Exposing Digital Forgeries in Ballistic Motion

Valentina Conotter, James F. O'Brien, and Hany Farid

***Abstract*—We describe a geometric technique to detect physically implausible trajectories of objects in video sequences. This technique explicitly models the three-dimensional ballistic motion of objects in free-flight and the two-dimensional projection of the trajectory into the image plane of a static or moving camera. Deviations from this model provide evidence of manipulation. The technique assumes that the object's trajectory is substantially influenced only by gravity, that the image of the object's center of mass can be determined from the images, and requires that any camera motion can be estimated from background elements. The computational requirements of the algorithm are modest, and any detected inconsistencies can be illustrated in an intuitive, geometric fashion. We demonstrate the efficacy of this analysis on videos of our own creation and on videos obtained from video-sharing web-sites.**

***Index Terms*—Digital Forensics, Video Forensics**

I. Introduction

Increasingly sophisticated video editing and special effects software has made it possible to create forged video sequences that appear to contain realistic dynamic motion. For example, video sharing websites are littered with titles like "Seriously Amazing Best Beer Pong Shots," "Dude Perfect Amazing Basketball Shots," and "Epic Pool Jump." These videos appear to show spectacular basketball shots, gravity-defying acrobatics, and the bone-crushing results of daredevil leaps and jumps. Some of these videos are real, but many are fake. In this paper we describe a forensic technique that is tailored to determine if video of a purportedly ballistic motion, such as a ball being thrown or a person jumping through the air, or a motorcycle soaring off of a ramp, is consistent with the geometry and physics of a free-falling projectile.

Posted videos are often of low-quality and typically have undergone a variety of post-processing and re-compression. As such, statistical techniques that focus on double-compression artifacts [1]–[3], interlaced and de-interlaced correlations [4], or sensor noise patterns [5]–[7] are unlikely to apply. In addition, forensic techniques based on detecting frame or region duplication [8], [9] will only apply when some form of duplication was necessary to create the fake.

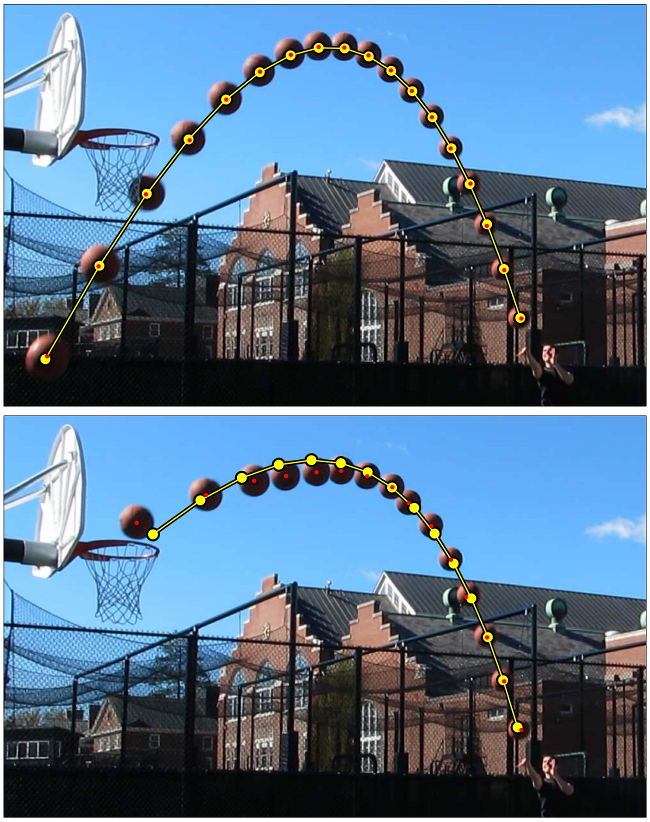


Fig. 1. An authentic (top) and fake video (bottom) each shown as a time-lapse composite. The small red dots specify the tracked position of the ball, and the larger yellow dots and solid lines denote the best fit model of a ballistic trajectory. A discrepancy between the model and the ball's actual positions in the lower image provides proof that the depicted motion is fake.

In contrast to these statistical techniques, we propose a geometric forensic technique that is largely insensitive to resolution, post-processing, compression, and re-compression. We begin by describing a plausible, albeit somewhat simplified, physical model for the expected trajectory of a projectile motion, and a basic imaging model for a static or moving camera. We then describe a technique to determine if the image of the trajectory of a projectile motion is consistent with this physical model. This technique makes minimal assumptions about the nature of the trajectory and camera, and requires only limited manual input. The core computational requirements of the algorithm are modest, and the analysis can be presented as a simple, intuitive geometric construction. We demonstrate the efficacy of this analysis on videos of our own creation and several obtained from video-sharing websites.

II. Related Work

The analysis of projectiles in video has not previously been considered in the forensic community. It has, however, been

Manuscript received March 16, 2011; revised August 14, 2011; accepted August 14, 2011. Date of publication August 22, 2011. This work was supported by gifts from Adobe, Autodesk, Intel, NVIDIA, Microsoft, and Pixar, by a grant from the National Science Foundation (CNS-0708209), by a grant from California Discovery Grant (COM09S- 156646), and by UC Lab Fees Research Program (09-LR-01-118889-OBRJ). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alex ChiChung Kot.

V. Conotter is with the Department of Engineering and Computer Science, University of Trento, Trento, Italy e-mail: conotter@disi.unitn.it.

J. O'Brien is with the Department of Computer Science, University of California, Berkeley, USA, e-mail: job@berkeley.edu.

H. Farid is with the Department of Computer Science, Dartmouth College, NH, USA, e-mail: farid@cs.dartmouth.edu.

Digital Object Identifier 10.1109/TIFS.2011.2165843

addressed in the robotics and computer vision communities. Most notably, in the development of ball-catching robots [10]–[12] and in the analysis of sporting events (*e.g.*, soccer, football, basketball) [13]–[16]. Unlike the approach described here, these techniques either require a stereo camera pair or require additional information about the scene geometry.

A more closely related approach considers the estimation of a projectile's trajectory from a single camera [17]. That approach, however, requires a calibrated camera and is only applicable when the camera is stationary. In contrast, the approach described here does not require a calibrated camera and is applicable to a stationary or moving camera (with unknown camera motion).

Because a projectile's motion is influenced by gravity, the vertical component of acceleration is roughly constant (assuming negligible air resistance or other external forces). In principle a human observer binocularly viewing a projectile could use this information to estimate the absolute size and distance to the projectile [18]. In practice, however, most observers do not consciously take advantage of this information [19]. These studies suggest that if observers are not able to take advantage of such constraints in a three-dimensional setting, then it is most likely that they will not be able to accurately reason about the two-dimensional projection of a projectile. It follows that relying on the visual system to authenticate a projectile's motion becomes even more unlikely when the camera moves along an arbitrary and unknown path.

III. Methods

Our approach to exposing inauthentic motion relies on four explicit assumptions:

1) The motion of the object in three-dimensional space is ballistic (*i.e.,* only gravitational acceleration) so that the trajectory of the object's center of mass will describe a simple parabola.
2) The moving object is sufficiently well-known so that the location of its center of mass can be estimated for each frame.
3) The scene is imaged under linear perspective projection.
4) Any movement or changes in the camera parameters during the sequence can be computed from the scene background so that all frames can be placed into a common coordinate system.

With regard to notation, scalars are denoted with lower-case italic letters s, vectors are denoted with lower-case bold-face letters $\mathbf{v}$, and matrices are denoted with upper-case bold-face letters $\mathbf{M}$. Image coordinates are denoted as $\tilde{\mathbf{p}}$, and world and camera coordinates are denoted as $\mathbf{p}$. Superscripts are used to denote the components of a vector $\mathbf{v} = \begin{bmatrix} v^x & v^y & v^z \end{bmatrix}$ or the rows of a matrix: $\mathbf{m}^k$ is the k^{th} row of matrix $\mathbf{M}$. Subscripts are most commonly used to denote position as a function of time, $\mathbf{p}_\tau$.

A. Trajectory Geometry

The center of mass of a projectile, in the absence of air resistance or any other external forces, follows a ballistic trajectory that can be described in three dimensions with a time-parametrized parabolic curve:

$$\mathbf{p}_\tau = \mathbf{p}_0 + \mathbf{v}\tau + \tfrac{1}{2}\mathbf{a}\tau^2 \tag{1}$$

where τ denotes time, $\mathbf{p}_0$ is the initial position, $\mathbf{v}$ is the initial velocity, and $\mathbf{a}$ is the acceleration (due to gravity). In the absence of any other external forces, the path of the projectile is planar. Assuming linear perspective projection under a pinhole camera model, the image of a projectile's trajectory, in homogeneous coordinates, is:

$$\tilde{\mathbf{q}}_\tau = \mathbf{H}\mathbf{p}_\tau, \tag{2}$$

where $\mathbf{H}$ is a 3×3 matrix embodying the planar perspective projection (*i.e.*, a homography).

