PERSPECTIVE DISTORTED VIDEO RESTORATION AND STABILIZATION FOR MOBILE DEVICES

Shuai Yang[1], *Jiaying Liu*[1], *Wenhan Yang*[1] *and Zongming Guo*[1,2]

[1]Institute of Computer Science and Technology, Peking University, Beijing, China

[2]Cooperative Medianet Innovation Center, Shanghai, China

ABSTRACT

Pictures/Videos taken by mobile devices may suffer degradation problems such as perspective distortion, noises and shakes. In this paper, we present a novel approach to solve the perspective distortion and camera shake issues. A perspective correction approach is proposed based on a pyramidal quadrangle detection method. An adaptive window-based autoregressive(AR) interpolation is brought up to enhance the warping result. We rectify the transformation model with smoothing motion vector to realize a video stabilization process that improves the video quality. Experimental results confirm the enhancement of the visual quality our method brings.

Index Terms— Perspective distortion, video stabilization, quadrangle detection, AR interpolation, mobile devices

1. INTRODUCTION

Mobile devices are widely used in daily life to record abundant information in forms of pictures and videos. These pictures/videos contain a variety of contents, ranging from PowerPoints on the projection screen to the posters on the billboard and the slogans on the banner as shown in Fig.1. However, due to the dissatisfactory shooting angles, lighting conditions and camera shake problems, they may suffer image degradation problems such as perspective distortion, noises and shakes. Sometimes, valuable information in these pictures/videos becomes hard to recognize.

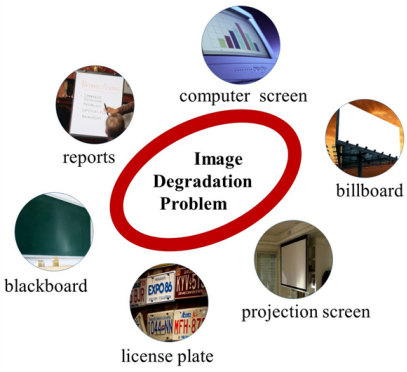


Fig. 1. Application scenarios of the proposed perspective distorted video restoration and stabilization approach

This work was supported by National High-tech Technology R&D Program (863 Program) of China under Grant 2014AA015205, National Natural Science Foundation of China under contract No.61472011 and Beijing Natural Science Foundation under contract No.4142021.

A targeted rectangular object in pictures/videos suffering perspective distortion can degenerate into a quadrangle. One way of solving this problem is the perspective correction method. Li *et al.* [1] used Hough transformation to identify the targeted quadrangle to estimate the transformation parameters. Yang *et al.* [2] proposed a perspective correction method based on diagonal scanning to detect four corners of the quadrangle. Perspective correction by calculating two vanishing points was proposed by Xu *et al.* [3]. However, line information and corner information are not fully utilized in these methods, leading to inaccurate results.

Most of the videos shot by mobile devices simultaneously suffer perspective distortion and shake problems. Research has been devoted to video stabilization to suppress video vibrations. Chang *et al.* [4] calculated optical flow to derive the global motion. A global motion estimation based on block matching was proposed by Chen *et al.* [5]. With the development of the feature extraction techniques, feature-based global motion estimations [6][7] were proposed to align two images for video stabilization. Achievements have been made in the research, but perspective correction integrated with video stabilization is rarely mentioned.

In this paper, we present an effective video restoration method for perspective correction and video stabilization to enhance the visual quality. The targeted quadrangle is first detected using a two-level pyramidal quadrangle detection method, which takes advantage of both line features and corner features. Then, we calculate transformation parameters and warp the image to correct distortion problems. In the warping process, an adaptive window-based autoregressive(AR) interpolation is proposed to reduce interpolation artifacts. Furthermore, based on our quadrangle detection algorithm, a video stabilization method is proposed. By calculating a smoothing motion vector for each frame, we rectify the transformation model to realize the motion compensation.

The rest of the paper is organized as follows: Section 2 describes the proposed perspective distorted video restoration and stabilization approach for mobile devices. Experimental results are shown in Section 3 and conclusions are summarized in Section 4.

