# A Novel Video Stitching Method for Multi-Camera Surveillance Systems

**Xiaoqing Yin[1], Weili Li[1], Bin Wang[2], Yu Liu[1], Maojun Zhang[1]**
[1] College of Information System and Management, National University of Defense Technology
Changsha, Hunan, 410073 – China
[2] Facility Design and Instrumentation Institute, China Aerodynamics Research and Development Center
Mianyang, Sichuan, 621000 - China
[E-mail: yinxiaoqing89@gmail.com]
*Corresponding author: Xiaoqing Yin

*Received April 21, 2014; revised July 11, 2014; accepted July 21, 2014; published October 31, 2014*

## Abstract

This paper proposes a novel video stitching method that improves real-time performance and visual quality of a multi-camera video surveillance system. A two-stage seam searching algorithm based on enhanced dynamic programming is proposed. It can obtain satisfactory result and achieve better real-time performance than traditional seam-searching methods. The experiments show that the computing time is reduced by 66.4% using the proposed algorithm compared with enhanced dynamic programming, while the seam-searching accuracy is maintained. A real-time local update scheme reduces the deformation effect caused by moving objects passing through the seam, and a seam-based local color transfer model is constructed and applied to achieve smooth transition in the overlapped area, and overcome the traditional pixel blending methods. The effectiveness of the proposed method is proved in the experiements.

***Keywords:*** Video stitching, two-stage seam searching, local update of seam, seam-based local color transfer model, smooth transition

**http://dx.doi.org/10.3837/tiis.2014.10.015**

## 1. Introduction

Video stitching provides high-resolution panoramic video of large field-of-view and has been widely used in multi-camera surveillance systems to monitor public places in wide areas [1-2]. However, in spite of satisfactory stitching results, most video stitching methods entail difficulty in meeting the increasing demand for real-time performance and visual quality [3-7]. Video stitching is performed in three steps: registration, seam-searching and blending.

In existing multi-camera surveillance systems, the locations of cameras remain stable and fixed. Thus, matching and registration of images from different cameras can be implemented only at the beginning and then keep changeless for the following frames. Meanwhile, real-time update is necessary for stitching seams according to the change in video content.

To obtain a satisfied result in video stitching with a natural transition from one frame to another, it is needed to decide which pixels to use and how to blend them in the overlapping area. Compared with simple techniques such as feathering and center-weighting, finding optimal seams in the overlapping area achieves better results. Although a variety of seam-searching algorithms have been applied in image stitching and obtain satisfactory visual quality, most of them fail to meet real-time performance, which limits their application in video stitching. In this paper we make improvements on an existing seam-searching algorithm to accelerate the computing speed while maintaining the accuracy in the meantime.

When moving objects pass through the seam, the sections on both sides of the seam are combined in the fusion strategy. The shape of the merged moving object may be deformed owing to the inevitable registration and calibration errors, as shown in the red box in **Fig. 1(b)**. As perceivable stitching quality is more sensitive to moving objects than to static objects, the deformation of moving objects results in the artifacts and impairs the subsequent object detection, recognition and tracking. To cope with this problem, a real-time local update strategy is associated with seam-searching process in our method and prevents the seams from passing through moving objects.

(a)Seam passes through the moving object (b) Deformation of the moving object

**Fig. 1.** Deformation of moving objects passed through by the stitching seam

As the images to be stitched are taken with different cameras, lighting condition and intensity differences may influence the consistency of the entire scene. Thus, color correction and smooth transition should be implemented in video stitching, but traditional methods are complex and make satisfactory results difficult to achieve. Applying color correction model is an effective technique, but conventional global color corection often fails to consider the local feature of color diffrence. Instead of taking a global color correction model, we introduce a

local model to make color correction within local regions obtained by image segmentation. The local color correction model makes better use of the information of the stitching seam and achieves smooth transition in the overlapping area.

The rest of the paper is organized as follows. In Section 2, we review previous research on video stitching, including seam searching, dealing with moving objects and color correction. In Section 3, we describe our video stitching method. A two-stage seam searching method based on enhanced dynamic programming algorithm is proposed, followed by a local update scheme of stitching seam to cope with moving objects. To improve the quality of smooth transition, a seam-based local color transfer model is constructed. In Section 4, we present the experimental results of video stitching using the proposed method and compare this method with conventional ones. In Section 5, we summarize the main conclusions and discuss directions for future research.

## 2. Related Work

In the overlapped areas of stitched images, it is required to obtain optimal seams which can decide how to choose and blend pixels from different images. Some conventional seam-searching algorithms, such as Graph Cuts [5], Dijkstra algorithm [6], traditional dynamic programming [8] have been used in static image stitching and achieved satisfactory stitching result. However, most of these algorithms often fail to obtain satisfactory performance in real-time video stitching, especially for huge amounts of video data, because of their computational complexity and large memory requirements.

Among these algorithms, enhanced dynamic programming [7] is less time-consuming because of low computing complexity and it provides more flexible extending directions for seam-searching than dynamic programming. Therefore, enhanced dynamic programming is an effective seam-searching algorithm. Although the conventional enhanced dynamic programming algorithm finds the global optimal seam, its real-time performance is still limited and it requires improvement when being used in video stitching.

When motion occurs in the overlapping area, the deformation of the merged moving object would reduce the accuracy of object detection, recognition and tracking. Image and video stitching with moving objects has been studied by several researchers [2, 9, 10]. Uyttendaele et al. [9] proposed a region of difference based method to deal with moving objects in the overlapped area. Mills presented a new technique to create consistent image mosaics in spite of the presence of moving objects [10]. In the seam-searching process, it is necessary to deal with moving objects separately to keep the consistency of moving parts.

Due to color and intensity differences of multiple cameras, incoherence may occur in the stitched image. Conventional methods including linear color transfer model [11, 12] is proposed to adjust the gain level of the images captured by different cameras. These methods use invariable color transfer models to deal with color difference in the overlapped area. Although the models are simple and obtain good performance, they fail to achieve smooth transition and generate a visible seam. Some conventional image blending techniques [13, 14] cause blur when blending pixels from different images. Instead of taking a global color correction model which often fails to consider the local feature of color difference, color correction can be made in local regions with similar color features. Image segmentation which has been studied by many researchers [15, 16] provides a new perspective for local color correction.

Based on the previous research about image and video stitching, we propose a novel real-time panoramic video stitching method for multi-camera surveillance system. The

proposed two-stage seam searching method based on enhanced dynamic programming can obtain satisfactory seam-searching result while achieving better real-time performance. In addition, a local update strategy is proposed for seam update, which can prevent the moving objects from passing through the seam, thereby maintaining the shape of the objects and avoiding the resulting deformation. A seam-based local color transfer model is constructed to achieve smooth transition in the overlapped area.

## 3. Video Stitching Method for Multi-Camera Surveillance Systems

### 3.1 Implementation of registration

Considering that the relative locations of cameras remain stable for most of the multi-camera surveillance systems, the alignment, registration and matching relations of different images are fixed. To improve the real-time performance of coordinate transformation, the one-to-one correspondence between coordinates can be stored in a pre-calculated look-up table (LUT).