Consider the special case where the optical axis of the camera is orthogonal to the plane of motion, the focal length is unit length, and the principal point is the image center. In this case, the z-component of the velocity is zero, and the x- and z-components of the acceleration are zero. The trajectory, Equation (1), then simplifies to:

$$\begin{bmatrix} p^x_\tau \\ p^y_\tau \\ p^z_\tau \end{bmatrix} = \begin{bmatrix} p^x_0 \\ p^y_0 \\ p^z_0 \end{bmatrix} + \begin{bmatrix} v^x \\ v^y \\ 0 \end{bmatrix} \tau + \frac{1}{2} \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \tau^2, \tag{3}$$

where g is gravity and the world coordinate system is defined such that the z-axis is the optical axis and the positive y-axis points upward. In addition, the world to image transformation is simply $\tilde{\mathbf{q}}_\tau = \mathbf{p}_\tau$ (*i.e.*, $\mathbf{H} = \mathbf{I}$). In non-homogeneous coordinates, this yields:

$$\tilde{q}^x_\tau = \frac{p^x_0 + v^x\tau}{p^z_0} \tag{4}$$

$$\tilde{q}^y_\tau = \frac{p^y_0 + v^y\tau - \frac{1}{2}g\tau^2}{p^z_0}. \tag{5}$$

Note that in this special case the projectile's path maps to a parabola in image coordinates, which can be seen more clearly by rewriting the above equations as:

$$\begin{bmatrix} \tilde{q}^x_\tau \\ \tilde{q}^y_\tau \end{bmatrix} = \begin{bmatrix} \frac{p^x_0}{p^z_0} \\ \frac{p^y_0}{p^z_0} \end{bmatrix} + \begin{bmatrix} \frac{v^x}{p^z_0} \\ \frac{v^y}{p^z_0} \end{bmatrix} \tau + \frac{1}{2} \begin{bmatrix} 0 \\ \frac{-g}{p^z_0} \end{bmatrix} \tau^2. \tag{6}$$

However under an arbitrary homography the image of a projectile's path will not necessarily be a parabola. Specifically:

$$\tilde{\mathbf{q}}_\tau = \mathbf{H}\mathbf{p}_\tau = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \mathbf{p}_\tau, \tag{7}$$

In non-homogeneous image coordinates, the projectile takes the form:

$$\tilde{q}^x_\tau = \frac{\mathbf{h}^1 \cdot \mathbf{p}_\tau}{\mathbf{h}^3 \cdot \mathbf{p}_\tau} = \frac{\mathbf{h}^1 \cdot (\mathbf{p}_0 + \mathbf{v}\tau + \frac{1}{2}\mathbf{a}\tau^2)}{\mathbf{h}^3 \cdot (\mathbf{p}_0 + \mathbf{v}\tau + \frac{1}{2}\mathbf{a}\tau^2)} \tag{8}$$

$$\tilde{q}^y_\tau = \frac{\mathbf{h}^2 \cdot \mathbf{p}_\tau}{\mathbf{h}^3 \cdot \mathbf{p}_\tau} = \frac{\mathbf{h}^2 \cdot (\mathbf{p}_0 + \mathbf{v}\tau + \frac{1}{2}\mathbf{a}\tau^2)}{\mathbf{h}^3 \cdot (\mathbf{p}_0 + \mathbf{v}\tau + \frac{1}{2}\mathbf{a}\tau^2)}, \tag{9}$$

where $\cdot$ is inner product and $\mathbf{h}^i$ is the i^{th} row of the homography $\mathbf{H}$. Note that these image coordinates follow a rational parabola described by a ratio of second-order polynomials, and

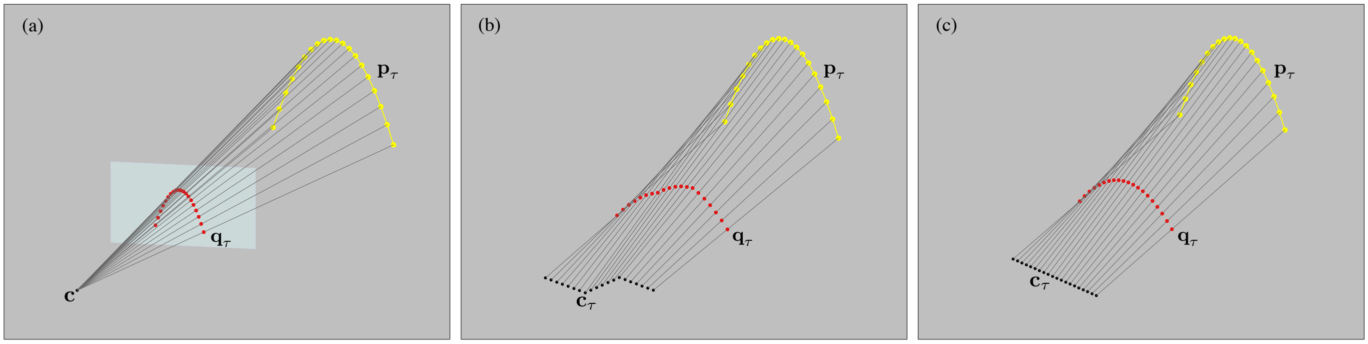


Fig. 2. For a stationary camera (a), a ballistic trajectory will appear as a rational parabola in the image plane (red dots). Lines from the center of projection $\mathbf{c}$ (black dot) through these points $\mathbf{q}_\tau$ on the image plane will intersect the object's true parabolic trajectory through space $\mathbf{p}_\tau$ (yellow dots). A scale ambiguity appears because any slice through the generalized cone formed by the lines would also produce a parabolic arc if that slice is parallel to the object's true parabolic trajectory. With a moving camera (b), the lines from the moving center of projection $\mathbf{c}_\tau$ (black points) through observed locations for the object do not form a cone, but they must still intersect the object's parabolic trajectory through space. In general, other slices through the lines will not form parabolae unless the camera's path is also a quadratic curve as shown in (c). In that case any isoparametric slice through the lines will also produce a parabolic trajectory. The scale ambiguity for a stationary camera is a special case of the parametric ambiguity for a camera on a quadratic path. For clarity, the image planes for each camera location in (b) and (c) are not shown.

cannot be expressed as a single second-order polynomial as in Equation (6).

Given a composite image from a still camera showing the path of an object, one could test the authenticity of the motion by verifying that it can be fit by a rational parabola using Equations (8) and (9). However that approach does not explicitly reveal the direction of gravity, provide a trajectory in world space, or generalize to a moving camera.

B. Projectile Estimation: Static Camera

In the previous section, the projectile was specified in a world coordinate system, Equation (1), and its projection was specified in an image coordinate system, Equation (2). In the following sections, we will specify all coordinates with respect to a common three-dimensional coordinate system in which the origin is the camera center, the image plane is f units from the origin (the focal length), and the optical axis is orthogonal to the image plane. With this notation, a projectile is specified as $\mathbf{p}_\tau$ in Equation (1), its projection is specified as a three-vector $\mathbf{q}_\tau$ in which the z-component $q^z_\tau = f$, and the camera center, $\mathbf{c}$, is the origin of the three-dimensional coordinate system. (See Fig. 2(a).)

For each moment in time $\tau = 1 \ldots n$, define a line from the camera center, $\mathbf{c}$, through the image of the projectile's center of mass, $\mathbf{q}_\tau$, as:

$$\mathbf{l}_\tau \quad = \quad \mathbf{c} + s_\tau(\mathbf{q}_\tau - \mathbf{c}), \tag{10}$$

where s_τ is a parametric variable for the line. If the projectile $\mathbf{p}_\tau$ follows a parabolic trajectory then there exists a value of the parametric variable at each moment in time that satisfies:

$$\begin{aligned} \mathbf{p}_\tau &= \mathbf{l}_\tau & (11)\\ \mathbf{p}_0 + \mathbf{v}\tau + \tfrac{1}{2}\mathbf{a}\tau^2 &= \mathbf{c} + s_\tau(\mathbf{q}_\tau - \mathbf{c}) & (12) \end{aligned}$$

for some values of s_τ. Expanding in terms of the individual components yields:

$$\begin{aligned} p^x_0 + v^x\tau + \tfrac{1}{2}a^x\tau^2 &= c^x + s_\tau(q^x_\tau - c^x) & (13)\\ p^y_0 + v^y\tau + \tfrac{1}{2}a^y\tau^2 &= c^y + s_\tau(q^y_\tau - c^y) & (14)\\ p^z_0 + v^z\tau + \tfrac{1}{2}a^z\tau^2 &= c^z + s_\tau(q^z_\tau - c^z) & (15) \end{aligned}$$

This system of equations is linear in terms of both the nine unknowns that specify the object's trajectory through space ($\mathbf{p}_0$, $\mathbf{v}$, and $\mathbf{a}$) and the n unknowns that specify the object's parametric location along the line from the camera (s_τ). With a sufficient number of frames ($n \geq 5$) where the projectile's position can be observed, it is possible to solve for these unknowns by performing a least-squares fit to the data. However, in the absence of noise, the linear system will always be rank deficient so that the least-squares solution is not unique. This deficiency corresponds to a scale ambiguity in the solution: any solution could be scaled about the origin and still satisfy the constraints, Fig. 2(a).

