J. Vis. Commun. Image R. 34 (2016) 1–11

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A survey on image mosaicing techniques ☆

Debabrata Ghosh *, Naima Kaabouch

Department of Electrical Engineering, University of North Dakota, United States

ARTICLE INFO

Article history:
Received 2 February 2015
Accepted 19 October 2015
Available online 30 October 2015

Keywords:
Image mosaicing
Registration
Blending
Geometric transformation
Homography
Low-level feature
Transition smoothening
Optimal seam

ABSTRACT

Image mosaicing, the process of obtaining a wider field-of-view of a scene from a sequence of partial views, has been an attractive research area because of its wide range of applications, including motion detection, resolution enhancement, monitoring global land usage, and medical imaging. A number of image mosaicing algorithms have been proposed over the last two decades. This paper provides an in-depth survey of the existing image mosaicing algorithms by classifying them into several groups. For each group, the fundamental concepts are first explained and then the modifications made to the basic concepts by different researchers are explained. Furthermore, this paper also discusses the advantages and disadvantages of all the mosaicing groups.

1. Introduction

Nowadays, image mosaicing is gaining a lot of interests in the research community for both its scientific significance and potential derivatives in real world applications. Image mosaicing is the alignment of multiple overlapping images into a large composition which represents a part of a 3D scene [1]. Mosaicing could be regarded as a special case of scene reconstruction where the images are related by planar homography only [2]. This is a reasonable assumption if the images exhibit no parallax effects, i.e. when the scene is approximately planar or the camera purely rotates about its optical center [3]. Using mosaicing it is possible to extend the field of view (FOV) of a camera by preserving the original resolution and without introducing undesirable lens deformation [4]. There have been a variety of new additions to the classic applications of image mosaicing that primarily aim to augment the FOV. Mosaic construction is finding its practices in many computer vision and computer graphics applications, such as motion detection and tracking [5–7], mosaic-based localization [8,9], resolution enhancement [10–12], and augmented reality [13,14]. Furthermore, video compression [15], video indexing [16], and image stabilization [17] are some of the prominent areas where mosaicing is creating significant impacts.

As shown in Fig. 1, mosaicing involves various steps of image processing: registration, reprojection, stitching, and blending. Registration refers to the establishment of geometric correspondence between a pair of images depicting the same scene. In order to register a set of images, it is required to estimate the geometric transformations which align the images with respect to a reference image within that set. The set may consist of two or more images taken of a single scene at different times, from different viewpoints, and/or by different sensors. The most general case of the transformation is the 8 degree of freedom planar homography [1]. The next step, following the registration, is reprojection which refers to the alignment of the images into a common coordinate system using the computed geometric transformations. The goal of the stitching step is to overlay the aligned images on a larger canvas by merging pixel values of the overlapping portions and retaining pixels where no overlap occurs. Errors propagated via geometric and photometric misalignments often result in undesirable object discontinuities and seam visibility in the vicinity of the boundary between two images. Thus, a blending algorithm needs to be used during or after the stitching step in order to minimize the discontinuities in the global appearance of the mosaic. The aforementioned registration step has been conceived to work with images with a single color band. Different techniques have been used by different mosaicing algorithms to deal with multiple color bands. For example, in [18–21] one of the color bands of the input RGB images are taken into consideration while obtaining the transformation parameters. In [22–24] on the other hand, the RGB images are first converted to grayscale and then transformation parameters are obtained. In either case, after finding the optimal transformation parameters, all the color bands are processed and

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author.
E-mail address: debabrata.ghosh@my.und.edu (D. Ghosh).

http://dx.doi.org/10.1016/j.jvcir.2015.10.014
1047-3203/

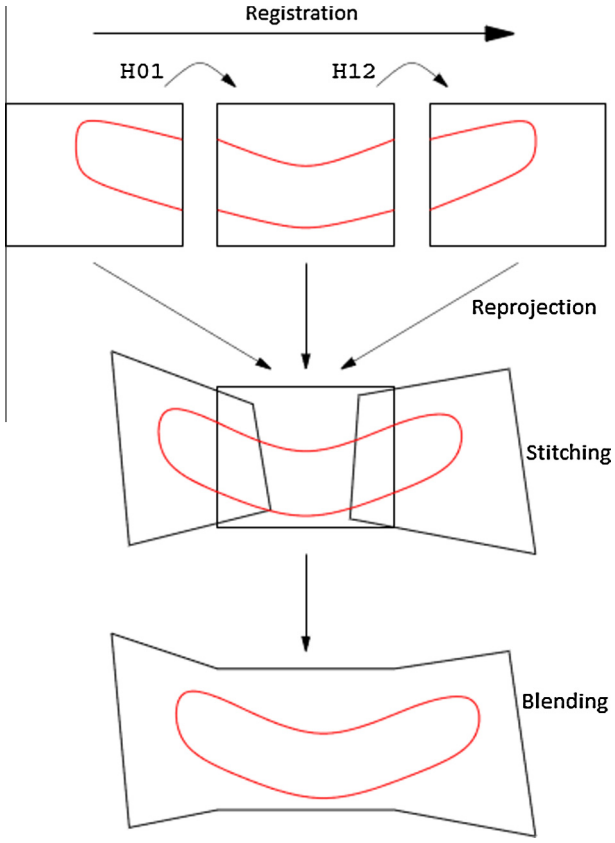

Fig. 1. Different steps of image mosaicing. Here H are the homography matrices between source images. Adapted from [1].

combined together during the reprojection step in order to produce color mosaic.

A number of mosaicing algorithms have been proposed in the literature [10,19,23–36]. Even though the state of the art indicates advancements in this research area in recent years, image mosaicing still remains a challenge because of several factors like registration and blending. Since pose and acquisition systems vary, the set of possible observations of a scene is immense [37]. Thus the task of determining the correspondences between observed images is complicated. Similarly, reducing the visible inconsistencies near the boundaries of the overlapping images is also challenging. While the majority of the recent works focus specifically on dealing with the aforementioned challenges, a comprehensive review of the existing algorithms remain mostly overlooked. Literature review shows that only a few review papers [38–42] on the existing image mosaicing techniques have been carried out. In [38,41,42], the authors review the existing mosaicing techniques based on a specific image registration method. [39] gives an overview of the different steps of image mosaicing techniques. However, the authors did not categorize the existing methods. [40] presents a review work in the field of document image mosaicing and retina image mosaicing only. Thus, none of the existing survey discuss the major categories of image mosaicing algorithms and ultimately fail to classify the most recent image mosaicing techniques. The continuous emergence of new image mosaicing algorithms in recent years necessitates such a review, which will be valuable guide to researchers and developers for selecting a suitable image mosaicing method for a specific application. In this paper, we classify the past and current mosaicing techniques based on image registration as well as image blending. For each of these classifications, we provide a comprehensive review of the major categories of the image mosaicing methods. The basics of these categories are first described. Then, for each of these basic categories, the evolving paths are discussed by providing the modifications that have been applied to the basic methods by different researchers. Since the current state-of-the-art is very large, only those works, which we think contributed significantly to the mosaicing literature, are discussed in this manuscript.

The rest of the paper is organized as follows: Section 2 provides the taxonomies of the existing mosaicing algorithms. Section 3 explains the classification of mosaicing methods based on image registration. Section 4 reviews the classification of mosaicing algorithms based on image blending. Finally, the paper comes to a conclusion in Section 5.

2. Classification of image mosaicing algorithms

Image registration and blending are the two significant research areas which directly influence the image mosaicing performance. Being the first and last step of image mosaicing, it is almost impossible to build a successful mosaicing algorithm without correctly implementing registration and blending algorithms. Though attempts have been made to overcome the registration errors by utilizing sophisticated blending algorithms, the significance of accurate registration in image mosaicing still remains unquestionable. In this paper, we focus on classification of the existing image mosaicing algorithms based on their registration methods, as well as based on their blending methods.

