Moving Obstacle Removal via Low Rank Decomposition

Yue-Yun Deng[1], Yan-Rui Xu[1], Dong Wang[1]*, Yue-Fang Gao[1], Xiao-Qiang Wu[2]
1 College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
2 Guangdong Electronics Industry Institute, Dongguan, China
E-mail: wdngng@163.com

***Abstract*—Obstacle removal is a classic problem in image processing. Because the content behind obstacle can't be deduced only from one image, we introduce a motion obstacle removal problem from an image sequence. It is regarded as a sparse problem with the obstacles as noise. First we match the images with image features to transform the images to the same camera coordinate system. The features of motion objects are not included, which are detected by the big offset of matching features in neighbor images. Then a matrix is constructed with each matched image as column. Without obstacles, each column should be the same. So we decompose the matrix into a low rank matrix and a sparse matrix. According to the low rank we can get the background image without obstacles. Compared to previous methods, the method can restore the background image correctly without any interaction. The experiments show the results are ideal.**

Keywords: Low rank decomposition; image completion; sparse representation; image matching; mesh deformation

I. INTRODUCTION

When we are taking photos, such situations always happen that the obstacles of unexpected people and animations pass through the scene suddenly. To avoid these obstacles, we have to adjust the shooting angle, which will change the content and layout of the scene; or just shoot directly where the obstacles stay in the scene. Anyway, the results are not what we want.

Obstacle removing is a classic problem in image processing. Although lots of work has been done, the results of the solutions from a single image are not satisfactory. Because obstacle removal is an ill-conditioned problem, the content behind obstacles can't be deduced accurately from a single image. In this paper, we will provide a new moving obstacle removal scheme based on image series. The images are taken at continuous time interval for the same scene. Because the objects are moving, the covered content in one image can be found in other images. We assume that the moving objects account for a small proportion of the whole image, and the blocked regions appear only in a few images. It is regarded as a sparse problem with the obstacles as noise, so a low rank decomposition method can be used to restore the background image.

Compared to previous methods, the method requires a series of photos. In fact, for an important scene, many people do in this way. Based on such an input, the method can accurately restore the background image, no matter how complex the content is. Another advantage is that the solution is fully automatic without any interaction.

II. RELATED WORK

Image restoration can be roughly divided into two categories with one based on diffusion and the other based on examples. The techniques based on diffusion are to fill small or narrow holes by diffusing neighbor image structure. The pioneer works include Bertalmio [1], Ballester [2], Bertalmio [3]. Because this kind of methods don't have texture synthesis technique, they can't fill a large unrestored region. In comparison, the example-based methods can restore the missing regions by mining the redundant information in the natural images. Some are texture synthesis methods based on examples [4, 5] and others are filling techniques based on prior structure [6]. These techniques rely on the underlying cues, which are usually invalid for large structures. In many real scenes, due to the shape change of the local scene, it is difficult to synthesize reasonable results by simple translate transformation of image patches. Mansfield [7] and Darabi [8] handled this problem by increasing similarity transformation, reflection transformation and the change of light. However, with the increase of dimensions and complexity, the search for the nearest neighbors becomes more difficult. Huang [9] found the appropriate transformation by analyzing the middle structure of the image. These methods above are only suitable for structural images. For more extensive applications, some methods with additional databases are given. Hays [10] retrieved similar semantic images from the database and fill the area to be repaired with one large region. Zhang [11] completed the region restoration by transforming similar regions. The results by those methods based on a single image just look reasonable visually, but usually are not consistent with the original content. In this paper, we refer to the idea in [12] to solve the problem of moving obstacle removal. With multiple images as input, transform the images to the same coordinate system, and take each transformed image as three columns to construct a matrix. Regarding motion objects as noise, decompose the matrix into a low rank matrix and a noise matrix by low rank decomposition. The mean of all columns of the low rank matrix is as the restored background image without motion objects. In their solution each image is transformed in the same way. However in our application because the depth of the whole scene may be very different, uniform transformation cannot achieve expected matching. We believe that the transformation of local region is similar, and the matching problem is solved by grid deformation technique.