Before describing a robust solution that copes with both this ambiguity and noise, we will first discuss the situation for a moving camera. Our solution treats both still and moving cameras using the same framework.

C. Projectile Estimation: Moving Camera

In the previous section we described the estimation of a projectile that was being imaged by a static camera. In that case, the projection of the projectile was a parabola subjected to a planar homography. However, if the camera is not stationary, then the projection of the projectile can take an arbitrary path in the image plane that results from combining image coordinates from different local coordinate systems. If the camera motion can be estimated, then these image coordinates can be described in a single common coordinate system. The resulting estimation of projectile motion then becomes equivalent to the static camera case with the difference being that the location of the camera center varies over time. We will arbitrarily use the coordinate system of the initial frame in a sequence as our common coordinate system.

Consider a camera undergoing a rigid-body transformation (rotation and translation) with fixed intrinsic parameters (focal length, principal point, etc.)[1]. The image of a projectile's trajectory can be expressed with respect to the common

[1] It is possible to contend with a varying focal length over time, but for notational simplicity we assume that the focal length is fixed.

coordinate system as $\mathbf{R}_\tau \mathbf{q}_\tau + \mathbf{t}_\tau$, where $\mathbf{q}_\tau$ is the image of the projectile in the camera's local coordinate system, and $\mathbf{R}_\tau$ and $\mathbf{t}_\tau$ are respectively the 3×3 rotation matrix and 3×1 translation vector that define the transformation between the local and common coordinate systems. The camera center is expressed with respect to the common coordinate system as $\mathbf{c}_\tau + \mathbf{t}_\tau$, where $\mathbf{c}_\tau$ is the location of the camera center in the local coordinate system.

Given the rotation and translation that relates a camera at each time to the common coordinate system, all coordinates can be expressed with respect to a single common coordinate system. As in the static case, for each frame a line is defined from the camera center, $\mathbf{c}_\tau$ (now a function of time), through the center of mass of the projection of the projectile into the image plane, $\mathbf{q}_\tau$, as:

$$\mathbf{l}_\tau = (\mathbf{c}_\tau + \mathbf{t}_\tau) + s_\tau(\mathbf{R}_\tau \mathbf{q}_\tau - \mathbf{c}_\tau) \tag{16}$$

where s_τ is the parametric variable. Note the similarity to Equation (10), with the exception that $\mathbf{q}_\tau$ and $\mathbf{c}_\tau$ undergo a coordinate transformation. If the projectile follows a parabolic trajectory, $\mathbf{p}_\tau$, then there exists a value of the parametric variable s_τ, at each moment in time that satisfies:

$$\mathbf{p}_\tau = (\mathbf{c}_\tau + \mathbf{t}_\tau) + s_\tau(\mathbf{R}_\tau \mathbf{q}_\tau - \mathbf{c}_\tau) \tag{17}$$

or

$$p_0^x + v^x\tau + \tfrac{1}{2}a^x\tau^2 = (c_\tau^x + t_\tau^x) + s_\tau(\mathbf{r}_\tau^1 \cdot \mathbf{q}_\tau - c_\tau^x) \tag{18}$$
$$p_0^y + v^y\tau + \tfrac{1}{2}a^y\tau^2 = (c_\tau^y + t_\tau^y) + s_\tau(\mathbf{r}_\tau^2 \cdot \mathbf{q}_\tau - c_\tau^y) \tag{19}$$
$$p_0^z + v^z\tau + \tfrac{1}{2}a^z\tau^2 = (c_\tau^z + t_\tau^z) + s_\tau(\mathbf{r}_\tau^3 \cdot \mathbf{q}_\tau - c_\tau^z) \tag{20}$$

where $\cdot$ is inner product and $\mathbf{r}_\tau^i$ is the i^{th} row of the rotation matrix $\mathbf{R}_\tau$. Note that, as in the previous section, this system of equations is linear in the unknown projectile parameters ($\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$), and the parametric variables (s_τ).

As before, with a sufficient number of frames ($n \geq 5$) where the projectile's position can be observed, it becomes possible to solve for these unknowns by performing a least-squares fit to the data. However, unlike the situation with a still camera there is no longer a necessary scale ambiguity. Because the camera now moves arbitrarily, scaling the solution would also require scaling the camera path as well, but the scale of that path will have been fixed when the camera's motion was estimated.

Nevertheless, specific combinations of camera and projectile paths may still fail to produce a fully determined system, Fig. 2(b)-(c). In particular, if the motion of the camera center is also described by a parabola then any isoparametric interpolation between the projectile's true trajectory and the camera's trajectory will also be a parabola. These extra solutions correspond to uniform scaling of the vector of s_τ values that would produce the true trajectory. This ambiguity is the general case of the scale ambiguity that appears for a static camera, and the static camera can be viewed as just a special case where the camera motion follows the trivial parabola.

D. Extrinsic Camera Calibration

Before continuing, we briefly describe how to estimate the required extrinsic camera transformations $\mathbf{R}_\tau$ and $\mathbf{t}_\tau$ needed to relate each video frame to a common coordinate system. Modern-day camera calibration techniques consist of three basic steps: (1) image features are automatically detected in each video frame; (2) the features are matched across video frames; and (3) the matched features are used to estimate the camera parameters: extrinsic ($\mathbf{R}_\tau$, $\mathbf{t}_\tau$) and intrinsic (focal length, principal point). A commonly used and effective technique for this last step is bundle adjustment [20].

Among the many available software tools for extrinsic camera calibration, we employ *Voodoo Camera Tracker* [21]. Within this software, we employ the SIFT operator [22] to extract image features and the RANSAC algorithm for matching features.

E. Size Constraints

Although the perspective projection of a parabola makes an ideal model for representing the image of ballistic trajectory, the previously mentioned scale ambiguity can lead to bad behavior when data are noisy or if the underlying motion is not actually ballistic. In these cases, the solution may be skewed dramatically in the otherwise unconstrained part of the solution space. This skew typically manifests as trajectories in planes that are nearly parallel to the view lines with unreasonably high velocities and accelerations.

To prevent these degenerate solutions, an optional constraint can be imposed based on the variation in size of a projectile over time. With the assumption that the actual projectile is of constant size, then its projected size in the image is inversely proportional to the distance between the object and camera center as measured orthogonal to the image plane. Accordingly, the additional constraints require that the trajectory's distance to the image plane vary based on measurements of the object's size in the input images.

Consider a spherical projectile with diameter d at position (x_1, y_1, z_1) relative to the camera center with the z-axis being perpendicular to the image plane. The projected size of this projectile will be $\tilde{d}_1 = fd/z_1$, where f is the camera focal length. As the projectile moves to another position (x_2, y_2, z_2), the projected size is $\tilde{d}_2 = fd/z_2$. The ratio of these projections is $\tilde{d}_2/\tilde{d}_1 = z_1/z_2$. Note that this ratio does not depend on the focal length f or diameter d.

For the static camera case, this basic constraint takes the form:

$$\frac{l_1^z - c^z}{l_k^z - c^z} = \frac{\tilde{d}_k}{\tilde{d}_1}, \tag{21}$$

where c^z is the z-component of the camera center $\mathbf{c}$, $\tilde{d}_1$ and $\tilde{d}_k$ are the measured sizes of the projectile in the image at times $\tau = 1$ and $\tau = k$, and l_1^z and l_k^z are the z-components of $\mathbf{l}_1$ and $\mathbf{l}_k$ as defined in Equation (10). This constraint expands to:

$$s_1 \left[\frac{q_1^z - c^z}{q_k^z - c^z}\right] - s_k \left[\frac{\tilde{d}_k}{\tilde{d}_1}\right] = 0. \tag{22}$$

Note that this constraint is linear in the unknown parametric variables s_1 and s_k. These linear constraints for all $k = 2 \ldots n$ can be included when solving for the trajectory parameters.

