to remove incorrect interpolated color differences and a final panoramic image.

5. Color balance and saturation

To make the final panorama more aesthetically pleasing, we apply additional processing to increase contrast, color balance, and color saturation. The additional processing also has the added benefit of making the collection process more tolerant to different lighting conditions.

We achieve color/white balance by using existing auto-white balance algorithm (employed in many image-editing applications, such as Adobe Photoshop). This algorithm computes histograms for each color channel (red, green, and blue) and then in each channel "discards" the 0.5% darkest and 0.5% lightest pixel colors from the histogram and then stretches the resulting histogram to [0, 255] effectively making all pixels that fell in the discarded ranges "black" and "white" respectively.

The above process has two effects. First, it increases the apparent contrast of the image. Second, it can cause a noticeable hue shift because the algorithm operates on the color channels individually. This is desirable for panoramas taken outdoors in natural light because it makes overly "cool" scenes appear warmer and vice versa--providing a good overall aesthetic improvement robustly under varying weather conditions. On the other hand, when the algorithm processes scenes with significantly dark scenes (such as panoramas in tunnels or those taken at night under artificial lighting) the hue shift will be much more pronounced and often produces very poor results. Thus, this color balance algorithm is only applied to images that is taken under decent lightly conditions. Figure 5 illustrates the effect of this algorithm.



a) Only our extended method



b) With more Color balance and saturation

Fig. 5. Example panoramas before a) and after b) color balance and saturation adjustment.

6. Conclusion

In this chapter we introduced a novel panorama-stitching algorithm. We described each stage of the algorithmic pipeline in detail, namely a channel-correction algorithm that normalizes luminance and chrominance of the projected images prior stitching; a seam-line finding algorithm that finds the optimal transition between two overlapping images; a color blending algorithm that blends the seam-lines; and finally, a color enhancement algorithm that adjusts the overall contrast, color, and saturation of the panoramic image.

In the algorithm, we particularly address the shortcomings of previous approaches, where poor results are often observed from the color-blending algorithm due to poorly aligned points from the individual images on the seam-lines. Our proposed algorithm removes outliers on the seam-lines due to misalignment by using histograms, color differences, and standard deviation of the color differences. The outliers are replaced by linearly interpolated points using the color differences of neighbor well-aligned seam points.

We have tested the proposed algorithm on millions of outdoor panoramas and it has been proven that the algorithm is robust under most lighting and weather conditions. The algorithm described in this paper is not only bounded to Earthmine's collection system but any system that generates panorama images.

7. References

Bellman, R., (1957). Dynamic Programming. Princeton Univ. Press.

Brown, M. and Lowe, D., (2007). Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, 74(1):59-73.

Chen, S. E., (1995). QuickTime VR-an image-based approach to virtual environment naviation. In ACM SIGGRAPH 1995 Conference Proceedings, pp. 29-38, Los Angeles.

Chon, J., Fuse, T., Shimizu, E., Shibasaki, R., (2007). Three-dimensional image mo saicking using multiple projection planes for 3-D visualization of roadside standing buildings. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 37 (4):771-783.

Chon, J., Kim, H., and Lin, C.S., (2010). Seam-line determination for image mosaicking: A technique minimizing the maximum local mismatch and the global cos., ISPRS Journal of Photogrammetry and Remote Sensing, 65(1): 86-92.

Chou, J.S., Qian, J., Wu, Z., Schramm, H., (1997). Automatic mosaic and display from a sequence of peripheral angiographic images. Proceedings SPIE Medical Imaging, 3034:1077-1087.

Coorg, S. and Teller, S., (2000). Sphereical Mosaics with Quaternions and Dense Correlation. International Journal of Computer Vision, 37(3):259-273.

Dijkstra, E.W., (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1: 269-271.

Efros, A., Freeman, W., (2001). Image quilting for texture synthesis and transfer. In: Proceedings International Conference on Computer Graphics and Interactive Techniques. pp. 341-346.

Genner, D.B., (2006). Generalized camera calibration including Fish-eye lenses, Int. J. Computer Vision, 68(3): 239-266.

Goldman, D. B. (2011). Vignette and exposure calibration and compensation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12):2276–2288.

Hartley, R. and Zisserman, A., (2003). Multiple View Geometry. 2nd Edition, Cambridge University Press.

Irani, M. and Anandan, P., (1998). Video Indexing based on mosaic representations. Proceedings of the IEEE, 86(5): 905-921.

Irani, M., Hsu, S., and Anandan, P. (1995). Video compression using mosaic representations. Signal Processing: Image Communication, 7:529-552.

Kass, M., Witkin ,A., and Terzopoulos, D., 1987. Snakes: Active contour models. *International Journal of Computer Vision*. 1(4): 321-331.

Kerschner, M., (2001). Seam-line detection in colour orthoimage mosaicking by use of twin snakes. ISPRS Journal of Photogrammetry and Remote Sensing, 56 (1):53-64.

