Feature-based Seamless and Panography Stitching Order Determination

Martin Pavlovski[1], Vladimir Ilievski[1], Tamara Dimitrova[1], Andrea Kulakov[1]

[1] Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University, Skopje

June 16, 2015

Abstract

The advancement in Artificial Intelligence and Robotics opens new areas of research into the Pattern Recognition and Computer Vision fields. Image stitching is a task in computer vision whose goal is to create natural-looking mosaics with high resolution and completed view, free of noise that occurs due to camera motion, illumination changes, etc., while being insensitive to the ordering of the input images. This paper introduces an algorithm for automatic image stitching. The automation refers to the determination of the stitching order. Additionally, OpenCV's seamless stitching method is compared with our newly proposed method for panography stitching, while the automatic arrangement is applied in both cases.

Keywords: Computer Vision, Machine learning, Seamless stitching, Panography stitching

1 Introduction

As Artificial Intelligence spreads in the everyday life, with robotics, pattern recognition, computer vision there is the need for spreading and evolving the techniques and domains of action. For many years the processing of cosmic images has been very difficult, first because of the way how they will be taken, and later, how they will be processed and interpreted. With the advancement of robotics, we have the tricopters, quadcopters, skywalkers, and many other devices that can observe and record the earth from different aerial perspective, this is not an issue. But another problem arises: these images are not taken in a sequence, there are shifts in different directions, changes in illumination. This problem is in the area of computer vision, where as an input are given the images and the task is to fuse them into a mosaic. There are many different techniques to solve this task and they differ in the success depending the descent of the images. In this paper we compare two different techniques towards solving this problem: seamless (smooth) stitching and panography (mosaic) stitching. Seamless stitching is a technique which is used to combine a set of partial images which have overlapping regions, in order to produce high-dimensional composite image. On the other hand we have implemented the panography, or mosaic stitching, which is a combination of overlapping images with same or different images, angles with goal to create joint panoramic view of the whole scene.

This paper is organized as follows. Section 3 introduces the concepts that we use to perform the images stitching, the process, and describes the techniques used. Section 4 describes the results, and finally in Section 6 are given the Conclusions and ideas for Future Work.

2 Definitions of Used Concepts

In the following section we define the main concepts, which take part in the proposed method, more formally. These concepts include modifications of well-known methods and refer to operations over a set of input images $M = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$. Each input image in M is represented as a matrix $\mathbf{M}_i = [m_{i_{rc}}]_{r_i \times c_i}$ where $r = 1, \ldots, r_i$ and $c = 1, \ldots, c_i$ for all $i = 1, \ldots, n$. The dimensions of these images may not be equal.

2.1 Measuring the similarity between images using the KNN classification method

For determining similarity between the images in M, the K-nearest-neighbour (KNN) classification method is used [9]. This well-known method can be based on different similarity measures. In our case, the function for measuring the similarity between two images $\mathbf{M}_i, \mathbf{M}_j \in M$ is denoted by $d(\mathbf{M}_i, \mathbf{M}_j)$, where $i, j = 1, \ldots, n$ and $i \neq j$.
With the 1-nearest neighbour rule, i.e. for $K = 1$, then an image $\mathbf{M}_i$ is the most similar one to a given image $\mathbf{M}_k$ if

$$d(\mathbf{M}_i, \mathbf{M}_k) = \min_j\{d(\mathbf{M}_j, \mathbf{M}_k)\} \tag{1}$$

where $i \neq k$, $j \neq k$ and $i, j, k = 1, \ldots, n$.
In addition, two different similarity measures are presented.

2.1.1 Histogram comparison

The first similarity measure is based on histogram comparison. In order to explain the idea behind this measure, we first need to give a definition of an image histogram.
Definition: An image histogram is a histogram that signifies the tonal distribution of an image. The image histogram of each image $\mathbf{M}_i$ $(i = 1, \ldots, n)$ is represented as a matrix $\mathbf{H}_i = [h_{i_{b_1 b_2}}]_{h_{bins} \times s_{bins}}$, where $b_1 = 1, \ldots, h_{bins}$ and $b_2 = 1, \ldots, s_{bins}$.

From now on, when the term "histogram" is mentioned, it is taken for granted that it refers to an image histogram. Also for this metric, the number of bins for hue h_{bins} and the number of bins for saturation s_{bins} are fixed i.e. $h_{bins} = 50$, $s_{bins} = 60$.