2. PROPOSED PERSPECTIVE DISTORTED VIDEO RESTORATION AND STABILIZATION APPROACH

In this section, the proposed perspective distorted video restoration and stabilization approach is presented. Fig.2

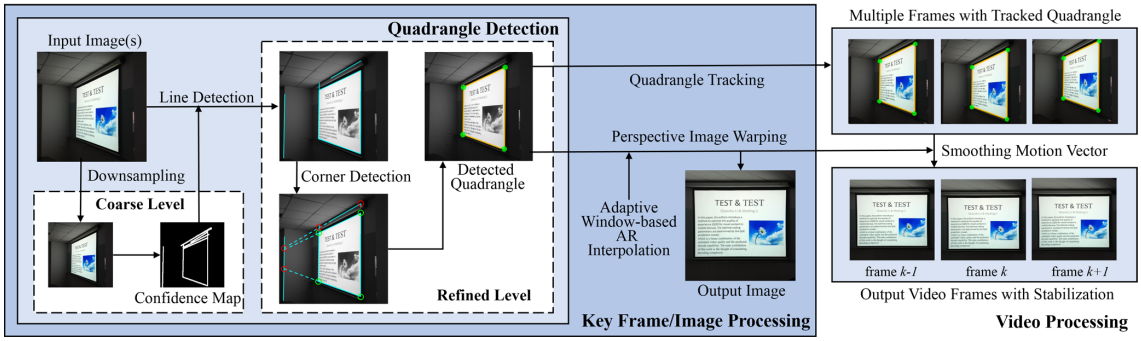


Fig. 2. Flow chart of the proposed perspective distorted video restoration and stabilization approach

shows the framework of our approach. In this paper, we denote the targeted quadrangular object as $\mathbf{Q}$ and its corresponding rectangular object after restoration as $\mathbf{R}$.

2.1. Quadrangle Detection in Image

The task of the quadrangle detection is to seek four vertexes coordinates of $\mathbf{Q}$ as precisely as possible. Since natural images contain much irrelevant information interfering with the quadrangle detection procedure, a two-level pyramidal implementation is proposed to solve this problem. In the coarse level, we seek a *Confidence Map* denoted as M_c that eliminates most of the irrelevant information but reserves the region where the edges of $\mathbf{Q}$ are located. In the refined level, line features and corner features are extracted to determine the position of $\mathbf{Q}$.

2.1.1. Coarse Level

The input key frame/image I is first down sampled to a low-resolution image I^l that reserves principal structural information. Then, a cluster of lines $\Omega_l = \{l_1, ..., l_k\}$ is obtained using a Hough line detection. Loose thresholds are set to ensure four edges of $\mathbf{Q}$ can be all obtained. Next, we select valid lines from Ω_l according to two criterions.

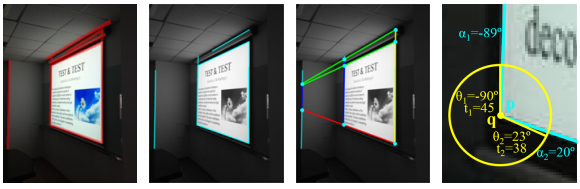
- Criterion 1 (Small Variation): The variation of l_i is defined as the number V_i of '01's and '10's along it. For each $l_i \in \Omega_l$, if $V_i \leq \beta_1 \cdot length(l_i)$, then $l_i \in \Omega'_l$;
- Criterion 2 (Effective Length): The effective length of l_i is defined as the number L_i of its longest continuous '1's. For each $l_i \in \Omega'_l$, if $L_i \geq \beta_2 \cdot length(l_i)$, then $l_i \in \Omega''_l$.

In the end, Ω''_l is extended by morphological dilation and is magnified to obtain M_c. Fig.3(a) shows M_c as the red region with $\beta_1 = 0.02$ and $\beta_2 = 0.67$.

2.1.2. Refined Level

In this section, we operate on I to detect a cluster of candidate quadrangles denoted as $\Omega_Q = \{Q_1, ..., Q_k\}$ using line features and determine $\mathbf{Q}$ from Ω_Q using corner features.