For the moving camera case, the constraint in Equation (22) takes the form:

$$s_1 \left[\frac{\mathbf{r}_1^3 \cdot \mathbf{q}_1 - c_1^z}{\mathbf{r}_k^3 \cdot \mathbf{q}_k - c_k^z} \right] - s_k \left[\frac{\tilde{d}_k}{\tilde{d}_1} \right] \;=\; 0, \qquad (23)$$

where $\mathbf{r}_k^3$ is the third row of the rotation matrix $\mathbf{R}_k$.

F. Trajectory Estimation

The constraints from the preceding sections can be assembled into a linear system of equations. For both the static and moving camera cases, this system will have a similar structure. This system can be over-constrained, either because the data cannot be fit by a parabola or because of noise caused by small measurement errors. This system can also be under-constrained due to a scale ambiguity, as previously described in Sections III-B and III-C. The following least-squares solution contends with all of these cases.

For the static camera case, the unknown parameters, $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$, and s_τ ($\tau = 1 \dots n$), are gathered into a length $n{+}9$ vector:

$$\mathbf{u} = [p_0^x \; v^x \; a^x \; p_0^y \; v^y \; a^y \; p_0^z \; v^z \; a^z \; s_1 \cdots s_n]^{\mathsf{T}} \quad . \qquad (24)$$

The constraints, Equations (13)-(15), are assembled into a linear system $\mathbf{Mu} = \mathbf{b}$ where:

$$\mathbf{M}=\begin{bmatrix} 1 & \tau_1 & \frac{\tau_1^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & c^x - q_1^x & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \tau_1 & \frac{\tau_1^2}{2} & 0 & 0 & 0 & c^y - q_1^y & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau_1 & \frac{\tau_1^2}{2} & c^z - q_1^z & 0 & \dots & 0 \\ 1 & \tau_2 & \frac{\tau_2^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c^x - q_2^x & \dots & 0 \\ 0 & 0 & 0 & 1 & \tau_2 & \frac{\tau_2^2}{2} & 0 & 0 & 0 & 0 & c^y - q_2^y & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau_2 & \frac{\tau_2^2}{2} & 0 & c^z - q_2^z & \dots & 0 \\ \vdots & & & & & & & \ddots & & & & & \vdots \\ 1 & \tau_n & \frac{\tau_n^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & c^x - q_n^x \\ 0 & 0 & 0 & 1 & \tau_n & \frac{\tau_n^2}{2} & 0 & 0 & 0 & 0 & 0 & \dots & c^y - q_n^y \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau_n & \frac{\tau_n^2}{2} & 0 & 0 & \dots & c^z - q_n^z \end{bmatrix} \qquad (25)$$

and

$$\mathbf{b} = [c^x \; c^y \; c^z \; c^x \; c^y \; c^z \cdots c^x \; c^y \; c^z]^{\mathsf{T}} \, .$$

For the moving camera case, Equations (18)-(20), the $\mathbf{c} - \mathbf{q}_\tau$ in the rows of $\mathbf{M}$ are replaced with $\mathbf{c}_\tau - \mathbf{R}_\tau \mathbf{q}_\tau$, and $\mathbf{b}$ is replaced with:

$$\mathbf{b} = [c_1^x{+}t_1^x \;\; c_1^y{+}t_1^y \;\; c_1^z{+}t_1^z \dots c_n^x{+}t_n^x \;\; c_n^y{+}t_n^y \;\; c_n^z{+}t_n^z]^{\mathsf{T}} \, .$$

If the optional size constraints from Section III-E are used, then $\mathbf{M}$ is extended by appending an additional $n-1$ rows with a corresponding number of zeros appended to $\mathbf{b}$. For a static camera these rows have the form:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{q_1^z - c^z}{q_2^z - c^z} & -\frac{\tilde{d}_2}{\tilde{d}_1} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{q_1^z - c^z}{q_3^z - c^z} & 0 & -\frac{\tilde{d}_3}{\tilde{d}_1} & \dots & 0 \\ \vdots & & & & & & \ddots & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{q_1^z - c^z}{q_n^z - c^z} & 0 & 0 & \dots & -\frac{\tilde{d}_n}{\tilde{d}_1} \end{bmatrix} . \qquad (26)$$

For the moving camera case, the $\mathbf{q}_\tau - \mathbf{c}$ are correspondingly replaced with $\mathbf{R}_\tau \mathbf{q}_\tau - \mathbf{c}_\tau$.

The least-squares solution to the system of constraints is given by $\mathbf{u} = \mathbf{M}^{+}\mathbf{b}$, where $\mathbf{M}^{+}$ denotes pseudo-inverse computed as $\mathbf{M}^{+} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^{\mathsf{T}}$, where $\mathbf{M} = \mathbf{USV}$ is the singular-value decomposition of $\mathbf{M}$.

Due to the possible scale ambiguity, this solution may not be unique. This occurs when the smallest-magnitude singular value in $\mathbf{S}$ is zero. Due to small amounts of measurement noise the smallest singular value may not be identically zero. We consider the smallest singular value to be zero if it is less than 0.1 times the second-smallest singular value.

If there is no zero singular value then $\mathbf{M}^{+}\mathbf{b}$ is our final solution. Otherwise the ambiguity is resolved by finding the solution where $\|\mathbf{a}\| = 9.8\text{m/s}^2$. Let $\mathbf{u}^*$ be the column of $\mathbf{V}$ corresponding to the zero singular value. The final solution is $\mathbf{u} = \mathbf{M}^{+}\mathbf{b} + \alpha\mathbf{u}^*$ where α is selected by solving the quadratic equation:

$$\left\| [0 \; 0 \; a^x \; 0 \; 0 \; a^y \; 0 \; 0 \; a^z \; 0 \cdots 0]^{\mathsf{T}} \cdot (\mathbf{M}^{+}\mathbf{b} + \alpha\mathbf{u}^*) \right\|^2 = 9.8^2 \quad . \qquad (27)$$

In the case of a static camera, this quadratic constraint on the acceleration will give us the scaling that corresponds to its size in the physical world. More importantly it also avoids the trivial solution of $\mathbf{u} = \mathbf{0}$ when $\mathbf{b} = \mathbf{0}$.

In the case of a moving camera where both the camera and the projectile follow a parabolic path, any linear blend between the two paths will satisfy Equation (17). Like a scale ambiguity, this situation will manifest as a zero singular value in $\mathbf{S}$ and can be handled with the quadratic acceleration constraint as described above. If the camera follows a parabolic path but the projectile does not, then the system will be fully determined with only a degenerate solution where all the s_i are zero. Note also that when used with a moving camera, Equation (27) assumes that $\mathbf{c}_\tau$ and $\mathbf{t}_\tau$ are expressed in units of meters. If other units have been used then the constant 9.8 should be adjusted accordingly.

G. Forensics

The estimation of the projectile motion described in the previous two sections yields two parametrizations of the projectile path. The first is the initial position, velocity and acceleration ($\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$) as specified in Equation (1). The second is a non-parametric representation specified by the variables s_τ.

For authentic video of a ballistic projectile, these representations should be in agreement, while an imperfectly faked projectile motion will yield inconsistencies in these representations. For the static camera case, we quantify the error in these representations as the average Euclidean distance between the pair of representations of the projectile motion:

$$\begin{aligned} E \quad &= \quad \frac{1}{n} \sum_{\tau=1}^{n} \|\mathbf{p}_\tau - \mathbf{l}_\tau\| && (28) \\ &= \quad \frac{1}{n} \sum_{\tau=1}^{n} \|(\mathbf{p}_0 + \mathbf{v}\tau + \tfrac{1}{2}\mathbf{a}\tau^2) - (\mathbf{c} + s_\tau(\mathbf{q}_\tau - \mathbf{c}))\|, && (29) \end{aligned}$$

where $\| \cdot \|$ denotes a vector 2-norm. Specifically, an error E above a specified threshold is taken to be evidence of tampering. For the moving camera case, this error takes on a similar form:

$$E = \frac{1}{n}\sum_{\tau=1}^{n} \quad \begin{aligned} &\|(\mathbf{p}_0 + \mathbf{v}\tau + \tfrac{1}{2}\mathbf{a}\tau^2) \\ &-((\mathbf{c}_\tau + \mathbf{t}_\tau) + s_\tau(\mathbf{R}_\tau\mathbf{q}_\tau - \mathbf{c}_\tau))\|. \end{aligned} \tag{30}$$

This average error will be large when an overall path does not correspond to a ballistic trajectory. For situations where only small parts of an otherwise correct motion have been altered, the maximum error may also be informative. Note, however, that because the error E is specified in world-space units, a reasonable threshold will depend on the overall scale of the scene and should be chosen accordingly. Furthermore, as shown for example in Figs. 3-6, a visual comparison can be made of the difference between the parametric and non-parametric solutions.

H. Summary

Here we summarize the algorithmic steps required to analyze a projectile's trajectory recorded from a static or moving camera.

1) **Tracking:** On the first video frame manually select a bounding box that is centered on the projectile's center of mass. On the remaining video frames automatically track the projectile's position (Section IV-A).
2) **Size Constraint (optional):** On each video frame manually estimate the size of the projectile (e.g., a ball's diameter). This step can be automated by using the results of the tracking in step 1.
3) **Extrinsic Calibration:** For a moving camera estimate the extrinsic camera transformations $\mathbf{R}_\tau$ and $\mathbf{t}_\tau$ at each time τ that relates each video frame to a common coordinate system (Section III-D).
4) **Trajectory Estimation:** Using the tracked projectile, the size constraint (optional), and the results of extrinsic calibration (for a moving camera), construct and solve the linear system $\mathbf{Mu} = \mathbf{b}$ (Section III-F). The solution of this linear system yields a parametric and non-parametric parametrization of the projectile's trajectory.
5) **Forensics:** Compute the error between the two trajectory parametrizations (Section III-G). An error above a specified threshold is considered to be evidence of a fake.

IV. Results

We evaluate the efficacy of our forensic technique on authentic and fake videos of our creation, and on several videos obtained from video-sharing websites such as YouTube. In each example, the optional size constraint described in Section III-E was used. In order to be able to directly compare the error metric, Equation (29) or (30), across different videos, the size constraint is weighted to yield the same average absolute difference, Equation (22). Specifically, each row of the matrix in Equation (26) is weighted by a scalar amount that controls the relative penalty contributed by the size constraint. An iterative approach was taken to determine this weighting. On each iteration the linear system is solved, and the weighting is decreased until the difference reaches a specified threshold. Because the trajectory estimation is scaled so that the acceleration term is 9.8m/s^2, Equation (27), the resulting error metric is specified in meters.

A. Tracking

The projection of a projectile's center of mass $\mathbf{q}_\tau$ on each video frame is estimated using a combination of manual and automatic tracking. A user first manually selects the approximate center of mass of the projectile on each frame. A user also places a bounding box around the projectile in the first frame, which is subsequently used as a template to refine the initial manually selected projectile locations. The bounding box is selected such that its center of mass corresponds to the projectile's center of mass. For each frame, the cross-correlation between the template and a small region around the user selected location is computed. The final projectile location is the position that maximizes the cross-correlation. The template is updated on each frame by automatically moving the selected bounding box to the next frame. This contends with the possible changing appearance and size of the projectile over time.

A specialized tracking algorithm is employed when the projectile is spherical (a basketball or soccer ball). Specifically, a circular Hough transform [23] is used to refine the location of the initially selected projectile location.

B. Simulations

We simulated ballistic trajectories of a projectile with varying initial positions relative to a fixed and stationary camera, varying velocities, small amounts of additive noise, and varying planes of motion relative to the camera. These trajectories had known world and image positions, so no tracking algorithm was required. In addition, the projectile was simulated as a point, so the optional size constraint was not used. Ten thousand random trajectories were generated from which the error, Equation (29), between the estimated parametric and non-parametric parametrizations was computed. The mean error for the authentic ballistic trajectories is 0.0096 with a standard deviation of 0.0039. Non-ballistic, yet visually plausible, trajectories were created by sampling a path along an ellipse. Ten thousand random elliptical trajectories were created. The mean error for the elliptical trajectories is 0.0446 with a standard deviation of 0.0150. In terms of discriminability, 98.82% of the parabolic trajectories have an error less than 0.02, while 97.96% of the elliptical trajectories have an error greater than 0.02, revealing good discrimination.

C. Static Camera

Images in Fig. 3 show results from videos recorded with a static camera. The images in panels (a) show four frames taken from each of the video sequences. Panel (b) shows the two parametrizations of the estimated trajectory: the parametric trajectory specified by initial position $\mathbf{p}_0$, velocity $\mathbf{v}$, and acceleration $\mathbf{a}$ is denoted with filled yellow dots and solid line, and locations along the non-parametric trajectory specified by the variables s_τ are denoted with open blue circles. The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Panel (c) shows the estimated parametric trajectory, denoted with filled yellow dots, projected into the image plane, and the tracked position of the projectile denoted with small red dots.

In the first column the different parametrizations of the projectile are in agreement, as would be expected for an authentic video. The different parametrizations are not in agreement for the second and third videos revealing them to be fakes. The mean error from Equation (29) for these three sequences are respectively 0.014m, 0.151m, and 0.175m, Table I. The errors for the fake videos are an order of magnitude larger than for the authentic video.

Note that in the first column of Fig. 3 the ball temporarily disappears from the field of view. Regardless, we can still set up the linear system, skipping frames where the ball cannot be observed, because the time-coding of the video establishes a consistent time parametrization. Once we have computed $\mathbf{p}_0$, $\mathbf{v}$, and $\mathbf{a}$, the full trajectory can then be estimated by extrapolating the ball's position during the time it is out of the field of view.

A total of six authentic and seven fake videos with static cameras were analyzed, a subset of which are shown in Figs. 3 and 7(a)-(b). The mean error averaged over all the authentic videos is 0.023m and the largest mean error for any of the authentic videos is 0.032m. The mean error averaged over the fake videos is an order of magnitude larger at 0.247m, and the smallest of the mean errors for any of the fake videos is 0.106m. With a threshold of 0.075m, the authentic and fake videos are perfectly separated in terms of the error metric used to assess authenticity. There is also a very clear qualitative distinction that can be seen when comparing the trajectory visualizations in panel (c) of Fig. 3.

Images in Fig. 4 show results from three YouTube videos. Based on the agreement of the estimated trajectories we conclude that the first two videos are authentic while the third is fake. The mean errors from Equation (29) for these sequences are 0.010m, 0.038m, and 0.307m, Table I. The errors for the fake video are an order of magnitude larger than for the authentic videos. Images in the first two columns of Fig 7(e) show results from two more YouTube videos. With a mean error of 0.033m and 0.036m, we conclude that these videos are authentic.

D. Moving Camera

Images in Fig. 5 show results from videos recorded with a moving camera. Panels (a) show four video frames from each motion sequence. Shown in panel (b) are the two parametrizations of the estimated trajectory: the parametric trajectory specified by initial position $\mathbf{p}_0$, velocity $\mathbf{v}$, and acceleration $\mathbf{a}$ is denoted with filled yellow dots and solid line, and the non-parametric trajectory specified by the variables s_τ is denoted with open blue circles. The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Shown in panel (c) is the estimated parametric trajectory, denoted with filled yellow dots, projected into the image plane, and the tracked position of the projectile denoted with small red dots. The underlying image in panel (c) was constructed by compositing the video frames into a single composite.

Fig	Col	μ (m)	σ (m)	max (m)	Truth	Predict
3	1	0.014	0.011	0.044	real	real
3	2	0.151	0.093	0.382	fake	fake
3	3	0.175	0.077	0.420	fake	fake
4	1	0.038	0.029	0.121	—	real
4	2	0.010	0.009	0.033	—	real
4	3	0.307	0.114	0.750	—	fake
5	1	0.027	0.019	0.082	real	real
5	2	0.082	0.072	0.341	fake	fake
5	3	0.266	0.173	0.934	fake	fake
6	1	0.024	0.016	0.077	—	real
6	2	0.056	0.035	0.162	—	real
6	3	0.119	0.064	0.365	—	fake

TABLE I. The error between the parametric and non-parametric parametrizations of the estimated projectile trajectory for a static camera (Figs. 3 and 4), Equation (29), and for a moving camera (Figs. 5 and 6), Equation (30). Shown are the mean μ, standard deviation σ, and maximum error, specified in meters. Shown in the last two columns are, respectively, the actual and predicted authenticity of each video. (Actual authenticity is not known for the YouTube videos).

