Automatic video mosaicking algorithm via dynamic key-frame

JI Yufeng¹, LI Weixing¹,* FENG Kai¹, XING Boyang¹, and PAN Feng¹,2
1. School of Automation, Beijing Institute of Technology, Beijing 100081, China; 2. Kunming-BIT Industry Technology Research Institute Co., Ltd, Kunming 650106, China

Abstract: Automatic video mosaicking is a challenging task in computer vision. Current researches consider either panoramic or mapping tasks on short videos. In this paper, an automatic mosaicking algorithm is proposed for both mapping and panoramic tasks based on the adapted key-frame on videos of any length. The speeded up robust features (SURF) and the grid motion statistic (GMS) algorithm are used for feature extraction and matching between consecutive frames, which are used to compute the transformation. In order to reduce the influence of the accumulated error during image stitching, an evaluation metric is put forward for the transformation matrix. Besides, a self-growth method is employed to stitch the global image for long videos. The algorithm is evaluated by using aerial-view and panoramic videos respectively on the graphic processing unit (GPU) device, which can satisfy the real-time requirement. The experimental results demonstrate that the proposed algorithm is able to achieve a better performance than the state-of-art.

Keywords: video mosaicking, image stitching, speeded up robust feature (SURF), panorama, mapping.

DOI: 10.23919/JSEE.2020.000005

1. Introduction

Video mosaicking is a valuable work in the computer vision and widely used in many fields like mapping, panoramic view of streets, visual simultaneous localization and mapping (SLAM), the navigation of unmanned vehicles and drones. There are several commercial software that can deal with the image stitching, such as Pix4DMapper, Photoscan and Microsoft Photosynth. However, they can only process a batch of images by an offline mode, which costs a lot of time. At present, almost all kinds of smartphones have the function of panoramic image stitching but only single-direction scanning. Therefore, it is necessary to study the real-time video mosaicking algorithm.

Image stitching is the process that two dependent images are stitched together to form a larger image. Zitova et al. [1] proposed the standard steps of image stitching: (i) extract features of two relevant images, (ii) compute the correspondence by using the features, (iii) compute the transformed image, and (iv) copy the transformed image to the reference image. Video mosaicking can be seen as the temporal expansion of the image stitching, which depends mostly on the quality of image stitching. However, there are also many factors affecting the mosaicking results, like the quality of the image and the transformation.

The video mosaicking can be divided into two scenes: the mapping and panorama. The fundamental difference lies in the consistency of the image planes, as shown in Fig. 1.

Fig. 1 Mapping and panoramic image

The mapping video is a set of aerial-view images which are captured by a fixed camera view; while the panoramic images are in a rotating view. Influenced by the camera and the shaking of unmanned aerial vehicles (UAVs), the images in the mapping videos are not ensured to be in the same plane. Our algorithm is able to estimate the change of the image plane, thus reducing the influence of the accumulated error.

2. Related work

In past few decades, many researchers were dedicated to the image stitching and video mosaicking. Zitova et al. [1] made a specific summary of this field and proposed the standard steps. Most works focus on the feature extraction and matching [2–5]. The most widely used feature and descriptor algorithms include speeded up robust features (SURF) [6], scale-invariant feature transform...
JI Yufeng et al.: Automatic video mosaicking algorithm via dynamic key-frame

(SIFT) [7], Harris [8], oriented brief (ORB) [9] and features from the accelerated segment test (FAST) [10]. The random sample and consensus (RANSAC) algorithm is used to compute the transformation between two consecutive frames [2,3,11–13]. Zhu et al. [11] proposed a robust line segment-based feature based on the Harris corner, which achieved high performance images with high textures. Yang et al. [14] proposed a matching method embracing semantic information of images which gains good performance on image matching, however, their algorithm can only handle two images, while the distortion is still not considered. Li et al. [2] used the FAST key point detector and the binary robust independent elementary features (BRIEF) descriptor for the feature extraction. Furthermore, they proposed a spatial and temporal filter to enhance the computing speed of RANSAC. The FAST and BRIEF are easy to compute, while the precision of image stitching is not high. Lin et al. [13] computed regional homography, which is used to estimate the global transform to mitigate the perspective distortion. The method is time-consuming as it aims to eliminate all local distortion regions. Now more and more algorithms deal with the accumulated error during the stitching process. Amiri et al. [3] proposed a mosaicking algorithm for panoramic videos, which used the SURF and binary robust invariant scalable keypoints (BRISK) descriptor to compute the matching between frames, and calculated the L2-norm between consecutive transform matrices to estimate the accumulated error. Besides, there are also many works fusing the sensor data into the mosaicking process to generate more accurate images. Szeliski et al. [15] proposed a stitching algorithm fused with the 3D pose of the camera, which makes a break through from the spherical stitching to the combinational stitching with image and camera pose. Bu et al. [12] explored an online video mosaicking framework fused with the camera pose, global positioning system (GPS) and mapping images based on monocular SLAM, which can generate both 2D map and 3D cloud points of the survey area. These methods use the data of other sensors to calibrate the video mosaicking process, which are more dependent compared with the pure-image-based methods.