978-1-5090-4400-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDH.2016.17

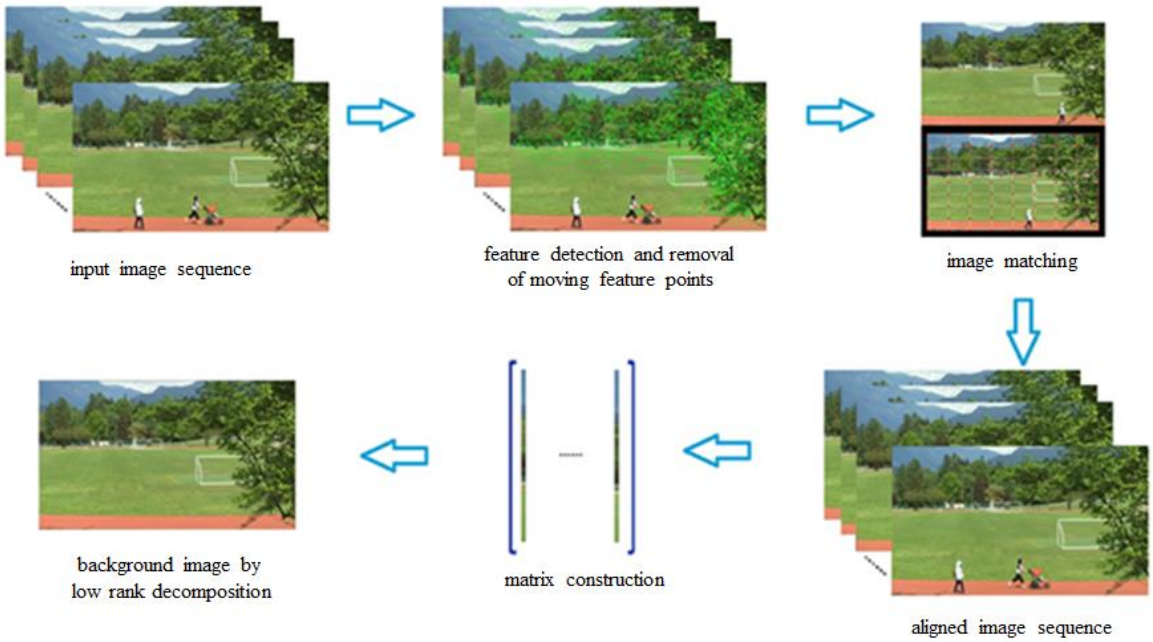


Figure 1. Overview

III. ALGORITHM MODEL

In the input image sequence, motion objects in different images cover different regions of the background and the same region is occluded only in a small proportion of images. The motion objects can be considered as noise and the occluded background region will be found in other images, so this is a sparse problem. If there is no motion objects, all the images should be the same after image matching. The problem can be solved by low rank decomposition. The whole algorithm is shown in Fig. 1. First feature matching for the input images is established. According to the position offset of the corresponding feature points in two adjacent images, the feature points of motion objects are removed. Then the reference image is selected based on the remaining feature points. The left images are aligned with the reference image and all the images are transferred to the same camera coordinate system. Last the common regions of all the images are found. With the common region of each image as a column, a matrix is constructed, which is further decomposed into a low rank matrix and a sparse matrix. The background image without motion objects is obtained from the low rank matrix. The whole algorithm consists of two parts: one is image matching and the other is low rank decomposition.

A. *Image Matching*

Image sequence can be got by continuous shooting in a short period of time. But, because the transformations such as translation and rotation are inevitable for the hand-held camera, there are not only the difference of position but the shape change for the same object in different images. It will affect the subsequent low rank decomposition seriously, which should be rectified by image matching.

Image matching depends on feature matching. Because the feature points in motion objects will interfere with image matching, they need to be removed. Given image sequence $\{\tau_1, \tau_2, \cdots, \tau_n\}$, n is the number of images, feature matching is based on Speed Up Robust Feature (SURF) technique. Set the positions of the corresponding feature points of image τ_i and image τ_{i+1} in Fig.2(c) are $\{P_1^i, P_2^i, \cdots, P_m^i\}$ and $\{P_1^{i+1}, P_2^{i+1}, \cdots, P_m^{i+1}\}$ respectively, m is the number of matching feature points in the two images. It is obvious that the position offsets of the feature points in motion object are larger than those of the left feature points in the two adjacent images. Based on the phenomenon, the feature points in motion object are identified. Set the average distance between the pairs of matched feature points $\bar{d} = \sum_{j=1}^{m} \left| P_j^{i+1} - P_j^i \right| / m$. For the k[th] pair of feature points, if $\left| P_k^{i+1} - P_k^i \right| > 2\bar{d}$, the feature point belongs to the motion object.