For the pixel in the stitched image, the coordinates of corresponding source pixels are stored in the LUT. It is needed to look up the corresponding coordinates in the LUT. Registration and alignment of images can be implemented through simple look-up operations instead of complex conventional algorithms.

Every pixel $(xc, yc)$ in the stitched image corresponds to a source pixel $(x_0^i, y_0^i)$ in the source image $S_i$, and $x_0^i, y_0^i$ can be expressed as: $x_0^i = I(x_0^i) + D(x_0^i)$, $y_0^i = I(y_0^i) + D(y_0^i)$, where $I(\cdot)$ and $D(\cdot)$ are the integral and the fractional part, respectively. The pixel value at $(xc, yc)$ can be obtained:

$$P(xc, yc) = S_i(x_0^i, y_0^i) = S_1 \cdot (1 - D(y_0^i)) + S_2 \cdot D(y_0^i) \tag{1}$$

where

$$\begin{cases} S_1 = S(I(x_0^i), I(y_0^i)) \cdot (1 - D(x_0^i)) + S(I(x_0^i) + 1, I(y_0^i)) \cdot D(x_0^i) \\ S_2 = S(I(x_0^i), I(y_0^i) + 1) \cdot (1 - D(x_0^i)) + S(I(x_0^i) + 1, I(y_0^i) + 1) \cdot D(x_0^i) \end{cases} \tag{2}$$

Static overlapped images are taken by the cameras, and alignment, registration and matching of these images can be achieved using SIFT and RANSAC algorithms [12]. The images are projected and transformed to form the stitched image. Thus the look-up table can be constructed and then applied to video stitching. Given the fixed and immovable layout of the cameras, the same LUT is used for different scenes when this platform is moved to another location.

The implementation of the LUT is shown in **Fig. 2**.

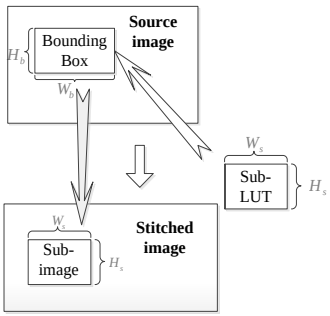



**Fig. 2.** Look-up table (LUT) implementation.

The entire LUT can be divided into a series of sub-tables (sub-LUT) that can be handled in parallel to improve the processing efficiency. The size of each sub-table is $W_s \times H_s$. A bounding box covering the corresponding part of the image can be obtained and loaded into memory.

### 3.2 Two-stage seam searching method based on enhanced dynamic programming

Although the traditional enhanced dynamic programming algorithm can obtain the global optimal seam, its computational complexity degrades the real-time performance. In this section, we introduce a two-stage seam searching method based on enhanced dynamic programming algorithm. This algorithm not only takes advantage of the traditional enhanced dynamic programming, but also improves the real-time performance. The proposed algorithm is described as follows:

Stage 1: Down-sample the original image with the sampling interval of $r$. Perform enhanced dynamic programming algorithm on the down-sampled image and obtain the optimal seam, which is then interpolated to the original resolution, as shown in **Fig. 3(a)**. The interpolated seam, which is the approximation of the final optimized seam, can be used as the initial value of stage 2;

Stage 2: Perform enhanced dynamic programming algorithm in the adjacent area along the initial seam with the width of $K$, and obtain the final optimized seam, as shown in **Fig. 3(b)**.

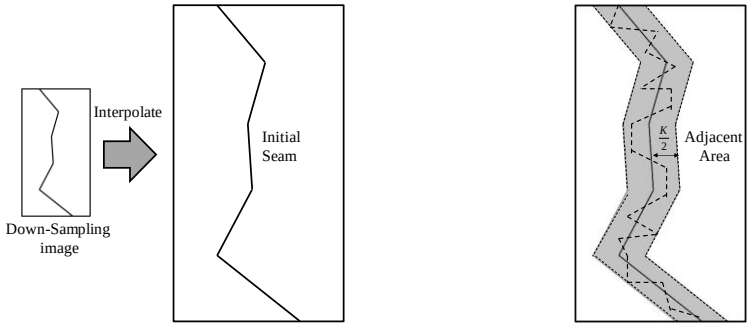



**Fig. 3.** Two-stage seam searching method based on enhanced dynamic programming algorithm.

The seam-searching process in the two stages can be implemented as follows:

(1)For each row of the overlapped area ( from the bottom to the top ), calculate the minimal accumulated cost $f(i, j)$ for every pixel $(i, j)$ as $f(i, j) = \min(f_L(i_l, j), f_R(i_r, j))$, where $f_L(i_l, j)$ is the minimal accumulated cost calculated from the left border to the right border and $f_R(i_r, j)$ is that calculated from the opposite direction, which can be obtained as follows:

$$\begin{cases} f_L(i_l, j) = \min(d(P_{i_l}) + f(P_{i_l})), i_l = 1, 2, 3, 4 \\ f_R(i_r, j) = \min(d(P_{i_r}) + f(P_{i_r})), i_r = 2, 3, 4, 5 \end{cases} \tag{3}$$

where $P_k (k = 1 \cdots 5)$ denotes the following points:

$$P_1 : (i, j-1), P_2 : (i+1, j-1), P_3 : (i+1, j), P_4 : (i+1, j+1), P_5 : (i, j+1).$$

The seam-searching process discussed above is illustrated in **Fig. 4.**

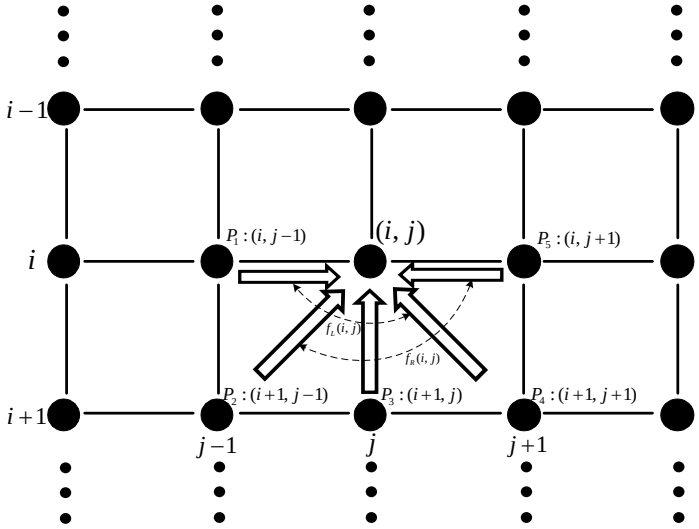



**Fig. 4.** Illustration of seam-searching process.

(2) Select the point $(1, j^*)$ that satisfies $f(1, j^*) = \min_j(f(1, j))$ as the starting point of the seam, and obtain the optimal seam according to the values of minimal accumulated cost.