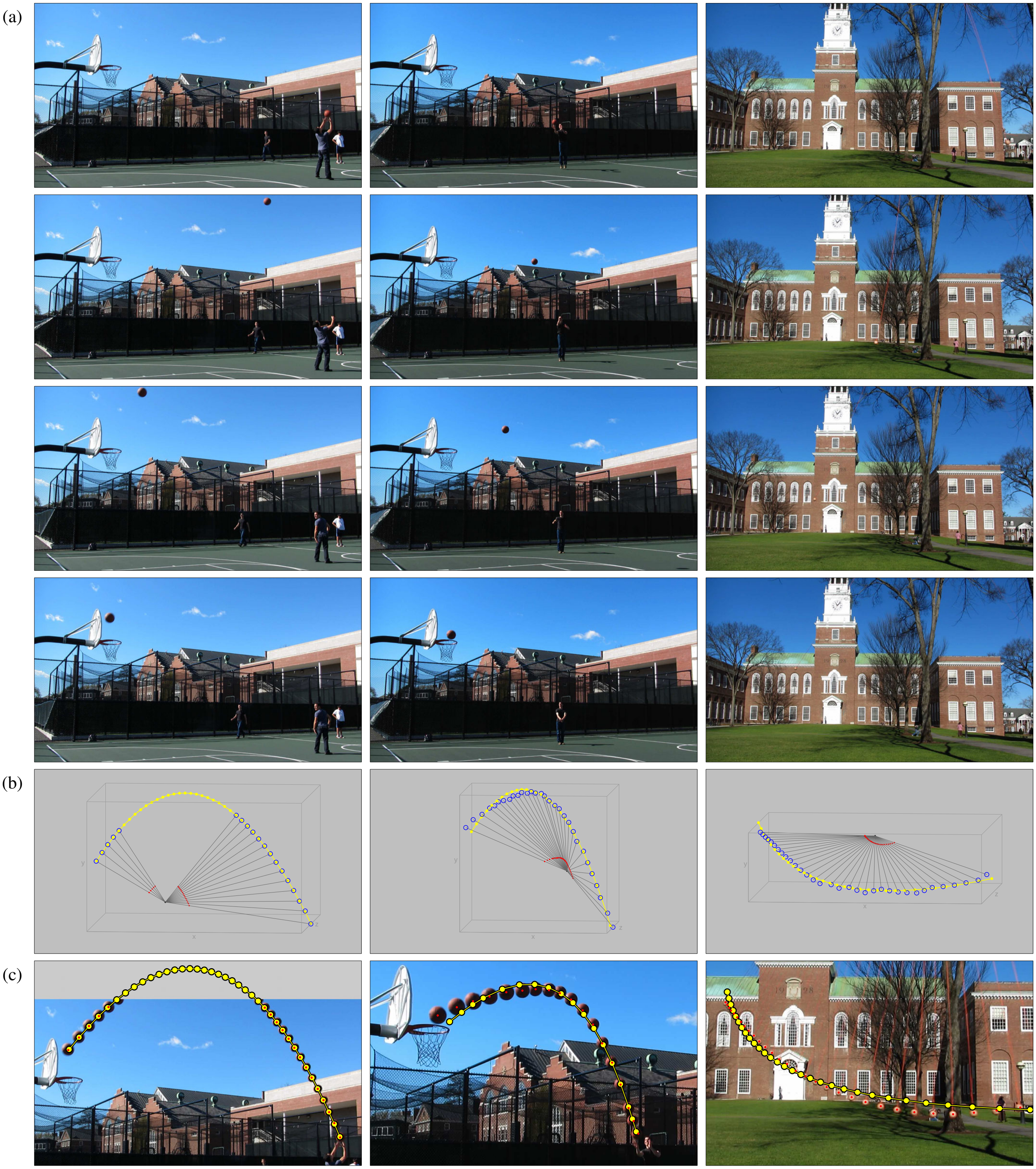
In the first column the different parametrizations of the projectile are in agreement, as would be expected for an authentic video. The different parametrizations are not in agreement for the second and third videos revealing them to be fakes. The mean error, Equation (30), for these three sequences are 0.027m, 0.082m, and 0.266m, Table I. The errors for the fake videos are significantly larger than for the authentic video.

A total of five authentic and six fake videos with a moving camera were analyzed, a subset of which are shown in Figs. 5 and 7(c)-(d). The mean error averaged over the authentic videos is 0.034m and the largest mean error for any of the authentic videos is 0.049m. The mean error for the fake videos is several times larger at 0.201m, and the smallest of the mean errors for any of the fake videos is 0.081m. With the same threshold as in the static camera case of 0.075m, the authentic and fake videos are perfectly separated in terms of the error metric used to assess authenticity. As with the static camera examples, there is also a very clear qualitative distinction that can be seen when comparing the trajectory visualizations.

Images in Fig. 6 show results from three YouTube videos. Based on the agreement of the estimated trajectories we conclude that the first two videos are authentic while the third is fake. The mean errors from Equation (29) for these three sequences are 0.056m, 0.024m, and 0.119m, Table I. The errors for the fake videos are significantly larger than for the authentic video. The image in the last column of Fig 7(e) shows results from one more YouTube video. With a mean error of 0.029m, we conclude that this video is authentic.

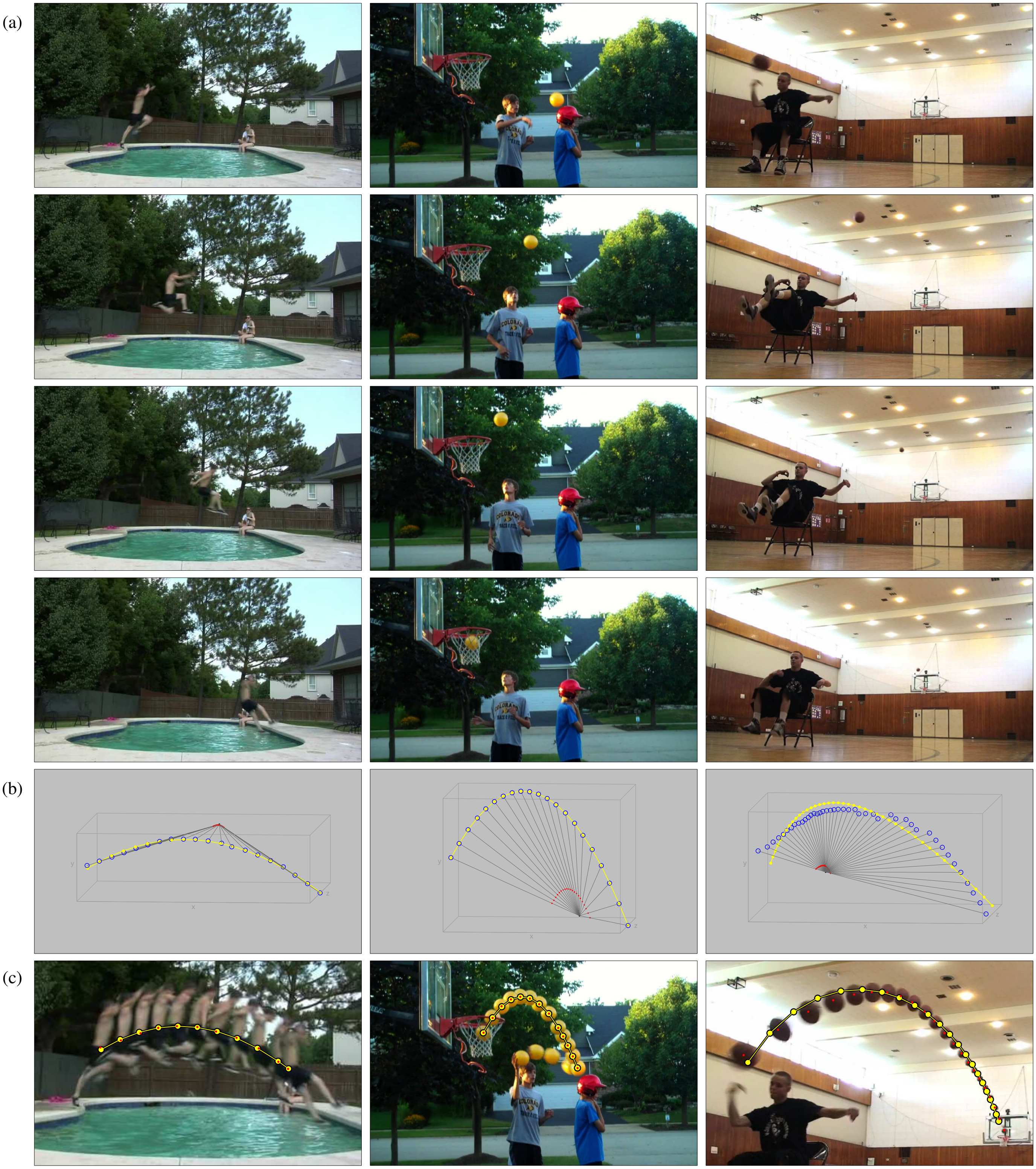
V. Discussion

We have described a geometric technique for detecting forged ballistic motion in video sequences. This technique explicitly models the three-dimensional trajectory of objects in free-flight and the two-dimensional imaging of the trajectory