In order to get a common region large enough for image matching, it needs to choose a suitable reference image. The camera coordinate system of ideal reference image should be in the middle of those of all images. It is the same for the positions of feature points in the images, which is used to find the reference image. Set the positions of all common feature points except those in motion objects from the 1[st] to n[th] images as $\{\{P_1^1, P_2^1, \cdots, P_c^1\}, \{P_1^2, P_2^2, \cdots, P_c^2\}, \cdots, \{P_1^n, P_2^n, \cdots, P_c^n\}\}$, where c is the number of those selected common feature points in the images, and c≤m. With the means of the

(a) image τ_i (b) image τ_{i+1} (c) position offsets of corresponding feature points in two images

Figure 2. Motion feature points detection

(a) Image τ_i with regular mesh (b) image τ_r (c) Image $\hat{\tau}_i$ with deformed mesh

Figure 3. Image matching

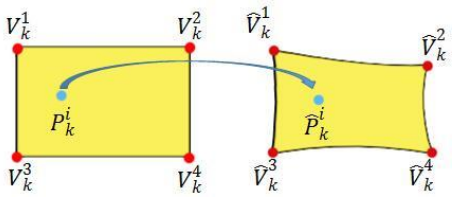


Figure 4. Mesh vertices and matching feature points

positions of these feature points as $\{\overline{P_1}, \overline{P_2}, \cdots, \overline{P_c}\}$, we compute the sum of differences between the means and the positions of these feature points in each image. That is, for image i, the sum of those differences is $d_i = \sum_k \left| P_k^i - \overline{P_k} \right|$. Set the smallest one $d_m = \arg\min_i d_i, i = 1,2,...,n$, the image with d_m as reference image τ_r.

With reference image τ_r, we use Liu et al's method in [13] to align those images. It is based on the idea of mesh deformation that after mesh deformation, the feature points in an image τ_i should have the same coordinates as the corresponding points in reference image τ_r and the mesh shape should be preserved as soon as possible. Concretely, for image τ_i, a regular rectangular mesh M_i is constructed, as shown in Fig. 3(a). For any feature point k in image τ_i with its coordinate P_k^i, it is in the mesh face with the coordinates of four vertices $\{V_k^1, V_k^2, V_k^3, V_k^4\}$, as shown in Fig. 4. Coordinate P_k^i can be represented by bilinear interpolation of the four vertices, which is $P_k^i = \sum_{j=1}^4 \omega_j V_k^j$ with $\sum_{j=1}^4 \omega_j = 1$. Based on this, we can obtain the coefficients ω_j, j=1,2,3,4. After mesh deformation, set the four vertices changed into $\{\hat{V}_k^1, \hat{V}_k^2, \hat{V}_k^3, \hat{V}_k^4\}$, the new coordinate of feature point k can be expressed as $\hat{P}_k^i = \sum_{j=1}^4 \omega_j \hat{V}_k^j$. Given the coordinate of feature point k in image τ_r as P_k^r, the coordinate $\hat{P}_k^i$ should be equal to P_k^r. Meanwhile, the mesh shape is preserved by triangular similarity constraint. For each mesh face, the four vertices can be divided into two triangles by one diagonal line. Let the coordinates of one of the triangles are $\{V_1, V_2, V_3\}$ and $\{\hat{V}_1, \hat{V}_2, \hat{V}_3\}$ respectively before and after deformation, the energy function is defined as:

$$E(\hat{V}) = \sum_k \left\| P_k^r - \sum_{j=1}^4 \omega_j \hat{V}_k^j \right\|^2 + \alpha \sum_t \left(\left\| \hat{V}_1 - \hat{V}_2 - sR_{90}(\hat{V}_3 - \hat{V}_2) \right\|^2 \right) \quad (1)$$

, $R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $s = \|V_1 - V_2\| / \|V_3 - V_2\|$, α is the weight coefficient. The function is a quadratic function about vertices set $\hat{V}$ in the deformed mesh $\hat{M}_i$, which can be solved by a sparse linear equation system. According to the deformed mesh, the image τ_i can be aligned with the reference image τ_r to get a new image $\hat{\tau}_i$, as shown in Fig. 3(c).