In the seam-searching process, the definition of the cost function is similar to the definition of cost value [7].

$$e = \alpha S_g + \beta S_c \tag{4}$$

where $S_g$ and $S_c$ measure the gradient smoothness and the color vector similarity of corresponding pixels in the overlapped region, respectively, and $\alpha, \beta$ are the weights used to balance the relative influence of the two factors. Thus the cost value in the overlapped area can be constructed as an image with pixel values indicating the gradient smoothness and the color vector similarity for each point.

However, the determination of the weights $\alpha, \beta$ as well as $S_g, S_c$ remains a problem. The values of the weights are important to the process of seam searching and improper selection of weights may cause imbalance of the two factors and degrade the seam-searching result and image stitching quality. To solve this problem, we use the following technique to determine the weights.

$S_g^*$ and $S_c^*$ can be defined as:

$$\begin{cases} S_g^* = \dfrac{\| \nabla(I_1 + I_2) \|}{\max(\| \nabla(I_1 + I_2) \|)} \\ S_c^* = \dfrac{\| I_1 - I_2 \|}{\max(\| I_1 - I_2 \|)} \end{cases} \tag{5}$$

where $I_1, I_2$ are the overlapped images, and $\| \nabla \cdot \|$ represents the norm of the gradient for each pixel. $\max(\| \nabla(I_1 + I_2) \|)$ and $\max(\| I_1 - I_2 \|)$ denote the maximum value of $\nabla(I_1 + I_2)$ and $\| I_1 - I_2 \|$, respectively.

If the two factors are considered to be of the same importance when calculating the value of cost function, then $\alpha = \beta = 0.5$ .Thus the relative influence of the two factors can be balanced. The cost value of the three testing overlapped areas in our experiments is shown in **Fig. 7**. The corresponding source images $I_1$, $I_2$ and $I_3$ are shown in **Fig. 8**.

### 3.3 Local update of stitching seam

Moving objects contained in the overlap area cause partial change in only part of the scene instead of the entire scene. For most parts, the video contents remain static and changing the segments of the seam passing through static area is unnecessary.

We divide the existing image into sections with the same height and each section contains a segment of the seam, as shown in **Fig. 11**. Each segment can be handled independently. When the moving objects appear in the overlapped region but are far from the seam, dealing with them immediately is not necessary. The adjacent area $R_a$ is defined as the pixels located along the seam with a horizontal distance less than $\Delta d$ :

$$R_a = \{P \,|\, \| X(P) - X(S_P) \| \leq \Delta d\} \tag{6}$$

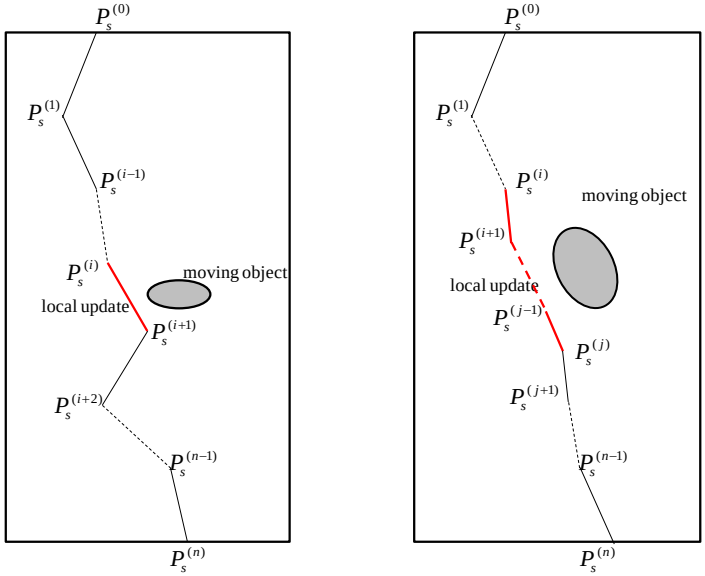
where $S_P$ is the seam point which is of the seam row as pixel $P$ . Moving objects approaching the seam result in difference in the adjacent area. Thus we only need to perform object detection in the adjacent region instead of the entire overlapped region, and make local modification to the seam if some moving objects pass through it. The code-book scheme [9] is applied in detecting moving objects in the adjacent region.

For the stitching seam $P_s = (P_s^{(1)}, P_s^{(2)}, \cdots, P_s^{(n)})$ , the strategy of local update can be listed as follows:

1. If the moving object is detected near the seam segment $(P_s^{(i)}, P_s^{(i+1)})$ and no moving objects are found for the adjacent segment $(P_s^{(i-1)}, P_s^{(i)})$ and $(P_s^{(i+1)}, P_s^{(i+2)})$ , update the seam segment $(P_s^{(i)}, P_s^{(i+1)})$ using the two-stage seam searching method.

2. If moving objects are detected near a series of seam segments $(P_s^{(i)}, P_s^{(i+1)}, \cdots, P_s^{(j)})$ , update the seam part $(P_s^{(i)}, P_s^{(j)})$ using the two-stage seam searching method.

The local update strategy 1 and 2 are illustrated in Fig. 5(a) and (b), respectively. The seam segments in red are the parts to be updated. The result for local update of stitching seam is shown in **Fig. 12**.




(a)Strategy 1 (b)Stategy 2

**Fig. 5.** Local update strategy for stitching seam

### 3.4 Seam-based local color transfer model

Although the color difference between neighbour images is effectively decreased, stitching seam still exist in the overlapped area. Considering that perceiving visual quality turns to be more sensitive to the the seams in the transition areas, a smooth scheme is required to eliminate the visible seams.

Instead of taking a global color correction model that often fails to consider the local feature of color diffrence, we introduce a local model to perform color correction within local regions. Segmentation for the overlapped areas is implemented by mean shift algorithm [14], as shown in **Fig. 15(a)**.

In our two-stage seam-searching approach, the optimal seam is obtained (**Fig. 15(b)**). Within the local regions which is not passed through by the seam, the color remains continuous and coherent, so adjusting the color in these regions is not necessary. We construct a seam-based local color transfer model and apply this method to the regions passed through by the seam. Instead of blending the pixels in two images, we use a linear model for color transfer on the pixels within each local region, which is able to adjust the coefficients according to the horizontal distance to the seam.

