Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 127 (2018) 344–352

Procedia Computer Science

www.elsevier.com/locate/procedia

The First International Conference On Intelligent Computing in Data Sciences

A dynamic mosaicking method based on histogram equalization for an improved seamline

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

[a] LIIAN Laboratory, Sidi Mohammed Ben Abdellah University, P.O.Box 1796 Atlas-Fes 30003, Morocco

[b] OMEGA, LERES Laboratory, Moualy Ismail University, Marjane 2, BP: 298, Meknes 50050, Morocco

Abstract

Image mosaicking is a process of combining multiple images to obtain an image with a larger field of view. Errors, like ghosting and parallax effects, could appear in the mosaic if the images contain dynamic objects. This paper introduces a new idea based on histogram equalization that was used to find an optimal seamline to avoid dynamic aspects in the images. This was achieved by firstly applying a color transfer algorithm to adjust the lightness between the images in the lab color space. After that a registration was done with SIFT and RANSAC. And finally, before searching the seamline, the overall intensity of the images were fixed by changing the intensity of the underexposed and overexposed areas with a histogram equalization method. As a result, the path of the seamline was improved. The present algorithm gives a fast result and a better detection of the dynamic objects, thus a better a mosaic result.

Keywords: Dynamic mosaicking; Ghosting; Seamline; Histogram equalization;

1. Introduction

A mosaic is created by following three steps: image registration, alignment and blending. First, image registration uses a keypoint detector and descriptors to find the projection matrix. Then, the alignment step projects all the images in the same coordinate system. And finally, the blending improves the quality of the mosaic result by minimizing the discontinuities in the overlap areas. Overall, mosaics can be divided into two types. Static mosaics, scenes without any motion [1, 2, 3, 4, 5], and dynamic mosaics, scenes with dynamic objects in the overlap areas of the images [6, 7, 8, 9]. Thereby, mosaics are used in different fields: augmented reality [10], video compression [6], video indexation [11], image stabilization [12], movement detection and segmentation [13], environment perception for robots [14] etc...

To obtain a static mosaic, some constraints must be fulfilled when capturing the images to avoid getting a failed mosaic: the scene must be static, the camera must rotate around its optical center, and the exposure between each

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

10.1016/j.procs.2018.01.131

image has to be small. When these constraints are not satisfied, ghosting and parallax errors appear. In fact, ghosting appears from the remnant of the dynamic objects that were cut, and parallax effects appear when the camera didn't rotate around its optical center. To avoid these errors, many methods were proposed for dynamic mosaicking. A couple of them on deghosting [15, 16, 17], others on managing the objects motion (extraction of background-foreground[9], optical flow[18]), and some on searching the optimal seamline [19, 20, 21, 22, 23, 24].

In this paper, an approach for finding an optimal seamline is introduced. The majority of the algorithms in this field first create a map that detects the dynamic objects, then the seamline is computed to avoid these objects. Mills and Dudek[20] used a weighted map that was computed from the intensity difference and the gradient difference of the images. The weights of the map were calculated using a linear exposure correction. With the same idea, Zeng et al.[21], created the CSDP (Combined Sift with Dynamic Programming) method by using a threshold on Mills et Dudek's approach[20], to identify all the dynamic elements. Next, they applied a weight on these elements to avoid them completely. In [22], Tang et al. combined a prominence map, computed in the CIELab color space with an Euclidian distance, and an improved energy map[23] to create their map. Most of these approaches use a dynamic algorithm to compute the optimal seamline that avoids all the dynamic objects, it can be a Dijkstra[25] or Snake[26] algorithm, or another way by using dynamic programming.

To fix the exposure between the images, the methods[20, 21] computed the bias and the gain between the images from the inlier points with a least square method. In this paper, we will present a method by using a color transfer[27] from an image to another, and a histogram equalization method to obtain a better detection of the dynamic objects. This will result in an optimal seamline. The main idea is to use the color transfer algorithm to adjust the intensities between the images and the histogram equalization to obtain a uniform intensity throughout each image, thus we will fix all the areas that are underexposed and overexposed when creating our map. In fact, these areas produce errors since they are often considered as part of the background, especially if the texture of the dynamic objects is similar to the background's texture. To solve this issue we will follow these steps:

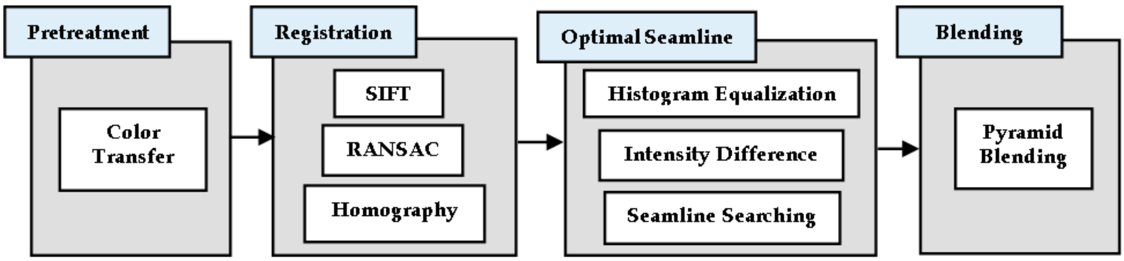

Fig. 1: Steps of the algorithm.

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

2. Dynamic mosaicking method

In this section, the steps of Fig. 1 are treated in detail.

2.1. Pretreatment

It's rare for images that are captured from cameras to have the same color intensities, even if they are from the same photography instrument. Often an exposure difference between the images exists due to the sensibility of the camera's lenses to light. That's why we used a Color Transfer[27] to alter the color of an image by using the colors of another image. This was done by computing the mean and the standard deviation of each channel in the CIELab color space, then using them to change the first image with equation (1).

Let Im_1 (Fig. 2.b) and Im_2 (Fig. 2.a) be the images for the mosaic, the pretreatment is done with the following steps: