Detecting Re-Projected Video

Weihong Wang and Hany Farid

Department of Computer Science
Dartmouth College
Hanover, NH 03755
www.cs.dartmouth.edu/~{whwang,farid}

Abstract. A common and simple way to create a bootleg video is to simply record a movie from the theater screen. Because the recorded video is not generally of high quality, it is usually easy to visually detect such recordings. However, given the wide variety of video content and film-making styles, automatic detection is less straight-forward. We describe an automatic technique for detecting a video that was recorded from a screen. We show that the internal camera parameters of such video are inconsistent with the expected parameters of an authentic video.

Key words: Digital Forensics, Digital Tampering

1 Introduction

Often only hours after their release, major motion pictures can find their way onto the Internet. A simple and popular way to create such bootleg video is to simply record a movie from the theater screen. Although these video are certainly not of the same quality as their subsequent DVD releases, increasingly compact and high resolution video recorders are affording better quality video recordings.

We describe how to automatically detect a video that was recorded from a screen. Shown in Fig. 1, for example, is a scene from the movie *Live Free Or Die Hard*. Also shown in this figure is the same scene as viewed on a theater screen. Note that due to the angle of the video camera relative to the screen, a perspective distortion has been introduced into this second recording. We show that this re-projection can introduce a distortion into the intrinsic camera parameters (namely, the camera skew which depends on the angle between the horizontal and vertical pixel axes). We leverage previous work on camera calibration to estimate this skew and show the efficacy of this technique to detect re-projected video.

2 Methods

We begin by describing the basic imaging geometry from 3-D world to 2-D image coordinates for both arbitrary points (Section 2.1) and for points constrained

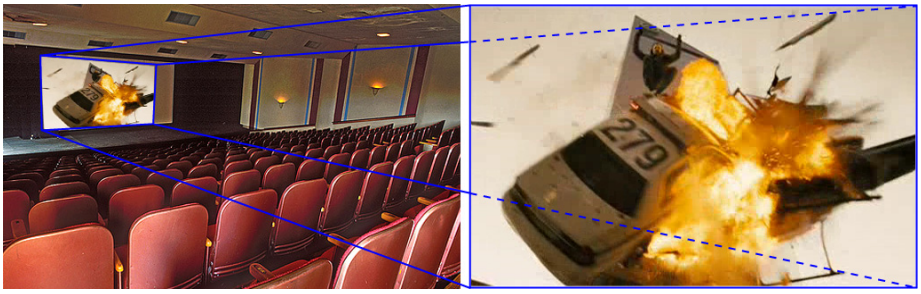


Fig. 1. Shown on the right is a scene from the movie *Live Free Or Die Hard*, and shown on the left is the same scene as viewed on a movie screen. A recording of the projected movie introduces distortions that can be used to detect re-projected video.

to a planar surface (Section 2.2). See [3] for a thorough treatment. We then describe the effect of a re-projection: a non-planar projection followed by a planar projection (Section 2.3). Such a re-projection would result from, for example, video recording the projection of a movie.

2.1 Projective Geometry: non-planar

Under an ideal pinhole camera model, the perspective projection of arbitrary points $\boldsymbol{X}$ (homogeneous coordinates) in 3-D world coordinates is given by:

$$\boldsymbol{x} = \lambda K M \boldsymbol{X}, \tag{1}$$

where $\boldsymbol{x}$ is the 2-D projected point in homogeneous coordinates, λ is a scale factor, K is the intrinsic matrix, and M is the extrinsic matrix.

The 3×3 intrinsic matrix K embodies the camera's internal parameters:

$$K = \begin{pmatrix} \alpha f & s & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2}$$

where f is the focal length, α is the aspect ratio, (c_x, c_y) is the principle point (the projection of the camera center onto the image plane), and s is the skew (the skew depends on the angle, θ, between the horizontal and vertical pixel axes: $s = f \tan(\pi/2 - \theta)$). For simplicity, we will assume square pixels ($\alpha = 1$, $s = 0$) and that the principal point is at the origin ($c_x = c_y = 0$) – these are reasonable assumptions for most modern-day cameras. With these assumptions, the intrinsic matrix simplifies to:

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

The 3×4 extrinsic matrix M embodies the transformation from world to camera coordinates:

$$M = (R \mid \boldsymbol{t}), \tag{4}$$

where R is a 3×3 rotation matrix, and $\boldsymbol{t}$ is a 3×1 translation vector.

2.2 Projective Geometry: planar

Under an ideal pinhole camera model, the perspective projection of points $\boldsymbol{Y}$ constrained to a planar surface in world coordinates is given by:

$$\boldsymbol{y} = \lambda K P \boldsymbol{Y}, \tag{5}$$

where $\boldsymbol{y}$ is the 2-D projected point in homogeneous coordinates, and $\boldsymbol{Y}$, in the appropriate coordinate system, is specified by 2-D coordinates in homogeneous coordinates. As before, λ is a scale factor, K is the intrinsic matrix, and P is the extrinsic matrix. The intrinsic matrix K takes the same form as in Equation (3). The now 3×3 extrinsic matrix P takes the form:

$$P = (\boldsymbol{p}_1 \ \boldsymbol{p}_2 \ \boldsymbol{t}), \tag{6}$$

where $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_1 \times \boldsymbol{p}_2$ are the columns of the 3×3 rotation matrix that describes the transformation from world to camera coordinates, and as before, $\boldsymbol{t}$ is a 3×1 translation vector.

2.3 Re-Projection

Consider now the effect of first projecting arbitrary points in 3-D world coordinates into 2-D image coordinates, and then projecting these points a second time. As described in Section 2.1, the first projection is given by:

$$\boldsymbol{x} = \lambda_1 K_1 M_1 \boldsymbol{X}. \tag{7}$$

The second planar projection, Section 2.2, is given by:

$$\boldsymbol{y} \ = \ \lambda_2 K_2 P_2 \boldsymbol{x} \ = \ \lambda_2 K_2 P_2 \left(\lambda_1 K_1 M_1 \boldsymbol{X} \right) \ = \ \lambda_2 \lambda_1 K_2 P_2 \left(K_1 M_1 \boldsymbol{X} \right). \tag{8}$$

The effective projective matrix $K_2 P_2 K_1 M_1$ can be uniquely factored (see Appendix A) into a product of an intrinsic, K, and extrinsic, M, matrix:

$$\boldsymbol{y} = \lambda_2 \lambda_1 \lambda K M \boldsymbol{X}. \tag{9}$$

Recall that we assumed that the camera skew (the $(1,2)$ entry in the 3×3 intrinsic matrix, Equation (2), is zero. We next show that that a re-projection can yield a non-zero skew in the intrinsic matrix K. As such, significant deviations of the skew from zero in the estimated intrinsic matrix can be used as evidence that a video has been re-projected.

We have seen that the re-projection matrix $K_2 P_2 K_1 M_1$ can be factored into a product of a scale factor and intrinsic and extrinsic matrices:

$$K_2 P_2 K_1 M_1 = \lambda K M. \tag{10}$$

Expressing each 3×4 extrinsic matrix M_1 and M in terms of their rotation and translation components yields:

$$\begin{aligned} K_2 P_2 K_1 (R_1 \mid \boldsymbol{t}_1) &= \lambda K (R \mid \boldsymbol{t}) \\ K_2 P_2 K_1 R_1 &= \lambda K R. \end{aligned} \tag{11}$$

Reshuffling[1] a few terms yields:

$$\begin{aligned} K^{-1} K_2 P_2 K_1 R_1 &= \lambda R \\ K^{-1} K_2 P_2 K_1 &= \lambda R R_1^T. \end{aligned} \tag{12}$$

Note that the right-hand side of this relationship is an orthogonal matrix – this will be exploited later. On the left-hand side, the left-most matrix is the inverse of the effective intrinsic matrix in Equation (2):

$$K^{-1} = \begin{pmatrix} \frac{1}{\alpha f} & -\frac{s}{\alpha f^2} & \frac{s c_y - c_x f}{\alpha f^2} \\ 0 & \frac{1}{f} & -\frac{c_y}{f} \\ 0 & 0 & 1 \end{pmatrix}. \tag{13}$$

And the product of the next three matrices is:

$$\begin{aligned} K_2 P_2 K_1 &= \begin{pmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_{11} & p_{21} & t_1 \\ p_{12} & p_{22} & t_2 \\ p_{13} & p_{23} & t_3 \end{pmatrix} \begin{pmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ &= \begin{pmatrix} f_1 f_2 p_{11} & f_1 f_2 p_{21} & f_2 t_1 \\ f_1 f_2 p_{12} & f_1 f_2 p_{22} & f_2 t_2 \\ f_1 p_{13} & f_1 p_{23} & t_3 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{q}_1^T \\ \boldsymbol{q}_2^T \\ \boldsymbol{q}_3^T \end{pmatrix}, \end{aligned} \tag{14}$$

where f_2 and f_1 are the focal lengths of the original projections, p_{1i} and p_{2i} correspond to the i^{th} element of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and t_i corresponds to i^{th} element of $\boldsymbol{t}_2$ (the third column of matrix P_2). The product of the four matrices on the left-hand side of Equation (12) is then:

$$K^{-1} K_2 P_2 K_1 = \begin{pmatrix} \left(\frac{1}{\alpha f} \boldsymbol{q}_1 - \frac{s}{\alpha f^2} \boldsymbol{q}_2 + \frac{s c_y - c_x f}{\alpha f^2} \boldsymbol{q}_3 \right)^T \\ \left(\frac{1}{f} \boldsymbol{q}_2 - \frac{c_y}{f} \boldsymbol{q}_3 \right)^T \\ \boldsymbol{q}_3^T \end{pmatrix}. \tag{15}$$

[1] Since the matrix R_1 is orthonormal $R_1^{-1} = R_1^T$.