As shown in Fig. 2, based on image registration methods, image mosaicing algorithms can be spatial domain-based or frequency domain-based. Spatial domain-based image mosaicing can further be grouped into area-based image mosaicing and feature-based image mosaicing. Feature-based image mosaicing can again be subdivided into low level feature-based image mosaicing and contour-based image mosaicing. Low level feature-based mosaicing can be divided into four classes: Harris corner detector-based mosaicing, FAST corner detector-based mosaicing, SIFT feature detector-based mosaicing, and SURF detector-based mosaicing. As shown in Fig. 3, based on the image blending techniques, mosaicing algorithms can be transition smoothening-based and optimal seam-based. Transition smoothening-based mosaicing can further be grouped into feathering-based, pyramid-based, and gradient-based mosaicing.

3. Classification of image mosaicing based on registration

Image registration is not only an important step of image mosaicing, but also the foundation of it. Registration of multi-source images, which are focused on the same target but produced from different sensors, different perspective, and different times, computes the optimal geometric transformation by looking into the correspondences between each pair of images. This process makes the multi-source images aligned into a common reference frame using the estimated geometric transformations. To the extent that corresponding points from multi-source images are aligned together, the registration is successful [43]. The aforementioned correspondences can be established either by matching templates between images, or by matching features extracted from images, or by utilizing the phase correlation property in the frequency domain. Different classes of image mosaicing algorithms based on the image registration are discussed in the following two subsections.

3.1. Spatial domain image mosaicing algorithms

Algorithms in this category use properties of pixels to perform registration, and, thus they are the most direct methods of image

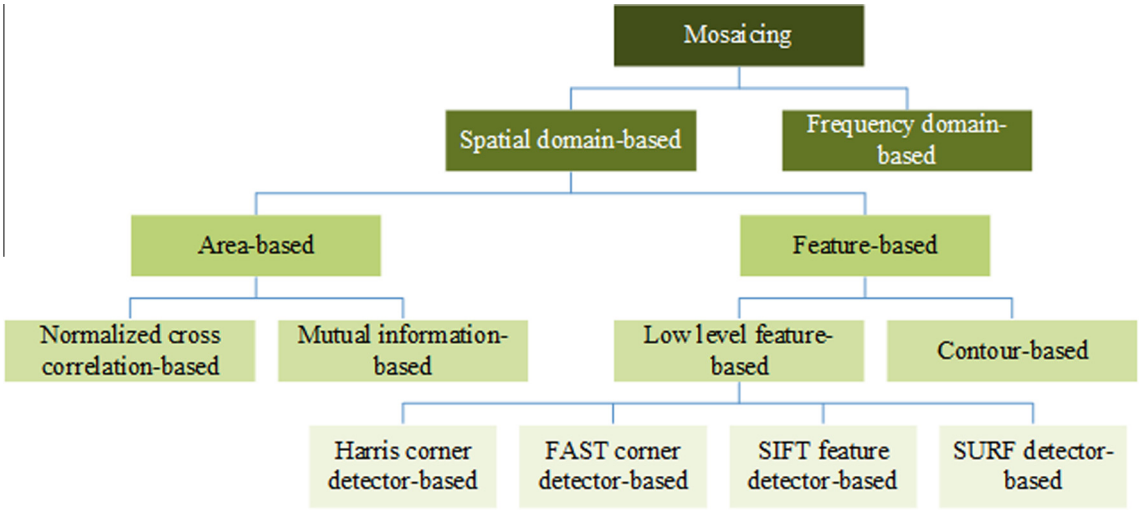

Fig. 2. Classification of mosaicing based on registration.

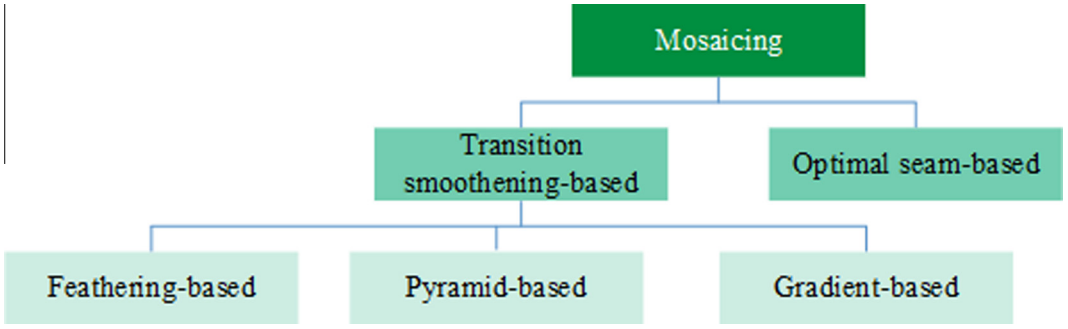

Fig. 3. Classification of mosaicing based on blending.

mosaicing. Majority of the existing image mosaicing algorithms fall into this category. Spatial domain-based image mosaicing can be either area-based or feature-based. Area-based image mosaicing algorithms rely on computation between "windows" of pixel values in the two images, which need to be mosaicked [18]. The fundamental approach is to shift the "windows" of the images relative to each other and see how much the pixels match. Subsequently, transformation parameters are obtained and used to warp and stitch the images. Area-based mosaicing algorithms are often referred to as pixel-based mosaicing, since they use pixel-to-pixel matching, as opposed to feature-to-feature matching used in the feature-based mosaicing. Two of the most commonly used area-based image mosaicing algorithms are normalized cross correlation-based mosaicing and mutual information-based mosaicing. Both of these methods provide a measure of image similarity, because larger values of these metrics result from matching areas or "windows".

3.1.1. Normalized Cross Correlation (NCC)-based mosaicing

This method computes similarity between the "windows" in the two images for each shifts. It is defined as [20]:

$$NCC(u) = \frac{\sum_i [I_1(x_i) - \overline{I_1}][I_2(x_i+u) - \overline{I_2}]}{\sqrt{\sum_i [I_1(x_i) - \overline{I_1}]^2 [I_2(x_i+u) - \overline{I_2}]^2}} \quad (1)$$

where

$$\overline{I_1} = \frac{1}{N}\sum_i I_1(x_i) \quad (2)$$

$$\overline{I_2} = \frac{1}{N}\sum_i I_2(x_i+u) \quad (3)$$

where $\overline{I_1}$ and $\overline{I_2}$ are the mean images of the corresponding "windows", $I_1(x,y)$ and $I_2(x,y)$ for the first and second images respectively. N is the number of pixels in the "window", $x_i = (x_i, y_i)$ is the pixel coordinate in the "windows", $u = (u, v)$ is the displacement or shift where NCC coefficient is calculated. The NCC coefficient values are always within the range [−1, 1]. The shift parameter corresponding to the peak NCC value represents the geometric transformation between the two images. Once geometric transformations are obtained between the image pairs, images are warped in the reference frame, and finally stitching is performed to generate the final mosaic. Methods within this category have the advantage of being computationally simple, however, at the cost of being particularly slow. Moreover, they perform accurately only when there are significant overlapping between the source images.

Several techniques [27,44–46] have been proposed to tackle the above mentioned problems. Nasibov et al. [27] employed a brightness correction matrix before the registration step in order to make the algorithm less sensitive to the illumination changes. In order to make the computation faster, Berberidis et al. [44] proposed an iterative algorithm for the spatial cross correlation to compute the displacements between the source images. Yet another method based on adjusting the correlation windows according to the scale and orientation of extracted interest-points from the source images was proposed by Zhao et al. [45] to increase the computation speed. In order to improve the performance of the algorithm in

the presence of non-rigid deformation, Vercauteren et al. [46] suggested the use of Riemannian statistics along with a scattered data fitting-based mosaicing.