Because of the camera's movement, the content of those images is not fully the same, it is needed to find the common region of all images from the newly generated image sequence $\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_n\}$. After matching, the same feature points have the same coordinates, so the method to find the common region is by the coordinates. We search the minimum x coordinates from the right boundary vertices of all the deformed meshes, the maximum x coordinates from the left, the minimum y coordinates from the bottom and the maximum x coordinates from the top. The four coordinates form a rectangular region, which is the common region for image sequence $\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_n\}$. The common region is labeled as $\{\bar{\tau}_1, \bar{\tau}_2, \cdots, \bar{\tau}_n\}$.

B. Matrix low rank decomposition

The generated image sequence $\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_n\}$ after image matching just likes it taken by fixed camera. If there are no motion objects, the content of all common regions $\{\bar{\tau}_1, \bar{\tau}_2, \cdots, \bar{\tau}_n\}$ should be the same. If we construct a matrix using $\{\bar{\tau}_1, \bar{\tau}_2, \cdots, \bar{\tau}_n\}$ with each RGB component of $\bar{\tau}_i$, i=1,2,...,n as a column, it will be a low rank matrix with rank 3. However, due to the presence of moving objects, no two common regions $\bar{\tau}_i, \bar{\tau}_j$, i,j=1,2,...,n,i≠j, are exactly the same and the matrix will be full rank. Although there are different, they occupy a small proportion of the whole image, which can be regarded as noise. That is, if we remove the noise, $\bar{\tau}_i, \bar{\tau}_j$ are the same and the matrix is reduced to a low rank matrix. So a low rank decomposition methods is given to solve the problem.

Set RGB three components of any image $\bar{\tau}_i$ in image sequence $\{\bar{\tau}_1, \bar{\tau}_2, \cdots, \bar{\tau}_n\}$ are $\bar{\tau}_i^R$, $\bar{\tau}_i^G$ and $\bar{\tau}_i^B$. By converting these components into column vectors separately, we construct a matrix A_E as:

$$A_E = [\mathrm{vec}(\bar{\tau}_1^R) \mid \mathrm{vec}(\bar{\tau}_1^G) \mid \mathrm{vec}(\bar{\tau}_1^B) \mid \cdots \mid \mathrm{vec}(\bar{\tau}_n^R) \mid \mathrm{vec}(\bar{\tau}_n^G) \mid \mathrm{vec}(\bar{\tau}_n^B)] \quad (2)$$

For matrix A_E, we hope to decompose it into a low rank matrix A and a sparse matrix E. That is, such a model needs to be solved:

$$\min_{A,E} \mathrm{rank}(A) + \lambda \|E\|_0 \quad \text{s.t.} \quad A_E = A + E, \quad (3)$$

where $\|\cdot\|_0$ represents the 0-norm operation, which counts the number of non-zero elements in matrix. Coefficient λ is a constant greater than zero, which balances the weight between the low rank and the sparse error. In (3), the rank function and the zero norm of the matrix are both non convex and discontinuous function, so it is difficult to solve them directly. It is a NP-hard problem and according to the idea of convex relaxation, the formula can be rewritten as:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad A_E = A + E, \quad (4)$$

where $\|A\|_*$ represents kernel norm of matrix A and $\|\cdot\|_1$ is one norm. That is, the rank of the matrix is approximated by the kernel norm and the zero norm is approximated by one norm. Thus the new (4) can be solved by the ALM algorithm [14]. Each column of the decomposed low rank matrix A represents the background information of the image. Considering the calculation error, we average all the columns to obtain the background image τ_b without motion objects.

IV. RESULT AND ANALYSIS

This algorithm is implemented by MATLAB programming on a computer, which has a 8G memory and a i3 dual core CPU. In the experiment, the number of image sequences is about 20 in each example. For the initial image sequence with 568*320 image size, the running time of the whole algorithm is about 300 seconds, among that, the image matching takes 50 seconds, and the matrix low rank decomposition takes 250 seconds. The weight coefficient of grid deformation in image matching is $\alpha = 1$. The λ in sparse matrix with low rank decomposition is $\lambda = 1/\sqrt{N}$, N is the number of rows of the matrix A_E, that is, the number of pixels in the final generated image τ_b.