Recall that $K^{-1}K_2P_2K_1 = \lambda RR_1^T$, Equation (12), and that R and R_1^T are each orthonormal. Since the product of two orthonormal matrices is orthonormal, $K^{-1}K_2P_2K_1$ is orthogonal (the rows/columns will not be unit length when $\lambda \neq 1$). This orthogonality constrains the above matrix rows as follows:

$$\boldsymbol{q}_3^T \left(\frac{1}{f}\boldsymbol{q}_2 - \frac{c_y}{f}\boldsymbol{q}_3 \right) = 0 \tag{16}$$

$$\left(\frac{1}{f}\boldsymbol{q}_2 - \frac{c_y}{f}\boldsymbol{q}_3 \right)^T \left(\frac{1}{\alpha f}\boldsymbol{q}_1 - \frac{s}{\alpha f^2}\boldsymbol{q}_2 + \frac{sc_y - c_x f}{\alpha f^2}\boldsymbol{q}_3 \right) = 0. \tag{17}$$

Solving Equation (16) for c_y yields:

$$c_y = \frac{\boldsymbol{q}_3^T \boldsymbol{q}_2}{\|\boldsymbol{q}_3\|^2}. \tag{18}$$

Substituting for c_y into Equation (17), followed by some simplifications, yields:

$$s = f\frac{\boldsymbol{q}_2^T \boldsymbol{q}_1 \|\boldsymbol{q}_3\|^2 - (\boldsymbol{q}_3^T \boldsymbol{q}_2)(\boldsymbol{q}_3^T \boldsymbol{q}_1)}{\|\boldsymbol{q}_2\|^2\|\boldsymbol{q}_3\|^2 - (\boldsymbol{q}_3^T \boldsymbol{q}_2)^2}. \tag{19}$$

Note that the skew, s, is expressed only in terms of the effective focal length f, the pair of intrinsic matrices K_1 and K_2, and the second transformation matrix P_2. We can now see under what conditions $s = 0$.

First, note that the denominator of Equation (19) cannot be zero. If $\|\boldsymbol{q}_2\|^2\|\boldsymbol{q}_3\|^2 - (\boldsymbol{q}_3^T \boldsymbol{q}_2)^2 = 0$ then, $\boldsymbol{q}_2 \propto \boldsymbol{q}_3$, in which case $K_2P_2K_1$ is singular, which it cannot be, since each matrix in this product is full rank. And, since $f \neq 0$, the skew is zero only when the numerator of Equation (19) is zero:

$$\begin{aligned} \boldsymbol{q}_2^T \boldsymbol{q}_1 \|\boldsymbol{q}_3\|^2 - (\boldsymbol{q}_3^T \boldsymbol{q}_2)(\boldsymbol{q}_3^T \boldsymbol{q}_1) &= 0 \\ f_1^2 p_{31}p_{32} - t_1t_2 + p_{33}^2 t_1t_2 + p_{31}p_{32}t_3^2 - p_{32}p_{33}t_1t_3 - p_{31}p_{33}t_2t_3 &= 0, \end{aligned} \tag{20}$$

where p_{3i} is the i^{th} element of $\boldsymbol{p}_3 = \boldsymbol{p}_1 \times \boldsymbol{p}_2$. Although we have yet to geometrically fully characterize the space of coefficients that yields a zero skew, there are a few intuitive cases that can be seen from the above constraint. For example, if the world to camera rotation is strictly about the z-axis, then $p_{31} = p_{32} = 0$ and $p_{33} = 1$, and the skew $s = 0$. This situation arises when the image plane of the second projection is perfectly parallel to the screen being imaged. As another example, if $\boldsymbol{t} = \pm f_1 \boldsymbol{p}_3$, then the skew $s = 0$. This situations arises when the translation of the second projection is equal to the third column of the rotation matrix scaled by focal length of the first projection – a perhaps somewhat unlikely configuration.

Although there are clearly many situations under which $s = 0$, our simulations suggest that under realistic camera motions, this condition is rarely satisfied. Specifically, we computed the skew, Equation (19), from one million randomly generated camera configurations. The relative position of the second camera to the planar projection screen was randomly selected with the rotation in the range $[-45, 45]$ degrees, X and Y translation in the range [-1000,1000],

Z translation in the range $[4000, 6000]$, and focal length in the range $[25, 75]$. The average skew was 0.295, and only 48 of the $1,000,000$ configurations had a skew less than 10^{-5} (in a similar simulation, the estimated skew for a single projection is on the order of 10^{-12}).

2.4 Camera Skew

From the previous sections, we see that re-projection can cause a non-zero skew in the camera's intrinsic parameters. We review two approaches for estimating camera skew from a video sequence. The first estimates the camera skew from a known planar surface, while the second assumes no known geometry.