3.1.2. Mutual Information (MI)-based mosaicing

Unlike NCC, which computes similarity based on image intensity values, mutual information measures similarity based on the quantity of information shared between two images. MI between two images $I_1(x,y)$ and $I_2(x,y)$ is expressed in terms of entropy as:

$$MI(I_1,I_2) = E(I_1) + E(I_2) - E(I_1,I_2) \tag{4}$$

where $E(I_1)$ and $E(I_2)$ are the entropies of $I_1(x,y)$ and $I_2(x,y)$, respectively. $E(I_1,I_2)$ represents the joint entropy between the two images. Entropy is a measure of variability of a random variable. Thus variability of $I_1(x,y)$ is expressed as:

$$E(I_1) = -\sum_g p_{I_1}(g)\log(p_{I_1}(g)) \tag{5}$$

where g are the possible gray level values of $I_1(x,y)$, and accordingly $p_{I_1}(g)$ is the probability distribution function of g. Similarly, the joint variability of $I_1(x,y)$ and $I_2(x,y)$ is expressed as:

$$E(I_1,I_2) = -\sum_{g,h} p_{I_1,I_2}(g,h)\log(p_{I_1,I_2}(g,h)) \tag{6}$$

where h indicates the possible gray level values of $I_2(x,y)$. $p_{I_1,I_2}(g,h)$ is the joint probability distribution function of g and h. Typically, the joint probability distribution between two images is measured as normalized joint histogram of the gray level values. It is observed that better the alignment between two images, higher the MI between them. Thus, two images are geometrically aligned by a transformation if the MI between them is maximum for that transformation. After the appropriate transformations are obtained between the image pairs, they are reprojected and stitched to get the final mosaic. These mosaicing methods have the advantage of being less sensitive to lighting and occlusion changes between source images. However, similar to NCC-based methods, these techniques have the disadvantages of being computationally slow, and requiring high degree of overlapping between input images.

Several of techniques [29,47,48] have been proposed to address the aforementioned shortcomings. In [29], Luna et al. used a stochastic gradient optimization along with MI-based similarity measure in order to make the algorithm faster. To increase the computation speed, Dame et al. [47] employed a B-spline function for normalized mutual probability density. They further used Newton's method to speed up the estimation of the displacement parameters. Concerning the drawback of MI-based mosaicing algorithms for low overlapping images, Césare et al. [48] proposed a template matching approach capable of explicitly acknowledging the plausibility of similarity between distant neighborhoods, and delaying definite block-to-block association to a step that globally evaluates their collective likelihood.

Unlike the area-based methods, feature-based mosaicing techniques use feature-to-feature matching in order to compute the geometric transformation between a pair of images. Thus these methods rely primarily on feature extraction algorithms which can detect salient features from the images. Salient features are subsets of the image domain, often in the form of isolated points, continuous curves or connected regions [49]. The general approach is to detect a few corresponding features from the source images, and then estimate homography using the reliable correspondences only. Using the homography matrices images are first warped and then stitched in a common reference coordinate. Since the features are used as the starting point, the overall algorithm will often be as good as the feature extraction algorithm is. Methods falling into this category give in general better results than the area-based techniques, however, at the cost of requiring high computation. Depending on the types of features extracted, feature-based mosaicing methods can be classified into low-level feature-based mosaicing and contour-based mosaicing.

3.1.3. Low-level feature-based mosaicing

These methods do not require images with large overlapping areas in order to mosaic them. This category of mosaicing algorithms rely on the computation of transformation using sparse set of low-level features. Commonly used low-level features include edge, corner, pixel, color, histogram etc. Irrespective of which low-level feature is chosen, it must be distinct and spread all over the image, and also it should be efficiently detectable in both the images [50]. The feature detector algorithm should be such that the number of common features detected from a set of images is sufficiently high even in the presence of the various geometric and radiometric changes. Additionally, the detector must have high repeatability rate, such that same features are detected in the overlapping regions between pair of images. Popular low-level feature extraction methods used in mosaicing literature include: Harris corner detector, Features from Accelerated Segment Test (FAST)-based corner detector, Scale Invariant Feature Transform (SIFT)-based feature detector, and Speeded Up Robust Feature (SURF)-based feature detector. Mosaicing algorithms based on these feature extraction techniques are discussed briefly below:

3.1.3.1. Mosaicing based on Harris corner detector. Harris corner detector detects corner points as robust low-level features from source images. Initially a local detection window in an image is chosen. Subsequently the variation in intensity that results by shifting the window by a small amount in different direction is determined as below [41]:

$$E(u,v) = \sum_i w(x_i,y_i)[I(x_i+u,y_i+v) - I(x_i,y_i)]^2 \tag{7}$$

where $w(x_i,y_i)$ is the window function for the detection window (x_i,y_i), $I(x_i,y_i)$ is the image intensity value at pixel location (x_i,y_i), and $I(x_i+u,y_i+v)$ is the shifted intensity with (u,v) shift. The local texture around pixel (x_i,y_i) is expressed as autocorrelation matrix Cas below:

$$C = \sum_i w(x_i,y_i)\begin{bmatrix} I_{x_i}^2 & I_{x_i}I_{y_i} \\ I_{x_i}I_{y_i} & I_{y_i}^2 \end{bmatrix} \tag{8}$$

where I_{x_i} and I_{y_i} are the first derivative of $I(x_i,y_i)$. Two large eigenvalues for the matrix C corresponds to a corner point. The center point of the window is characterized as a corner point. For more robustness, a "cornerness" measure R is used to eliminate the edge points as below [30]:

$$R = Det(C) - \alpha Tr^2(C) \tag{9}$$

where $Tr(C)$ is the trace of C and α is within the range $0.04 \leqslant \alpha \leqslant 0.06$. Corner points are detected as local maxima of R above a predefined threshold T. After the Harris corner points are detected from both the images, correspondences are established either by NCC or by any other Sum of Squared Difference (SDD) method. Subsequently, the geometric motion parameters are calculated and images are warped into a global reference frame in order to stitch them all. Mosaicing algorithms using Harris corner detector are computationally simple and accurate.

Harris corner detector almost always finds closely crowded features. However, this can be overcome by counting the number of features in the neighborhood and then accordingly exclude some of the points, as has been done in [23]. One major problem with the Harris corner detector-based mosaicing methods is that large

changes in rotation often generates ghosting in the mosaic output. [30] dealt with this by utilizing a luminance center-weighting algorithm which is used following a slope clustering algorithm for Harris corner point matching. Another problem related to the uncertainty in choosing a local detection window was addressed in [51], where the authors used region segmentation and matching in order to limit the search window to potential homologous points.

3.1.3.2. Mosaicing based on FAST corner detector. FAST algorithm is a corner detection algorithm which is computationally more efficient and faster than most of the other low-level feature extraction methods; thus mosaicing methods based on this algorithms are particularly suitable for real-time image processing applications. Initially a circle of sixteen pixels is considered around each corner candidate. According to the FAST algorithm, the candidate is a corner if there exists a set of n contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel plus a threshold, or all darker than the intensity of the candidate pixel minus the threshold, as shown in Fig. 4. The number n is usually chosen twelve. In order to increase the speed of FAST algorithm, a corner response function (CRF) is used. CRF gives the numerical value of the "cornerness" of a corner point based on image intensities in the local neighborhood [41]. Corners are detected as local maxima for the CRF function computed over the entire image. Following the detection, corner point matching is performed for each pair of frames. Sometimes a Bag-of-Words (BoW) algorithm is used to represent each image as a set of corner descriptors to speed up the matching process as in [31]. Then, homography matrices are computed and finally the images are projected into a common coordinate to get the final mosaic.

Choosing an optimal threshold is often a fundamental challenge of the FAST corner detector-based algorithms. However, it can be addressed by incorporating a robust threshold selection algorithm as in [52]. For matching the corner points from successive frames, they further proposed a threshold learning method together with a region-based gray correlation. Another major issue of the FAST-based algorithms is that they are not particularly robust to increased degree of variations. For that, extending the sampling area beyond the sixteen pixels around each candidate point [53] could be considered as a promising approach, since it gives the FAST corner points more distinctiveness and, in turn, makes them invariant to larger variations.