This measure can be expressed as a function of two parameters $\mathbf{M}_i$ and $\mathbf{M}_j$ $(i, j = 1, \ldots, n)$ which returns a numerical parameter that signifies how well $\mathbf{H}_i$ and $\mathbf{H}_j$ match with each other, where $\mathbf{H}_i$ and $\mathbf{H}_j$ are the histograms of $\mathbf{M}_i$ and $\mathbf{M}_j$ respectively. But to compare these two histograms we can choose from 4 different ways to express how well both histograms match. That way, 4 different variants of this similarity measure are defined.

1. Correlation

$$d(\mathbf{M}_i, \mathbf{M}_j) = -\frac{\sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} (h_{i_{b_1 b_2}} - \overline{\mathbf{H}}_i)(h_{j_{b_1 b_2}} - \overline{\mathbf{H}}_j)}{\sqrt{\sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} (h_{i_{b_1 b_2}} - \overline{\mathbf{H}}_i)^2 \sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} (h_{j_{b_1 b_2}} - \overline{\mathbf{H}}_j)^2}} \tag{2}$$

where $\overline{\mathbf{H}}_i = \frac{1}{h_{bins}+s_{bins}} \sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} h_{i_{b_1 b_2}}$ and $\overline{\mathbf{H}}_j = \frac{1}{h_{bins}+s_{bins}} \sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} h_{j_{b_1 b_2}}$

2. Chi-Square

$$d(\mathbf{M}_i, \mathbf{M}_j) = \sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} \frac{(h_{i_{b_1 b_2}} - h_{j_{b_1 b_2}})^2}{h_{i_{b_1 b_2}}} \tag{3}$$

3. Intersection

$$d(\mathbf{M}_i, \mathbf{M}_j) = -\sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} \min\{h_{i_{b_1 b_2}}, h_{j_{b_1 b_2}}\} \tag{4}$$

4. Bhattacharyya distance

$$d(\mathbf{M}_i, \mathbf{M}_j) = \sqrt{1 - \frac{1}{\sqrt{\overline{\mathbf{H}}_i \, \overline{\mathbf{H}}_j}} \sum_{b_1=1}^{h_{bins}} \sum_{b_2=1}^{s_{bins}} \sqrt{h_{i_{b_1 b_2}} h_{j_{b_1 b_2}}}} \tag{5}$$

The metrics (2) and (4) can be found in [10] while (3) and (5) are presented in [11] and [2] respectively. The open source library of programming functions for computer vision also knows as OpenCV offers an adequate implementation of these measures [7].

2.1.2 The number of "good" matches

The second similarity measure includes the concept of feature detection and feature extraction using two of the well-knows algorithms for these purposes SIFT [5] and SURF [1]. But in order to give a clear explanation of this measure, we need to define the term *keypoint*. In our case, a keypoint which was extracted from an image $\mathbf{M}$ refers to the vector $\mathbf{k} = [x \ y \ s \ \alpha]^T$, where x and y are the coordinates of the keypoint, s is the diameter of the useful keypoint adjacent area and α is the angle of the keypoint i.e. the computed orientation of the keypoint (-1 if not applicable). Let $\mathbf{p}_i = [\mathbf{k}_{i_1} \ \mathbf{k}_{i_2} \ \ldots \ \mathbf{k}_{i_{||\mathbf{p}_i||}}]^T$ and $\mathbf{p}_j = [\mathbf{k}_{j_1} \ \mathbf{k}_{j_2} \ \ldots \ \mathbf{k}_{j_{||\mathbf{p}_j||}}]^T$ are the vectors which contain the extracted keypoints from the images $\mathbf{M}_i$ and $\mathbf{M}_j$ respectively. The function $dist(\mathbf{k}_{i_k}, \mathbf{k}_{j_l})$ gives the distance between the matched keypoints $\mathbf{k}_{i_k}$ and $\mathbf{k}_{j_l}$ i.e.

$$dist(\mathbf{k}_{i_k}, \mathbf{k}_{j_l}) = \begin{cases} v, & \text{if } \mathbf{k}_{i_k} \text{ and } \mathbf{k}_{j_l} \text{ are matched keypoints} \\ -1, & \text{otherwise} \end{cases} \quad , \ v \in \mathbb{N}_0{}^+ \tag{6}$$