Skew estimation I: Recall that the projection of a planar surface, Equation (5), is given by:

$$\boldsymbol{y} = \lambda K P \boldsymbol{Y} = \lambda H \boldsymbol{Y}, \tag{21}$$

where $\boldsymbol{y}$ is the 2-D projected point in homogeneous coordinates, and $\boldsymbol{Y}$, in the appropriate coordinate system, is specified by 2-D coordinates in homogeneous coordinates. The 3×3 matrix H is a non-singular matrix referred to as a homography. Given the above equality, the left- and right-hand sides of this homography satisfy the following:

$$\begin{aligned} \boldsymbol{y} \times (H\boldsymbol{Y}) &= \mathbf{0} \\ \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \times \left(\begin{pmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} \right) &= \mathbf{0}. \end{aligned} \tag{22}$$

Note that due to the equality with zero, the multiplicative scalar λ, Equation (21), is factored out. Evaluating the cross product yields:

$$\begin{pmatrix} y_2(h_{13}Y_1 + h_{23}Y_2 + h_{33}Y_3) - y_3(h_{12}Y_1 + h_{22}Y_2 + h_{32}Y_3) \\ y_3(h_{11}Y_1 + h_{21}Y_2 + h_{31}Y_3) - y_1(h_{13}Y_1 + h_{23}Y_2 + h_{33}Y_3) \\ y_1(h_{12}Y_1 + h_{22}Y_2 + h_{32}Y_3) - y_2(h_{11}Y_1 + h_{21}Y_2 + h_{31}Y_3) \end{pmatrix} = \mathbf{0}. \tag{23}$$

This constraint is linear in the unknown elements of the homography h_{ij}. Reordering the terms yields the following system of linear equations:

$$\begin{pmatrix} 0 & 0 & 0 & -y_3Y_1 & -y_3Y_2 & -y_3Y_3 & y_2Y_1 & y_2Y_2 & y_2Y_3 \\ y_3Y_1 & y_3Y_2 & y_3Y_3 & 0 & 0 & 0 & -y_1Y_1 & -y_1Y_2 & -y_1Y_3 \\ -y_2Y_1 & -y_2Y_2 & -y_2Y_3 & y_1Y_1 & y_1Y_2 & y_1Y_3 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{21} \\ h_{31} \\ h_{12} \\ h_{22} \\ h_{32} \\ h_{13} \\ h_{23} \\ h_{33} \end{pmatrix} = \mathbf{0}$$

$$A\boldsymbol{h} = \mathbf{0}. \tag{24}$$

A matched set of points $\boldsymbol{y}$ and $\boldsymbol{Y}$ appear to provide three constraints on the eight unknowns elements of $\boldsymbol{h}$ (the homography is defined only up to an unknown scale

factor, reducing the unknowns from nine to eight). The rows of the matrix, A, however, are not linearly independent (the third row is a linear combination of the first two rows). As such, this system provides only two constraints in eight unknowns. In order to solve for $\boldsymbol{h}$, we require four or more points with known image, $\boldsymbol{y}$, and (planar) world, $\boldsymbol{Y}$, coordinates that yield eight or more linearly independent constraints. From four or more points, standard least-squares techniques, as described in [3, 5], can be used to solve for $\boldsymbol{h}$: the minimal eigenvalue eigenvector of $A^T A$ is the unit vector $\boldsymbol{h}$ that minimizes the least-squares error.

We next describe how to estimate the camera skew from the estimated homography H. This approach is a slightly modified version of [15]. Recall that H can be expressed as:

$$H = KP = K\left(\boldsymbol{p}_1\ \boldsymbol{p}_2 \mid \boldsymbol{t}\right). \tag{25}$$

The orthonormality of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, yields the following two constraints:

$$\boldsymbol{p}_1^T \boldsymbol{p}_2 = 0 \qquad \text{and} \qquad \boldsymbol{p}_1^T \boldsymbol{p}_1 = \boldsymbol{p}_2^T \boldsymbol{p}_2, \tag{26}$$

which in turn imposes the following constraints on H and K:

$$\begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \end{pmatrix}^T K^{-T}K^{-1} \begin{pmatrix} h_{21} \\ h_{22} \\ h_{23} \end{pmatrix} = 0 \tag{27}$$

$$\begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \end{pmatrix}^T K^{-T}K^{-1} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \end{pmatrix} = \begin{pmatrix} h_{21} \\ h_{22} \\ h_{23} \end{pmatrix}^T K^{-T}K^{-1} \begin{pmatrix} h_{21} \\ h_{22} \\ h_{23} \end{pmatrix}. \tag{28}$$

For notational ease, denote $B = K^{-T}K^{-1}$, where B is a symmetric matrix parametrized with three degrees of freedom (see Equation (41) in Appendix B):

$$B = \begin{pmatrix} b_{11} & b_{12} & 0 \\ b_{12} & b_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{29}$$

Notice that by parametrizing the intrinsic matrix in this way, we have bundled all of the anomalies of a double projection into the estimate of the camera skew. Substituting the matrix B into the constraints of Equations (27)-(28) yields the following constraints:

$$\begin{pmatrix} h_{11}h_{21} & h_{12}h_{21} + h_{11}h_{22} & h_{12}h_{22} \\ h_{11}^2 - h_{21}^2 & 2\left(h_{11}h_{12} - h_{21}h_{22}\right) & h_{12}^2 - h_{22}^2 \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{22} \end{pmatrix} = -\begin{pmatrix} h_{13}h_{23} \\ h_{13}^2 - h_{23}^2 \end{pmatrix}. \tag{30}$$

Each image of a planar surface enforces two constraints on the three unknowns b_{ij}. The matrix $B = K^{-T}K^{-1}$ can, therefore, be estimated from two or more views of the same planar surface using standard least-squares estimation. The

desired skew can then be determined (see Appendix B) from the estimated matrix B as:

$$s = -f\frac{b_{12}}{b_{11}}. \tag{31}$$

Note that the estimate of the skew, s, is scaled by the focal length, f. Since a camera's skew depends on the focal length, it is desirable to work with this normalized skew.