3.1.3.3. Mosaicing based on SIFT feature detector. SIFT algorithm is a low-level feature detection algorithm which detects distinctive features (also called "keypoints") from images. The SIFT descriptor is invariant to translations, rotations and scaling transformations in the image domain and robust to moderate perspective transformations and illumination variations. SIFT's operation is based on

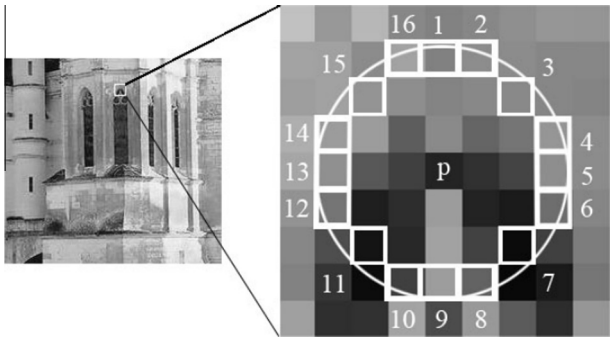

Fig. 4. Candidate feature detection for FAST algorithm, adapted from [53].

five major steps: scale-space construction, scale-space extrema detection, keypoint localization, orientation assignment, and defining keypoint descriptors. Initially, a scale space is constructed by convolving an image repeatedly using a Gaussian filter with changing scales and grouping the outputs into octaves as [54]:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{10}$$

where $*$ is the convolution operator, $G(x,y,\sigma)$ is a Gaussian filter with variable scale σ, and $I(x,y)$ is the input image. After the scale space construction is complete, difference-of-Gaussian (DoG) images are computed from adjacent Gaussian-blurred images in each octave as [54]:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{11}$$

Following that, candidate keypoints are identified as local extrema of DoG images across the scales. The scale space and DoG scale space construction as well as extrema detection in DoG scale space is illustrated in Fig. 5. In the next step, low contrast keypoints and edge response points along the edges are discarded using accurate keypoint localization. The keypoints are then assigned one or more orientations based on local image gradient directions as [54]:

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{12}$$

where $\theta(x,y)$ represents the gradient direction for $L(x,y,\sigma)$. A set of orientation histograms is formed over the neighborhoods of each keypoint. Finally, a normalized 128-dimensional vector is computed for each keypoint as its descriptor [11]. In order to find the initial matching keypoints from two images, nearest neighbor of a keypoint in the first image is identified from a database of keypoints for the second image [54,55]. Following the initial matching, RANSAC algorithm is used to remove the outliers and to compute the transformation parameters between a pair of frames. Finally, images are warped using the transformation parameters and stitched to generate the mosaic image. SIFT based image mosaicing algorithms are particularly suitable for stitching high resolution images under variety of changes (rotation, scale, affine, etc.), however, at the cost of high processing time.

Several researchers have made variations to the above mentioned SIFT-based mosaicing method in order to further improve its performance. For example, in [10] the authors proposed switching between Kanade–Lucas–Tomasi (KLT) tracker and SIFT matching to find the correspondences between successive frames depending on their amount of overlapping. In [32], the author exploited a deformation vector propagation algorithm in the gradient domain to reduce the intensity discrepancy between the mosaicked images. Similarly, a bundle adjustment algorithm along with a modified-RANSAC algorithm capable of developing a probabilistic model is used in [56] to eliminate registration error and make the matching process more accurate.

3.1.3.4. Mosaicing based on SURF feature detector. SURF algorithm is a scale and rotation invariant local feature detector. Like SIFT, this algorithm is also based on scale space theory. However, SURF uses Hessian matrix of the integral image to estimate local maxima across different scale spaces [57]. The Hessian matrix of an image I with scale σ at any point (x,y) is defined as [58]:

$$H(x,y,\sigma) = \begin{pmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{pmatrix} \tag{13}$$

where $L_{xx}(x,y,\sigma)$, $L_{yy}(x,y,\sigma)$, and $L_{xy}(x,y,\sigma)$ are the convolutions of I in point (x,y) with Gaussian second order filters $\frac{\partial^2}{\partial x^2}G(x,y,\sigma)$, $\frac{\partial^2}{\partial y^2}G(x,y,\sigma)$, and $\frac{\partial^2}{\partial x \partial y}G(x,y,\sigma)$ respectively. While computing

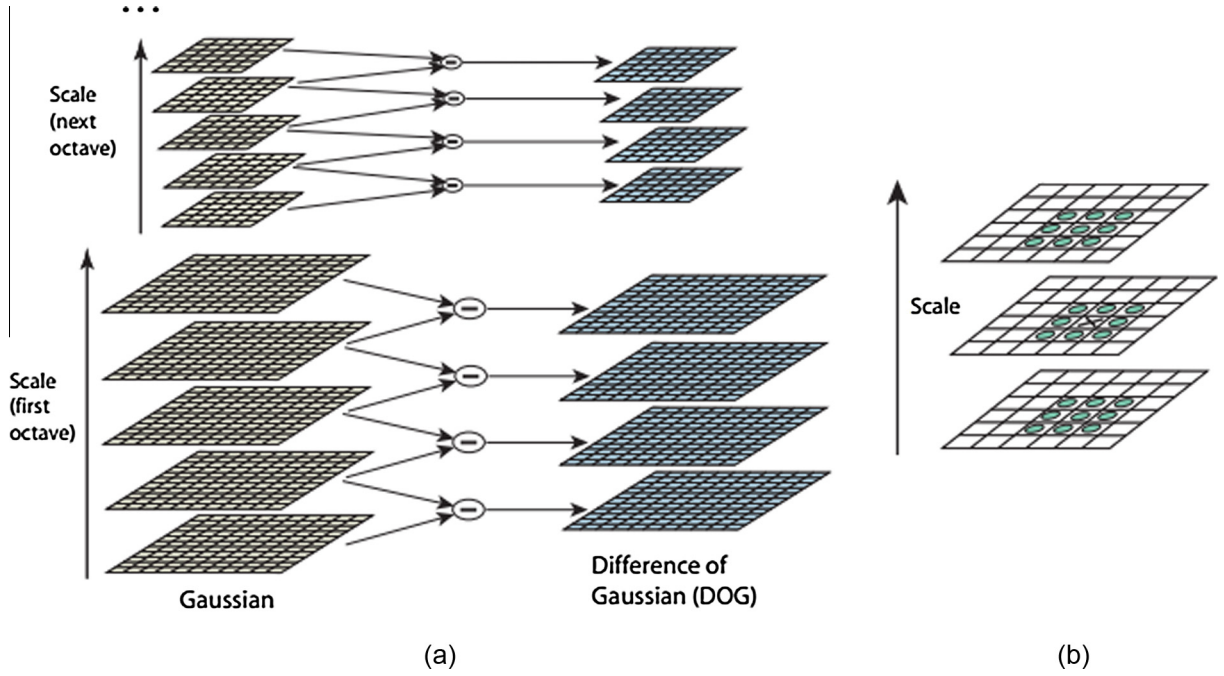

Fig. 5. (a) Scale space and DoG scale space construction and (b) extrema detection in DoG scale space by looking into 26 neighbors, adapted from [54].

Hessian matrix at each pixel, the Gaussian filter operations are approximated by operations using box filters as shown in Fig. 6. The response at each pixel is computed as the determinant of the Hessian matrix. Following that, a thresholding and a $3 \times 3 \times 3$ local maxima detection window are used for non-maxima suppression. The local maxima are then interpolated in scale space to achieve keypoints with their location and scale values. In order to assign orientation for each keypoint, Haar-wavelet responses are calculated within a circular neighborhood around each keypoint. A vector is then formed by summing up all the responses within 60-degree window. The longest vector is assigned as orientation to the keypoint. In order to assign descriptor vector to each keypoint, a square neighborhood region around the keypoint is selected. It is then split into smaller sub-regions. Sum of the Haar-wavelet responses from all the sub-regions are then used to generate a 64 dimensional descriptor vector [50]. After finding the matching keypoints from a pair of images, RANSAC algorithm is used to eliminate false matches as well as to calculate the homography matrices. Once homography matrices are achieved, images are warped and stitched to get the final mosaic. SURF-based mosaicing techniques are faster than SIFT-based techniques. However, they perform poorly under certain variations (particularly color, illumination, some affine transformation).

The process of determining the SURF descriptors as mentioned above has sometimes been modified by some authors. For example, in [59] the local maxima is searched beyond a $3 \times 3 \times 3$ neighborhood in the present scale and two immediately adjacent scales in order to make the feature descriptors more distinctive. In [60], the authors proposed dividing the SURF descriptor window into eight sub-regions while assigning descriptor vector. This technique increases the matching speed at the cost of increased number of false matches. However, the authors show that the use of RANSAC guarantees elimination of most of those incorrect matches.

Often multiple low-level feature extraction methods are used together in image mosaicing algorithms in order to use their respective benefits. Joshi et al. [61] proposed a mosaicing algorithm which uses both Harris corner detector and SURF detector for extracting distinctive features from source images. Feature-based mosaicing algorithm proposed by Bind et al. [62] used both SIFT and SURF based feature detector to detect interest points from images. Kang et al. [63] and Zhu et al. [64] used Harris corner detector and SIFT detector in their feature-based mosaicing algorithm.

3.1.4. Contour-based mosaicing

This type of mosaicing algorithms is based on extraction of high-level features from images. Unlike the low-level features, these features are more natural to human perception and therefore they are high-level. High-level feature extraction mostly concerns finding the shapes or textures in an image. Shape extraction

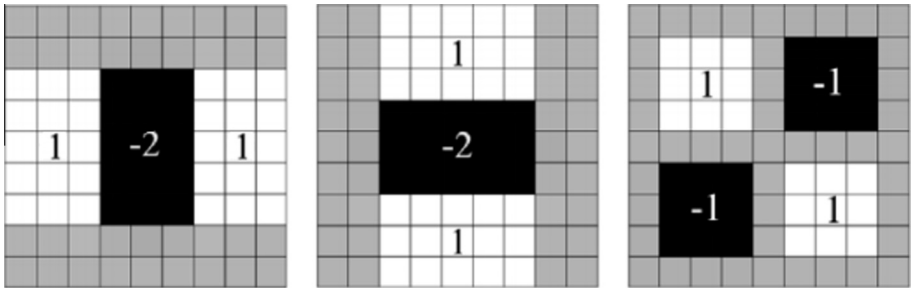

Fig. 6. Left to right: Approximation of Gaussian second order partial derivatives in *x*, *y*, and *xy* directions taken from [59].

implies finding their position, orientation, and size [49]. Usually regions of different structures are extracted as high level image features. Then these features are matched to find correspondences, which are later used to compute the transformation parameters. Different techniques are used to eliminate the false matches. Finally, warping and blending are performed to generate the mosaic output. The use of high-level features significantly increase the computation in these types of mosaicing algorithms. However, they are particularly suitable to work under larger and complicated motion parameters, and even under multi-layer registration.

Some of the notable contributions in high-level feature-based mosaicing include [65–67]. In [65], the authors used a wide baseline algorithm together with an adaptive region expansion method to achieve robust registration using high-level features. Prescott et al. [66] proposed extracting regions of image structures using a threshold technique and then computing area-based similarity matching for registration. Contour extraction using a segmentation algorithm, followed by finding their centroids for image registration was used in [67].

3.2. Frequency domain image mosaicing algorithms

Unlike spatial domain-based image mosaicing algorithms, methods classified in this category require computation in the frequency domain in order to find the optimal transformation parameters between a pair of images. These algorithms use the property of phase correlation for registering images. Let $I_1(x,y)$ and $I_2(x,y)$ are two images having some overlapping areas. Let's further assume that (x_0, y_0) is the translation between the images. Thus,

$$I_2(x,y) = I_1(x - x_0, y - y_0) \tag{14}$$

The corresponding Fourier transforms $F_1(u,v)$ and $F_2(u,v)$ are related by:

$$F_2(u,v) = F_1(u,v).e^{-j(ux_0+vy_0)} \tag{15}$$

The cross-power spectrum of the two images is defined as:

$$\frac{F_1^*(u,v)F_2(u,v)}{|F_1^*(u,v)F_2(u,v)|} = e^{-j(ux_0+vy_0)} \tag{16}$$

where $F_1^*(u,v)$ is the complex conjugate of $F_1(u,v)$. The shift theorem guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. (x_0, y_0) could be solved in two different ways. One way is to work directly in the frequency domain. However, this technique is sensitive to noise. A better approach is to take inverse Fourier transform of the above equation and get an impulse function $\delta(x - x_0, y - y_0)$, which is approximately zero everywhere except at the displacement (x_0, y_0) as shown in Fig. 7. With the displacement (translational) parameters the two images are warped and finally stitched to get a mosaic. Mosaicing algorithms based on this technique are usually efficient because of the use of shift property of Fourier transform and the use of Fast Fourier Transform (FFT). However, they suffer from being overly sensitive to noise. Additionally, accurate registration often requires significant overlapping between source images.

The above explained method of image mosaicing has sometimes, as in [34,68,69], been modified to make it suitable for handling transformations other than translation. A log-polar transformation is utilized in [34] to find the scale and translational parameters. A two-step method is proposed in [68]. The first step computes the rotation angle by finding the maximum peak while rotating the target image with an incremental angle. Using the computed rotation angle and phase correlation, the second step determines the translational displacement. In [69], the authors suggested changing the rotation and scale parameters to translational parameters using Fourier–Mellin transform.

The comparative overview of different categories of mosaicing algorithms based on image registration is presented in Table 1. Table 2 highlights the processing times of different mosaicking papers surveyed based on image registration.

4. Classification of image mosaicing based on blending

Similar to registration, image blending is a significant step for successful implementation of mosaicing. Stitching multiple images together to create a seamless mosaic requires the use of a suitable blending algorithm. Blending is often referred to as photometric registration, which is vital to equalize color and luminance appearance in a composite image. There are several reasons (difference in camera exposure, variation in scene illumination, presence of moving objects between frames, geometric misalignments, etc.) which may lead to inconsistencies in the final mosaic image. The visibility of such inconsistencies can be minimized by choosing an appropriate blending algorithm. This way, the final mosaic would be visibly free of annoying seams, giving it a consistent global appearance. The following two subsections discuss the classification of image mosaicing algorithms based on the image blending methods used by them.

4.1. Image mosaicing algorithms using transition smoothing-based blending

Mosaicing algorithms within this category attempt to minimize the visibility of seams by smoothing the common overlapping regions of the combined images. The information of the overlapping region between two images is fused in such a way that the boundaries of the images involved become imperceptible. Even though a totally indistinguishable transition may be achieved, the content and coherency of the overlapping region is not guaranteed, as the information is fused without taking into account the

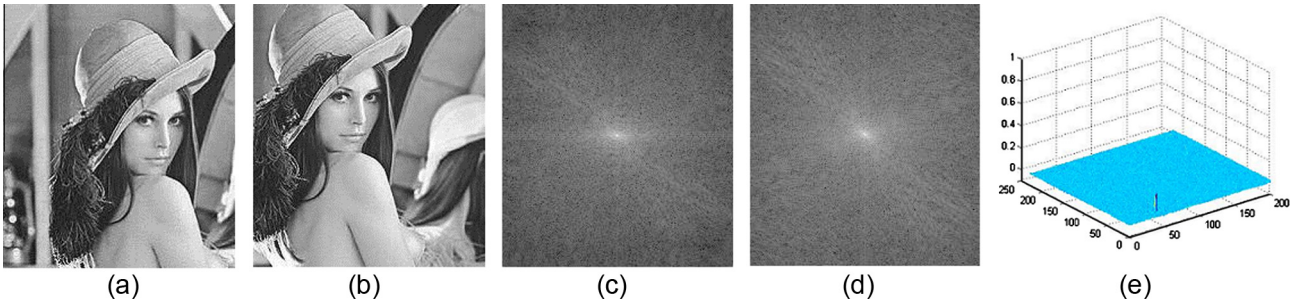

Fig. 7. Use of cross-power spectrum to detect transformation. (a) and (b) Source images with displacement between them; (c) and (d) spectrum of (a) and (b); and (e) impulse function indicating displacement between the source images, adapted from [34].

Table 1
Comparative overview of different categories of mosaicing methods based on image registration.

Category	Advantages	Disadvantages
NCC-based	No high level structural analysis required, and can be applied directly to image data	Flat similarity due to self-similarity of images, and good only for images with large overlapping
MI-based	Good for multimodal analysis and less sensitive to illumination and occlusion changes	Slow and causes registration error when images have small overlapping
Harris corner detector-based	Simple and accurate computation	Needs prior knowledge of window size and good only for moderate changes in scale and rotation
FAST corner detector-based	Accurate and fast computation	Not robust to high degree of noise, and prior knowledge about threshold required
SIFT feature detector-based	Efficient for high resolution images and offers invariance to various transformations	Computationally expensive
SURF detector-based	Fast computation, good for real-time applications	Poor performance under certain transformations (e.g. color, illumination)
Contour-based	Efficient when large and complicated motion involved	Computationally expensive because of the use of high-level features
Frequency domain-based	Efficient because of FFT	Overly sensitive to noise and accuracy relies on large overlapping

Table 2
Processing times for different mosaicing papers surveyed based on image registration.

Category	Methods	Processing time
NCC-based	[27,44–46]	No information available
MI-based	[29,47,48]	Processing time of 1.5 min for stitching a pair of images (using 3.2 GHz Pentium IV processor with 2 GB RAM) [29] Processing time of 1s for video mosaicing 40 160 $*$ 100 frames using 2.4 GHz Pentium processor [47]
Harris corner detector-based	[30,51,23]	Processing time of 2 ms for stitching a pair of 512 $*$ 512 images (using XC3S5000 FPGA board) [23]
FAST corner detector-based	[31,52,53]	Processing time of 20 ms for registering a pair of multimodal real images (using 2.5 GHz Pentium Dual-Core processor with 3 GB RAM) [52] Processing time of 437 ms for stitching a pair of 512 $*$ 512 images [53]
SIFT feature detector-based	[54,11,55,10,32,56]	Processing time of 2 s for stitching 10 100 $*$ 67 images using 2.4 GHz Pentium Core 2 Duo processor with 4 GB RAM [11] Processing time of 31 m for stitching 10 images using 3.3 GHz Pentium dual core processor with 6 GB RAM [10]
SURF detector-based	[58,50,59,60]	Processing time of 400 ms for feature detection and description (using 3 GHz Pentium IV processor) [58] Processing time of 23.6 s for stitching a set of 9 1280 $*$ 720 images [59] Processing time of 11.865 s for matching a pair of 1600 $*$ 1200 images [60]
Contour-based	[49,65–67]	Processing time of 3 s for registering a pair of 1024 $*$ 768 images (using 2.4 GHz Pentium Dual-Core processor with 4 GB RAM) [49]
Frequency domain-based	[34,68,69]	Processing time of 3.9 s for feature detection (using 3 GHz Pentium IV processor) [34]

content of the scene [70]. Thus, most often, these mosaicing methods generate mosaic with blurry transitions in the boundary regions. Popular methods which use transition smoothing for their blending operation include feathering, pyramid blending, and gradient-based blending. Mosaicing algorithms based on these techniques are discussed briefly as follows.

4.1.1. Mosaicing algorithms using feathering-based blending

Mosaicing algorithms within this category perform blending operation by taking an average value in each pixel of the overlapping region. However, the simple average method fails when exposure differences, misalignments, and presence of moving object are very obvious in the input images. A superior method is to use weighted averaging along with a distance map. Pixels near the center of an image are weighted heavily and those near the edges are weighted lightly. This is done by computing a distance map in terms of Euclidean distance of each valid pixel (mask) from its nearest invalid pixel as [20]:

$$w_k(x) = \left\| \arg\min_{y} \left\{ \|yy\| \middle| \widetilde{I}_k(x+y) \ is \ invalid \right\} \right\| \tag{17}$$

where $\widetilde{I}_k(x)$ are the warped images and $w_k(x)$ are the weights of the images. Finally, the mosaic image is generated as a weighted combination of the input images. Examples of composite images formed of six color images using simple average blending and feathering are shown in Fig. 8. Mosaicing algorithms which use the aforementioned technique perform reasonably well under exposure differences. However, it is difficult in practice to achieve a balance between smoothing out low-frequency exposure differences and preserving sharp enough transitions to prevent blurring. Furthermore, these methods suffer from ghosting artifacts.

Examples of mosaicing methods using feathering-based blending include [56,60,71]. [56,71] used alterations of the above mentioned method for finding the weights of images in the overlapping region. In [56], the aforementioned weight is measured by computing the distance of the overlapping pixels from the borders of the left and the right images. In [71], the authors used weighted average of the pixel color values in the overlapping region.

4.1.2. Mosaicing algorithms using pyramid-based blending

In an attempt to perform the blending operation in a more robust way, these mosaicing algorithms convert the input images into band-pass pyramids as shown in Fig. 9. Mask image associated with each source image is then created. Mask creation can be made automatic by using grassfire transform as used in [72]. Then the mask image is converted into a low-pass pyramid by using a Gaussian kernel [20]. The resultant blurred and subsampled masks are treated as weights to perform per-level feathering. The final mosaic is then achieved by interpolating and summing up the results from per-level feathering as:

$$LO(x,y) = GM(x,y) * LI_1(x,y) + (1 - GM(x,y)) * LI_2(x,y) \tag{18}$$

where $LI_1(x,y)$ and $LI_2(x,y)$ are the Laplacian pyramids of the warped source images $I_1(x,y)$ and $I_2(x,y)$. $GM(x,y)$ is the Gaussian pyramid of the mask image $M(x,y)$ and $LO(x,y)$ is the Laplacian pyramid of the output image $O(x,y)$.

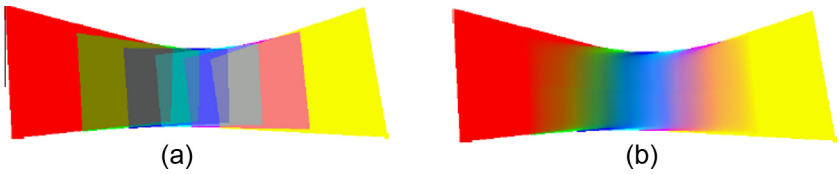

Fig. 8. (a) Blended image using simple averaging and (b) blended image using feathering, adapted from [1].

Fig. 9. (a) Low-pass pyramid and (b) band-pass pyramid, adapted from [73].

Sometimes, all the strips are combined in a single blending step when pyramids for multiple narrow strips are required, as proposed in [35]. Algorithms using the above method achieve reasonable balance between smoothing out low frequency components and preserving sharp enough transitions to prevent blurring [74]. Edge duplication is also eliminated noticeably using these methods. However, double contouring and ghosting effects become significant when the registration error is significant.

4.1.3. Mosaicing algorithms using gradient-based blending

Another group of transition smoothening method is based on gradient domain blending. These methods are based on the idea that by suitably mixing the gradient of images, it is possible to mosaic image regions convincingly. In general, the gradients across seams are set to zero for smoothing out the color differences. Since humans are more sensitive to gradients than image intensities, mosaicing methods using this technique generate visually more pleasant results compared to the other two techniques discussed before. However, working exclusively in the gradient domain requires higher computational resources to deal with large data sets. Furthermore, for best performance, the alignment of images through registration needs to be almost perfect.

Notable work in this category was performed by Levin et al. [19], Xiong [75], and Szeliski et al. [76]. In [19], the authors developed two approaches called GIST (gradient domain image stitching). One of the approaches is based on minimizing a cost function that evaluates the dissimilarity measure between the derivatives of the mosaic and the derivatives of the source images. The other approach is based on inferring a mosaic by optimization over image gradients. In [75], the authors used a gradient domain object moving and region filling algorithm to eliminate the visible artifacts arising from moving objects in the scene. Algorithm based on assigning low resolution offset map to each source image followed my Poisson's blending was proposed by Szeliski et al. [76].

Table 3
Comparative overview of different categories of mosaicing methods based on image blending.

Category	Advantages	Disadvantages
Feathering-based	Fast and good performer under exposure differences	Output often suffer from blur and ghosting effect
Pyramid-based	Good in preventing blur and edge duplication	Suffers from double contouring and ghosting when registration error significant
Gradient-based	Output visually more appealing than other methods	High computation required and registration error must be small for good performance
Optimal seam-based	Good in dealing with moving objects and parallax	Transition obvious when there are exposure differences

Table 4
Processing times for different mosaicing papers surveyed based on image blending.

Category	Methods	Processing time
Feathering-based	[56,71,60]	No information available
Pyramid-based	[72,35,74]	No information available
Gradient-based	[19,75,76]	Processing time of 2 m for stitching a pair of $200 * 300$ images (using Pentium processor) [19] Processing time of 16 s for stitching 9 frames (using 2.4 GHz Intel Core 2 Duo processor with 4 GB RAM) [76]
Optimal seam-based	[36,77,78]	Processing time of 129 ms for stitching 900 images [36] Processing time of 2 s for stitching a pair of images [77]

4.2. Image mosaicing algorithms using optimal seam-based blending

This type of mosaicing algorithms attempt to minimize the visibility of seams by looking for optimal seams in the joining boundaries between the images. The objective of optimal seam technique is to allocate the optimal location of a seam line by looking into the overlapping region between a pair of images. The seam line placement should be such that it minimizes the photometric differences between the two sides of the line. At the same time the seam line should be able to determine the contribution of each of the images in the final mosaic. Once the placement and the contribution information are obtained, each image is copied to the corresponding side of the seam. When the difference between the two images on the seam line is zero, no seam gradients are produced in the mosaic. Unlike the mosaicing methods using transition smoothing-based blending, optimal seam-based mosaicing algorithms consider the information content of the scene in the

overlapping region, allowing to deal with problems like moving objects or parallax. However, no information is fused in the overlapping region, thus the transition between the images can be easily noticeable when there are global intensity or exposure difference between the frames.

Different optimal seam finding methods have been used in mosaicing literature. For example, in [36] a modified region-of-difference method is used. In [77], authors proposed the use of an algorithm based on watershed segmentation and graph cut optimization. Another method based on dynamic programming and grey relational analysis is used in [78].

A general comparison of different categories of mosaicing algorithms based on image blending is presented in Table 3. Table 4 highlights the processing times of different mosaicing papers surveyed based on image blending.

5. Conclusion

Image mosaicing is an important task in the field of computer vision. Success of a mosaicing algorithm depends primarily on registration and blending methods. This paper provides a classification of image mosaicing methods based on image registration and blending algorithms. In addition to providing the description of the different categories, this paper discusses the advantages and disadvantages of each category. From the discussion, it is obvious that there is no single best image mosaicing category. At the same time, the continuous advent of new mosaicing methods in recent years makes it really difficult to choose an appropriate mosaicing algorithm for a specific purpose. Hence, this paper aims at providing a guide for selecting a suitable mosaicing method for a specific application. Although an extensive research has been done in the area of mosaicing, there are still problems to be addressed. For example, mosaicing in the presence of images with significant parallax is still a challenge. Another problem is the processing time. All mosaicing techniques are time consuming and cannot run on low power and low frequency devices. Addressing these issues would be where the future research might be directed at.

References

[1] D.P. Capel, Image Mosaicing and Superresolution, Springer Science & Business Media, 2004.
[2] D. Ghosh, S. Park, N. Kaabouch, W. Semke, Quantitative evaluation of image mosaicing in multiple scene categories, in: IEEE International Conference on Electro/Information Technology (EIT), 2012, pp. 1–6.
[3] P. Azzari, L. Di Stefano, S. Mattoccia, An evaluation methodology for image mosaicing algorithms, Adv. Concepts Intell. Vis. Syst. 5259 (2008) 89–100.
[4] A. Bevilacqua, P. Azzari, A fast and reliable image mosaicing technique with application to wide area motion detection, Image Anal. Recogn. 4633 (2007) 501–512.
[5] K.S. Bhat, M. Saptharishi, P.K. Khosla, Motion detection and segmentation using image mosaics, in: IEEE International Conference on Multimedia and Expo, pp. 1577–1580, 2000.
[6] S.T. Sreyas, J. Kumar, S. Pandey, Real time mosaicing and change detection system, in: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, 2012, pp. 53.
[7] M. Vivet, B. Martínez, X. Binefa, Real-time motion detection for a mobile observer using multiple kernel tracking and belief propagation, in: Pattern Recognition and Image Analysis, 2009, pp. 144–151.
[8] A.E.S. Lucas, C. Christo, M.P. Silva, C. Cardeira, Mosaic based flexible navigation for AGVs, in: IEEE International Symposium on Industrial Electronics (ISIE), 2009, pp. 3545–3550.
[9] T. Suzuki, Y. Amano, T. Hashizume, Vision based localization of a small UAV for generating a large mosaic image, in: Proceedings of SICE Annual Conference, 2010, pp. 2960–2964.
[10] A. Nemra, N. Aouf, Robust invariant automatic image mosaicing and super resolution for UAV mapping, in: International Symposium on Mechatronics and its Applications, 2009, pp. 1–7.
[11] D. Ghosh, N. Kaabouch, R.A. Fevig, Robust spatial-domain based super-resolution mosaicing of cubesat video frames: algorithm and evaluation, Comput. Inform. Sci. 7 (2014) 68.
[12] D. Ghosh, N. Kaabouch, W. Semke, Super-resolution mosaicing of unmanned aircraft system (UAS) surveillance video frames, Int. J. Sci. Eng. Res. 4 (2013).
[13] P. Azzari, L. Di Stefano, F. Tombari, S. Mattoccia, Markerless augmented reality using image mosaics, Image Signal Process. 5099 (2008) 413–420.
[14] J. Seokhee, G.J. Kim, Mosaicing a wide geometric field of view for effective interaction in augmented reality, in: IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, pp. 265–266.
[15] M. Irani, S. Hsu, P. Anandan, Video compression using mosaic representations, Signal Process.: Image Commun. 7 (1995) 529–552.
[16] C. Yun-Hee, S. Yeong Kyeong, C. Tae-Sun, Image mosaicing with automatic scene segmentation for video indexing, in: International Conference on Consumer Electronics, 2002, pp. 74–75.
[17] M. Ramachandran, R. Chellappa, Stabilization and mosaicing of airborne videos, in: IEEE International Conference on Image Processing, 2006, pp. 345–348.
[18] S. Ghannam, A.L. Abbott, Cross correlation versus mutual information for image mosaicing, Int. J. Adv. Comput. Sci. Appl. (IJACSA) 4 (2013).
[19] A. Levin, A. Zomet, S. Peleg, Y. Weiss, Seamless image stitching in the gradient domain, Comput. Vis.-ECCV (2004) 377–389.
[20] R. Szeliski, Image alignment and stitching: a tutorial, Found. Trends Comput. Graph. Vis. 2 (2006) 1–104.
[21] A. Behrens, M. Guski, T. Stehle, S. Gross, T. Aach, Intensity based multi-scale blending for panoramic images in fluorescence endoscopy, in: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2010, pp. 1305–1308.
[22] G. Guandong, J. Kebin, A new image mosaics algorithm based on feature points matching, in: International Conference on Innovative Computing, Information and Control, 2007, pp. 471–471.
[23] K.-I. Okumura, S. Raut, Q. Gu, T. Aoyama, T. Takaki, I. Ishii, Real-time feature-based video mosaicing at 500 fps, in: International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 2665–2670.
[24] S. Park, D. Ghosh, N. Kaabouch, R.A. Fevig, W. Semke, Hierarchical multi-level image mosaicing for autonomous navigation of UAV, in: IS&T/SPIE Electronic Imaging, 2012, pp. 830116–830116-9.
[25] L. Miao, Y. Yue, Automatic document image mosaicing algorithm with hand-held camera, in: International Conference on Intelligent Control and Information Processing (ICICIP), 2011, pp. 1094–1097.
[26] H. Shejiao, H. Yaling, C. Zonghai, J. Ping, Feature-based image automatic mosaicing algorithm, in: in: World Congress on Intelligent Control and Automation, 2006, pp. 10361–10364.
[27] A. Nasibov, H. Nasibov, F. Hacizade, Seamless image stitching algorithm using radiometric lens calibration for high resolution optical microscopy, in: International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009, pp. 1–4.
[28] T. Vercauteren, A. Perchant, X. Pennec, N. Ayache, Mosaicing of confocal microscopic in vivo soft tissue video sequences, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI, 2005, pp. 753–760.
[29] R. Miranda-Luna, C. Daul, W.C. Blondel, Y. Hernandez-Mier, D. Wolf, F. Guillemin, Mosaicing of bladder endoscopic image sequences: distortion calibration and registration algorithm, IEEE Trans. Biomed. Eng. 55 (2008) 541–553.
[30] G. Gao, K. Jia, A new image mosaics algorithm based on feature points matching, in: International Conference on Innovative Computing, Information and Control, 2007, pp. 471–471.
[31] T. Botterill, S. Mills, R. Green, Real-time aerial image mosaicing, in: International Conference of Image and Vision Computing New Zealand (IVCNZ), 2010, pp. 1–8.
[32] L. Yao, Image mosaic based on SIFT and deformation propagation, in: IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, 2008, pp. 848–851.
[33] G. Jun-Hui, Z. Jun-Hua, A. Zhen-Zhou, Z. Wei-Wei, L. Hui-Min, An approach for X-ray image mosaicing based on Speeded-up robust features, in: International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012, pp. 432–435.
[34] F. Yang, L. Wei, Z. Zhang, H. Tang, Image mosaic based on phase correlation and Harris operator, J. Comput. Inform. Syst. 8 (2012) 2647–2655.
[35] M. Vivet, S. Peleg, X. Binefa, Real-time stereo mosaicing using feature tracking, in: IEEE International Symposium on Multimedia (ISM), 2011, pp. 577–582.
[36] M. El-Saban, M. Izz, A. Kaheel, M. Refaat, Improved optimal seam selection blending for fast video stitching of videos captured from freely moving devices, in: IEEE International Conference on Image Processing (ICIP), 2011, pp. 1481–1484.
[37] S.Z. Kovalsky, G. Cohen, J.M. Francos, Registration of joint geometric and radiometric image deformations in the presence of noise, in: IEEE/SP Workshop on Statistical Signal Processing, 2007, pp. 561–565.
[38] D. Vaghela, K. Naina, A review of image mosaicing techniques, Int. J. Adv. Res. Comput. Sci. Manage. Stud. 2 (3) (2014).
[39] P. Jain, V.K. Shandliya, A review paper on various approaches for image mosaicing, Int. J. Comput. Eng. Res. 3 (4) (2013).
[40] R. Abraham, P. Simon, Review on mosaicing techniques in image processing, in: International Conference on Advanced Computing and Communication Technologies (ACCT), 2013, pp. 63–68.
[41] H. Joshi, M.K. Sinha, A survey on image mosaicing techniques, in: International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2(2), 2013.
[42] M.H.M. Patel, A.P.P.J. Patel, A.P.M.S.G. Patel, Comprehensive study and review of image mosaicing methods, Int. J. Eng. Res. Technol. (2012).