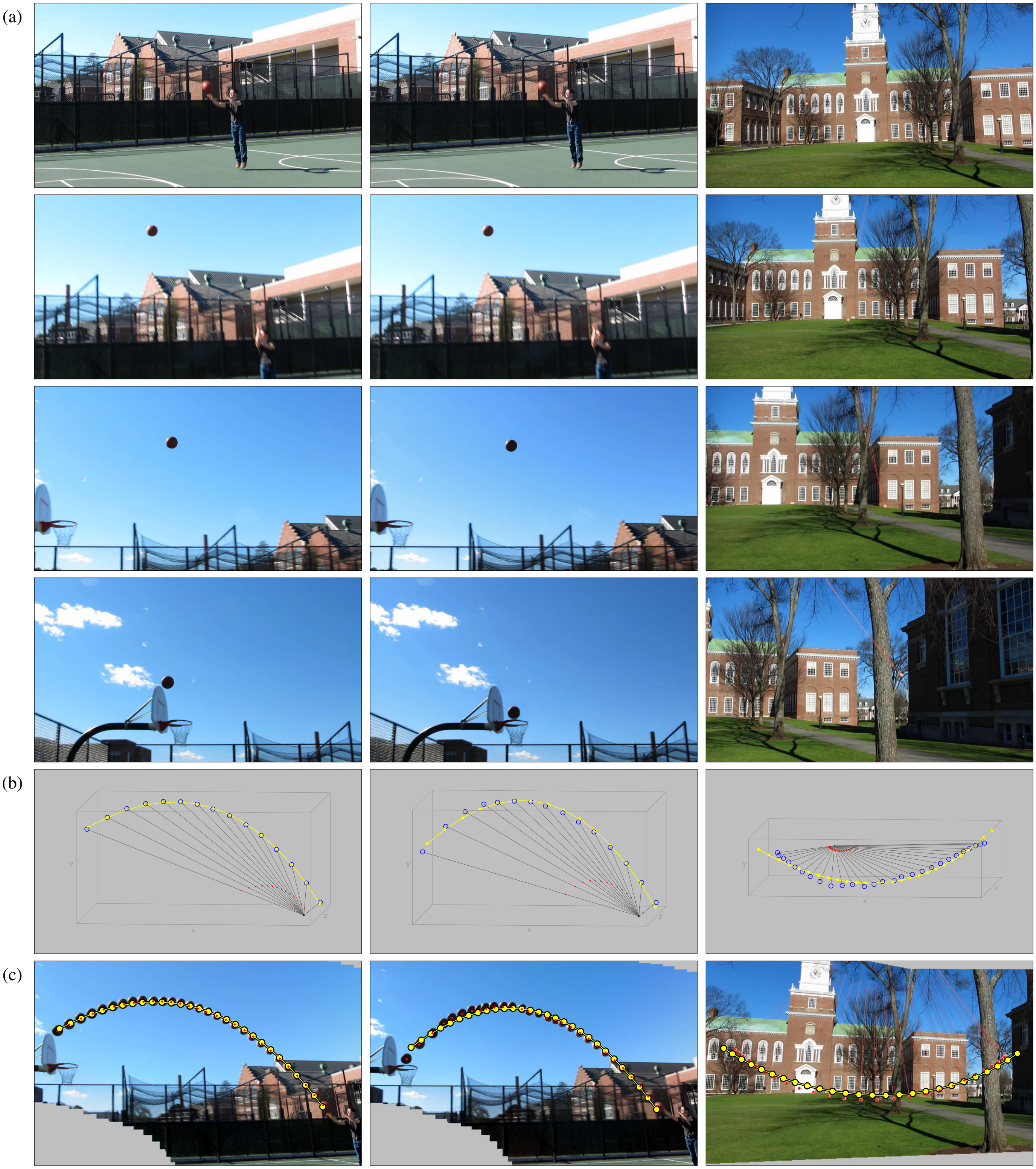
http://www.youtube.com/watch?v=z7NpkDCDYjg *http://www.youtube.com/watch?v=RLJ7aNkIv7I* *http://www.youtube.com/watch?v=ucGLQ8wzxNM*

Fig. 3. One authentic video (left), one fake video (middle), and a non-ballistic motion (right) all from a static camera. Shown in panel (a) are four sample video frames. Shown in panel (b) are the two parametrizations of the estimated trajectory: the parametric trajectory specified by $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$ (filled yellow dots and solid line), and the non-parametric trajectory specified by the variables s_τ (open blue circles). The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Note that the two parametrizations are in agreement for the authentic video but not for the fake videos. Shown in panel (c) is the estimated parametric trajectory projected into the image plane (filled yellow dots) and the tracked position of the projectile (small red dots). These locations are in agreement for the authentic video but not for the other videos.



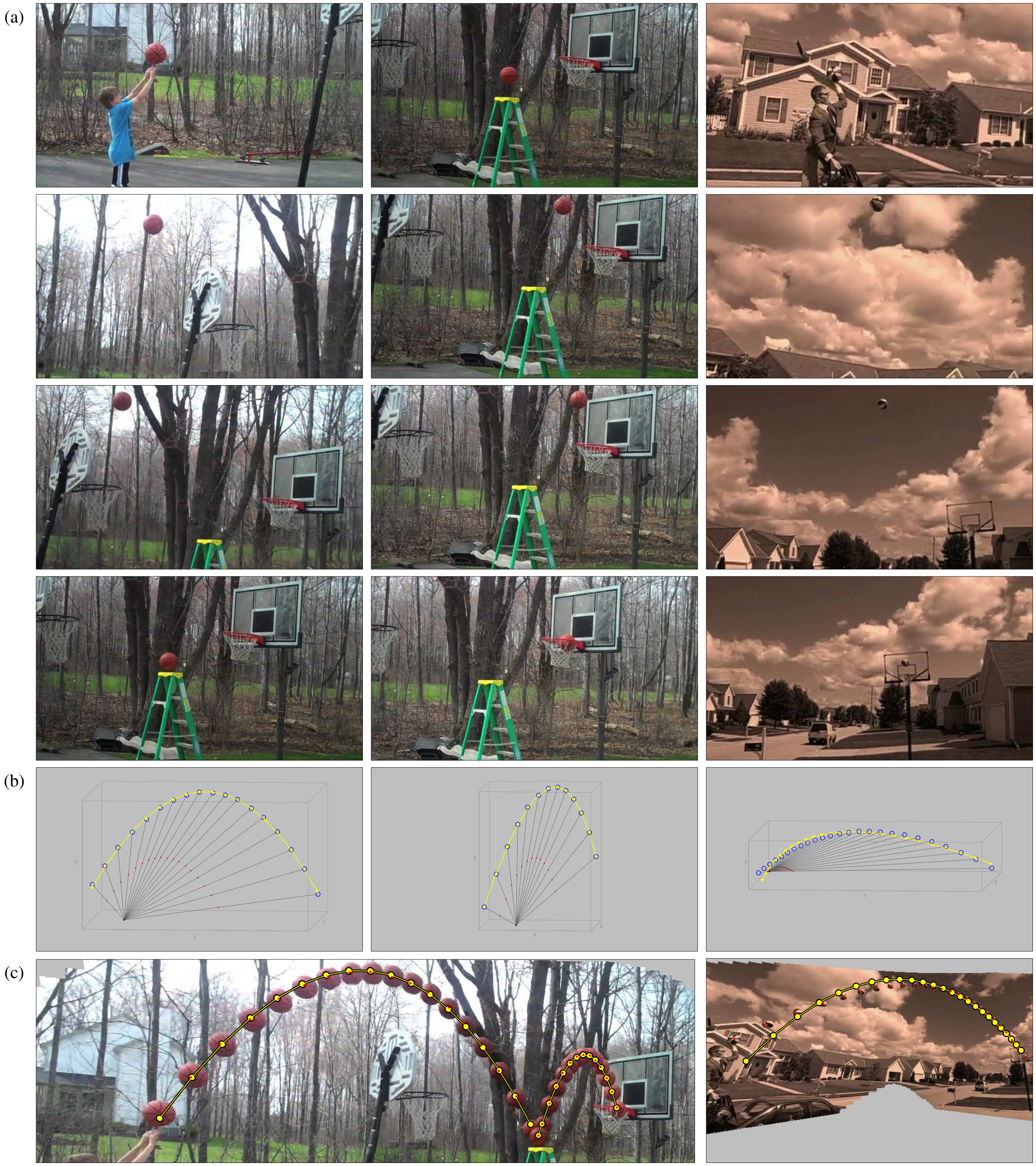
http://www.youtube.com/watch?v=wpYM3NkMYMs *http://www.youtube.com/watch?v=kDUHTioJi7A* *http://www.youtube.com/watch?v=WbaH52JI3So*