Line features: The candidate edges are detected in the area of M_c using the Hough line detection on I. Then, according to the slopes and the spatial positions of the candidate edges, we classify them into four groups. By taking one edge from each group, we get four edges that define a candidate quadrangle $Q_i \in \Omega_Q$. The candidate edges and the candidate quadrangles are shown in Fig.3(b)(c).



(a) (b) (c) (d)

Fig. 3. Key procedures of our quadrangle detection method. (a) Confidence Map. (b) Candidate edges. (c) Result of grouping. The blue, green, yellow and red lines are classified into group left-edge, right-edge, top-edge and bottom-edge respectively. (d) The corner detected by ACJ method.

Corner features: For each $Q_i \in \Omega_Q$ with four vertexes noted as $\mathbf{p}_{i,j}$ $(j = 1, 2, 3, 4)$, we find corners in a small neighborhood around $\mathbf{p}_{i,j}$ using a contrario junction detection [8]. According to [8], as shown in Fig.3(d), a corner is a structure defined as $\{\mathbf{q}, \{t_i, \theta_i\}_{i=1}^M\}$. It is characterized by its center $\mathbf{q}$ and M branches around $\mathbf{q}$. Each branch is expressed as its direction θ and strength t. The branch strength is a measure of how well its direction agrees with the branch. We define the probability of the intersection $\mathbf{p}$ of two edges with directions of α_1 and α_2 to be a valid corner by:

$$P(\mathbf{p}) = (t_{i1} \cdot \tau(\alpha_1, \theta_{i1}) + t_{i2} \cdot \tau(\alpha_2, \theta_{i2}))/(t_{i1} + t_{i2}), \quad (1)$$

$$i1 = \arg\max_{i1} \tau(\alpha_1, \theta_{i1}), i2 = \arg\max_{i2} \tau(\alpha_1, \theta_{i2}),$$

$$\tau(\alpha, \theta) = \max(|\cos(\alpha - \theta)| - |\sin(\alpha - \theta)|, 0),$$

where $\{\mathbf{q}, \{t_i, \theta_i\}_{i=1}^M\}$ is the corner detected around $\mathbf{p}$, τ measures the matching degree of two angles, the $i1$-th and $i2$-th branches are the best matching branches with the edges. Finally, $\mathbf{Q} = \arg\max_{Q_i \in \Omega_Q} \sum_{j=1}^4 P(\mathbf{p}_{i,j})$ is determined.

2.2. Perspective Transformation Model with Stabilization

Given four vertexes coordinates of $\mathbf{Q}$, the vertexes coordinates of $\mathbf{R}$ are calculated. The perspective transformation parameters are deduced and rectified using a smoothing motion vector to realize the video stabilization procedure.

2.2.1. *Quadrangle Tracking*

For multiple frames, $\mathbf{Q}$ is detected in the key frames and then tracked in the following frames for perspective correction. Let I_i be the i-th frame of the input video. In our work, a pyramidal version of Lucas-Kanade optical flow computation [9] is applied to track four vertexes between I_k and I_{k+1}. In Lucas-Kanade process, a subpixel computation is enabled making the result more accurate.

2.2.2. *Video Stabilization with Smoothing Motion Vector*

The vertexes coordinates $\{(x_i, y_i)\}_{i=1}^4$ of $\mathbf{Q}$ are obtained in Section 2.1. The vertexes coordinates $\{(x'_i, y'_i)\}_{i=1}^4$ of $\mathbf{R}$ are determined by its size and the central point C. Given the size of $\mathbf{Q}$ and the particular aspect ratio, the size of $\mathbf{R}$ can be estimated. The central point C of $\mathbf{R}$ is determined according to two different alignment strategies:

- *Center Alignment*: set C as the central point of I;
- *Floating Alignment*: set C as the central point of $\mathbf{Q}$.

The first strategy eliminates global motions thus leaving no shakes. The second strategy reserves global motions but may suffer undesired video jitters. To suppress jitters while maintaining global motions, we apply a motion smoothing method based on quadrangle tracking in the second alignment strategy. To be more specific, let $C_k = (\bar{x}_k, \bar{y}_k)$ be the central point of $\mathbf{Q}$ in I_k. A motion vector T_i^j from I_i to I_j is defined as $T_i^j = C_j - C_i$. In the *floating alignment*, we seek the smoothing motion vector $S_k = (\delta x_k, \delta y_k)$ of I_k by:

$$S_k = \sum_{k-\mu \le i \le k+\mu} T_k^i \otimes G(k-i), \tag{2}$$

where $G(x)$ is the Gaussian kernel, $\otimes$ is the convolution operator and μ is the smoothing factor. Larger μ yields a smoother result. In the *center alignment*, $S_k = \mathbf{0}$.

Finally, we merge the motion composition into the perspective correction process by rectifying the affine perspective transformation model of I_k:

$$x = \frac{ax' + by' + c}{gx' + hy' + 1} + \delta x_k, \; y = \frac{dx' + ey' + f}{gx' + hy' + 1} + \delta y_k, \tag{3}$$

where (x', y') is a coordinate in the restored frame I'_k, (x, y) is the corresponding coordinate in I_k. a, b, c, d, e, f, g, and h are parameters of the affine perspective transformation model. Given $\{(x_i, y_i)\}_{i=1}^4$ of $\mathbf{Q}$ and $\{(x'_i, y'_i)\}_{i=1}^4$ of $\mathbf{R}$, eight parameters can be calculated by solving the linear system.

2.3. Image Warping using Proposed Interpolation

For each pixel (x', y') in I', its corresponding coordinate (x, y) in I is calculated using (3). Since x and y are usually not integers, the adaptive window-based AR interpolation is used to magnify I' by factor of q, and the value of the pixel (x', y') is set to the value of the pixel round(qx, qy) in the magnified image. In real implementation, we set $q = 4$.

In the natural images, 'patch sifting' is represented as a general phenomenon. Adjacent image patches in local exhibit similar spatial distribution of intensity. Along the image

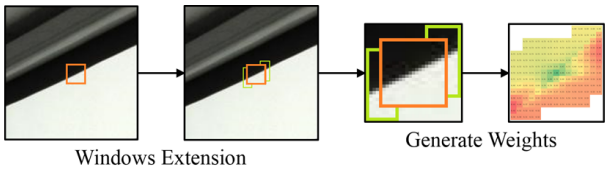


Fig. 4. Key procedures of the proposed adaptive window-based AR interpolation

isophote, the intensity spatial distribution of successive adjacent image patches shows similarity and changes gradually. Based on this, the adaptive window-based AR interpolation method is proposed. As shown in Fig.4, the algorithm estimates the isophote by local neighboring similar patches information and then extends the interpolation window along the isophote direction. An irregular window that contains similar adjacent patches is built. Modulated with the similarity metric based on patch-geodesic distance, the objective function is obtained. Finally, the function can be solved iteratively.

Let $\boldsymbol{X}$ be a vector consisting of four LR pixels in the interpolation window. Let $\boldsymbol{Y}$ be a vector consisting of HR pixels in the interpolation window. Let $\boldsymbol{M}$ and $\boldsymbol{N}$ be vectors consisting of the covariance coefficients between pixels. Let $\boldsymbol{W}$ be the diagonal matrix composed of the similarity probability based on the patch-geodesic distance. We obtain the objective function:

$$\hat{\boldsymbol{Y}} = \arg\min_{\boldsymbol{Y}} ||\boldsymbol{W}(\boldsymbol{M}\boldsymbol{Y} - \boldsymbol{N}\boldsymbol{X})||_2^2, \tag{4}$$

$$\boldsymbol{M} = \left[\boldsymbol{I}_{(k_1+k_2)\times(k_1+k_2)}, \boldsymbol{M}^1_{(k_3)\times(k_1+k_2)}, \lambda\boldsymbol{M}^2_{(k_1)\times(k_1+k_2)}\right]^T,$$

$$\boldsymbol{N} = \left[\boldsymbol{N}^1_{(k_1+k_2)\times(k_3+k_4)}, \boldsymbol{N}^2_{(k_3)\times(k_3+k_4)}, \mathbf{0}_{(k_1)\times(k_3+k_4)}\right]^T,$$