, where ($k = 1, \ldots, ||\mathbf{p}_i||$ and $l = 1, \ldots, ||\mathbf{p}_j||$).
Now, let $dist_{min}$ be the minimum distance between all matched points. This means that $dist_{min} \leq dist(\mathbf{k}_{i_k}, \mathbf{k}_{j_l})$ for all $k = 1, \ldots, ||\mathbf{p}_i||$ and $l = 1, \ldots, ||\mathbf{p}_j||$. If G_{ij} represents the set which contains only the "good" matches which refer to the matches whose distance is less than $2dist_{min}$ or a small arbitrary value in the event that the minimum distance is very small, then the mentioned similarity measure can be expressed through the cardinality of this set i.e.

$$d(\mathbf{M}_i, \mathbf{M}_j) = -|G_{ij}| \tag{7}$$

where $i, j = 1, \ldots, n$ and $i \neq j$.

3 Proposed Method

In this section we review the two techniques for stitching images: the seamless and the panography stitching.

3.1 Seamless stitching

The *seamless* or sometimes called *smooth stitching* is a type of stitching technique which is used to combine a set of partial images, having some overlapping regions, in order to produce high-dimensional composite image. We cannot assume that these images are created under the same illumination conditions. The difference in the intensity between the parts of a panoramic image, lens distortion or motion in the scene can lead to the appearance of false edges and seam between the images which is considered to be the main issue in the stitching process. In our case, an already developed method for seamless stitching is used. It is based on a stitching pipeline proposed in [3], while then whole implementation can be found in the "High Level Functionality" module of the OpenCV documentation [8].

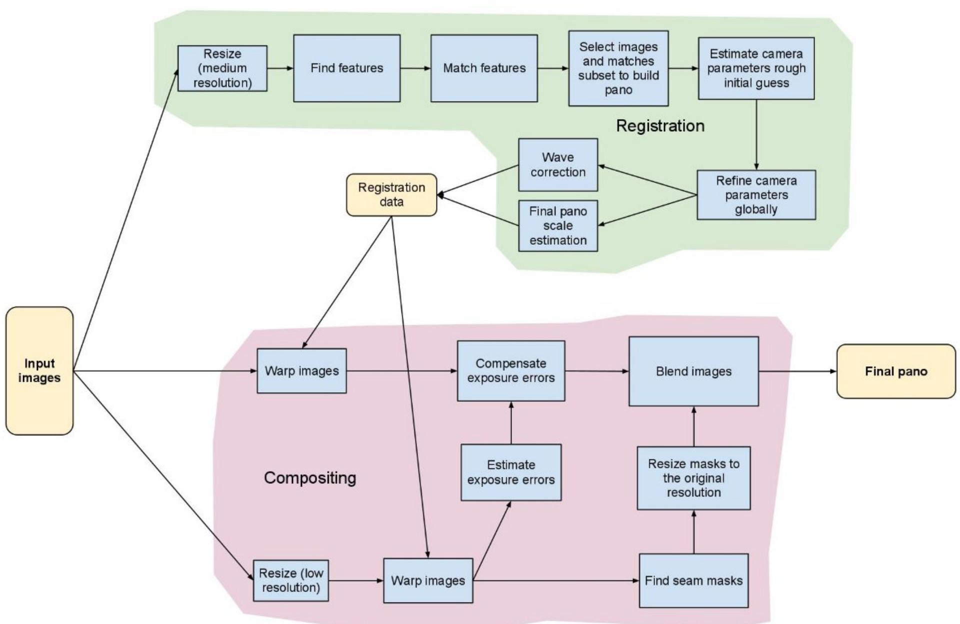

Figure 1: Stitching pipeline [6]

3.2 Panography

Panography as a special form of image stitching combines and assembles overlapping images with same of even different dimensions, thus creating a joint panoramic view of the whole scene known as *panograph*. In essence, the images that are used for producing the panograph are translated, optionally rotated and scaled before being blended [10].