[43] J.M. Fitzpatrick, D.L. Hill, C.R. Maurer Jr., Image registration, Handbook Med. Imag. 2 (2000) 447–513.
[44] K. Berberidis, I. Karybali, A new efficient cross-correlation based image registration technique with improved performance, in: Proceedings of the European Signal Processing Conference, 2002, pp. 3–6.
[45] F. Zhao, Q. Huang, W. Gao, Image matching by normalized cross-correlation, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2006, pp. II–II.
[46] T. Vercauteren, A. Meining, F. Lacombe, A. Perchant, Real time autonomous video image registration for endomicroscopy: fighting the compromises, in: Biomedical Optics (BiOS), 2008, pp. 68610C–68610C-8.
[47] A. Dame, E. Marchand, Video mosaicing using a mutual information-based motion estimation process, in: IEEE International Conference on Image Processing (ICIP), 2011, pp. 1493–1496.
[48] C. de Cesare, M.-J. Rendas, A.-G. Allais, M. Perrier, Low overlap image registration based on both entropy and mutual information measures, in: OCEANS, 2008, pp. 1–9.
[49] M.B. Islam, M.M.J. Kabir, A new feature-based image registration algorithm, Comput. Technol. Appl. 4 (2013) 79–84.
[50] V.S. Bind, Robust Techniques for Feature-Based Image Mosaicing, National Institute of Technology Rourkela, 2013.
[51] E. Zagrouba, W. Barhoumi, S. Amri, An efficient image-mosaicing method based on multifeature matching, Mach. Vis. Appl. 20 (2009) 139–162.
[52] J. Jiao, B. Zhao, S. Wu, A speed-up and robust image registration algorithm based on fast, in: IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2011, pp. 160–164.
[53] X. Wang, J. Sun, H.-Y. Peng, Efficient panorama mosaicing based on enhanced-FAST and graph cuts, in: Recent Advances in Computer Science and Information Engineering, vol. 128, 2012, pp. 757–762.
[54] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[55] D. Liqian, J. Yuehui, Moon landform images fusion and Mosaic based on SIFT method, in: International Conference on Computer and Information Application (ICCIA), 2010, pp. 29–32.
[56] Y. Li, Y. Wang, W. Huang, Z. Zhang, Automatic image stitching using sift, in: International Conference on Audio, Language and Image Processing, 2008, pp. 568–571.
[57] Y. Lei, W. Xiaoyu, Z. Jun, L. Hui, A research of feature-based image mosaic algorithm, in: International Congress on Image and Signal Processing (CISP), 2011, pp. 846–849.
[58] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: Computer vision–ECCV, 2006, pp. 404–417.
[59] N. Geng, D. He, Y. Song, Camera image mosaicing based on an optimized SURF algorithm, Indones. J. Electr. Eng. 10 (2012) 2183–2193.
[60] R. Wen, C. Hui, L. Jiaju, X. Yanyan, R. Haeusler, Mosaicing of microscope images based on SURF, in: International Conference on Image and Vision Computing New Zealand, 2009, pp. 271–275.
[61] H. Joshi, K. Sinha, Novel techniques image mosaicing based on image fusion using harris aand SURF, in: International Conference on Computer Science and Information Technology, 2013.
[62] V.S. Bind, P.R. Muduli, U.C. Pati, A robust technique for feature-based image mosaicing using image fusion, Int. J. Adv. Comput. Res. 3 (2013) 263.
[63] K. Peng, M. Hongbing, An automatic airborne image mosaicing method based on the SIFT feature matching, in: International Conference on Multimedia Technology (ICMT), 2011, pp. 155–159.
[64] J. Zhu, M. Ren, Image mosaic method based on SIFT features of line segment, Comput. Math. Methods Med. 2014 (2014).
[65] J. Xiao, Y. Zhang, M. Shah, Adaptive region-based video registration, in: IEEE Workshops on Application of Computer Vision, 2005, pp. 215-220.
[66] J. Prescott, M. Clary, G. Wiet, T. Pan, K. Huang, Automatic registration of large set of microscopic images using high-level features, in: IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006, pp. 1284–1287.
[67] M. Deshmukh, U. Bhosle, A survey of image registration, Int. J. Image Process. (IJIP) 5 (2011) 245.
[68] H. Xie, N. Hicks, G.R. Keller, H. Huang, V. Kreinovich, An IDL/ENVI implementation of the FFT-based algorithm for automatic image registration, Comput. Geosci. 29 (8) (2003) 1045–1055.
[69] C. Wang, Y. Cheng, C. Zhao, Robust subpixel registration for image mosaicing, in: Chinese Conference on Pattern Recognition, 2009, pp. 1–5.
[70] R. Prados Gutiérrez, Image Blending Techniques and their Application in Underwater Mosaicing, Springer, 2014.
[71] D.K. Jain, G. Saxena, V.K. Singh, Image mosaicing using corner techniques, in: International Conference on Communication Systems and Network Technologies (CSNT), 2012, pp. 79–84.
[72] Y. Xiong, K. Turkowski, Registration, calibration and blending in creating high quality panoramas, in: IEEE Workshop on Applications of Computer Vision Proceedings, 1998, pp. 69–74.
[73] P. Liang, X. Zhiwei, D. Jiguang, Joint edge detector based on Laplacian pyramid, in: International Congress on Image and Signal Processing (CISP), 2010, pp. 978–982.
[74] A. Pandey, U.C. Pati, A novel technique for non-overlapping image mosaicing based on pyramid method, in: IEEE India Conference (INDICON), 2013, pp. 1–6.
[75] Y. Xiong, Eliminating ghosting artifacts for panoramic images, in: IEEE International Symposium on Multimedia, 2009, pp. 432–437.
[76] R. Szeliski, M. Uyttendaele, D. Steedly, Fast Poisson blending using multi-splines, in: IEEE International Conference on Computational Photography (ICCP), 2011, pp. 1–8.
[77] N. Gracias, M. Mahoor, S. Negahdaripour, A. Gleason, Fast image blending using watersheds and graph cuts, Image Vis. Comput. (2009) 597–607.
[78] H. Wen, J. Zhou, An improved algorithm for image mosaic, in: International Symposium on Information Science and Engineering, 2008, pp. 497–500.