Fig.5 shows an example of image sequence preprocessing. Figure 5(a) is one of the three input image sequences, the middle one is the reference image. Fig. 5(b) is the rectified reference image by using grid deformation in feature matching. Fig. 5(c) is a common region obtained from a sequence of matched images. In Fig. 5(a) in the three images, the contents of the image have changed as well as the moving people. The tree leaves in the left side of image are gradually increased from the first to the third image, while the contents in the right side of image are gradually decreased, that means the camera is moving from left side to right side. After matching correction, the three images are converted to the same camera coordinate system, and the results are shown in Fig. 5(b), where the black region corresponds to the contents that are not photographed in the middle reference image, which means the camera is tilted toward the upper left in shooting process. The common region images (5(c)) are obtained by cutting. It can be seen from the figure, the contents of the scene are exactly the same except the moving people. At this time the image can be understood as in the case of a fixed camera, the photos are taken at different moments.

Fig. 6 gives a group of examples of moving obstacle removing. In Fig. 6(a), there are 6 images in the input image sequence, arranged in two columns, motion objects are in different position in the scene. Images with the rectangular frame are to be processed in [6], the contents in the rectangular frame are removed, and will be restored by using

(a) (b) (c)

Figure 5. Image sequence preprocessing.(a) Input image sequence;(b) Match Image; (c) Extract common region

the method in [6]. Fig. 6(b) shows the result of the removal of obstacles where in each example, the upper graphs are our results, the lower graphs are the results of the method in [6]. The method in literature [6] relies on the local information to deduce the content of the region to be restored, because of the uncertainty and complexity of the region to be restored, a gap exists between the results and reality. As shown in the figure, the contents of the region to be restored are not accurately recovered. In our results, because of the occlusion of the contents can be fully showed in other images, motion objects are completely removed, and the block region is also well restored.

V. CONCLUSION

In this paper we introduce a method on moving obstacle removal based on low rank decomposition. Compared with the traditional single image restoration method, it has the following characteristics: (1) it is completely automatic without any interaction; (2) there is no special requirement for the content of the image, which is applicable to any image; (3) the occluded background region can be restored accurately. Although it requires an image sequence as input, it is easy to acquire by hand-held cameras like smartphones because it doesn't require to fix the camera. In the next, we will relax the constraints on obstacles and try to solve the problem on static obstacles.

ACKNOWLEDGMENT

The paper is supported by NSF of China (No. 61202294), Joint Funds of NSF of China (No. U1301253), Science and Technology Planning Project of Guangdong Province, China (No. 2016A020210086) and Guangdong Innovative Research Team Program (No.201001D0104726115).

REFERENCES

[1] BERTALMIO, M., SAPIRO, G., CASELLES, V., AND BALLESTER, C. 2000. Image inpainting. ACM Trans. on Graphics (Proc. Of Siggraph) 19, 3, 417-424.

[2] BALLESTER, C., BERTALMIO, M., CASELLES, V., SAPIRO, G., AND VERDERA, J. 2001. Filling-in by joint interpolation of vector fields and gray levels. IEEE TIP 10, 8, 1200-1211.

[3] BERTALMIO, M., VESE, L., SAPIRO, G., AND OSHER, S. 2003. Simultaneous structure and texture image inpainting. IEEE TIP 12, 8, 882-889.

[4] EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. ACM Trans. on Graphics (Proc.of Siggraph) 20, 3, 341-346.

[5] EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In ICCV.

[6] CRIMINISI, A., PEREZ ´ , P., AND TOYAMA, K. 2004. Region filling and object removal by exemplar-based image inpainting. IEEE TIP 13, 9, 1200-1212.

[7] MANSFIELD, A., PRASAD, M., ROTHER, C., SHARP, T., KOHLI, P., AND VAN GOOL, L. 2011. Transforming image completion. In BMVC.

[8] DARABI, S., SHECHTMAN, E., BARNES, C., GOLDMAN, D. B., AND SEN, P. 2012. Image Melding: Combining Inconsistent Images using Patch-based Synthesis. ACM Trans. on Graphics (Proc. of Siggraph) 31, 4.

[9] JIABIN, H., SINGBING K., NARENDRA A., AND JOHANNES K. 2014. Image Completion using Planar Structure Guidance. ACM Trans. on Graphics (Proc. of Siggraph) 33, 4.