For each local region passed through by the seam in the overlapped area, we consider the pixels on the seam for the two neighbor images. For overlapped images 1 and 2, the overlapped area in the stitched image is divided into two parts ( Parts 1 and 2, as shown in **Fig. 15(b)** ) by the stitching seam, that belongs to source image 1 and 2, respectively. The overlapped area is segmented into local regions by mean shift algorithm. Within each local region, the color of pixels remains coherent and continuous, so color correction is conducted within each local region instead of the entire overlapped area.

$$SP^{(i)} = \left[ p_{s1}^{(i)}, p_{s2}^{(i)}, \cdots, p_{sm}^{(i)} \right] = \begin{bmatrix} p_{s1\_R}^{(i)}, & p_{s2\_R}^{(i)}, & \cdots, & p_{sm\_R}^{(i)} \\ p_{s1\_G}^{(i)}, & p_{s2\_G}^{(i)}, & \cdots, & p_{sm\_G}^{(i)} \\ p_{s1\_B}^{(i)}, & p_{s2\_B}^{(i)}, & \cdots, & p_{sm\_B}^{(i)} \end{bmatrix} \quad (i = 1,2) \tag{7}$$

where $SP^{(i)}$ is the set of $m$ seam points for image $i\,(i=1,2)$ and $p_{st}^{(i)}$ $(i=1,2; t=1\cdots m)$ is one of the seam points. $p_{st\_j}^{(i)}$ denotes channel $j$ for pixel $p_{st}^{(i)}$. Solve the linear models for R, G, and B channels, respectively, and get the coefficients:

$$\left[ \alpha_j^{(1)*}, \beta_j^{(1)*} \right] = \left\{ \left[ X_{sj}^{(1)} \right]^T X_{sj}^{(1)} \right\}^{-1} \left[ X_{sj}^{(1)} \right]^T X_{sj} \ (j = R,G,B) \tag{8}$$

where:

$$X_{sj}^{(i)} = \begin{bmatrix} p_{s1\_j}^{(i)}, & 1 \\ p_{s2\_j}^{(i)}, & 1 \\ \vdots & , \vdots \\ p_{sm\_j}^{(i)}, & 1 \end{bmatrix}, \quad X_{sj} = \begin{bmatrix} p_{s1\_j}, & 1 \\ p_{s2\_j}, & 1 \\ \vdots & , \vdots \\ p_{sm\_j}, & 1 \end{bmatrix} (\, i = 1,2;\ j = R,G,B \,) \tag{9}$$

$p_{st\_j}(t=1\cdots m; j=R,G,B)$ is the weighted combination of seam points in source image 1 and 2:

$$p_{st\_j} = (1 - \frac{n_c(t)}{N_c}) p_{st\_j}^{(1)} + \frac{n_c(t)}{N_c} p_{st\_j}^{(2)} \ (t = 1\cdots m; j = R,G,B) \tag{10}$$

$n_c(t)$ and $N_c$ denote the column index of $p_{st}^{(i)}$ and the width of the overlapped area, respectively. For each local region in Part 1, the linear color transfer model for each channel $j = R,G,B$ at pixel $x$ is defined as:

$$c_j(x) = \alpha_j^{(1)}(x) + \beta_j^{(1)}(x) c_j^{(1)}(x) \quad (j = R,G,B; x \in Part\ 1) \tag{11}$$

where $c_j(x)$ and $c_j^{(i)}(x)$ $(i = 1,2; j = R,G,B)$ denote the color values of channel $j$ at pixel $x$ for the stitched image and source image 1 and 2, respectively. Instead of keeping constant values, the coefficients $\alpha_j^{(1)}(x)$ and $\beta_j^{(1)}(x)$ adjust as follows:

$$\begin{cases} \alpha_j^{(1)}(x) = \alpha_0^{(1)} + n_c(x)\Delta\alpha_j^{(1)}(x) \\ \beta_j^{(1)}(x) = \beta_0^{(1)} + n_c(x)\,\Delta\beta_j^{(1)}(x) \end{cases} (j = R,G,B) \tag{12}$$

where $n_c(x)$ is the column index of pixel $x$, and $d_s(x)$ is the horizontal distance to the seam (number of pixels) at pixel $x$. $\alpha_0^{(1)} = 1, \beta_0^{(1)} = 0$ and $\Delta\alpha_i(x)$, $\Delta\beta_i(x)$ are obtained as follows:

$$\begin{cases} \Delta\alpha_j^{(1)}(x) = \dfrac{\alpha_j^{(1)*} - \alpha_0^{(1)}}{n_c(x) + d_s(x)} \\ \Delta\beta_j^{(1)}(x) = \dfrac{\beta_j^{(1)*} - \beta_0^{(1)}}{n_c(x) + d_s(x)} \end{cases} (j = R,G,B) \tag{13}$$

Similarly, we can obtain the color transfer model at pixel $x$ for Part 2 as follows:

$$c_j(x) = \alpha_j^{(2)}(x) + \beta_j^{(2)}(x) c_j^{(2)}(x) \quad (j = R,G,B; x \in Part\ 2) \tag{14}$$

where the values of coefficients are obtained as follows:

$$\begin{cases} \alpha_j^{(2)}(x) = \alpha_0^{(2)} + d_s(x)\Delta\alpha_j^{(2)}(x) \\ \beta_j^{(2)}(x) = \beta_0^{(2)} + d_s(x)\,\Delta\beta_j^{(2)}(x) \end{cases} (j = R,G,B) \tag{15}$$

$$\begin{cases} \Delta\alpha_j^{(2)}(x) = \dfrac{\alpha_0^{(2)} - \alpha_j^{(2)*}}{N_c - n_c(x) + d_s(x)} \\ \Delta\beta_j^{(2)}(x) = \dfrac{\beta_0^{(2)} - \beta_j^{(2)*}}{N_c - n_c(x) + d_s(x)} \end{cases} (j = R,G,B) \tag{16}$$

where $\alpha_0^{(2)} = 1, \beta_0^{(2)} = 0$ and $\alpha_j^{(2)*}, \beta_j^{(2)*}$ are expressed as:

$$\left[\alpha_j^{(2)*}, \beta_j^{(2)*}\right] = \left\{\left[X_{sj}^{(2)}\right]^T X_{sj}^{(2)}\right\}^{-1} \left[X_{sj}^{(2)}\right]^T X_{sj} \ (j = R,G,B) \tag{17}$$

The result of the seam-based local color transfer model is illustrated in **Fig. 16(c)**.

## 4. Experimental Results and Analysis

The procedure of the proposed algorithm is illustrated in **Fig. 6**.

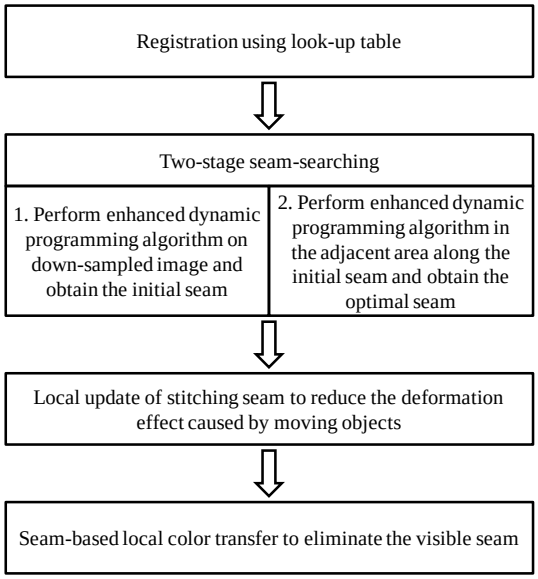


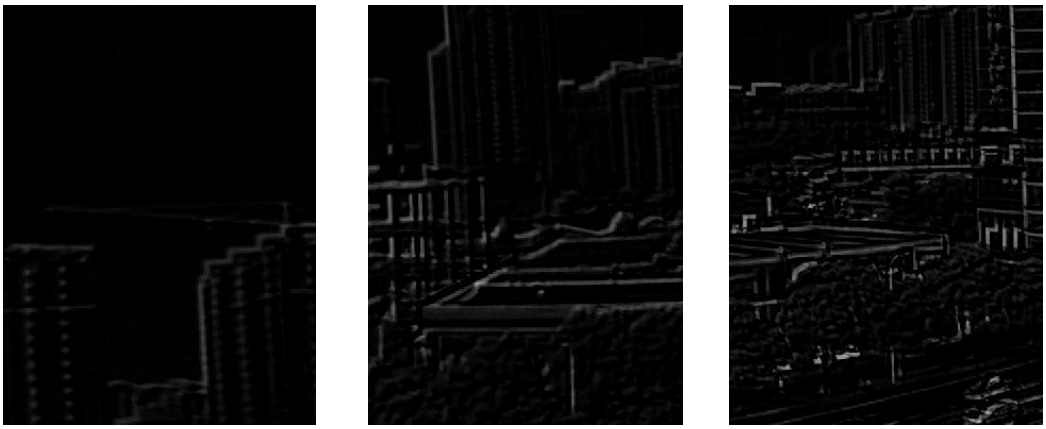

**Fig. 6.** Flowchart of the proposed algorithm

The experiments are conducted on a multi-camera sampling platform using HIKVISION DS-2CD3210(D)-I5 720p cameras. The proposed video stitching algorithms are implemented on a 2.00 GHz PC with 2GB memory. In **Fig. 8**, the results of different seam-searching algorithms for three overlapped areas: image $I_1, I_2, I_3$ are compared, with (a)-(c), (d)-(f) and (g)-(i) illustrating the seams obtained through dynamic programming, enhanced dynamic programming and the proposed algorithm, respectively. In these algorithms, the stitching seams are illustrated with red labels.

Compared with the dynamic programming and enhanced dynamic programming algorithm, the seam-searching result of the proposed algorithm is also satisfactory. Meanwhile, the proposed seam-searching algorithm can effectively reduce the computational complexity, as illustrated in **Table 1**.

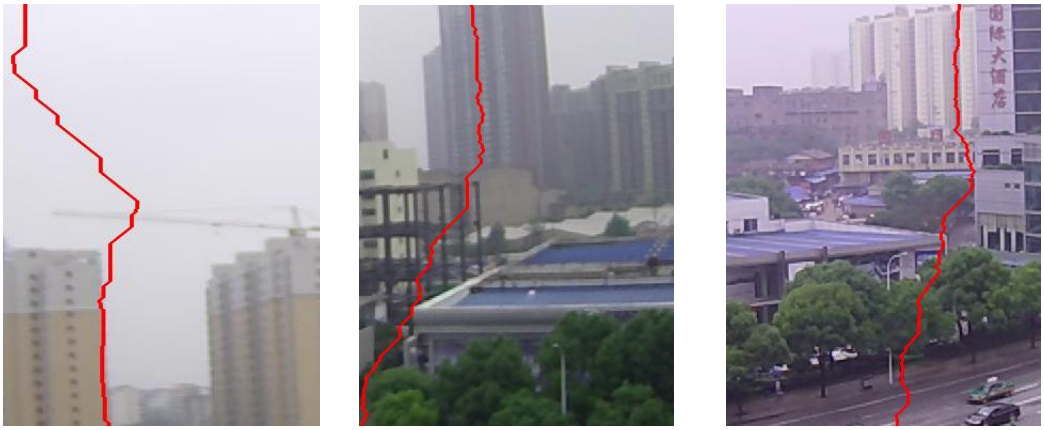
Video stitching based on the proposed seam-searching algorithm obtains almost the same satisfactory result as enhanced dynamic programming, while the dynamic programming algorithm suffers worse stitching results. As shown in **Fig. 1**, when the seam obtained by dynamic programming passes through unsmooth region (**Fig. 1(b)**), deformation and incoherence occur in the red pane. Seam-searching using the proposed algorithm is more flexible in the searching direction because the seam can extend on the horizontal direction. Thus the unsmooth region can be detoured by the seam. The overlapped areas have a dimension of $640 \times 480$, based on the surveillance videos.

The comparison of merging results for the proposed method and the traditional dynamic programming algorithm are shown in **Fig. 9** and **Fig. 10**. Structure deformations occur in the red boxes in **Fig. 10** for dynamic programming, while the proposed algorithm can obtain satisfactory merging result.

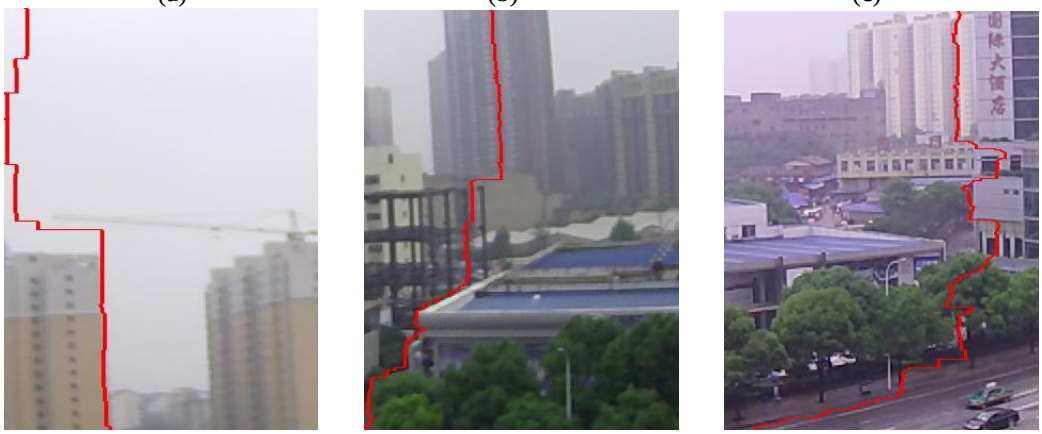


(a) Cost value of $I_1$ (b) Cost value of $I_2$ (c) Cost value of $I_3$

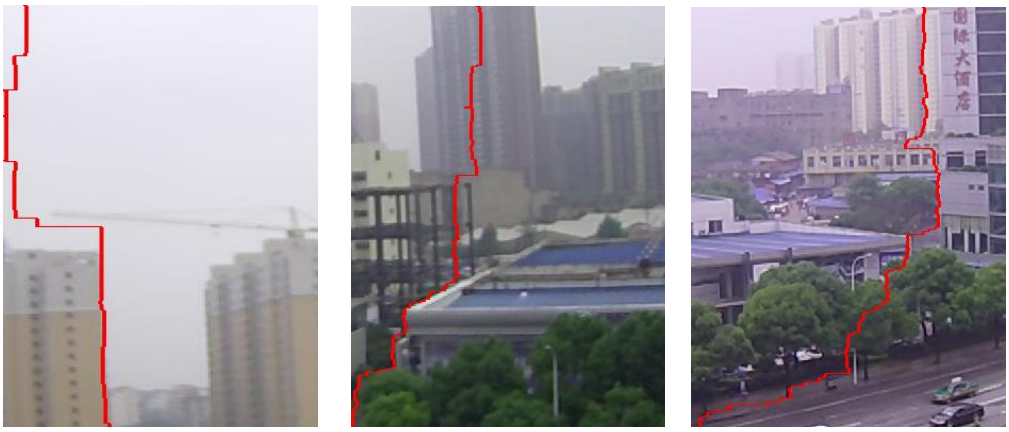
**Fig. 7.** Cost value of overlapped areas for images $I_1, I_2, I_3$



(a) (b) (c)

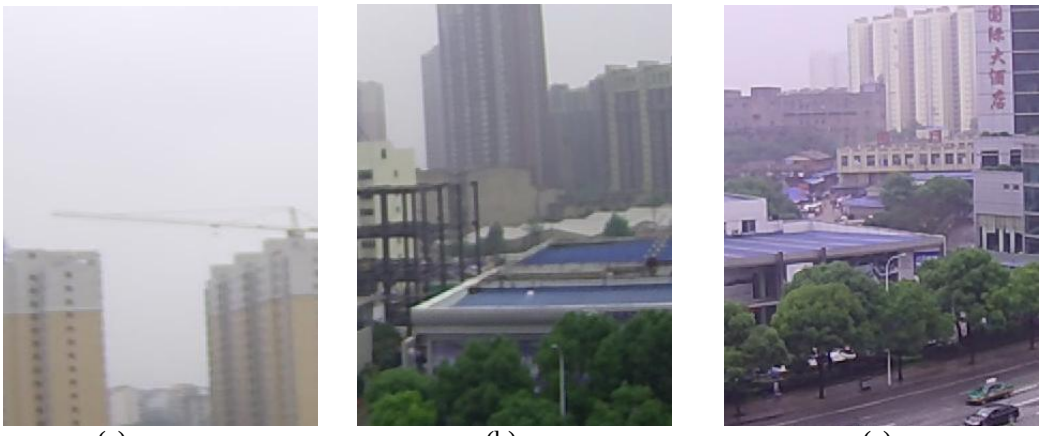


(d) (e) (f)



(g) (h) (i)

**Fig. 8.** Comparison of seam-searching results. (a)-(c), (d)-(f), (g)-(i) are the seam-searching results of dynamic programming, enhance dynamic programming, and the proposed algorithm, respectively. Three columns of images correspond to image $I_1, I_2, I_3$, respectively.



(a) (b) (c)

**Fig. 9.** Merging results of the proposed algorithm

(a) (b) (c)

**Fig. 10.** Merging results of the dynamic programming

Given the combinations of parameters $r$ and $K$, the computing time of seam-searching and the accumulated cost of stitching seam obtained by Equation (3) are shown in **Tables 1** and **2**, respectively.

**Table 1** and **2** indicate that the proposed seam-searching algorithm overcomes conventional methods in terms of computing time, and the accuracy of this algorithm is also satisfactory. Although the value of accumulated cost of stitching seam is slightly higher than that of enhanced dynamic programming owing to information loss in the down-sampling process, it achieves better real-time performance and hardly influences the stitching quality.

**Table 1**. Computing time of seam-searching

| **Algorithm** | | | | **Computing Time (ms)** | | |
|---|---|---|---|---|---|---|
| **Image** | | | | $I_1$ | $I_2$ | $I_3$ |
| Dynamic Programming | | | | 156.478 | 162.323 | 158.549 |
| Enhanced Dynamic Programming | | | | 265.125 | 271.342 | 269.640 |
| Proposed Algorithm | $r=3$ | $K=20$ | Stage 1 | 31.390 | 32.901 | 31.164 |
| | | | Stage 2 | 25.064 | 26.995 | 24.870 |
| | | | Total | 56.454 | 59.896 | 56.034 |
| | | $K=30$ | Stage 1 | 31.309 | 32.946 | 31.113 |
| | | | Stage 2 | 34.943 | 34.931 | 33.639 |
| | | | Total | 66.252 | 67.877 | 64.752 |
| | | $K=40$ | Stage 1 | 31.311 | 32.984 | 31.173 |
| | | | Stage 2 | 47.387 | 44.613 | 43.248 |
| | | | Total | 78.698 | 77.597 | 74.421 |
| | | $K=50$ | Stage 1 | 31.365 | 32.932 | 31.885 |
| | | | Stage 2 | 55.891 | 56.987 | 56.267 |
| | | | Total | 87.256 | 89.919 | 88.152 |
| | | $K=60$ | Stage 1 | 31.391 | 32.993 | 31.123 |
| | | | Stage 2 | 70.924 | 69.251 | 71.956 |
| | | | Total | 102.315 | 101.244 | 102.079 |
| | $r=4$ | $K=20$ | Stage 1 | 19.411 | 19.746 | 19.406 |
| | | | Stage 2 | 22.493 | 23.514 | 22.891 |
| | | | Total | 42.904 | 43.260 | 42.297 |
| | | $K=30$ | Stage 1 | 19.409 | 19.750 | 19.467 |
| | | | Stage 2 | 32.550 | 36.152 | 34.583 |
| | | | Total | 53.959 | 55.902 | 54.050 |
| | | $K=40$ | Stage 1 | 19.472 | 19.754 | 19.413 |
| | | | Stage 2 | 47.781 | 48.013 | 46.220 |
| | | | Total | 66.253 | 67.767 | 65.633 |
| | | $K=50$ | Stage 1 | 19.431 | 19.710 | 19.439 |
| | | | Stage 2 | 56.504 | 59.123 | 57.862 |
| | | | Total | 77.935 | 78.833 | 77.301 |
| | | $K=60$ | Stage 1 | 19.351 | 19.513 | 19.401 |
| | | | Stage 2 | 71.123 | 72.183 | 69.240 |
| | | | Total | 89.474 | 91.696 | 88.641 |
| | $r=5$ | $K=20$ | Stage 1 | 12.953 | 12.872 | 13.449 |
| | | | Stage 2 | 22.858 | 23.313 | 22.769 |
| | | | Total | 35.811 | 36.185 | 36.218 |
| | | $K=30$ | Stage 1 | 12.984 | 12.823 | 13.410 |
| | | | Stage 2 | 32.287 | 32.972 | 33.087 |
| | | | Total | 45.271 | 45.795 | 46.497 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | $K=40$ | Stage 1 | 12.943 | 12.895 | 13.491 |
| | | | Stage 2 | 43.230 | 43.460 | 43.482 |
| | | | Total | 56.173 | 56.355 | 56.973 |
| | | $K=50$ | Stage 1 | 12.937 | 12.864 | 13.491 |
| | | | Stage 2 | 54.923 | 55.192 | 55.420 |
| | | | Total | 67.860 | 68.056 | 68.911 |
| | | $K=60$ | Stage 1 | 12.959 | 12.842 | 12.473 |
| | | | Stage 2 | 66.401 | 67.056 | 66.505 |
| | | | Total | 79.360 | 79.898 | 78.978 |

