Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 148 (2019) 618–626

Procedia Computer Science

www.elsevier.com/locate/procedia

Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018)

A dynamic mosaicking method for finding an optimal seamline with Canny edge detector

Saadeddine Laaroussi[a,*], Aziz Baataoui[a,b], Akram Halli[c], Satori Khalid[a]

[a] *LIIAN Laboratory, Sidi Mohammed Ben Abdellah University, P.O.Box 1796 Atlas-Fes 30003, Morocco*
[b] *Faculty of Science & Tecchniques, Moulay Ismail University, BP 509 Boutalamine 52000 Errachidia, Morocco*
[c] *OMEGA, LERES Laboratory, Moualy Ismail University, Marjane 2, BP: 298, Meknes 50050, Morocco*

Abstract

Image mosaicking is a process of assembling multiple images to create an image with a larger field of view. Therefore, to avoid failures and to solve errors that could occur during this process, a new method for dynamic mosaicking is presented. This method finds regions that are similar between the images and regions that are not alike, then an optimal seamline is computed by going through the similar regions as much as possible and by avoiding the not common regions. In fact, this is done by combining Canny edge detector and the outliers and inliers from the RANSAC method to identify these areas. In addition, by highlighting these regions in an intensity difference, a map that reveals which areas to avoid and to frequent was created. Thus, an optimal seamline was found. The experimental results show that the proposed approach is capable of generating robust mosaics against ghosting and parallax effects.

Keywords: Dynamic mosaicking ; Ghosting ; Seamline ; Edge Detector Canny;

1. Introduction

Image mosaicking is a process of assembling multiple images to combine them and to obtain a new image with a larger field of view. With the current technology, it is not possible to capture images with a large field of view and a good resolution, that is why mosaicking methods are used to get a better mosaic quality. In general, there are two types of mosaics, the first one uses static methods where the scene doesn't include any motion [1, 2, 3, 4, 5] while the latter uses dynamic methods where the scene contains different dynamic events [6, 7, 8, 9]. Furthermore, image mosaics (panoramas) are widely used in many fields of computer vision where it is necessary to have a larger field

* Corresponding author. Tel.: +212-671902303.
E-mail address: saadeddine.laaroussi@usmba.ac.ma

10.1016/j.procs.2019.01.050

view, some of them are: augmented reality [10], video compression [6], video indexation [11], image stabilization [12], motion detection and tracking [13], environment perception for robots [14] etc...

Usually mosaics are created by following three primary steps: image registration, alignment and blending. First, image registration computes the geometric transformation, then the alignment step projects the images into the same coordinate system and lastly the blending step improves the quality and removes the discontinuities of the resulting mosaics. Algorithms used for image registration or blending may vary. However in some cases, it is beneficial to add a pretreatment[15] step to simplify the creation of the mosaic.

Most of the time, dynamic aspects exist in images. As a matter of fact, these aspects are often different from an image to another, since the objects change their position or motion. Consequently, these dynamic elements create errors that could lead to the failure of the mosaic such as: ghosting and parallax effects where ghosting is created from the remnant of the dynamics objects and parallax effects are distortions that appear when the scene is not planar.

There are many approaches that have treated this subject in various ways. Some have used a deghosting method [16, 17, 18], while others tried to manage the motion of the objects by extracting the background from the foreground [9] or by using optical flow [19]. And some used an approach for searching an optimal seamline [20, 21, 22, 23, 25] that will divide the images by avoiding the dynamic aspects. In this paper, we will use a seamline based approach. In fact, A seamline is a line that will connect the images in their overlap regions by creating the least amount of discontinuities. In this field, we find Mills and Dudek [21]. First, they applied an exposure correction by computing the bias and the gain between the images. Then they did a weighted sum between the intensity difference and the gradient difference of the overlap areas. The weights were computed from the bias and the gain. This created an energy map for computing the seamline. In a same way, Zeng et al. [22], used the same idea and added a threshold to add a weight on the areas that have a high intensity in the map. This criteria enabled them to find an optimal seamline. However with a different approach, Tang et al. [23] used a combination of a prominence map, which was computed in the CIELab color space with a Euclidean distance, and an optimized gradient map based on an energy function [24] to find the seamline. Differently, Pan et al. [25] used a mean shift segmentation [26] to find regions they named PR (Preferred regions), these regions are areas that the seamline must not pass through. In most cases, the path of the seamline was computed with a Djikstra algorithm [27], a snake algorithm [28] or in another way by using dynamic programming.