Skew estimation II: We showed in the previous section how to estimate a camera's skew from two or more views of a planar surface. This approach has the advantage that it affords a closed-form linear solution, but has the disadvantage that it only applies to frames that contain a known planar surface. Here we review a related approach that does not require any known world geometry, but requires a non-linear minimization.

Consider two frames of a video sequence with corresponding image points given by $\boldsymbol{u}$ and $\boldsymbol{v}$, specified in 2-D homogeneous coordinates. It is well established [3] that these points satisfy the following relationship:

$$\boldsymbol{v}^T F \boldsymbol{u} = 0, \tag{32}$$

where F, the fundamental matrix, is a 3×3 singular matrix ($\text{rank}(F) = 2$). Writing the above relationship in terms of the vector and matrix elements yields:

$$\begin{pmatrix} v_1 & v_2 & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \\ f_{13} & f_{23} & f_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ 1 \end{pmatrix} = 0$$

$$u_1v_1f_{11} + u_2v_1f_{21} + v_1f_{31} + u_1v_2f_{12} + u_2v_2f_{22} + v_2f_{32} + u_1f_{13} + u_2f_{23} + f_{33} = 0.$$

Note that this constraint is linear in the elements of the fundamental matrix f_{ij}, leading to the following system of linear equations:

$$\begin{pmatrix} u_1v_1 & u_2v_1 & v_1 & u_1v_2 & u_2v_2 & v_2 & u_1 & u_2 & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} = 0$$

$$A\boldsymbol{f} = \boldsymbol{0}. \tag{33}$$

Each pair of matched points $\boldsymbol{u}$ and $\boldsymbol{v}$ provides one constraint for the eight unknown elements of $\boldsymbol{f}$ (the fundamental matrix is defined only up to an unknown scale factor reducing the unknowns from nine to eight). In order to solve for the components of the fundamental matrix, $\boldsymbol{f}$, we require eight or more matched pairs of points [6, 2]. Standard least-squares techniques can be used to solve for $\boldsymbol{f}$: the minimal eigenvalue eigenvector of A^TA is the unit vector $\boldsymbol{f}$ that minimizes the least-squares error.

We next describe how to estimate the camera skew from the estimated fundamental matrix F. We assume that the intrinsic camera matrix K, Equation (3),

is the same across the views containing the matched image points. The essential matrix E is then defined as:

$$E = K^T F K. \tag{34}$$

Since F has rank 2 and the intrinsic matrix is full rank, the essential matrix E has rank 2. In addition, the two non-zero singular values of E are equal [4]. This property will be exploited to estimate the camera skew. Specifically, as described in [8], we establish the following cost function to be minimized in terms of the camera focal length f and skew s:

$$C(f, s) = \sum_{i=1}^{n} \frac{\sigma_{i1} - \sigma_{i2}}{\sigma_{i2}}, \tag{35}$$

where σ_{i1} and σ_{i2} are, in descending order, the non-zero singular values of E from n estimated fundamental matrices (each computed from pairs of frames throughout a video sequence), and where K is parametrized as:

$$K = \begin{pmatrix} f & s & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{36}$$

Note that since only the relative differences in the singular values of E are considered, the arbitrary scale factor to which E is estimated does not effect the estimation of the skew. As before, by parametrizing the intrinsic matrix in this way, we have bundled all of the anomalies of a double projection into the estimate of the camera skew. The cost function, Equation (35), is minimized using a standard derivative-free Nelder-Mead non-linear minimization.

3 Results

We report on a set of simulations and sensitivity analysis for each of the skew estimation techniques described in the previous sections. We then show the efficacy of these approaches on a real-video sequence. In each set of simulations we provide the estimation algorithm with the required image coordinates. For the real-video sequence we briefly describe a point tracking algorithm which provides the necessary image coordinates for estimating the camera skew.

3.1 Simulation (skew estimation I):

Recall that a minimum of four points with known geometry on a planar surface viewed from a minimum of two views are required to estimate the camera skew. We therefore randomly generated between 4 and 64 points on a planar surface and generated a video sequence of this stationary surface. In all of the simulations, the first projection was specified by the following camera parameters: the planar surface was 2000 units from the camera, between successive frames the

rotation about each axis was in the range $[-2.5, 2.5]$ degrees and the translation in each dimension was in the range $[-50, 50]$, and the camera focal length was in the range $[25, 75]$ (but fixed for each sequence). For the second projection, the camera was placed a distance of 5000 units from the first projected image and underwent a motion in the same range as the first camera. We randomly generated $10,000$ such sequences as imaged through a single projection, and $10,000$ sequences as imaged through a double projection (re-projection).