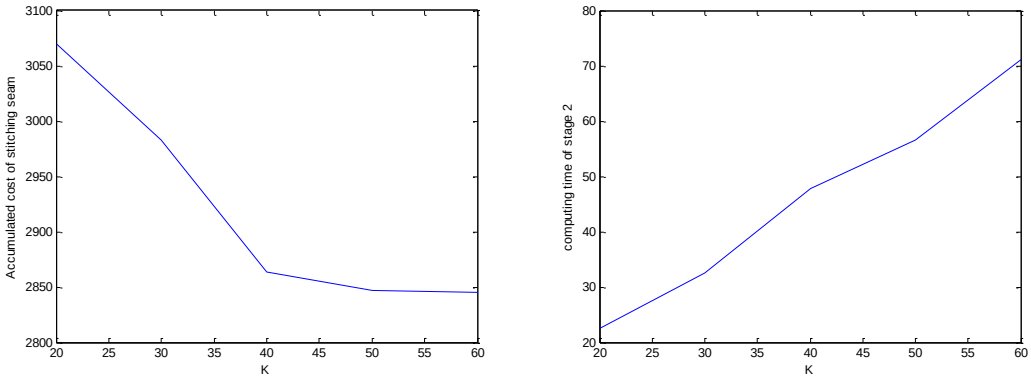
**Table 2.** Accumulated cost of stitching seam

| **Algorithm** | | | **Accumulated Cost of Stitching Seam** | | |
|---|---|---|---|---|---|
| **Image** | | | $I_1$ | $I_2$ | $I_3$ |
| Dynamic Programming | | | 3469 | 5623 | 6105 |
| Enhanced Dynamic Programming | | | 2858 | 4172 | 5289 |
| Proposed Algorithm | $r=3$ | $K=20$ | 3063 | 4348 | 5357 |
| | | $K=30$ | 2976 | 4292 | 5341 |
| | | $K=40$ | 2847 | 4256 | 5304 |
| | | $K=50$ | 2847 | 4221 | 5303 |
| | | $K=60$ | 2843 | 4201 | 5301 |
| | $r=4$ | $K=20$ | 3070 | 4369 | 5383 |
| | | $K=30$ | 2983 | 4312 | 5367 |
| | | $K=40$ | 2864 | 4278 | 5313 |
| | | $K=50$ | 2847 | 4225 | 5309 |
| | | $K=60$ | 2845 | 4201 | 5304 |
| | $r=5$ | $K=20$ | 3213 | 4361 | 5380 |
| | | $K=30$ | 3194 | 4335 | 5359 |
| | | $K=40$ | 3051 | 4303 | 5336 |
| | | $K=50$ | 2915 | 4246 | 5327 |
| | | $K=60$ | 2890 | 4232 | 5321 |

**Table 1** and **2** show that the computing time of seam searching and the accumulated cost of stitching seam are sensitive to the settings of parameters $r$ and $K$. Less accumulated cost of stitching seam corresponds to higher visual quality of the stitched image. Considering both computing time and accumulated cost, the combination $r=4, K=60$ generates the best result and the corresponding seam-searching results are shown in **Fig. 8(g-i)**. Compared with enhanced dynamic programing, the computing time of the proposed method is reduced by 66.4% on average for the input images, while the seam-searching performance is maintained.

With the increase of sampling interval $r$, the computing time of stage 1 drops dramatically because of the decrease of pixels to be processed. However, the accumulated cost of stitching seams are also on the increase due to the error of the initial seam in stage 1, which is caused by information loss in the down-sampling process.

Larger values of $K$ contributes to the decrease of accumulated cost, and the descent speed declines due to the increasing accuracy for seam optimization. The computing time of stage 2 is proportional to the choice of $K$. For image 1, when $r=4$ the variation of accumulated cost and computing time of stage 2 with the increase of $K$ is shown in **Fig. 11(a)** and **(b)**.




(a) Accumulated cost of stitching seam (b) Computing time of stage 2

**Fig. 11.** The variation of accumulated cost and computing time.

**Fig. 12.** Dividing the existing seam into sections.

As shown in **Fig. 12**, the entire seam is divided into sections. The results for local update of stitching seams in video surveillance are shown in **Fig. 13**, preventing the seam from passing through the moving objects. In our experiment, $\Delta d = \frac{w}{6}$, where $w$ is the width of the overlapped area, and the seam is segmented into 9 parts. Compared with **Fig. 1**, the stitching results of the moving car passed through by the seam are shown in **Fig. 13(a)** and **(b)**, indicating that the proposed algorithm can achieve better stitching result.

(a) Local update of the seam (b) Stitching result of (c)

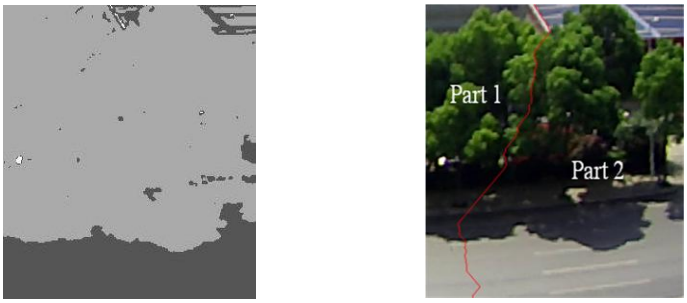
**Fig. 13.** local update of stitching seam

Smooth transition can be achieved using the seam-based local color transfer model. The two source images are shown in **Fig. 14**. Segmentation is implemented using Mean shift algorithm and the result is shown in **Fig. 15(a)**, with the different colors indicating labels for regions. **Fig. 15(b)** illustrates the seam-searching result.

As shown in **Fig. 16(a)**,**(b)**, the method [12] cannot eliminate the visble seam and the blending method [13] produce blurring result as a result of registration error,. Based on the results of segmentation, the seam-based local color transfer model is applied to the local regions passed through by the seam. Compared with results of traditional smooth transition methods in **Fig. 16(a)** and **(b)**, more natural and smooth transition result is obtained using the proposed model. Blurring and visible seams in the overlapped region are eliminated.

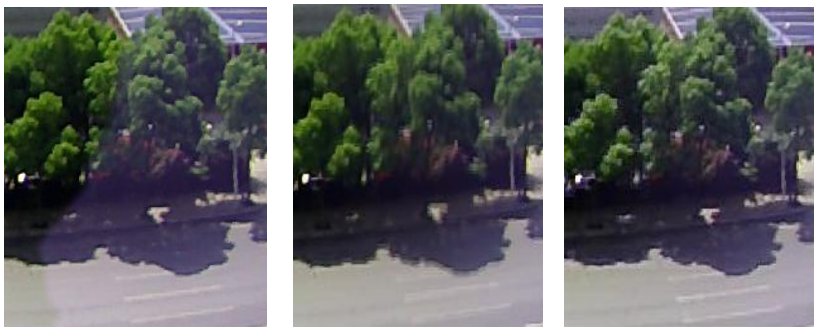
(a) Source image 1 (b) Source image

**Fig. 14.** Source images for smooth transition




(a) Segmentation result (b) Stitching seam

**Fig. 15.** Segmentation and seam-searching for the overlapped area



(a) Visible seam[12] (b) Blurring effect[13] (c) Proposed model

**Fig. 16.** Comparison of smooth transition methods

The final results of video stitching with moving objects are displayed in **Fig. 17**.