This article proposes a new approach for finding the optimal seamline. Two contributions will be presented. The first contribution resides in using regions that were created with a Canny edge detector and then combining them with an intensity difference to create a map where the seamline is computed. In fact, by using a SIFT keypoint detector and the RANSAC method, two types of keypoints will be detected: inliers and outliers. These keypoints and the edges from the canny edge detector will create two types of regions: Outlier regions, regions that contain outliers and inlier regions that contain inliers. In addition, by using thresholds, it is possible to identify clearly the dynamic elements from the static elements in the images since inlier regions are regions that are common to all the images, and the outliers regions are going to be regions that are specific to an image. And the second contribution is a seamline searching approach. Since the outlier and inlier regions are known, the seamline needs to go through a maximum of inlier regions by passing through the least amount of outlier regions. The proposed approach is more efficient in images with a large number of keypoints. The different steps of the method are shown in Fig.1.

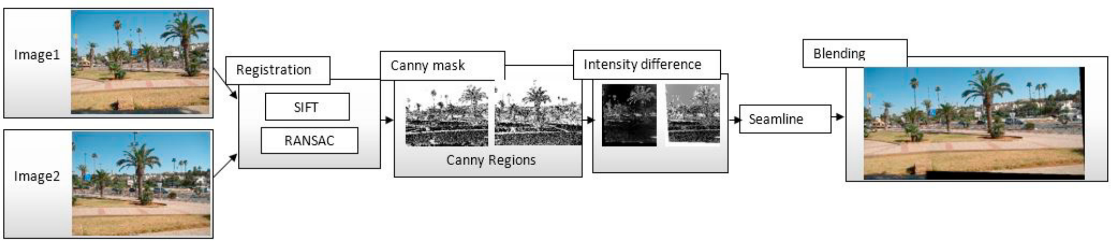


Fig. 1: Steps of the algorithm.

The remainder of this paper is organized as follows: the second part develops the different steps of the proposed method. The experimental results are described in the third section, and the last section presents a conclusion.

2. Proposed approach

In this section, the steps of Fig. 1 are treated in detail. Let Im_1 (Fig. 2.a) and Im_2 (Fig. 2.b) be the images that are going to be stitched.

2.1. Image registration

Image registration [29] is the first step to create a mosaic and to put the images into the same coordinate system. Keypoint detector SIFT[30] (Scale Invariant Feature Transform) is used due to its robustness and efficiency compared to other keypoint detectors[31] then RANSAC is used to compute the geometric transformation and the inliers (matching points) and outliers (false matching points).

2.2. Canny mask

In this step the Canny mask that will show the inlier and outlier regions is created. To find the regions in a precise way, we used an edge detector, Canny edge detector [32]. First, the detector is used to obtain the edges of the input images. Then, by using the inliers and outliers computed by SIFT and a flood fill algorithm [33], we will fill the regions containing an inlier or an outlier with a chosen value, for example 1 for the outlier regions and 0.1 for the inlier regions. The masks are created as follows:

- Removing the noise from Im_1 and Im_2 by using a Gaussian filter.
- Getting the edges M_1 and M_2 by using Canny on Im_1 and Im_2.
- Using a dilatation on M_1 and M_2 to fill any empty pixel between the edges.
- For each inlier and outlier, a flood fill algorithm is applied on the region that contains the point if it is still empty, with the outliers having a priority over the inliers. The chosen value is then applied for the outlier and inlier regions. (Fig. 2.c, Fig. 2.d)
- Applying the geometric transformation.

The canny masks are going to be used so that the outlier regions will lighten the dynamic elements and inlier regions will darken the common areas. In addition, This step was done before using the geometric transformation to avoid precision errors that could occur in the edge detection.

After the geometric transformation of M_1 and M_2, the canny mask M is created by fusing the information of M_1's and M_2's overlap regions. In fact, if a point belongs to an outlier region, then in the fusion it will belong to an outlier region, but if a point belongs to an inlier region then this point needs to belong to an inlier region in all the images to belong to an inlier region in the fusion. With 1 being the value for outliers and 0.1 the value for inliers, the Canny mask M can be computed with equation (1).

$$M_{ij} = \begin{cases} 1 & M_{1ij} = 1 \quad \| \quad M_{2ij} = 1 \\ 0.1 & M_{1ij} = 0.1 \quad \&\& \quad M_{2ij} = 0.1 \\ 0 & other \end{cases} \tag{1}$$

Fig. 2: (a) Left image to be stitched (b) Right image to be stitched (c) Left image Canny mask (d) Right image Canny mask.

2.3. Improved Intensity Difference

The intensity difference [34] performs a difference between the images (Fig. 4.a). By using this difference, we will manage to assign high values to the dynamic objects and low values to the common areas. The intensity difference is computed with this formula:

$$\delta_{ij} = \frac{abs(Im'_{1ij} - Im'_{2ij})}{max(Im'_{1ij}, Im'_{2ij})} \tag{2}$$

Where Im'_{1ij} and Im'_{2ij} are the Gaussians of the overlap areas of the images and $\delta_{ij} \in [0, 1]$. From this part on, all the transformations are done on the overlap areas of the images.

To improve the intensity difference (Fig. 4.a) and thus have a better detection of the dynamic elements, the previously computed canny mask M is going to be used. Then pixels that are in outlier regions are going to be highlighted and those that are in inlier regions are going to be darkened. This was done by using equation (3). Even though some areas that don't contain motion were highlighted, they might create parallax effects so it is safer to avoid them.

$$\delta'_{ij} = \begin{cases} 0.5 * M_{ij} + 0.5 * \delta_{ij} & M_{ij} = 1 \\ 0.5 * \delta_{ij} & M_{ij} = 0.1 \\ \delta_{ij} & M_{ij} = 0 \end{cases} \tag{3}$$

Furthermore two thresholds have been used, $Thrd1 \in [0.6, 0.7]$ to highlight high values and $Thrd2 \in [0.1, 0.2]$ for small values. This way pixels that are higher than $Thrd1$ are considered as pixels that contain motion and pixels that are lower than $Thrd2$ are considered as safe pixels (Fig. 4.b). Equation (4) shows the details of this step.

$$\delta'_{ij} = \begin{cases} 1 & \delta'_{ij} >= Thrd1 \\ 0 & \delta'_{ij} <= Thrd2 \\ \delta'_{ij} & else \end{cases} \tag{4}$$

2.4. Optimal seamline

An optimal seamline [35] is a seamline that will pass through a path without creating any discontinuities and without cutting objects in half, therefore to obtain an optimal seamline, passing through inlier regions and avoiding outlier regions will be necessary since inlier regions are regions that both images have while outlier regions only exist in one image. The following steps show how to compute the optimal seamline:

- For each point in the first row of δ'_{ij}, find the points that have the smallest value. These points are going to be the start of the different seamlines.
- For each one of these points, the next point of the seamline is chosen from the five points near the seed point (Fig. 3.), the next point is going to be the point with the smallest intensity. The loop ends when the seamline has passed through the entire image in height. In this step, many seamlines are going to be created.
- Since the seamlines won't have the same amount of points, the ratio of points that have a value of 1 and the ratio of points that have a value of 0 is computed for each seamline.

- Choosing the seamline with the smallest ratio of points with a value of 1 and the highest ratio of points with a value of 0. (Fig. 4.c)

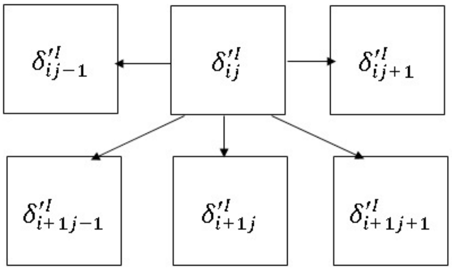


Fig. 3: The path that a seed point can take.

(a) (b) (c)

Fig. 4: (a)Intensity difference δ_{ij} (b) Intensity difference δ_{ij} after fusion with M (c) Optimal seamline in green.

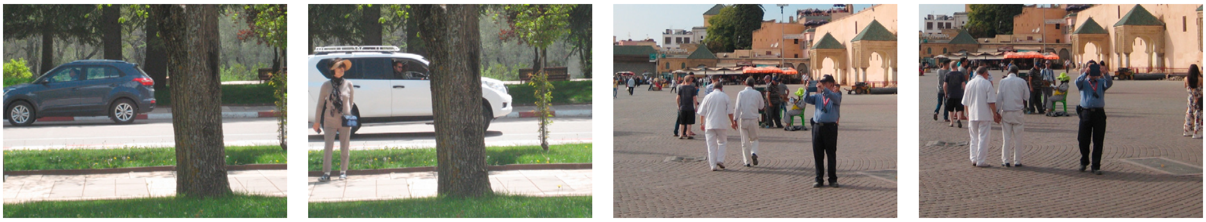
2.5. Blending

The blending is the final step of the mosaic creation. It will improve the quality of the mosaic by obtaining a continuous effect near the seamline. The pyramid blending[36] (also known as multi band blending) was used. This method uses Gaussian pyramids to create Laplacian images (LoG) and combines the Laplacian pyramids by using a mask. This mask is created by using the previously computed seamline, and shows the points that right from the seamline by giving them a value of 1 and left from the seamline by giving them a value of 0.

3. Experimental results

To show the effectiveness of this approach, we used three sets of images of real scenes captured with a digital camera. Each set contains a pair of two images. The first set (Fig. 2.a, Fig. 2.b) contains two images with dynamic aspects that are far, the second set (Fig. 5.a) is composed of two images with a small amount of dynamic elements, and the third set (Fig 5.b) has two images with many motions. The presented method is then compared by three criteria with the Mills [21] method and CSDP(Combined SIFT with Dynamic Programming) [22] method. The criteria are the path of the seamline, the quality of the mosaic and the execution time.

Our approach, the Mills [21] method and the CSDP [22] method were coded in Java with the use of the OpenCV library with a portable PC that has a microprocessor Intel Core i7-3632QM CPU at 2.20 GHz. The images were captured by using the digital camera Canon Digital IXUS 60.

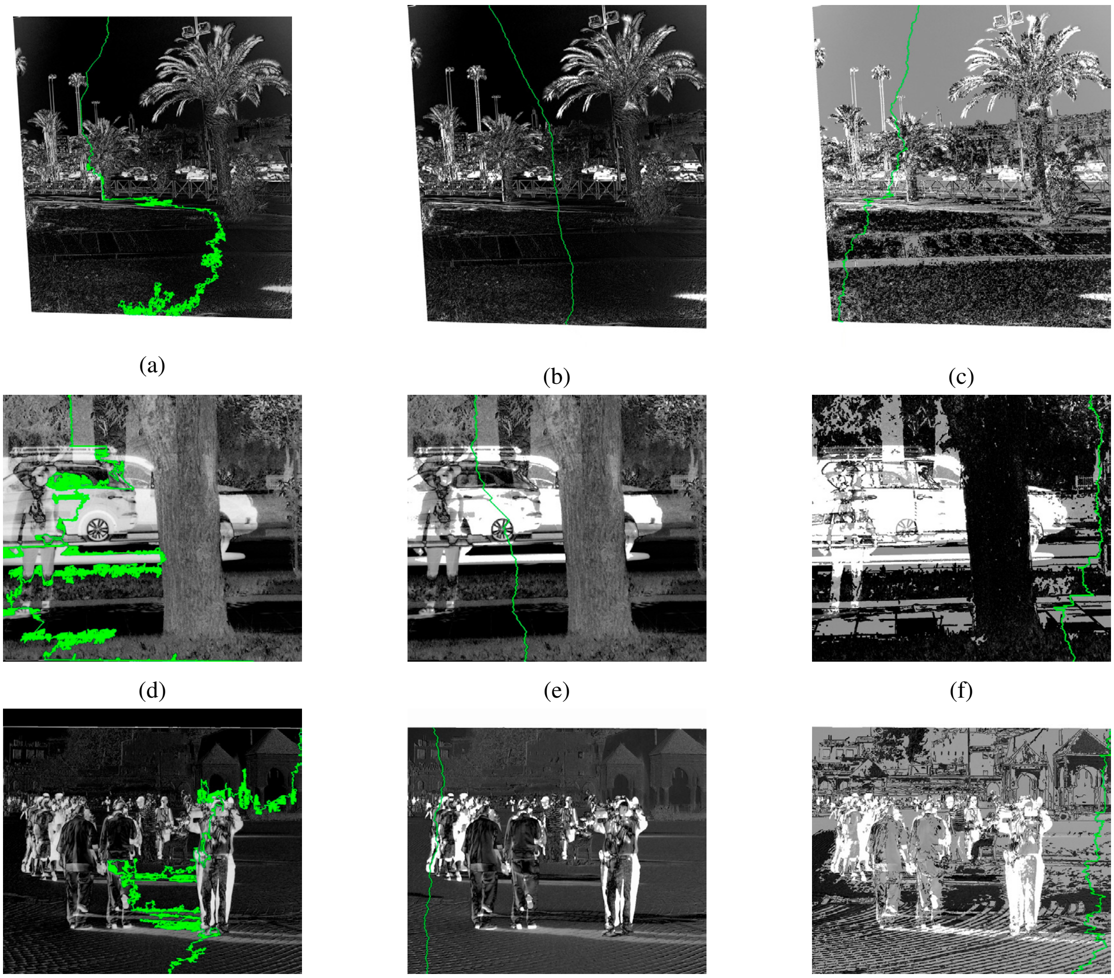


(a) (b) (c) (d)

Fig. 5: Second set, (a) First image (b) Second image. Third set, (c) First image (d) Second image

3.1. Seamline comparison

Figure 6 shows the different seamlines of the three approaches(Mills [21], CSDP [22] and the proposed approach) that are going to be compared.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

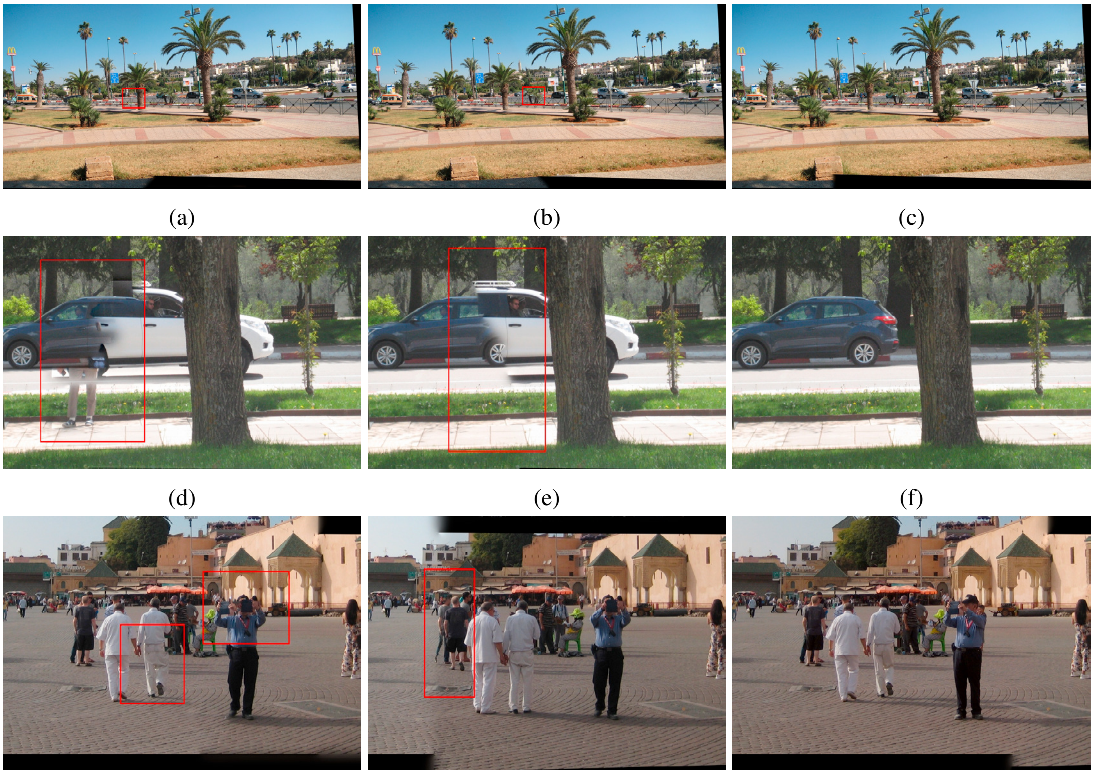
Fig. 6: Intensity difference and seamline: (a, d, g) Mills (b, e, h) CSDP (c, f, i) Proposed approach.

In this subsection we will compare the seamlines of the three different approaches. The Mills [21] method and CSDP [22] method both use an exposure correction. They compute the bias and the gain of the images to, first adjust the intensities between the images and second to weight the intensity difference and gradient difference. Furthermore, the CSDP approach uses a threshold to highlight the dynamic aspects and a weight to raise the value of these areas. To find the path of the seamline, the Mills and Dudek approach uses a Djikstra algorithm whereas the CSDP method uses a dynamic way by choosing a point from the three point underneath the seed point with the smallest value. To compare these approach with ours, the figure (Fig. 6.) shows the three δ and the seamlines, for each set, for all the three approaches.

By analyzing the intensity differences in Figure 6, since the Mills and CSDP approach don't detect the dynamic aspects in a precise way, the difference on the detection is noticeable with our approach showing the positions exactly. It is also visible that the other approaches' seamlines pass through some dynamic objects. In figure (6.a, 6.d, 6.g) the Mills seamline passes through a lot of objects, in (6.b, 6.e, 6.h) the CSDP passed between dynamic objects and sometimes cut through them, while in (6.c, 6.f, 6.i) our seamline passed through inlier region and thus avoided outliers regions. Since Mills and CSDP cut through objects, ghosting will appear. By using the outlier regions the detection of the dynamic aspects in our approach was more precise. The inlier regions managed to guide the seamline through areas where it's safe to pass and thus the seamline was more optimal. In fact, it is because of the fact even if a region doesn't have a value 1, it was lightened because it contained an outlier point which is visible in (6.i) for example where the starting points for the seamlines were reduced considerably.

3.2. *Mosaic result comparison*

The results of the mosaics are shown of the three methods(Mills [21], CSDP [22] and the proposed approach) are shown in Figure 7.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 7: Final mosaic, (a, d, g) Mills (b, e, h) CSDP (c, f, i) Proposed approach.

Figure (Fig. 7) shows the mosaic result of the three approaches (Mills [21], CSDP [22] and the proposed approach). For the three sets, as predicted, the Mills and CSDP approach created ghosting effects since the seamline cut through objects while our approach avoided them. The effects were highlighted with a red square. The first set (Fig 2.a) is a simple set where the dynamic elements are far but still the Mills (Fig. 7.a) and CSDP(Fig. 7.b) approach created some slight ghosting effects but the proposed approach didn't (Fig. 7.c). The second set of images (Fig 5.a) represents images with a small amount of dynamic elements, Mills (Fig 7.d) and CSDP (7.e) didn't manage to avoid all the objects, while our approach (7.f) managed to avoid them thanks to the inliers regions. The images of the third set (Fig. 5.b) are images with a high amount of dynamic elements, it's noticeable that for Mills (Fig. 7.g) the seamline passed through a lot of dynamic objects, that's because the Mills approach only uses an intensity difference which was not enough to avoid the dynamic elements of the image, while the CSDP method (Fig. 7.h) passed between dynamic objects and still created ghosting. On the other hand, our seamline (7.i) passed through the building that was an inlier region and thus was an invisible seamline and didn't create any ghosting.