Kim, S. J. and Pollefeys, M., (2008). Robust Radiometric Calibration and Vignetting Correction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(4):562-576.

Kim, W.S., Steinke, R.C., Steele, R.D., Ansar, A.I., (2004). Camera Calibration and Stereo Vision Technology Validation Report. JPL D-27015.

Kumar, R., Anandan, P., Irani, M.,Bergen, J., and Hanna, K. (1995). Representation of scenes from collections of images. In IEEE Workshop on Representaions of Visual Scenes, pp. 10-17, Cambridge, Massachusetts.

Lee, M.-C., ge Chen, W., lung Bruce Lin, C., Gu, C., Markoc, T., Zabinsky, S. I., and Szeliski, R. (1997). A layered video object coding system using sprite and affine motion model. IEEE Transactions on Circuits and Video Technology, 7(1):130-145.

Lewis, J. P., (1995). Fast Normalized Cross-Correlation. Vision Interface, pp. 120-123.

Leymarie, F. and Levine, M.D., 1993. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 15(6): 617-634.

Lowe, D.G., (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2):91-110.

McGlone, C., E. Mikhail and J. Bethel, (2004). Manual of Photogrammetry, 5th Edition, ASPRS.

Milgram, D.L., (1975). Computer methods for creating photomosaics. IEEE Transactions on Computers, C-24(11):1113-1119.

Milgram, D.L., (1977). Adaptive techniques for photomosaicking. IEEE Transactions on Computers, C-26(11):1175-1180.

Poynton, C. ,(2002). Digital Video and HD: Algorithms and Interfaces. Morgan Kaufmann; 1st edition (December 2002)

Tepdosio, L. and Bender, W., (1993). Salient video stills: Content and context preserved. In ACM Multimedia 93, pp. 39-46, Anaheim, California

Szeliski, R., (1996). Video mosaics for virtual environments. Computer Graphics Applications 16 (3): 22-30.

Szeliski, R., (2011). Computer Vision: Algorithms and Applications. Springer.

Szeliski, R. and Shum, H.-Y., (1997). Creating full view panoramic image mosaics and texture-mapped models. In ACM SIGGRAPH 1997 Conference Proceedings, pp. 251-258, Los Angeles.

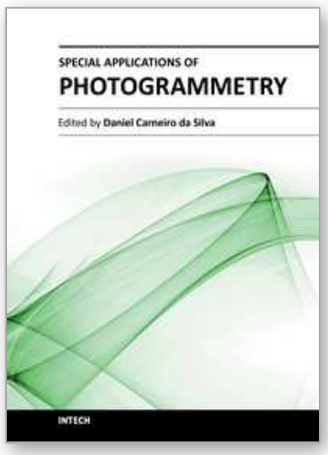
Tiilikainen, N.P., (2007). A Comparative Study of Active Contour Snakes. Copenhagen University, Denmark.

Williams, D.J. and Shah, M., (1990). A fast algorithm for active contours. *CVGIP: Image Understanding*. 55(1): 14-26.

Xiong, Y. and Pulli, K., (2010). Color Matching for High-Quality Panoramic Images on Mobile Phones. IEEE Transactions on Consumer Electronics, 56(4), 2592-2600.

Yakimovsky, Y. and Cunningham, R.T., (1978). a system for extracting three-dimensional measurements from a stereo pair of TV cameras, Computer Graphics and Image Processing, 7, 195-210.

Zomet, A., Levin, A., Peleg, S., and Weiss, Y., (2006). Seamless Image Stitching by Minimizing False Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 15(4): 969-977.



Special Applications of Photogrammetry

Edited by Dr. Daniel Carneiro Da Silva

ISBN 978-953-51-0548-0
Hard cover, 136 pages
Publisher InTech
Published online 25, April, 2012
Published in print edition April, 2012

Photogrammetry is widely accepted as one of the best surveying methods to acquire tridimensional data without direct contact with the object, but its high operational costs in equipment and personnel somewhat limit its application in mapping. However, with the development of digital photogrammetry in the 1990's, it was possible to introduce automated processes and reduce the personnel costs. In the following years, the cost of computer hardware, digital cameras and positioning sensors has been lowering, making photogrammetry more accessible to other engineering fields, such as architecture, archeology and health fields. This book shows the results of the work of researchers from different professional backgrounds, which evaluate the uses of photogrammetry, including issues of the data, processing, as well as the solutions developed for some surveying types that can be extended to many applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jaechoon Chon, Jimmy Wang, Tom Slankard and John Ristevski (2012). High-Quality Seamless Panoramic Images, Special Applications of Photogrammetry, Dr. Daniel Carneiro Da Silva (Ed.), ISBN: 978-953-51-0548-0, InTech, Available from: http://www.intechopen.com/books/special-applications-of-photogrammetry/high-quality-seamless-panoramic-images

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen