Modelling Radial Distortion Chains for Video Recapture Detection

Marco Visentini-Scarzanella [1], Pier Luigi Dragotti [2]

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London
SW7 2AZ London, United Kingdom
[1] marcovs@imperial.ac.uk
[2] p.dragotti@imperial.ac.uk

***Abstract*—This paper presents a novel cue for automatic recapture detection of videos. The problem of recapture detection is important to the field of digital forensics as recapture is often an indicator of prior tampering activity. In this paper, we tackle the problem by considering the deformation underwent by geometric primitives, such as straight lines, when processed along recapture chains. We mathematically derive a general curve model for straight lines deformed after single capture under a radial distortion model. The model is then extended to the case of recapture, demonstrating how to automatically classify videos on a *per-frame* basis from its compliance with a low-order radial distortion model. Finally, we test our model with a practical detector to automatically extract deformed straight lines for classification, which is applied to synthetic sequences.**

I. Introduction

The recent surge in popularity of digital forensics can be attributed primarily to the widespread perception of a fundamental technological gap: while graphics software has been relentlessly evolving enabling users to convincingly tamper or even synthetically render images [1], algorithmic solutions able to automatically identify the tampering have essentially been outpaced.

With the current ease of duplicating, tampering and massively distributing images and videos through social networks, deception is within anyone's reach, eroding the public trust on digital content. Modern forgery detection methods have focused on a variety of 'footprints': traces left by different operators during acquisition, compression and editing of the material. A comprehensive review of digital forensics methods can be found in [2] and [3] for approaches related to videos and images respectively.

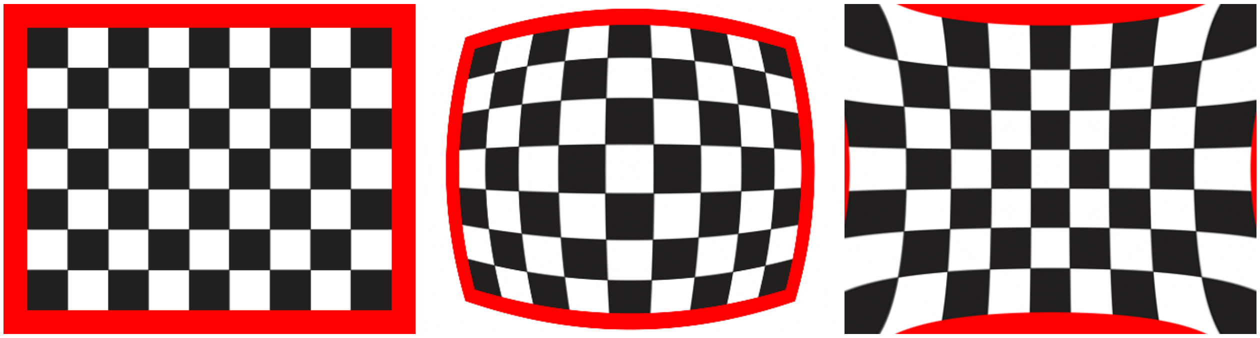
However, one of the easiest methods to elude forensics analysis consists in recapturing forged images either from a printout or directly from the LCD screen. In these cases, the sensitive numerical information relative to the most common footprints used for tampering detection such JPEG compression coefficients, Photo Response Non-Uniformity (PRNU) and Colour Filter Array (CFA) artifacts [3] are destroyed during the recapture process, thus fooling automatic algorithms. Moreover, whenever due care is taken during the recapture [4], it has been shown that human observers are in general poor at differentiating between originals and recaptured images [5]. Recapture detection is therefore essential as an indicator of prior tampering activity, coming in aid to the forensic analyst whenever both footprint detectors and human observers might be ineffective.

Existing approaches for recapture detection have been proposed for both video and images. On video sequences, recaptured videos are identified from estimation of non-zero skew parameters in the camera intrinsic matrix [6], we note that the proposed method requires for accuracy a geometric configuration where the camera is off-axis relative to the target display. In [7], combing artifacts introduced by interlaced scanning are exploited as recapture footprints, but can be avoided altogether by recapture with modern progressive devices. In [8] ghosting artifacts due to the phase difference between original video and camera recapture are used as a cue for classification of recaptured videos from LCD screens. More recently, the work in [9] proposes a method to detect video recapture by identifying the high-frequency jitter introduced by physiological tremor and other environmental factors.

Regarding image recapture, several works have focused on forensic analysis of scanned prints, exploiting colour, sensor noise features and specularities [10], [11], [12]. Other methods have turned their attention to detecting recapture directly from LCD screens: in [5], a combination of colour features, wavelet statistics and local binary patterns is fed into a SVM classifier for recapture detection. The method in [13] is able to recover the acquisition history from the edges of an image after recapture, but it requires a database of acquisition devices.

In this work we examine a novel cue for automatic video recapture detection based on the radial lens distortion of primitive image features. In particular, our contribution is twofold: first, we provide a mathematical expression for the shape of straight lines in space after 2 consecutive chains of spatial transformation, perspective projection and radial distortion, outlining a procedure to extend it to N consecutive chains. Based on our theoretical modelling, we also develop a practical detector and classifier able to discriminate between original and recaptured images without any prior information

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
???-?-????-????-?/10/$??.??



(a) (b) (c)

Fig. 1: Effects of first-order radial distortion. (a) Original grid. (b) Barrel ($k_0 > 0$) and (c) pin-cushion distortions ($k_0 < 0$).

about the devices. The strength of the proposed approach lies in its generality as it does not assume particular recapture conditions (e.g. handheld or off-axis cameras) or display devices (interlaced screens or LCD). The method is applied on a per-frame basis, and can be therefore applied to individual images as well, provided enough features are present.

Our paper is organised as follows: in Section II, we derive the expression for distorted edges after consecutive recaptures. In Section III, the pipeline of our edge classifier is described, while in Section IV, results are presented. We then conclude and outline avenues for further research in Section V.

II. Mathematical modelling of recapture chains

The effect of lens distortion on images has been thoroughly analysed in many computer vision and image processing articles [14], [15]. Furthermore, several works have used lens-based footprints, but contemplating exclusively the scenarios of camera identification [16], [17], [18] and video splicing detection [19]. However, the composite effect of multiple lens distortions on geometric primitives to detect multiple recapture is yet to be characterised.

Generally, lens distortion is a nonlinear operation shifting all projected elements with cooordinates (x, y) by $(\delta(x), \delta(y))$, where:

$$\begin{cases} \delta(x) = x(k_0 r^2 + k_1 r^4 + \cdots) \\ \qquad + (p_0(r^2 + 2x^2) + 2p_1 xy)(1 + q_0 r^2 + q_1 r^4 + \cdots), \\ \delta(y) = y(k_0 r^2 + k_1 r^4 + \cdots) \\ \qquad + (p_1(r^2 + 2y^2) + 2p_0 xy)(1 + q_0 r^2 + q_1 r^4 + \cdots). \end{cases} \tag{1}$$

In Eq. (1) above, $r = \sqrt{x^2 + y^2}$ is the pixel distance from the center of the image plane, k_i are the coefficients of radial distortion, while p_i, q_i are the coefficients affecting the amount of decentering distortion [20]. Generally, different reports in the literature [14][21] agree on assuming the distortion function to be dominated by the low-order radial distortion coefficients, i.e. $(p_i, q_i) = 0, \forall i \in \mathbb{N}$, $k_i = 0, \forall i \geq 1, i \in \mathbb{N}$. In Fig. 1, we show the effect of a first-order radial distortion model (i.e. $k_i = 0, \forall i \geq 1, i \in \mathbb{N}$) on a calibration grid image.

In order to develop a footprint detector for image recapture, we will now derive the effect of radial distortion in recapture chains on geometric primitives such as straight lines. Without any loss of generality, starting from a horizontal 3D line with coordinates $(X, Y, Z) = (s, 0, 0), -\infty < s < \infty$, rotated and translated in space by a 3×3 matrix $\mathbf{R}$ and a 3×1 vector $\mathbf{t}$ respectively, the line's coordinates (X', Y', Z') after the spatial transform can be expressed as:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \mathbf{R} \begin{pmatrix} s \\ 0 \\ 0 \end{pmatrix} + \mathbf{t} = \begin{pmatrix} \alpha_x s + \beta_x \\ \alpha_y s + \beta_y \\ \alpha_z s + \beta_z \end{pmatrix}, \tag{2}$$