Let $\mathbf{P} = [p_{rc}]_{d_1 \times d_2}$ be a panograph used to assemble the input images $\mathbf{M}_i$ $(i = 1, \ldots, n)$. At each time step t $(t = 0, \ldots, n-1)$, one of these images joins the panograph. The panograph constructed after t time steps is denoted by $\mathbf{P}^{(t)} = [p^{(t)}{}_{rc}]_{d_1 \times d_2}$. At the beginning (when $t = 0$), the panograph is equal to the first image in the set of input images i.e.

$$\mathbf{P}^{(0)} = \mathbf{M}_1 \tag{8}$$

Now, for each time step $t = 1, \ldots, n-1$ the following phases need to be passed:

3.2.1 Rotation of the current image

Let $\mathbf{k}_{P^{(t-1)}{}_{best}} = [x_{P^{(t-1)}{}_{best}}\ y_{P^{(t-1)}{}_{best}}\ s_{P^{(t-1)}{}_{best}}\ \alpha_{P^{(t-1)}{}_{best}}]^T$ and $\mathbf{k}_{M_{t+1}{}_{best}} = [x_{M_{t+1}{}_{best}}$ $y_{M_{t+1}{}_{best}}\ s_{M_{t+1}{}_{best}}\ \alpha_{M_{t+1}{}_{best}}]^T$ are the keypoints extracted from $\mathbf{P}^{(t-1)}$ and $\mathbf{M}_{t+1}$ respectively which correspond to the best match (the match with minimum distance between its keypoints) in the set G_{ij} explained in 2.1.2. Then,

$$\mathbf{M}_{t+1}{}' = \mathbf{R}_{t+1, \alpha_{P^{(t-1)}{}_{best}} - \alpha_{M_{t+1}{}_{best}}} \mathbf{M}_{t+1} \tag{9}$$

where $\mathbf{R}_{t+1, \alpha_{P^{(t-1)}{}_{best}} - \alpha_{M_{t+1}{}_{best}}}$ is the rotation matrix used for rotating the image $\mathbf{M}_{t+1}$ through an angle $\alpha_{P^{(t-1)}{}_{best}} - \alpha_{M_{t+1}{}_{best}}$.

3.2.2 Adding transparent padding around the panograph proportionally

$$\begin{aligned} \mathbf{P}^{(t-1)'} &= \left[\left[\mathbf{0}_{d_1 \times \delta} \mid \mathbf{P}^{(t-1)} \right] \mid \mathbf{0}_{d_1 \times \delta} \right] \\ \mathbf{P}^{(t-1)''} &= \left[\left[\mathbf{0}_{(d_2+2\delta) \times \delta} \mid (\mathbf{P}^{(t-1)'})^T \right] \mid \mathbf{0}_{(d_2+2\delta) \times \delta} \right]^T \end{aligned} \tag{10}$$

where $\delta = \sqrt{(r_{t+1}{}')^2 + (c_{t+1}{}')^2}$ is the offset used to create the transparent space around the panograph.

3.2.3 Stitching the current image to the panograph and removing the transparent space afterwards

Let $x_{t-1_{start}} = \delta + x_{P^{(t-1)}{}_{best}} - x_{M_{t+1}{}_{best}}$ and $y_{t-1_{start}} = \delta + y_{P^{(t-1)}{}_{best}} - y_{M_{t+1}{}_{best}}$ are the coordinates of the starting point in the panograph $\mathbf{P}^{(t-1)''}$ from where $\mathbf{M}_{t+1}{}'$ is about to be stitched. Then $p^{(t-1)}{}_{rc}{}'''$ can be expressed as a function of t i.e.

$$p^{(t-1)}{}_{rc}{}''' = s(t) \tag{11}$$

where

$$s(t) = \begin{cases} m_{t+1_{(r-\delta-y_{t-1_{start}})}}{}', & \text{if } y_{t-1_{start}} \le r \le y_{t-1_{start}} + r_{t+1} \wedge\ x_{t-1_{start}} \le c \le x_{t-1_{start}} + c_{t+1} \\ & \wedge\ m_{t+1_{(r-\delta-y_{t-1_{start}})}}{}' \text{ is not in one of the black borders caused by} \\ & \text{the rotation of } \mathbf{M}_{t+1} \\ p^{(t-1)}{}_{rc}{}'', & \text{otherwise} \end{cases} \tag{12}$$

for all $r = 1, \ldots, d_1 + 2\delta$ and $c = 1, \ldots, d_2 + 2\delta$.