where λ is the Lagrange multiplier, k_1, k_2, k_3, k_4 are the numbers of four types of pixels in the interpolation window. As shown in Fig.5, gray HR pixels can only be the centers in the diagonal direction constraints. Blue HR and red LR pixels can be the centers in both the diagonal direction and the cross direction constraints. Black LR pixels only engage in other pixels' constraints. In Fig.5, $k_1 = 4$, $k_2 = 19$, $k_3 = 11$, $k_4 = 27$. $\boldsymbol{I}$ is the unit matrix and $\mathbf{0}$ is the zeros matrix. $\boldsymbol{N}^2 = [\boldsymbol{I}_{k_3\times k_3} \quad \mathbf{0}_{k_3\times k_4}]$. The elements of other submatrixs in $\boldsymbol{M}$ and $\boldsymbol{N}$ are defined as:

$$m_1(i,j) = \begin{cases} \hat{r}_k, Y_i \in \{X_{i\otimes k} | k = 1,2,3,4\} \\ 0, \text{other} \end{cases},$$

$$m_2(i,j) = \begin{cases} 1, i = j \\ -\hat{s}_k, Y_j \in \{Y_{i\oplus k} | k = 1,2,3,4\} \\ 0, \text{other} \end{cases},$$

$$n_1(i,j) = \begin{cases} \hat{r}_k, X_i \in \{Y_{i\otimes k} | k = 1,2,3,4\} \\ 0, \text{other} \end{cases}.$$

where $k \in \{1,2,3,4\}$ is used to represent the k-th neighborhood in the AR model. $x_{i\otimes k}$ and $y_{i\otimes k}$ represents the k-th

neighborhood of x_i and y_i in the diagonal direction. $x_{i\oplus k}$ and $y_{i\oplus k}$ represents the k-th neighborhood of x_i and y_i in the cross direction. r_k represents the parameters in diagonal direction and s_k represents the parameters in cross direction.

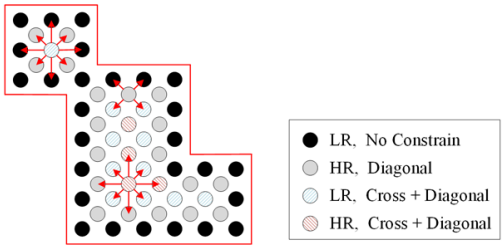


Fig. 5. An example of the window extension result and the distributions of the pixels.

Then a close-formed resolution can be obtained:

$$\hat{Y} = (M^T W^2 M)^{-1} M^T W^2 N x. \quad (5)$$

3. EXPERIMENTAL RESULTS

The proposed algorithm is implemented on MATLAB R2013b platform. Our perspective correction method is compared with the CaptureBoard[1]. Pictures and videos in different application scenarios are taken by mobile phones for perspective correction and stabilization. Part of the experimental results are shown in this section. The whole pictures, videos and experimental results have been released on our website[2].

Fig. 6. Comparison of the quadrangle detection results. From left to right: *pic* 0, *pic* 1 and *pic* 2. From top to bottom: CaptureBoard and the proposed method.

The quadrangle detection results are shown in Fig.6. Almost all detection results are ideal except a less-than-ideal case in *pic* 0. The proposed method excludes the taskbar from the desktop in *pic* 0. CaptureBoard does well in *pic* 0, but fails to detect the accurate positions of the projection screens in *pic* 1 and *pic* 2. As a result, CaptureBoard cannot solve the distortion problem well as shown in Fig.7. On the contrary, the quadrangular screens are successfully corrected by the proposed method. Since our method is designed to detect general quadrangles and the detection result is indeed a valid quadrangle, the performance of the proposed method in *pic* 0 is acceptable. Moreover, higher-quality warping results are provided by our method compared to CaptureBoard.

Then, we show the performance of two alignment strategies and test the proposed video stabilization method. In the experiment, we set $\mu = 6$. As shown in Fig.8(a), the input frames contain a screen dropping severely at the 4-th frame. The *center alignment* strategy eliminates all motions, and the *floating alignment* strategy does nothing about video jitters. After motion smoothing, video jitters are suppressed effectively, and the screen moves down steadily. The red lines in Fig.8 help find out the computer screen's motion fluctuation.