In the first simulation, we tested the sensitivity to additive noise. As described above, 30 frames were generated each containing 4 points on a planar surface. Noise in the range of $[0, 1]$ pixels was added to the final image coordinates. The skew was estimated from 15 pairs of frames, where each frame at time t was paired with a frame at time $t + 15$. A sequence was classified as re-projected if one or more of the image pairs yielded an estimated skew greater than 0.1. While this type of voting scheme yields slightly higher false positive rates, it also significantly improves the detection accuracy. In the absence of noise, 0 of the $10,000$ singly projected sequences were classified as re-projected, and 84.9% of the re-projected sequences were correctly classified. With 0.5 pixels of noise, 17.2% of the singly projected sequences were incorrectly classified as re-projected, and 87.4% of the re-projected sequences were correctly classified. Shown in Fig. 2(a) are the complete set of results for additive noise in the range of $[0, 1]$ pixels. Note that even with modest amounts of noise, the false positive rate increases to an unacceptable level. We next show how these results can be improved upon.

In this next simulation, we tested the sensitivity to the number of known points on the planar surface. The noise level was 0.5 pixels, and the number of known points was in the range $[4, 64]$. All other parameters were the same as in the previous simulation. With the minimum of 4 points, 16.2% of the singly projected sequences were incorrectly classified while 87.7% of the re-projected sequences were correctly classified (similar to the results in the previous simulation). With 6 points, only 0.33% of the single projection sequences were incorrectly classified, while the accuracy of the re-projected sequences remained relatively high at 84.6%. Shown in Fig. 2(b) are the complete set of results – beyond 8 points, the advantage of more points becomes negligible.

In summary, from 6 points, with 0.5 pixels noise, in 30 frames, re-projected video can be detected with 85% accuracy, and with 0.3% false positives.

3.2 Simulation (skew estimation II):

Recall that a minimum of eight points viewed from a minimum of two views are required to estimate the camera skew. We therefore generated between 8 and 128 points with arbitrary geometry and generated a video sequence of this stationary cloud of points. In all of the simulations, the first and second projection were generated as described in the previous section. We randomly generated $10,000$ sequences as imaged through a single projection, and $10,000$ sequences as imaged through a double projection (re-projection). As before, a sequence was classified as re-projected if the estimated skew was greater than 0.1.

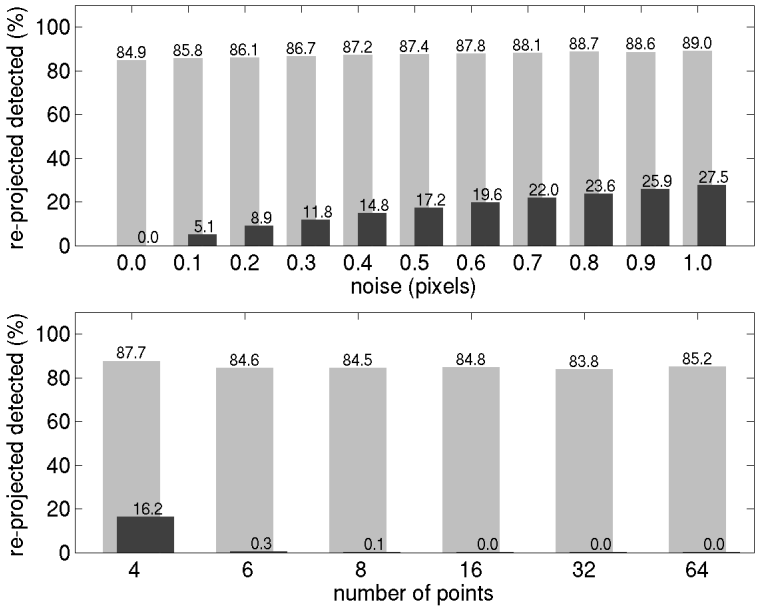


Fig. 2. Skew Estimation I: Detection accuracy (light gray) and false positives (dark gray) as a function of noise (top) and the number of points (bottom).

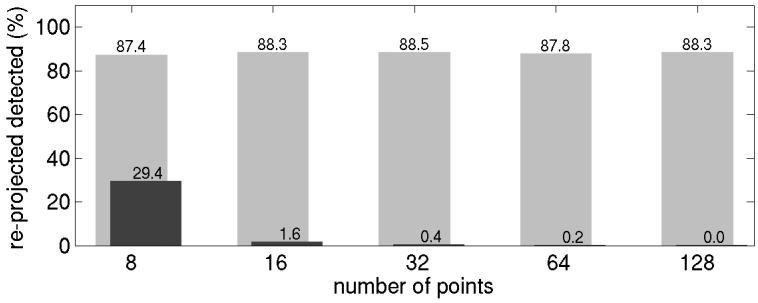
In the first simulation with the minimum of 8 points, 2 frames, and with no noise, 0 of the $10,000$ singly projected sequences were classified as re-projected, and 88.9% of the re-projected sequences were correctly classified. With even modest amounts of noise, however, this minimum configuration yields unacceptably high false positives. We find that the estimation accuracy is more robust to noise when the skew is estimated from multiple frames (i.e., multiple fundamental matrices in Equation (35)). In the remaining simulations, we estimated the skew from 5 fundamental matrices, where each frame t is paired with the frame at time $t + 15$.