3.2.4 Completing the panograph in the t-th time step

After the stitching phase, lets denote the row number of the first non-transparent pixel, which can be seen if we look at the panograph $\mathbf{P}^{(t-1)'''}$ from the bottom up, by r_{lower} and the row number of the first non-transparent pixel, which can be seen if we look at the same panograph from its top to its lowest point, by r_{upper}. For the column numbers of the first non-transparent pixel on the left and the one on the right we use the notations c_{left} and c_{right} respectively. Then, the panograph in the t-th time step $\mathbf{P}^{(t)}$ is completed with the following

$$p^{(t)}{}_{rc} = p^{(t-1)}{}_{r_{upper}+r-1, c_{left}+c-1}{}''' \tag{13}$$

for all $r = 1, \ldots, r_{lower} - r_{upper} + 1$ and $c = 1, \ldots, c_{right} - c_{left} + 1$.

3.3 Automated general image stitching algorithm

The urge motivation for automatizing the whole stitching process is the work on fully automated panoramic image stitching, presented in [3]. Thus, the general method for image stitching that we propose is based on an automated general image stitching algorithm. This algorithm consolidates the concepts of similarity measurement between images and seamless or mosaic stitching.

For a given set of input images, the algorithm uses an iterative technique for stitching images, thus resulting with a final assemble which contains every input image, whether it was transformed or it retained its original form. Each iteration consists of four simple steps:

- Find an image, from the set of unprocessed images (a set of images that have not been stitched yet), which is most similar to the currently constructed composite image or panograph.
- Stitch the chosen image to the composite image (if 3.1 is used as a stitching technique) or panograph (if 3.2 is used as a stitching technique).
- Remove the chosen image from the set of unprocessed images.
- Check if the the termination criterion is satisfied i.e. check if the set of unprocessed images is empty.

Our implementation of the AGIS algorithm can be found in the repository [4]. According to this implementation, the algorithm can be used in three different modes: STANDARD, FAST and COMPROMISED.

Algorithm 1 A pseudocode for the algorithm which was briefly described in this subsection

procedure AGIS –Algorithm
  Input: $M = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ ▷ set of input images
  $t \leftarrow 0$ ▷ initialize the time step
  $randValue \leftarrow$ **rand()** ▷ generate a random number
  $\mathbf{P}^{(t)} \leftarrow \mathbf{M}_{randValue}$ ▷ randomly initialize the composite image or panograph
  $U \leftarrow \{\mathbf{M}_{randValue}\}$ ▷ initialize the set of unprocessed images
  while $|U| > 0$ **do** ▷ while the set of unprocessed images is not empty, keep stitching
    ▷ each of these images to the composite image or panograph
    $minDist \leftarrow \infty$
    $indexOfMostSimilarImage \leftarrow -1$
    for $i \leftarrow 1$ to n **do**
      if $\mathbf{M}_i \in U$ **then**
        if $d(\mathbf{P}^{(t-1)}, \mathbf{M}_i) < minDist$ **then**
          $minDist \leftarrow d(\mathbf{P}^{(t-1)}, \mathbf{M}_i)$
          $indexOfMostSimilarImage \leftarrow i$
        end if
      end if
    end for
    $\mathbf{P}^{(t)} \leftarrow stitch(\mathbf{P}^{(t)}, \mathbf{M}_{indexOfMostSimilarImage})$
    $U \leftarrow U - \{\mathbf{M}_{indexOfMostSimilarImage}\}$
    $t \leftarrow t + 1$
  end while
  Return $\mathbf{P}^{(t)}$ ▷ return the finalized composite image or panograph
end procedure

4 Results and Discussions

In this section we give a brief summary of the results from few examples of image stitching. Each example includes images that are taken in various situations and from different angles. Some of the images that refer to a same example, may even have different dimensions.

Example 1:

Figure 2: The input images used in Example 1

Figure 3: A composite image which is created by using the COMPROMISED stitching mode for seamless stitching of the input images, while using SIFT as main feature detector and extractor in the stitching process.

Figure 4: A panograph which is created by using the STANDARD stitching mode for panography or mosaic stitching of the input images. In this case, SIFT is used again for feature detection and feature extraction in the stitching process.