**Fig. 17.** Results of video stitching

## 5. Conclusion

In this paper, a novel video stitching method for multi-camera surveillance systems is presented. To achieve better real-time performance for seam-searching process, a two-stage seam searching method based on enhanced dynamic programming is proposed and satisfactory seam searching result is obtained. A local update scheme for stitching seams is proposed to eliminate deformation results. A seam-based local color transfer model is constructed to achieve smooth transition between different images. Experiments have shown the effectiveness of the proposed method. For future research, hardware implementation of this method will be designed to accelerate the process of video stitching. Other efficient stitching algorithms will be further explored.

## References

[1] Wei Xu and Jane Mulligan, "Panoramic video stitching from commodity HDTV cameras," *Multimedia Systems*, Vol. 19, No. 5, 407-426, 2013. Article (CrossRef Link).

[2] Kwang Wook Lee, Seung Won Jung, Seung Kyun Kim, and Sung Jea KO, "A Novel Content-Aware Stitching Algorithm for Real-time Video Sequences," *IEICE Transactions on Information and System*, Vol. E94-D, No.2, pp. 357-362, 2011. Article (CrossRef Link).

[3] Yong Yang , Wenjuan Zheng, Shuying Huang, "A Novel Automatic Block-based Multi-focus Image Fusion via Genetic Algorithm," *KSII Transactions on Internet and Information Systems (TIIS)*, Vol.7 No.7, 1671-1689, 2013. Article (CrossRef Link).

[4] Cuiyin Liu , Peng Cheng , Shu-qing Chen , Cuiwei Wang , Fenghong Xiang, "A Novel Multifocus Image Fusion Algorithm Based on Nonsubsampled Contourlet Transform," *KSII Transactions on Internet and Information Systems (TIIS)*, Vol.7 No.3, 539-557, 2013. Article (CrossRef Link).

[5] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, 1222 - 1239 , 2001. Article (CrossRef Link).

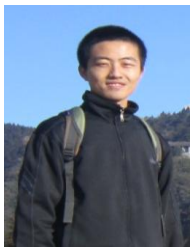
[6] J. Davis, "Mosaics of scenes with moving objects," in *Proc. of Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 354 - 360, 1998. Article (CrossRef Link).

[7] Hua Gu, Yue Yu, and Weidong Sun, "A New Optimal Seam Selection Method for Airborne Image Stitching," in *Proc. of Proceedings of International Workshop on Imaging Systems and Techniques*, pp. 159-163, 2009. Article (CrossRef Link).

[8] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics*, Vol. 26, No. 3, pp. 1-9, 2007. Article (CrossRef Link).

[9] M. Uyttendaele, A. Eden. and R. Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image

Mosaics," in *Proc. of Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, pp. 509-516, 2001. Article (CrossRef Link).
[10] Alec Mills and Gregory Dudek, "Image stitching with dynamic elements," *Image and Vision Computing*, Vol. 27, No. 10, pp. 1593-1602, 2009. Article (CrossRef Link).
[11] M. Brown and D. G. Lowe. "Recognising Panoramas," in *Proc. of Proceedings of International Conference Computer Vision (ICCV)*, Vol. 2, pp. 1218-1225, 2003. Article (CrossRef Link).
[12] Richard Szeliski, "Image Alignment and Stitching: A Tutorial," *Technical Report MSR-TR-2004-92*, pp. 1-68, 2004. Article (CrossRef Link).
[13] Peter J. B. and Edward H. A., "A Multi-resolution Spline with Application to Image Mosaics," *ACM Transactions on Graphics*, Vol. 2, No. 4, pp. 217-236, 1983. Article (CrossRef Link).
[14] R. Szeliski, "Image mosaicing for tele-reality applications," in *Proc. of Proceedings of 2nd IEEE Workshop on Applications of Computer Vision,* 1994, pp. 44-53. Article (CrossRef Link).
[15] D. Comaniciu and P. Meer, "Mean shift: A Robust Approach toward feature space analysis," *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603-619, 2002. Article (CrossRef Link).
[16] Young-Seob Jeong, Chae-Gyun Lim, Byeong-Soo Jeong, Ho-Jin Choi, "Topic Masks for Image Segmentation," *KSII Transactions on Internet and Information Systems (TIIS)*, Vol.7, No.12, 3274-3292, 2013. Article (CrossRef Link).
[17] Jingbo Zhou, Shangbing Gao, Zhong Jin, "A New Connected Coherence Tree Algorithm For Image Segmentation," *KSII Transactions on Internet and Information Systems (TIIS)*, VOL. 6, NO. 4, 547-565, 2012. Article (CrossRef Link).
[18] M. El-Saban, M. Izz, A. Kaheel and M. Refaat, "Improved optimal seam selection blending for fast video stitching of videos captured from freely moving devices," in *Proc. of Proceedings of 18th IEEE International Conference on Image Processing*, pp. 1481 - 1484, 2011.
Article (CrossRef Link).



**Xiaoqing Yin** received the B.S., and M.S. degrees in Multimedia Information System and Virtual Reality from National University of Defense Technology, Changsha, Hunan, China, in 2011 and 2013, respectively. Currently, he is working toward the Ph.D. degree with the College of Information System and Management, National University of Defense Technology. His current research interests include image and video stitching, object detection and recognition.

**Weili Li** received the B.S. degrees in System Engineering from National University of Defense Technology, Changsha, Hunan, China, in 2012. Currently, she is working toward the M.S. degree in the College of Information System and Management, National University of Defense Technology. Her current research interests include image and video stitching, debluring and denoising.

**Bin Wang** received the B.S., M.S. and Ph.D. degrees in Control Science and Engineering from College of Information System and Management, National University of Defense Technology, Changsha, Hunan, China, in 2006, 2008 and 2013, respectively. He is currentlyworking in the Facility Design and Instrumentation Institute, China Aerodynamics Research and Development Center, Mianyang, Sichuan. His current research interests include machine learning, object detection and recognition, image and video denoising.

**Yu Liu** received the B.S. degree from NorthWestern Polytechnical University, Xi'an, China in 2005 and the M.S. degree on image processing and Ph.D. degree on computer graphics from University of East Anglia, Norwich, UK, in 2007 and 2011, respectively. He is currently a lecturer in the Department of System Engineering, College of Information System and Management, National University of Defense Technology, Changsha, China. His current research interests include image and video processing, computer graphics, and visual-haptic technology.

**Maojun Zhang** received the B.S. and Ph.D. degrees in System Engineering from National University of Defense Technology, Changsha, China, in 1992, and 1997 respectively. He is currently a professor in the Department of System Engineering, College of Information System and Management, National University of Defense Technology. In 2001, he became a visiting professor at the University of Ottawa, Canada. His research interests include computer vision, image and video stitching, information system engineering, system simulation and virtual reality technology.