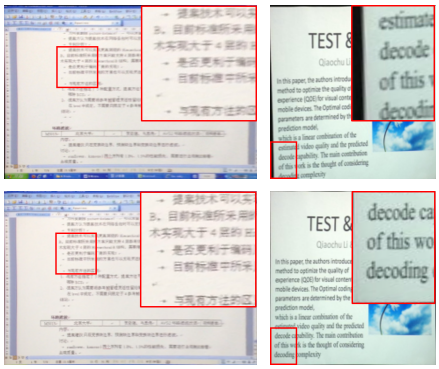


Fig. 7. Comparison of the perspective correction results. From left to right: *pic* 0 and *pic* 1. From top to bottom: CaptureBoard and the proposed method.

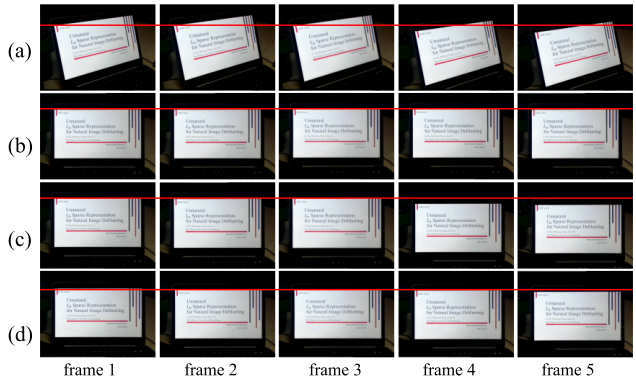


Fig. 8. Video perspective correction and stabilization results. (a) The original video frame. (b) The *center alignment* strategy. (c) The *floating alignment* strategy. (d) The stabilization result.

4. CONCLUSION

This paper presents a novel perspective distorted video restoration and stabilization approach for mobile devices. A two-level pyramidal quadrangle detection method is proposed to locate the position of the targeted quadrangular object. An adaptive window-based AR interpolation is proposed for image warping. A practical motion smoothing algorithm is proposed to remove the undesired video jitters. Experimental results demonstrate the improvement of the visual quality our algorithm brings.

[1] http://itunes.apple.com/en/app/captureboard/id542229387

[2] http://www.icst.pku.edu.cn/course/icb/Projects/PICE.html

5. REFERENCES

[1] R. Li, M. Y. Fort, and G. C. Anagnostopoulos, "Multi-Stage Automatic License Plate Location and Recognition", *Advances of Machine Learning in Theory and Applications,* 2008.

[2] S. J. Yang,C. C. Ho, J. Y. Chen, and C. Y. Chang, "Practical Homography-Based Perspective Correction Method for License Plate Recognition", *Proc. of Information Security and Intelligence Control,* pp. 198-201, 2012.

[3] X. W. Xu, Z. Y. Wang, Y. Q. Zhang, and Y. H. Liang, "A Skew Distortion Correction Method for 2D Bar Code Images Based on Vanishing Points", *Proc. of IEEE International Conference Machine Learning and Cybernetics,* vol. 3, pp. 1652-1656, 2007.

[4] H. C. Chang, S. H. Lai, and K. R. Lu, "A Robust Real-Time Video Stabilization Algorithm," *Visual Communication and Image Representation,* vol. 17, no. 3, pp. 659-673, 2006.

[5] H. H. Chen, C. K. Liang, Y. C. Peng, and H. A. Chang, "Integration of Digital Stabilizer with Video Codec for Digital Video Cameras," *IEEE Trans. Circuits and System for Video Technology,* vol. 17, no. 7, pp. 801-813, 2007

[6] J. Yang, D. Schonfeld, C. Chen, and M. Mohamed, "Online Video Stabilization Based on Particle Filters," *Proc. of IEEE International Conference on Image Processing,* pp. 1545-1548, 2006.

[7] A. Adams, N. Gelfand, and K. Pulli, "Viewfinder Alignment" *Computer Graphics Forum,* vol. 2, no. 27, pp. 597-606, 2008.

[8] G. S. Xia, J. Delon, and Y. Gousseau, "Accurate Junction Detection and Characterization in Natural Images", *International Journal of Computer Vision,* vol. 106, no. 1, pp. 31-56, 2014.

[9] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proc. of International Joint Conference on Artificial Intelligence,* vol. 81, pp. 674-679, 1981.