where $(\alpha_x, \alpha_y, \alpha_z)$ and $(\beta_x, \beta_y, \beta_z)$ are parameters given by the rotation and translation respectively. A pin-hole perspective projection is then applied to the rotated line, projecting its points to the ideal camera coordinates (x', y'):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f \begin{pmatrix} \frac{X'}{Z'} \\ \frac{Y'}{Z'} \end{pmatrix} = f \begin{pmatrix} \frac{\alpha_x s + \beta_x}{\alpha_z s + \beta_z} \\ \frac{\alpha_y s + \beta_y}{\alpha_z s + \beta_z} \end{pmatrix}, \tag{3}$$

where f is the camera focal length. The distorted coordinates $(\hat{x}, \hat{y})$ are then obtained by adding the distortion $\delta(x')$ and $\delta(y')$ from Eq. (1) to the projected coordinates (x', y'):

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x' \left(1 + \delta(x')\right) \\ y' \left(1 + \delta(y')\right) \end{pmatrix}. \tag{4}$$

Finally, the observed pixel coordinates (u, v) are obtained through an affine transformation of the distorted image coordinates $(\hat{x}, \hat{y})$ which are modified according to the camera's intrinsic parameters:

$$\begin{cases} u = f\hat{x} + \lambda_0 \hat{y} + u_0, \\ v = \lambda_1 f \hat{y} + v_0, \end{cases} \tag{5}$$

where λ_0 and λ_1 are the pixel skew parameter and aspect ratio respectively, while (u_0, v_0) are the pixel coordinates of the optical center. In all our following expressions, we assume that $\lambda_1 = 1$ for convenience, as it does not substantially change our conclusions. Under the reasonable assumption that on standard cameras the pixel skew can be neglected (i.e. $\lambda_0 = 0$), we can substitute in the expression for (x', y') in Eq. (5):

$$\begin{cases} u - u_0 = fx'\left(1 + \delta\left(x'\right)\right) = f\dfrac{\alpha_x s + \beta_x}{\alpha_z s + \beta_z}\left(1 + \delta\left(x'\right)\right), \\ v - v_0 = fy'\left(1 + \delta\left(y'\right)\right) = f\dfrac{\alpha_y s + \beta_y}{\alpha_z s + \beta_z}\left(1 + \delta\left(y'\right)\right). \end{cases} \tag{6}$$

The significance of the expressions in Eq. (6) consists in the fact that the observed coordinates (u, v) along a distorted line can be described by a rational polynomial in s of degree 3 assuming a first-order radial distortion model. However, by considering the recapture of a 2D distorted line defined as per Eq. (5), and displayed on a plane related to the recapture device by a rotation and translation $\mathbf{R}$ and $\mathbf{t}$ respectively, we obtain again the 3D coordinates (X', Y', Z') of the line after spatial transform:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \mathbf{R}\begin{pmatrix} f\left(1 + \delta\left(x'\right)\right)\frac{\alpha_x s + \beta_x}{\alpha_z s + \beta_z} + u_0 \\ f\left(1 + \delta\left(y'\right)\right)\frac{\alpha_y s + \beta_y}{\alpha_z s + \beta_z} + v_0 \\ 0 \end{pmatrix} + \mathbf{t}. \tag{7}$$

By repeating the same procedure followed earlier, we can derive the counterpart of Eq. (6) for the recapture case. In the expression below, we define $\hat{\delta}_i$ to be the distortion element in the i^{th} recapture scaled by the focal length and the perspective foreshortening term, which in the case of the first capture is equivalent to $\hat{\delta}_1(\cdot) = f\frac{1+\delta(\cdot)}{\alpha_z s + \beta_z}$:

$$\begin{cases} u - u_0 = f_2\hat{\delta}_2\left(x'\right)\left(f_1\hat{\delta}_1\left(x'\right)\left(\mu_x s + \nu_x\right) + \xi_x\right), \\ v - v_0 = f_2\hat{\delta}_2\left(y'\right)\left(f_1\hat{\delta}_1\left(y'\right)\left(\mu_y s + \nu_y\right) + \xi_y\right). \end{cases} \tag{8}$$

In the expression above, μ, ν and ξ are unknown parameters resulting from the initial spatial transform, and f_i is the focal length of the camera at the i^{th} recapture stage. It can be seen that unless $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$, Eq. (8) is a rational polynomial of degree at least 9 in s assuming a first-order radial distortion function. This can be seen by considering that the second distortion $\hat{\delta}_2$ is performed on already distorted coordinates, i.e. $\hat{\delta}_2$ is a function of $\hat{\delta}_1$, and the two distortions are further compounded together as indicated by the multiplication of the two distortion factors. By iterating the same process, the polynomial will rapidly increase in degree as each stage compounds the previous distortion.

III. A practical detector for recaptured videos

The main message from the previous section lies in the demonstration that consecutive recapture will introduce distortions of increasingly high order. Therefore, any state-of-the-art method for distortion correction assuming a reasonably low distortion order will not succeed in completely restoring the original straight lines. We exploit this error in our practical algorithm by extracting distorted straight lines from incoming frames, and testing them for compliance with a low order distortion model. However, because of the strong distortion, traditional line detection methods such as the Hough transform are bound to fail, as they tend not to be robust to deviations from perfectly straight segments. On the other hand, the problem arises of automatically distinguishing a distorted straight line from a genuine curved line.

Our approach is based on the idea of detecting continuous smooth segments that exhibit small deviations from a straight line. In Fig. 2, the pipeline for our proposed approach is shown. Given a frame from an input video sequence, the edges are first detected and converted into a binary mask. While a Hough transform will not be able to identify entire distorted segments, these can be approximated in a piecewise fashion through a set of straight lines. The Hough transform is therefore applied to the edge map and the Hough lines are then clustered together, so that multiple lines with similar coefficients describing the same curved edge can be associated to the same cluster. Lines from the clusters are then merged together by considering the intersections of the lines from each cluster with the original edge map. The segmented curves are finally used to estimate the overall distortion function for the frame under the assumption of a first-order radial distortion model, which is then used to obtain a corrected version of the image. Deviations between the first- and higher-order orthogonal projections of the lines are then used to provide incremental evidence towards single or double capture.

A. Edge detection

The first task is to detect straight segments in the input frame. Given an input frame I, the image derivatives along the x and y directions are calculated with a 3×3 Scharr filter. Similarly to other works, the computed derivatives are then combined together within a monotonically decreasing gradient function $g(x, y), g \in [0, 1]$, which tends to 1 for smooth areas and to 0 for perfect edges:

$$g(x, y) = \frac{1}{1 + \gamma\left(\left(\dfrac{dI}{dx}\right)^2 + \left(\dfrac{dI}{dy}\right)^2\right)}, \tag{9}$$

where the constant γ was empirically set to $\frac{1}{50}$. The detected edges are then thresholded in order to preserve only areas of strong gradient in the image. The threshold is set manually for the first frame in the sequence, and then kept constant for the remainder of the video input, although an extension to vary it automatically to preserve an initial quantity of detected edges in the image would be straightforward to implement.

The output of this stage is a binary map of detected edges. The next step consists of isolating distorted straight lines from genuinely curved segments. To this end, the Hough transform of the binary edge map is computed in order to identify straight segments within the detected edges. Hough lines with little support corresponding to short straight segments are automatically filtered out. The reasoning stems from the observation that for conventional capture devices the deformation will be mild and warped straight lines will still be roughly straight and suitable to be approximated with a limited number of piecewise straight segments from the Hough transform. On the other hand, curved lines will exhibit a much higher

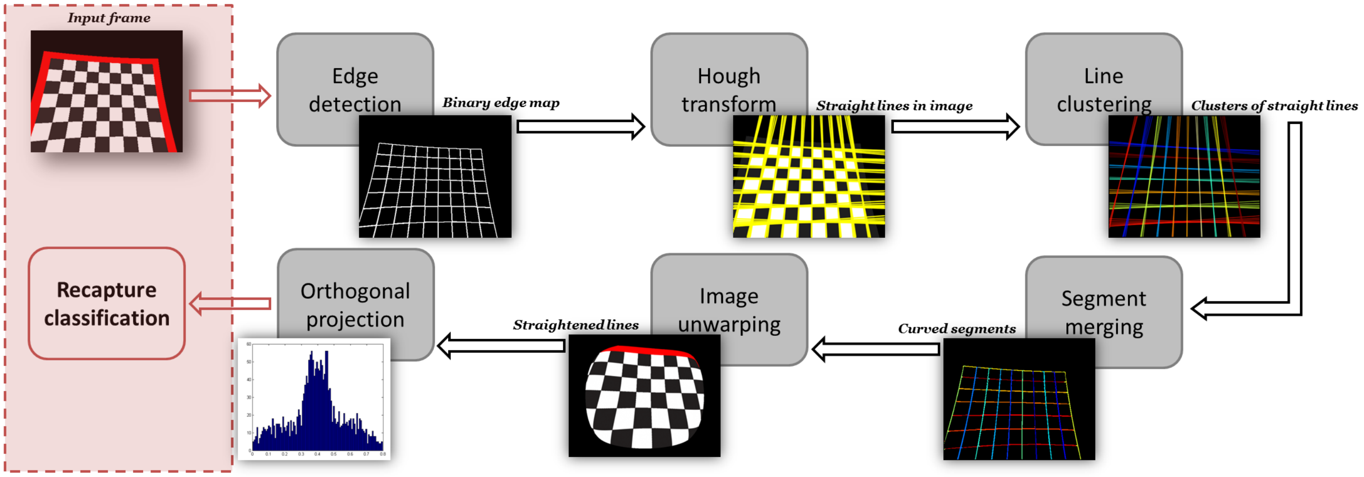


Fig. 2: Workflow for the proposed video recapture detector.

curvature, thus will require a higher number of Hough lines for approximation, each with a lower support in the image.

B. Distorted line identification

Hough lines belonging to the same segment will be characterised by similar linear coefficients. Indeed, even Hough lines belonging to parallel edges will be characterised by an identical slope but distinct y-intercepts. The discrepancies in the slope and y-intercepts are used to automatically cluster Hough lines into groups describing each separate edge in the binary map. The Hough line coefficients are first calculated and the lines separated in two groups depending on whether the line intersects first with the x or y axis. This first grouping is carried out in order to have a better posed clustering initialisation. Lines within the two groups are then clustered using K-means, where each cluster defines the lines belonging to a curved edge. The clusters from a distorted synthetic calibration grid are shown in different colours in Fig. 2.

Identification of the curved lines is then carried out by considering the intersections of the lines in each cluster with the initial binary edge map. This is necessary since we are interested in finding which portion of the curved segment each Hough line in the cluster represents. The interesctions are detected within a small user-defined tolerance threshold which in our experiments has been set to ± 2 pixels. The identified intersection points are then filtered to eliminate disconnected points from spurious intersections of the clustered Hough lines with other edges in the edge map. All the intersection points assigned to a cluster are then labelled as a single curved segment and used for distortion estimation and classification.

C. Distortion estimation

Depending on the number of edges present in each frame, the steps outlined above might need to be repeated for several consecutive frames before a sufficient number of edges is accumulated for estimating the lens radial distortion function. Given a large enough set of N curved segments $\mathbf{l}_i = (\mathbf{x}_i, \mathbf{y}_i), i \in [0, \cdots, N]$ from the previous stage of the algorithm, these are used for an automatic estimation of the lens radial distortion coefficients k_j according to the method proposed in [22].

In light of the considerations made in Section II and [14], [21], a first-order radial distortion model is imposed within the optimisation scheme. The method minimises the following function to find the distortion coefficient $\hat{k}_0$:

$$E(\hat{k}_0) = \frac{1}{N} \sum_{i=0}^{N} \left(S^i_{xx} S^i_{yy} - \left(S^i_{xy}\right)^2 \right), \tag{10}$$