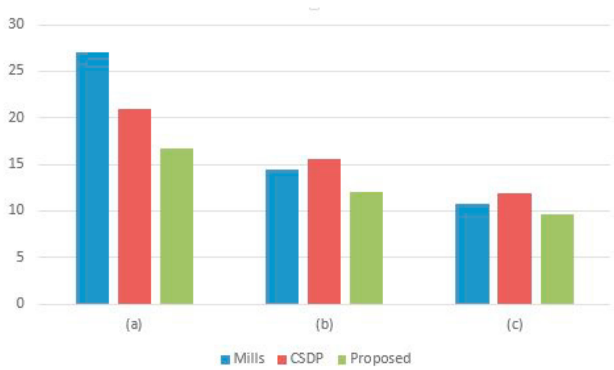


Fig. 8: Execution time (in seconds) of the three approaches. (a) First set (b) Second set (c) Third set.

The execution time of the three sets for the three approaches is shown in (Fig. 8). The proposed approach is slightly faster than the others and this is due to the fact that inliers and outliers were computed in the image registration step and were later on used to detect the dynamic objects and also to a better seamline calculation.

4. Conclusion

In this paper, we proposed an idea that is based on creating a canny mask by using the canny edge detector and combining him with the inliers and outliers that were computed with the RANSAC method. In fact by using the outliers and inliers regions, it was possible to obtain a better detection of the dynamic elements and of the areas that could create distortions and parallax effects. As a result, by having the seamline go through regions that were darkened and by avoiding highlighted areas, it allowed us to compute a better seamline. And finally, the effectiveness of the approach was shown in the different comparisons that were done between the Mills approach, the CSDP method and our approach on the detection of dynamic elements in the intensity differences, the seamlines, execution time and the final mosaics.

Acknowledgements

This work was supported by the national center of scientific and technology research (CNRST, Centre National pour la Recherche Scientifique et Technique).

References

[1] S. Chen and QuickTime VR.(1995) "Image-based approach to virtual environment navigation." *In SIGGRAPH'95* **29** : 29–38.

[2] R. Szeliski and H.Y. Shum.(1997) "Creating full view panoramic image mosaics and environment maps." *In SIGGRAPH'97 24th annual conference on Computer graphics and interactive techniques* : 251-258.

[3] M. Brown and D. Lowe.(2007) "Automatic panoramic image stitching using invariant features." *International Journal of Computer Vision* **74** (1): 59-73.

[4] A. Baataoui, A. Laraqui, A. Saaidi, K. Satori, A. Jarrar, and Med. Masrar.(2015) "Image Mosaicing Using a Self-Calibration Camera." *3D Research (Springer)* **6** (2): 19-34.

[5] A. Laraqui, A. Baataoui, A. Saaidi, A. Jarrar, Med. Masrar and K. Satori.(2017) "Image Mosaicing Using Voronoi Diagram. Multimedia Tools and Applications (Springer)" **76**: 8803-8829.

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

[7] H.S. Sawhney and S. Ayer.(1996) "Compact representations of videos through dominant and multiple motion estimation." *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **18**: 814-830.

[8] Z. Qi and J.R. Cooperstock.(2008) "Depth-based image mosaicing for both static and dynamic scenes." *In ICPR'08 19th International Conference on Pattern Recognition*.

[9] A. Bartoli, N. Dalal and R. Horaud.(2004) "Motion Panoramas." *In Journal of Computer Animation and Virtual Worlds* **15**: 501-517.

[10] P. Azzari, L. Di Stefano, F. Tombari and S. Mattoccia.(2008) "Markerless Augmented Reality Using Image Mosaics." *Image and Signal Processing* **5099**: 413-420.

[11] C. Yun-Hee, S. Yeong Kyeong and C. Tae-Sun.(2002) "Image mosaicing with automatic scene segmentation for video indexing." *International Conference on Consumer Electronics*: 74-75.

[12] M. Ramachandran, and R. Chellappa.(2006) "Stabilization and Mosaicing of Airborne Videos." *IEEE International Conference on Image Processing*: 345-348.

[13] K. S. Bhat, M. Saptharishi and P. K. Khosla.(2000) "Motion detection and segmentation using image mosaics." *IEEE International Conference on Multimedia and Expo*: 1577-1580.

[14] X. Gu, P. Song, Y. Rao, Y. G. Soo, C. F. Yeong, J. T. C. Tan et al.(2016) "Dynamic Image Stitching for Moving Object." *IEEE International Conference on Robotics and Biomimetics*, Qingdao, China.

[15] S. Laaroussi, A. Baataoui, A. Halli and S. Khalid (2018) "A dynamic mosaicking method based on histogram equalization for an improved seamline." *Proc. of the first inter. conf. on intel. comp. in data sciances ICDS2017* **127**: 344-352.

[16] M. Uyttendaele, A. Eden and R. Skeliski.(2001) "Eliminating ghosting and exposure artifacts in image mosaics." *Computer Vision and Pattern Recognition*.

[17] W. Zhang, S. Hu and K. Liu.(2017) "Patch-Based Correlation for Deghosting in Exposure Fusion." *Information Sciences* **415**-**416**: 19-27.

[18] W. Zhang, S. Hu, K. Liu and J. Yao.(2016) "Motion-free exposure fusion based on inter-consistency and intra-consistency." *Information Sciences* **376**: 190-201.

[19] N. Rebire, M. Auclair-Fortier and F. deschłnes.(2008) "Image mosaicing using optical flow registration." *Pattern Recognition*.

[20] M. Duplaquet.(1998) "Building large image mosaics with invisible seam lines." *Proc. SPIE 3387, Visual Information Processing* 2: 369.

[21] A. Mills and G. Dudek.(2009) "Image stitching with dynamic elements." *Image Vis. Comput.* **27**(10): 1593-1602.

[22] L. Zeng, S. Zhang, J. Zhang and Y. Zhang.(2014) "Dynamic image mosaic via SIFT and dynamic programming." *Machine Vision and Applications archive* **25** (5): 1271-1282.

[23] Y. Tang, J. Shin and H. Liao.(2012) "De-ghosting Method for Image Stitching." *International Journal of Digital Content Technology and its Applications (JDCTA)*.

[24] Y. Tang and H. Jiang.(2009) "Highly efficient image stitching based on energy map." *Image and Signal Processing, 2nd International Congress*.

[25] J. Pan, Q. Zhou and M. Wang.(2014) "Seamline Determination Based on Segmentation for Urban Image Mosaicking." *IEEE Geoscience and Remote Sensing Letters* **11** (8): 1335-1339.

[26] D. Comaniciu, P. Meer," Mean Shift : A Robust Approach Toward Feature Space Analysis."

[27] L. Yu, E. J. Holden, M. C. Dentith and H. Zhang.(2011) "Towards the automatic selection of optimal seam line locations when merging optical remote-sensing images." *International Journal of Remote Sensing* **33** (4): 1000-1014.

[28] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley.(2001) "Color transfer between images." *IEEE Computer Graphics and Applications* **21** (5): 34-41.

[29] Mahesh and M .V. Subramanyam.(2012) "Automatic Feature Based Image Registration Using SIFT Algorithm." *Computing Communication and Networking Technologies (ICCCNT)*.

[30] D. Lowe.(2004) "Distinctive image features from scale-invariant keypoints." *International Journal Computer Vision* **60**(2): 91-110.

[31] L. Juan and O. Gwun.(2009) "A comparison of SIFT, PCA-SIFT and SURF." *International Journal of Image Processing (IJIP)* **3** (4): 143-152.

[32] J. Canny (1986) "A Computational Approach To Edge Detection" *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**(6): 679698.

[33] J. Dunlap," Queue-Linear Flood Fill: A Fast Flood Fill Algorithm," http://www.codeproject.com/KB/GDI-plus/ queuelinearfloodfill.aspx

[34] J. Davis.(1998) "Mosaics of scenes with moving objects." *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*: 354360.

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

[36] M. Brown and D. G. LOWE.(2007) "Automatic panoramic image stitching using invariant features" *International journal of computer vision* **74** (1): 59-73.