3. The proposed algorithm

In this paper, we address the accuracy of video mosaicking for mapping and panorama tasks. An automatic video mosaicking algorithm is proposed based on the SURF, grid motion statistic (GMS) matching [16] and dynamic key frame. The framework is shown in Fig. 2.

The features and descriptors are computed by using SURF, the matching is executed based on the GMS. Then, the transform is estimated by using RANSAC by the matched key points. We evaluate the transform matrix based on our designed metric. If the transform is good, we directly compute the warped image and paste the warped image onto the reference image; otherwise, the new key frame is selected by using the warped area in the global image produced in the last iteration, then the transform of the current frame is recomputed relative to the key frame. The stitching boundary is computed according to the matrix, and the current size of the new stitched image is computed for the dynamic increment. The main contribution includes:

(i) A general video mosaicking algorithm is proposed. The present algorithms are aimed at either mapping or panorama. We combine these two scenes together, making the algorithm more widely used.

(ii) A metric to evaluate the transform matrix is designed. The transform of the current frame is analyzed, and we dynamically adjust the matrix to adapt to different scenes.

(iii) The dynamic growth algorithm is employed for the global image, which satisfies the mosaicking of videos of any length by pre-computing the feature size of the global image.

3.1 Image matching based on SURF and GMS

The SURF is efficient on feature extraction and description. Inspired by SIFT, the SURF is several times faster than the SIFT, while the precision is nearly not affected. The SURF searches for stable edge pixels across the image by using the Hessian matrix, and counts the main directions of feature vectors in a range of angles, then computes the descriptors of these vectors in the neighborhood of the main directions.
GMS is an excellent feature matching algorithm with high precision and robustness [16]. The coarse matching using Brute-force (BF) is computed firstly, then GMS filters the mismatches and picks the correct matches from the results of BF according to the probability of the matching point to be true. It is considered in GMS that the true matching point has many other true matching points in its neighborhood region. The score of the ith region satisfies the Bernoulli distribution:

$$S_i \sim \begin{cases} B(Kn, p_t), & x_i \text{ is true} \\ B(Kn, p_f), & x_i \text{ is false} \end{cases} \quad (1)$$

where $K$ is the number of disjoint regions, $x_i$ is the feature point, $n$ is the number of matches in the ith region, $p_t$ and $p_f$ represent the true probability and false probability respectively. The mean and standard deviations of (1) are

$$\begin{align*}
m_t &= Kn p_t, s_t = \sqrt{Kn p_t (1 - p_t)} \\
m_f &= Kn p_f, s_f = \sqrt{Kn p_f (1 - p_f)}.
\end{align*} \quad (2)$$

The objective of GMS is to maximize the following function:

$$P = \frac{m_t - m_f}{s_t + s_f} = \sqrt{Kn} \frac{p_t - p_f}{\sqrt{p_t (1 - p_t)} + \sqrt{p_f (1 - p_f)}}. \quad (3)$$

In practice, two images are divided into several grids respectively, and the region with the largest number of matches is picked to calculate the score by (1). The matches of this region are considered to be true if the score is larger than a threshold.

The matching precision of SURF and GMS methods is much higher than that of SURF and BF methods, as shown in Fig. 3. We choose two dependent frames to compute the feature key points and descriptors by using SURF, and compute feature correspondence by using the GMS and BF respectively. The top 30 of the best matches are visualized by using yellow lines.

3.2 Transformation matrix and evaluation metric

Image stitching is a process to warp the current image by using the linear transform so that it has the overlap with the reference image as large as possible. The overlap ratio of the corresponded pixel is an important evaluation metric of the stitching [14]. However, we focus on a new metric for the transform, or, its matrix. The transform can be formulated as follows:

$$Y = HX \quad (4)$$

where $X$ represents the original image, $H$ is the matrix of the transform, and $Y$ is the warped image.