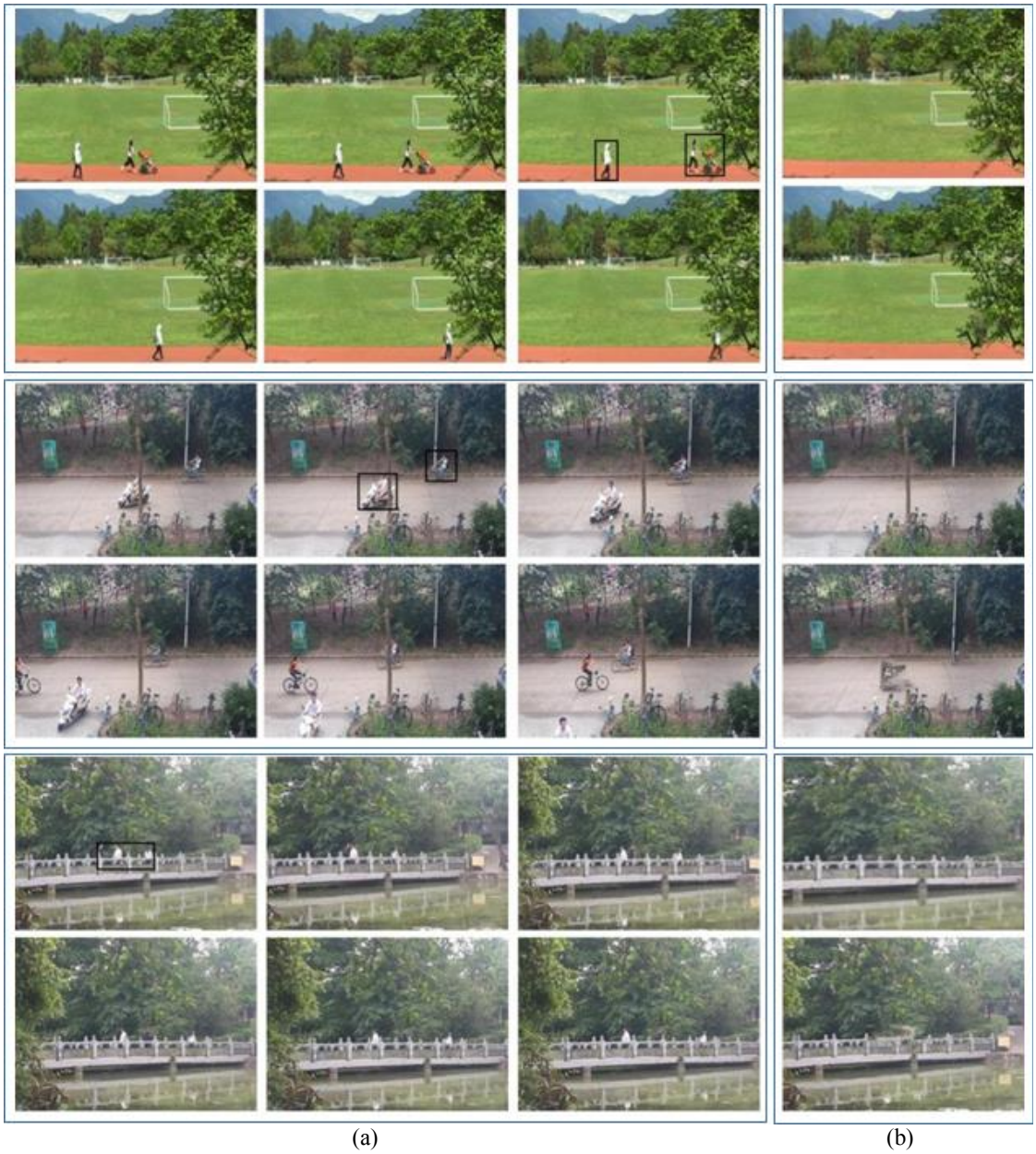
[10] HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. ACM Trans. on Graphics (Proc. of Siggraph) 26, 3, 4.

[11] ZHANG, Y., XIAO, J., HAYS, J., AND TAN, P. 2013. FrameBreak: Dramatic image extrapolation by guided shift-maps. In CVPR.

[12] PENG Y., GANESH A., WRIGHT J., XU W., AND MA Y.. 2010. RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images. In CVPR.

[13] LIU S., YUAN L., TAN P., AND SUN J. 2013. Bundled Camera Paths for Video Stabilization. ACM Transactions on Graphics (Proceeding of SIGGRAPH) 32, 4.

[14] Lin Z, Chen M, Ma Y. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices[J]. Eprint Arxiv, 2010,9.



(a) (b)

Figure 6. Moving obstacle removal. (a) Input image sequence; (b) removal result. Each example occupies two rows. The upper image in (b) is our result, and the lower is the result of the method in [6].