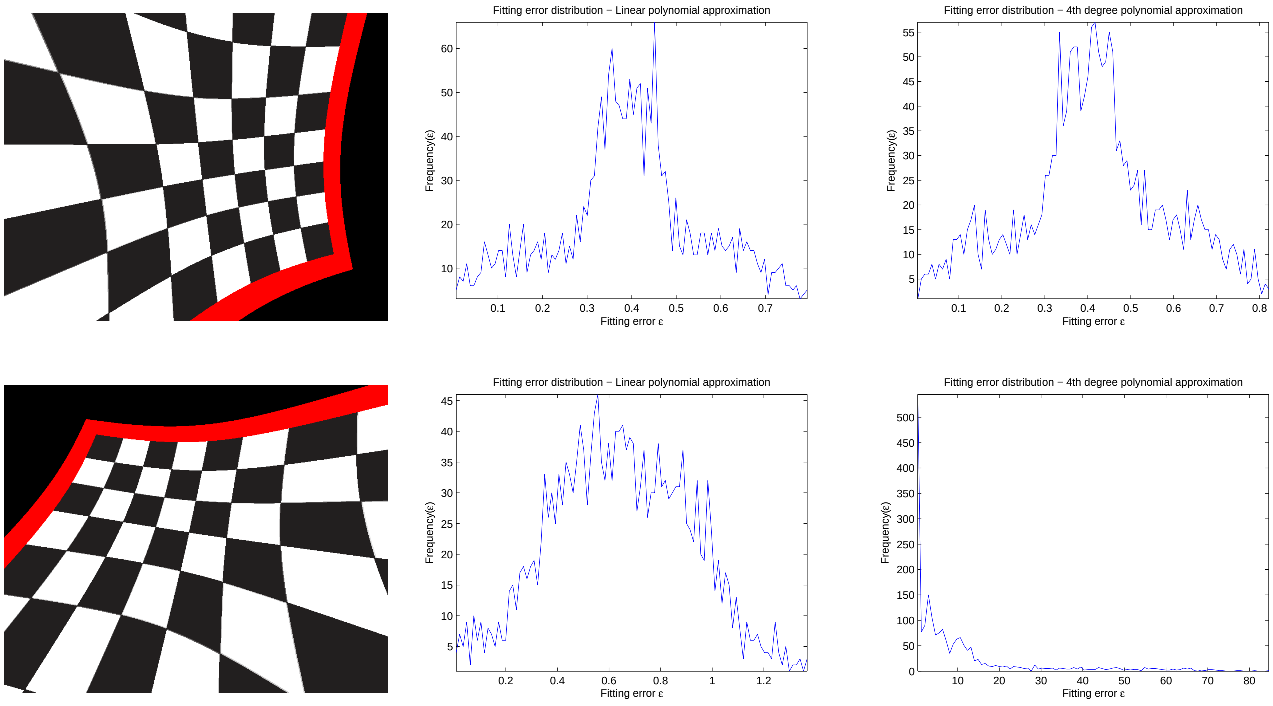
where the S^i elements are the covariance matrix entries for the i^{th} line calculated from the coordinates of the points on the line corrected according to the current estimate $\hat{k}_0$. The function in (10) has been proven to reach its minimum whenever the corrected points lie along a straight line. For more details, please refer to [22].

The estimated radial distorted coefficients are then used in conjunction with the known cordinates of the distorted segments in order to recover the corrected segments coordinates $\mathbf{l}^*_i = (\mathbf{x}^*_i, \mathbf{y}^*_i)$ and passed onto the final stage of the classification scheme.

D. Automatic classification

The fundamental idea behind the classification phase is that the first-order distortion model assumed in the energy minimisation in (10) is sufficient to describe the distortion behaviour under single capture but unsuitable for the high-order nonlinearities introduced during recapture. As a consequence, we expect the detected lines after undistortion with the estimated parameters to be closer to perfectly straight edges in the single capture case rather than in recaptured sequences.

In our algorithm, we test the compliance of the corrected lines $\mathbf{l}^*_i = (\mathbf{x}^*_i, \mathbf{y}^*_i)$ both with first and higher-order polynomials. Generally this can be accomplished by projecting the vector $\mathbf{y}^*_i$ onto the subspace spanned by the vectors ${\mathbf{x}^*_i}^M, {\mathbf{x}^*_i}^{M-1}, \cdots, \mathbf{1}$, where M is the order of the polynomial for the fitting. By defining the $N \times M$ matrix $\mathbf{A}$ as the stacked



(a)

(b)

(c)

Fig. 3: Automatic classification. Top and bottom: (a) Frames from single capture and recaptured sequences. Fitting error distribution with a (b) linear and (c) 4^{th} degree polynomial fitting for single capture and recaptured frames.

$\mathbf{x}_i^{*M}, \mathbf{x}_i^{*M-1}, \cdots, \mathbf{1}$ vectors, the $M \times 1$ vector $\mathbf{c}_i$ containing the polynomial coefficients can be calculated as:

$$\mathbf{c}_i = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}_i^*. \qquad (11)$$

For the projection, we only consider portions of the detected curves away from the image boundaries in order to avoid excessive curvature artifacts due to the undistortion algorithm. Once all coefficients are estimated for all the corrected lines and for both first and higher order polynomial approximations, the distributions of the fitting errors are computed for both approximation orders. For single capture cases, high-order coefficients will be negligible and the error distributions will be very similar. On the other hand, significant differences will be observed in the error distributions from recaptured sequences, as it is shown for simulated data in Fig. 3 (b) and (c). Such discrepancies are measured by calculating the Jensen-Shannon divergence between the two orders of error distributions, which has the desirable properties of being symmetric and being bound within the interval $(0, 1)$. Depending on whether the measured divergence is greater than a threshold τ determined offline with training sequences, the input acquisition chain is classified as either single capture or recapture.

IV. Experimental results

The method has been tested on automatically generated synthetic sequences. A virtual rendering environment was created where the single capture sequence depicts a standard calibration grid in 3D at a randomly generated position and orientation. This is projected within the field of view of the virtual camera for each frame. To generate the recaptured sequence, another random 3D translation and rotation was applied to the 2D image plane from the first capture for each frame, and the scene projected again in 2D with a second virtual camera. Both virtual cameras were set according to a first-order radial distortion model.

In our experiment, lines were detected and analysed on a per-frame basis on video sequences consisting of 50 frames. Example frames from the test sequences are shown in Fig. 3 (a). It is interesting to note how despite the strong distortion no appreciable differences between single capture and recaptured frames can be deduced from visual inspection alone. In Fig. 3 (b), the linear polynomial fitting error distributions obtained from all the frames in the sequences are shown. Here, it can be noticed that the distribution of the recaptured sequence, as well as having a higher average error value, presents a distinctly different shape compared to the single capture case. However, due to the limited size of the image domain and the generally low curvature of the segments, these errors alone will be too small for automatic classification.

Regarding the fitting error distributions for higher order polynomials, these are shown in Fig. 3 (c). In the single capture case, the error fitting distribution is virtually identical to the one shown in (b) as the linear approximation is sufficient to capture the distortion coefficients and the high order polynomial coefficients will be close to zero. Conversely, the high-order nonlinearities introduced during the recapture operation will not be corrected during the undistortion phase, as a low-order distortion model is assumed. This will in turn leave nonzero high order polynomial coefficients in the projection, resulting in the discrepancies between the distributions in Fig. 3.

Quantitative evaluation was carried out by 50-fold cross validation for each sequence, corresponding to a classification test for each frame with the classification threshold trained based on the remaining 49 frames. As shown in Fig. 4, the distributions of the JS divergence are perfectly separated between the two sequences, resulting in perfect classification.

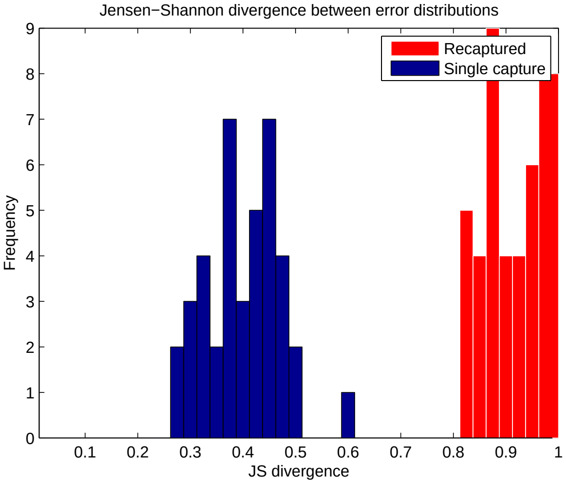


Fig. 4: Distribution of the Jensen-Shannon divergences for single and double capture sequences during cross-validation.

V. Conclusions and further work

In this paper we have proposed a model for automatic recapture of video sequence based on radial distortion artifacts. The effect of recapture chains on geometric primitives such as straight lines was mathematically derived as the introduction of high-order components in the distorted points coordinates. A practical detector and classifier was then proposed to extract curved edge features from images, estimate the lens distortion coefficients and classify the sequence based on its compliance to a low-order lens radial distortion model. Future work will concentrate on fusing this cue with existing video recapture methods and to test it extensively on real video data.

Acknowledgments

This work was supported by the REWIND Project funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme (FP7) of the European Commission, under FET-Open grant number: 268478.

References

[1] F. Pan and J. Huang, “Discriminating computer graphics images and natural images using hidden markov tree model,” in *Digital Watermarking*, 2011, vol. 6526, pp. 23–28.

[2] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, “An overview on video forensics,” *APSIPA Transactions on Signal and Information Processing*, vol. 1, 11 2012.

[3] A. Piva, “An overview on image forensics,” *ISRN Signal Processing*, vol. 2013, pp. 1–22, 2013.

[4] H. Muammar and P. L. Dragotti, “An investigation into aliasing in images recaptured from an lcd monitor using a digital camera,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May To appear.

[5] H. Cao and A. Kot, “Identification of recaptured photographs on lcd screens,” in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010, pp. 1790–1793.

[6] W. Wang and H. Farid, “Detecting re-projected video,” in *10th International Workshop on Information Hiding*, Santa Barbara, CA, USA, May 2008, pp. 72–86.

[7] J.-W. Lee, M.-J. Lee, T.-W. Oh, S.-J. Ryu, and H.-K. Lee, “Screenshot identification using combing artifact from interlaced video,” in *12th ACM workshop on Multimedia and Security (MMSEC)*, Rome, Italy, September 2010, pp. 49–54.

[8] P. Bestagini, M. Visentini-Scarzanella, M. Tagliasacchi, P. L. Dragotti, and S. Tubaro, “Video recapture detection based on ghosting artifact analysis,” in *IEEE International Conference on Image Processing (ICIP)*, Under review.

[9] M. Visentini-Scarzanella and P.-L. Dragotti, “Video jitter analysis for automatic bootleg detection,” in *IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, Banff, Canada, September 2012, pp. 101–106.

[10] N. Khanna, G.-C. Chiu, J. Allebach, and E. Delp, “Forensic techniques for classifying scanner, computer generated and digital camera images,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008, pp. 1653–1656.

[11] H. Gou, A. Swaminathan, and M. Wu, “Intrinsic sensor noise features for forensic analysis on scanners and scanned images,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 476–491, September 2009.

[12] H. Yu, T.-T. Ng, and Q. Sun, “Recaptured photo detection using specularity distribution,” in *15th IEEE International Conference on Image Processing (ICIP)*, San Diego, CA, USA, October 2008, pp. 3140–3143.

[13] T. Thongkamwitoon, H. Muammar, and P.-L. Dragotti, “Identification of image acquisition chains using a dictionary of edge profiles,” in *20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August 2012, pp. 1757–1761.

[14] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, November 2000.

[15] A. Fitzgibbon, “Simultaneous linear estimation of multiple view geometry and lens distortion,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Kauai, HI, USA, June 2001, pp. 125–132.

[16] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, “Automatic source camera identification using the intrinsic lens radial distortion,” *Opt. Express*, vol. 14, no. 24, pp. 11 551–11 565, Nov 2006.

[17] A. Dirik, H. Sencar, and N. Memon, “Digital single lens reflex camera identification from traces of sensor dust,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 539–552, September 2008.

[18] L. T. Van, S. Emmanuel, and M. Kankanhalli, “Identifying source cell phone using chromatic aberration,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Beijing, China, July 2007, pp. 883–886.

[19] H. R. Chennamma and L. Rangarajan, “Image splicing detection using inherent lens radial distortion,” *arXiv:1105.4712*, 2011.

[20] J. Salvi, X. Armangue, and J. Batlle, “A comparative review of camera calibrating methods with accuracy evaluation,” *Pattern Recognition*, vol. 35, no. 7, pp. 1617 – 1635, 2002.

[21] R. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[22] L. Alvarez, L. Gomez, and J. Sendra, “An algebraic approach to lens distortion by line rectification,” *Journal of Mathematical Imaging and Vision*, vol. 35, no. 1, pp. 36–50, 2009.