To realize the stitching of the continuous image, we define a key frame $F = X_0$, which is usually initialized as the first frame of the video. We calculate the transform between two consecutive frames in order to compute the transform of the current frame relative to the key frame

$$\begin{align*}
X_i &= H_{i,i-1}X_{i-1} \\
X_{i-1} &= H_{i-1,i-2}X_{i-2}.
\end{align*} \quad (5)$$

then we get

$$X_i = \prod_{1}^{i} H_{i,i-1}X_{0} = \prod_{1}^{i} H_{i,i-1}F \quad (6)$$

which means that the transform of any frame relative to the key frame is the accumulated product of all transforms of the consecutive frames

$$H_i = \prod_{1}^{i} H_{i,i-1}. \quad (7)$$

In this way, we can achieve the stitching of videos of any length.

The RANSAC is adopted for the homography of two consecutive frames. In general, the homography $H$ is a $3 \times 3$ matrix

$$H = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \quad (8)$$

and it can be divided into three parts:

$$H = \begin{bmatrix} L & T \\ P \end{bmatrix} \quad (9)$$

where

$$L = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, T = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}, P = \begin{bmatrix} a_{31} & a_{32} & 1 \end{bmatrix}.$$  

$L$ produces the linear transforms like rotation, scaling and shearing, $T$ produces the shifting, and $P$ produces the perspective effect. We can also use the affine transform, which

![Fig. 3 SURF+BF (up) and SURF+GMS (down)](image-url)
does not consider the difference between the image planes, and we have \( P = [0 \ 0 \ 1] \) in affine transformation.

As for mapping and panorama tasks, we should use homography and affine respectively. For mapping, all images are expected on the same plane, while this is nearly impossible in fact with the distortion and shaking of cameras, thus we need to adjust the planes dynamically using the homography. On the other hand, in panorama tasks, we use the affine operation because all the images are expected on a spherical surface instead of a plane, thus we cannot forcibly put all images into a single plane, which will cause severe distortion and lead to a failure. In Fig. 4, we show two failed cases where we use homography for panorama and affine for mapping. We can see clearly in the upper image of Fig. 4, the distortion becomes explosively larger and larger in the right part of the image because of the over perspective effect, which is going to fail after several frames. We call this over-perspective. The below part is a mapping image with the distortion and shaking of cameras, and the camera is moving along a large loop while taking this video. We use the affine transformation for this mapping video [3], and the result shows that there is an obvious shifting inside the box, which we call poor-perspective. In this way, we believe that it is necessary to evaluate the transform and adapt to different scenes.

The homography \( H \) estimated by RANSAC contains accumulated errors which will be magnified in (7). We define a metric to evaluate the perspective effect of the transform by using

\[
\rho = \sqrt{a_{31}^2 + a_{32}^2}
\]

where \( a_{31} \) and \( a_{32} \) are the first two values of the last row of \( H \). If \( \rho \) is larger than a threshold \( \tau \), it means the perspective effect is too strong which may lead to cause severe distortion. Empirically, we find it is effective for \( \tau \in (0.0001, 0.0004) \).

### 3.3 Self-adapted stitching based on dynamic key frame

In order to eliminate the accumulated error of the homography when the perspective effect is strong enough. We need to select a new key frame and re-compute the homography relative to the key frame.

In the proposed method, we crop the new key frame from the global image. The global position of the key frame is recorded, which is used to compute the global position of the warped image. We crop a patch centered at the recorded position with the fixed size as a new frame, if the homography is bad. Besides, we also update the key frame for long videos in each \( t \) frame even if the homography is good, in case for shifting during the stitching, especially when there are large loops in the motion of the camera. Empirically, \( t \) depends on the frame ratio of the video. We make \( t = 2f_{num} \) in our experiment, where \( f_{num} \) is the number of frames per second.

We use the sequences in Fig. 4 to validate the proposed method with the dynamic key frame, the results are shown in Fig. 5.
In the upper part of Fig. 5, we can see that the severe distortion is eliminated, and the proposed method stitches this video as a panorama correctly. In the lower part, the shifting shown in Fig. 4 is partially eliminated and the mapping video is stitched more precisely.