In the second simulation, the number of points were in the range $[8, 128]$, with 0.5 pixels of additive noise, and 5 fundamental matrices. With the minimum of 8 points the false positive rate is 29.4%, while with 32 points, the false positive rate falls to 0.4%. In each case, the detection accuracy is approximately 88%. Shown in Fig. 3(a) are the complete set of results for varying number of points.

In the third simulation, the number of points was 32, with 0.5 pixels of noise, and with the number of fundamental matrices (i.e., pairs of frames) in the range $[1, 20]$. As shown in Fig. 3(b), increasing the number of fundamental matrices reduces the false positives while the detection accuracy remains approximately the same.

In summary, from 32 points, with 0.5 pixels noise, in 5 fundamental matrices, re-projected video can be detected with 88% accuracy, and with 0.4% false

(a)



(b)

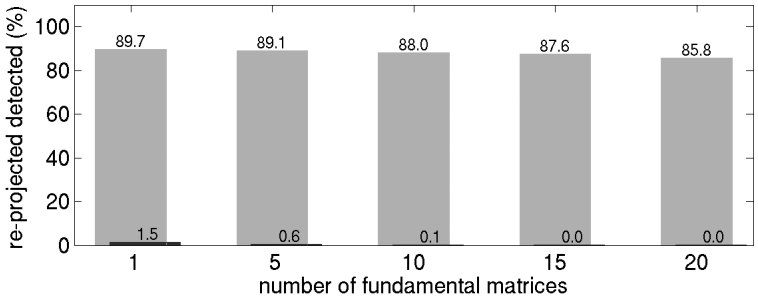


Fig. 3. Skew Estimation II: Detection accuracy (light gray) and false positives (dark gray) as a function of the number of points (top) and the number fundamental matrices (bottom).

positives. This is similar to the accuracy for the skew estimation from points on a planar surface. The advantage here, however, is that this approach does not require known geometry of points on a planar surface.

3.3 Real Video

Shown in Fig. 4 are three frames of a 42 frame segment from the movie *Live Free Or Die Hard*. These frames were digitized at a resolution of 720×304 pixels. Superimposed on each frame are 64 features tracked across all frames. We employed the KLT feature tracker [7, 11, 10] which automatically selects features using a Harris detector, and tracks these points across time using standard optical flow techniques. We manually removed any features with clearly incorrect tracking, and any points not on the buildings or street (the estimation of a fundamental matrix requires points with a rigid body geometry). These tracked features were then used to estimate the skew (method II). The 42 frames were grouped into 21 pairs from which the skew was estimated (each frame at time t was paired with the frame at time $t + 21$). The estimated skew was 0.029, well below the threshold of 0.1.

This 42-frame segment was then displayed on a 20 inch LCD computer monitor with 1600×1200 pixel resolution, and recorded with a Canon Elura video

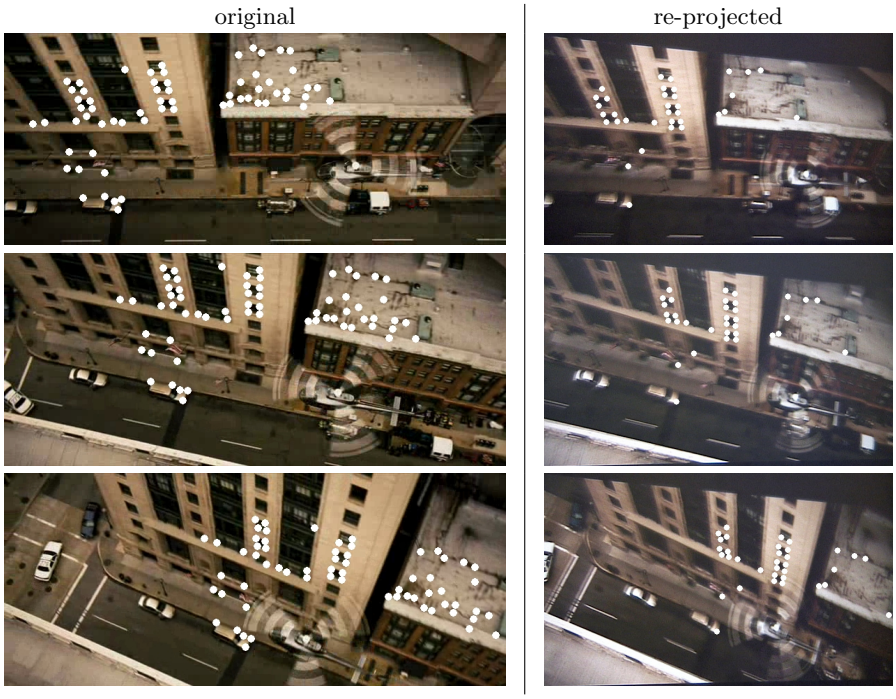


Fig. 4. Shown are the first, middle, and last frame of a 42-frame segment of the movie *Live Free or Die Hard*. On the left is the original digitized video, and on the right is the re-projected video. The white dots denote the tracked features used to estimate the camera skew using method II.

camera at a resolution of 640×480. As above, features were tracked in this video segment, from which the skew was estimated. The estimated skew was 0.25, an order of magnitude larger than the skew from the authentic video and well above our threshold of 0.1.

4 Discussion

We have described how to detect a re-projected video that was recorded directly from a projection on a movie or television screen. We have shown that such a re-projection introduces a skew into the camera's intrinsic parameters, which does not normally occur with authentic video. The camera skew can be estimated from two or more video frames from either four or more points on a planar surface, or eight or more arbitrary points. In addition, from four or more points, the camera skew can be estimated from only a single image. This technique can be applied to detect if a single image has been re-photographed (an often cited technique to circumvent some forensic image analysis, e.g., [9]).

Acknowledgments. This work was supported by a Guggenheim Fellowship, a gift from Adobe Systems, Inc., a gift from Microsoft, Inc., a grant from the National Science Foundation (CNS-0708209), a grant from the U.S. Air Force (FA8750-06-C-0011), and by the Institute for Security Technology Studies at Dartmouth College under grants from the Bureau of Justice Assistance (2005-DD-BX-1091) and the U.S. Department of Homeland Security (2006-CS-001-000001). Points of view or opinions in this document are those of the author and do not represent the official position or policies of the U.S. Department of Justice, the U.S. Department of Homeland Security, or any other sponsor.

References

1. R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *European Conference on Computer Vision*, pages 579–587, 1992.
2. R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
3. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
4. T. S. Huang and O. D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, 1989.
5. M.K. Johnson and H. Farid. Metric measurements on a plane from a single image. Technical Report TR2006-579, Department of Computer Science, Dartmouth College, 2006.
6. H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, (10):133–135, 1981.
7. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
8. P. R. S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Computer Vision and Pattern Recognition*, 1999.
9. A.C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.
10. Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
11. Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.
12. W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *ACM Multimedia and Security Workshop*, 2006.
13. W. Wang and H. Farid. Exposing digital forgeries in interlaced and de-interlaced video. *IEEE Transactions on Information Forensics and Security*, 3(2):438–449, 2007.
14. W. Wang and H. Farid. Exposing digital forgeries in video by detecting duplication. In *ACM Multimedia and Security Workshop*, 2007.
15. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

Appendix A

In this appendix, we show that the product of matrices $K_2P_2K_1M_1$ in Equation (8) can be uniquely factored into a product of an upper triangular matrix (intrinsic) and an orthonormal matrix augmented with a fourth column (extrinsic). We begin by expressing the second extrinsic matrix M_1 in terms of its rotation and translation components:

$$K_2P_2K_1M_1 = K_2P_2K_1\left(R_1 \mid \boldsymbol{t}_1\right). \tag{37}$$

Multiplying R_1 and $\boldsymbol{t}_1$ each by the 3×3 matrix $(K_2P_2K_1)$ yields:

$$K_2P_2K_1M_1 \;=\; \left(K_2P_2K_1R_1 \mid K_2P_2K_1\boldsymbol{t}_1\right) \;=\; \left(K_2P_2K_1R_1 \mid \boldsymbol{t}'\right). \tag{38}$$

Consider now the 3×3 matrix $K_2P_2K_1R_1$. Since each of these matrices is non-singular, their product is non-singular. As such, this matrix can be uniquely factored (within a sign), using RQ-factorization, into a product of an upper triangular, U, and orthonormal, O, matrix:

$$K_2P_2K_1M_1 \;=\; \lambda\left(UO \mid \tfrac{1}{\lambda}\boldsymbol{t}'\right) \;=\; \lambda U\left(O \mid \tfrac{1}{\lambda}U^{-1}\boldsymbol{t}'\right) \;=\; \lambda KM, \tag{39}$$

where $K = U$, $M = (O \mid \frac{1}{\lambda}U^{-1}\boldsymbol{t}')$, and where λ is chosen so that the $(3,3)$ entry of U has unit value. Note that this factorization leads to a product of a scale factor, λ, an intrinsic matrix K, and an extrinsic matrix, M, as in Equation (9).

Appendix B

In this appendix, we show how to determine the skew, s, in Equation(31), from the estimated matrix $B = K^{-T}K^{-1}$. The intrinsic matrix is parametrized as:

$$K = \begin{pmatrix} f & s & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{40}$$

Applying the matrix inverse and multiplication yields:

$$B \;=\; K^{-T}K^{-1} \;=\; \begin{pmatrix} \frac{1}{f^2} & -\frac{s}{f^3} & 0 \\ -\frac{s}{f^3} & \frac{s^2+f^2}{f^4} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{41}$$

from which:

$$-\frac{b_{12}}{b_{11}} = \frac{s}{f}, \tag{42}$$

where b_{ij} denotes the entries of the matrix B, Equation (29).