Determination of the stitching order in 3	
Index of corresponding image	Number of "good" matches
0	/
1	16
2	11
3	2
4	2

Determination of the stitching order in 4	
Index of corresponding image	Number of "good" matches
0	/
1	19
2	14
3	9
4	3

Example 2:

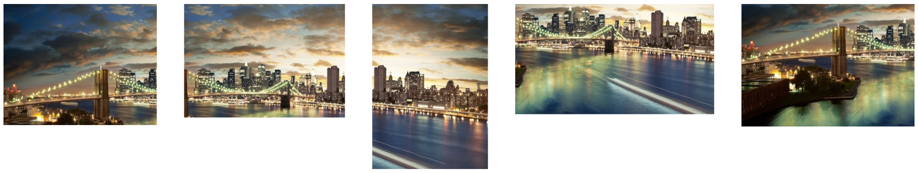

Figure 5: The input images used in Example 2

Figure 6: stitching type: seamless stitching; stitching mode: STANDARD; main feature detector and extractor: SIFT;

Figure 7: stitching type: panography or mosaic stitching; stitching mode: STANDARD; main feature detector and extractor: SIFT;

Determination of the stitching order in 6	
Index of corresponding image	Number of "good" matches
2	/
1	7
3	16
0	11
4	10

Determination of the stitching order in 7	
Index of corresponding image	Number of "good" matches
0	/
4	16
3	10
1	6
2	8

Example 3:

Determination of the stitching order in 9	
Index of corresponding image	Number of "good" matches
4	/
1	13
2	8
0	4
5	4
6	4
7	5
3	0

Figure 8: The input images used in Example 3

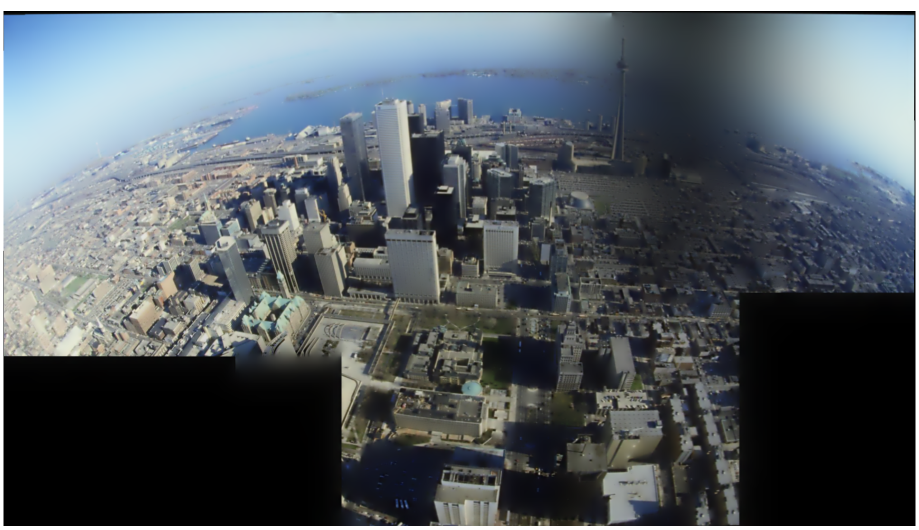

Figure 9: stitching type: seamless stitching; stitching mode: STANDARD; main feature detector and extractor: SIFT;

Figure 10: stitching type: panography or mosaic stitching; stitching mode: STANDARD; main feature detector and extractor: SIFT;

Determination of the stitching order in 10	
Index of corresponding image	Number of "good" matches
0	/
1	37
2	39
5	10
6	14
7	4
4	3
3	14

According to the experimental results, the arrangement of images showed as accurate in most cases. Additionally, the usage of our newly proposed method for image panography resulted in the creation images which were more completed that the ones generated by OpenCV's seamless stitching method.

5 Conclusions and Future Work

The significance and importance of this field is that is applied to many different problems and serves as a very strong tool for visualization. It is applied on many different problems, such as: fusion of cosmic, mountain, pole, different region images and many others.

As a future work, we plan on trying different combinations and techniques in order to improve the automation process and obtain better and faster image stitching.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[2] A Bhattachayya. On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35:99–109, 1943.

[3] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[4] GitHub. Automated general image stitching algorithm. `https://github.com/MartinPavlovski/Automated-general-image-stitching-algorithm`, 2015. [Online; accessed 16-June-2015].

[5] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[6] OpenCV. High Level Functionality stitching pipeline, 2000.

[7] OpenCV. Image Processing histogram comparison, 2000.

[8] OpenCV. OpenCV high level functionality, 2000.

[9] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[10] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[11] Minhaz Fahim Zibran. Chi-squared test of independence. *Department of Computer Science, University of Calgary, Alberta, Canada.[online].[Cited 2010-08-12]*, 2007.