It is inevitable for blurring and shadow in the global image, and the feature matching may fail if we directly crop the key frame from the global image. In this case, we use the affine operation between the current and last frames. The matrix of the affine transform satisfies \( P = [0 \ 0 \ 1] \) and \( \rho \) is unchanged after the accumulated product of the matrices in (7). We use this as a replacement for cropping the key frame, until the feature matching succeeds.

### 3.4 Self-growth of global image

The transform between any frame and the key frame can be computed. Given the original coordinates of the corners of the image, we can compute the warped coordinates \( z' \) by (4) as
\[
z' = H z
\]  
then we can adjust the position of the global image \( G \) dynamically. As discussed, the coordinate is used to crop the key frame from \( G \), while now we use it to calculate the increment of the width and height and the new position of \( G \). In practice, we firstly create an empty image \( G' \), then paste \( G \) and the warped tile \( Y_i \) to \( G' \), thus realizing the dynamic growth of the stitched image, as shown in Fig. 6.

![Fig. 6 Self-growth algorithm](image)

The self-growth algorithm is the key to realize the mosaicking with the camera moving at any direction and angle. The proposed algorithm is described as follows.

---

**Algorithm 1** Key frame video mosaicking

Input: A sequence of images

\[
t = 0
\]

for \( i = 1 \) to \( N \) do

Compute match between \( X_i \) and \( X_{i-1} \)

Compute homography from \( X_{i-1} \) to \( X_i \)

Estimate the quality of the homography

if \( \rho < \tau \) or \( t == f \) then

Crop new key frame \( K \)

Compute match between \( X_i \) and \( K \)

if no match then

Compute affine from \( X_{i-1} \) to \( X_i \)

else

Compute homography from \( K \) to \( X_i \)

if \( t == f \) then

\[
t = 0
\]

end if

\[
H = H_i
\]

else

\[
H = H H_i
\]

end if

\[
t = t + 1
\]

Compute the warped frame by (4)

Compute warped coordinates by (11)

end for

Output: Stitched image

---

### 4. Experiments and evaluation

We execute the proposed algorithm on several sequences. As there is no public dataset for video mosaicking and the quality is mostly judged by the subjective assessment for the stitched image. There are metrics for stitching of two images [14]; however, they are not suitable for the video mosaicking with a large set of images. We select several mapping sequences from [12], panorama sequences from [3] and our videos.

The proposed algorithm is developed and implemented on Ubuntu16.04 with Intel Corei7 2.8 GHz CPU and NVidia GTX1050 GPU. OpenCV3.3.0 and CUDA8.0 libraries are used.

#### 4.1 Video mosaicking results

We select two mapping sequences from [12] and one taken by our UAV; one panorama video from [3] and one taken by the smartphone. In our experiment, we make \( \tau = 0.0003 \) and align the height of each frame to 360 when extracting features. The result shown in Fig. 7 is a comparison of the proposed algorithm, method in [3] and Pix4DMapper. It can be seen that the precision of the proposed method on mapping sequences approaches the result.
JI Yufeng et al.: Automatic video mosaicking algorithm via dynamic key-frame

of Pix4DMapper, which takes more than 15 min for a sequence with no more than 200 frames. Pix4DMapper is quite good at dealing with mapping images; however, it has no ability for panorama images. The quality of stitching of the proposed method is also obviously better than the method in [3], which is not good at dealing with very long aerial-view videos. It is because the method in [3] does not consider using a new key frame to eliminate the accumulated error, while the proposed method evaluates the transform matrix and dynamically chooses the key frame, which can mitigate the accumulated error and satisfy the long-term video mosaicking. Besides, the method in [3] cannot produce maps from videos of arbitrary lengths, while the proposed algorithm solves this problem via the dynamic growth.

![Fig. 7 Results of the proposed method, method in [3] and Pix4DMapper](image)

### 4.2 Running time of our algorithm

The main time-consuming process of video mosaicking is feature extraction and matching. With the acceleration of GPU device, the proposed method satisfies the real-time requirement. Table 1 lists the speed of the proposed algorithm with CPU and GPU respectively. The second column of Table 1 represents the seconds for feature extraction and matching, the third column is the mean frame rate per-second. The speed of 22 frames per second (FPS) is enough for real-time requirements.

| Device   | SURF+GMS/s | FPS |
|----------|------------|-----|
| CPU      | 0.128      | 8   |
| GPU      | 0.027      | 22  |

### 5. Conclusions

In this paper, a general video mosaicking algorithm is proposed, which merges the mapping and panorama tasks together. The metric for the transform matrix evaluation is designed to reduce the accumulated error. The key frame is
updated dynamically after the matrix evaluation if the matrix is bad. The key frame is chosen in each frame in case of shifting. The self-growth of the global image is used for videos of any length.

There are still several flaws in the proposed method. The metric for the matrix is simply based on the perspective effect and lacks generality. We seek for stronger metrics for the evaluation, which considers more conditions, thus improve the precision of the proposed method.

References

[1] ZITOVA B, FLUSSE R J. Image registration methods: a survey. Image and Vision Computing, 2003, 21(11): 977 – 1000.
[2] LI J, YANG T, YU J Y, et al. Fast aerial video stitching. International Journal of Advanced Robotic Systems, 2014, 11(10): 167 – 177.
[3] AMIRI A J, MORADI H. Real-time video stabilization and mosaicking for monitoring and surveillance. Proc. of the 4th International Conference on Robotics and Mechatronics, 2016: 613 – 618.
[4] BROWN M, LOWE D G. Automatic panoramic image stitching using invariant features. International Journal of Computer Vision, 2007, 74(1): 59 – 73.
[5] HEINZE N, ESSWEIN M, KRUGER W, et al. Automatic image exploitation system for small UAVs. Proc. of the International Society for Optics and Photonics, 2008: 1 – 10.
[6] BAY H, ESS A, TUYTELAARS T, et al. Speeded-up robust features (SURF). Computer Vision and Image Understanding, 2008, 110(3): 346 – 359.
[7] DAVID G L. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91 – 110.
[8] HARRIS C, STEPHENS M J. A combined corner and edge detector. Proc. of the Alvey Vision Conference, 1988: 147 – 152.
[9] RUBL E E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF. Proc. of the International Conference on Computer Vision, 2011: 2564 – 2571.
[10] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection. Proc. of the European Conference on Computer Vision, 2006: 430 – 443.
[11] ZHU J, REN M. Image mosaic method based on SIFT features of line segment. Computational and Mathematical Methods in Medicine, 2014: 1 – 11.
[12] BU S, ZHAO Y, WAN G, et al. Map2DFusion: real-time incremental UAV image mosaicking based on monocular SLAM. Proc. of the International Conference on Intelligent Robots & Systems, 2016: 4564 – 4571.
[13] LIN C C, PANKANTI S U, RAMAMURTHY K N, et al. Adaptive as-natural-as-possible image stitching. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1155 – 1163.
[14] YANG Y, WANG D S, SONG W J, et al. Dynamic video stitching method embracing image semantics. Systems Engineering and Electronics, 2018, 40(12): 2845 – 2854. (in Chinese)
[15] SZELISKI R, SHUM H Y. Creating full view panoramic image mosaics and environment maps. Proc. of the 24th Annual Conference on Computer Graphics and Interactive Techniques, 1997: 251 – 258.
[16] BIAN J, LIN W Y, MATSUSHITA Y, et al. GMS: grid-based motion statistics for fast, ultra-robust feature correspondence. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4181 – 4190.

Biographies

Jì Yufeng was born in 1994. He received his B.S. degree from Beijing Institute of Technology in 2016. Since 2016, he has been a postgraduate student majoring in control science and engineering in Beijing Institute of Technology. His research interests include machine learning and objects detecting, and tracking of UAVs.
E-mail: jyf131462@163.com

Li Weixing was born in 1976. He received his M.S. degree in pattern recognition and intelligent control from Beijing Institute of Technology. He is an associate professor at School of Automation in Beijing Institute of Technology, China. His research interests include pattern recognition, video analysis and machine learning.
E-mail: liweixing@bit.edu.cn

Feng Kai was born in 1996. He received his B.S. degree from Beijing Institute of Technology in 2018. Since 2018, he has been a postgraduate student majoring in control science and engineering in Beijing Institute of Technology. His research interests include computer vision and deep learning.
E-mail: bitfengkai@163.com

Xing Boyang was born in 1990. He received his B.S. degree from Beijing Institute of Technology in 2012. Since 2015, he has been a doctoral student majoring in control science and engineering in Beijing Institute of Technology. His research interests include computer vision and deep learning.
E-mail: golaced@163.com

PAN Feng was born in 1978. He received his M.S. and Ph.D. degrees in pattern recognition and intelligent control from Beijing Institute of Technology. He is an associate professor at School of Automation in Beijing Institute of Technology. His research interests include intelligent computing, robotics and control, and computer vision.
E-mail: andropanfeng@126.com