Fig. 4. Three YouTube videos from a static camera. Shown in panel (a) are four sample video frames. Shown in panel (b) are the two parametrizations of the estimated trajectory: the parametric trajectory specified by $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$ (filled yellow dots and solid line), and the non-parametric trajectory specified by the variables s_τ (open blue circles). The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Shown in panel (c) is the estimated parametric trajectory projected into the image plane (filled yellow dots) and the tracked position of the projectile (small red dots). Based on the agreement of the estimated trajectories we conclude that the first two videos are authentic while the third is fake.



http://www.youtube.com/watch?v=4n8B1jWjsq4 *http://www.youtube.com/watch?v=3v93eqwwkV0* *http://www.youtube.com/watch?v=PumYuLlHDoc*

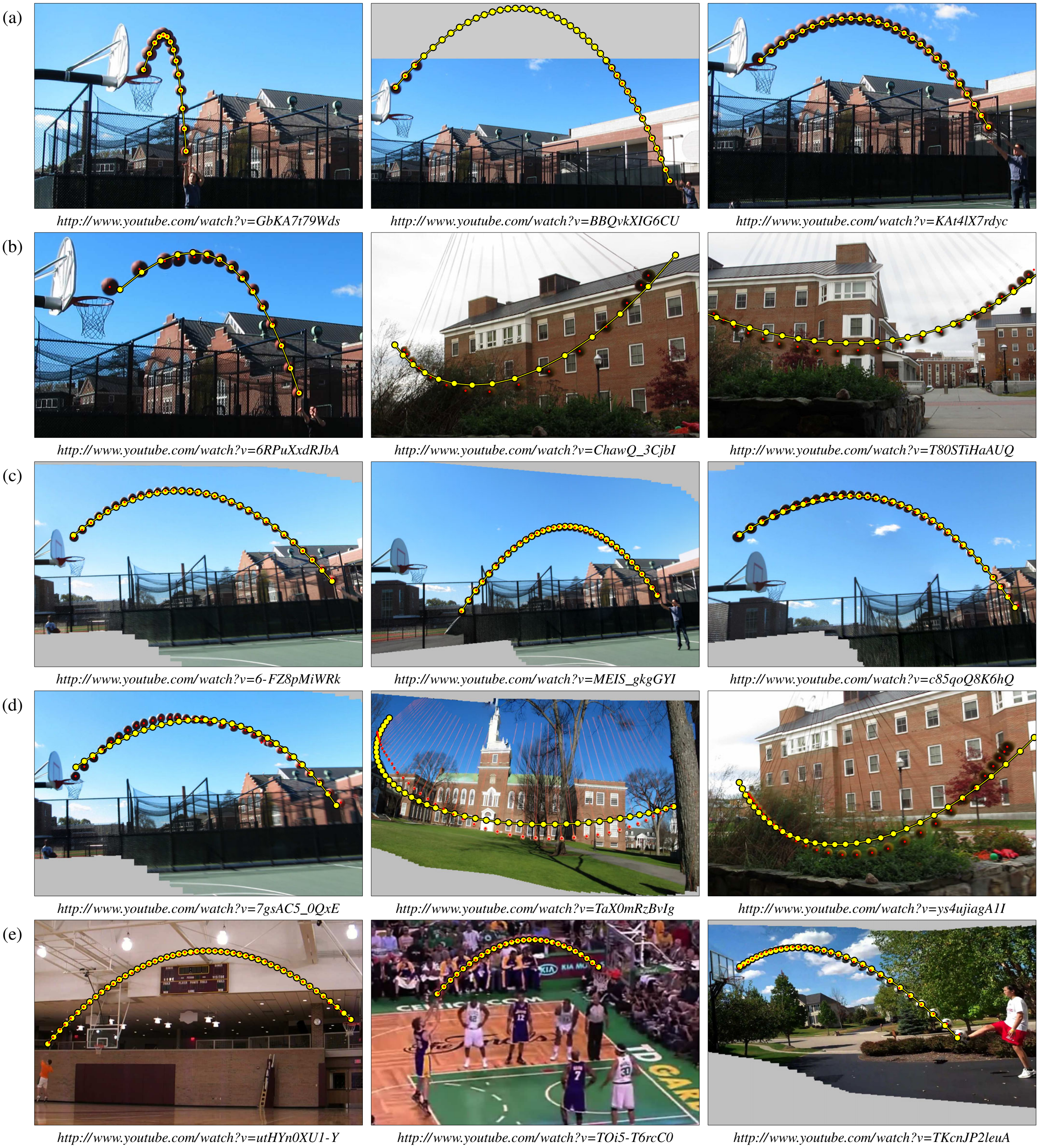
Fig. 5. One authentic video (left), one fake video (middle), and a non-ballistic motion (right) all from a moving camera. Shown in panel (a) are four sample video frames. Shown in panel (b) are the two parametrizations of the estimated trajectory: the parametric trajectory specified by $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$ (filled yellow dots and solid line), and the non-parametric trajectory specified by the variables s_τ (open blue circles). The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Note that the two parametrizations are in agreement for the authentic video but not for the fake videos. Shown in panel (c) is the estimated parametric trajectory projected into a rectified composite image plane (filled yellow dots) and the tracked position of the projectile (small red dots). These locations are in agreement for the authentic video but not for the other videos.



http://www.youtube.com/watch?v=2JwbubAX-Aw

http://www.youtube.com/watch?v=EXPOfStElLA

Fig. 6. Two YouTube videos from a moving camera. Shown in panel (a) are four sample video frames. Shown in panel (b) are the two parametrizations of the estimated trajectory: the parametric trajectory specified by $\mathbf{p}_0$, $\mathbf{v}$, $\mathbf{a}$ (filled yellow dots and solid line), and the non-parametric trajectory specified by the variables s_τ (open blue circles). The small black dot corresponds to the camera center and the small red dots correspond to the projection of the projectile in each video frame. Shown in panel (c) is the estimated parametric trajectory projected into a rectified composite image plane (filled yellow dots) and the tracked position of the projectile (small red dots). Based on the agreement of the estimated trajectories we conclude that the first video is authentic while the second is fake.



http://www.youtube.com/watch?v=GbKA7t79Wds *http://www.youtube.com/watch?v=BBQvkXIG6CU* *http://www.youtube.com/watch?v=KAt4lX7rdyc*

http://www.youtube.com/watch?v=6RPuXxdRJbA *http://www.youtube.com/watch?v=ChawQ_3CjbI* *http://www.youtube.com/watch?v=T80STiHaAUQ*

http://www.youtube.com/watch?v=6-FZ8pMiWRk *http://www.youtube.com/watch?v=MEIS_gkgGYI* *http://www.youtube.com/watch?v=c85qoQ8K6hQ*

http://www.youtube.com/watch?v=7gsAC5_0QxE *http://www.youtube.com/watch?v=TaX0mRzBvIg* *http://www.youtube.com/watch?v=ys4ujiagA1I*

http://www.youtube.com/watch?v=utHYn0XU1-Y *http://www.youtube.com/watch?v=TOi5-T6rcC0* *http://www.youtube.com/watch?v=TKcnJP2leuA*

Fig. 7. Additional test videos. Shown in panel (a) are three authentic videos from a static camera. Shown in panel (b) are three fake videos from a static camera. Shown in panel (c) are three authentic videos from a moving camera. Shown in panel (d) are three fake videos from a moving camera. Shown in panel (e) are three videos from YouTube, the first two taken from a static camera and the third from a moving camera. For all the examples, the estimated parametric trajectory projected into the image plane (filled yellow dots) and the tracked position of the projectile (small red dots) are shown. The two estimated trajectories are in agreement for authentic videos, but not for manipulated sequences.

by a static or moving camera. We have shown that the three-dimensional trajectory can be directly and reliably estimated from a video sequence. Deviations from this model provide evidence of manipulation. This forensic analysis makes minimal assumptions, requires limited user input, and is computationally efficient. We have verified the efficacy of this technique in large-scale simulations, and on numerous real-world examples.

Once a moving camera is calibrated for its extrinsic parameters, the analysis of a video from a moving camera and static camera rely on the same underlying analysis. In the case of a static camera, the primary source of error in the analysis is the localization of the projectile on each frame, particularly in the presence of motion blur. In the case of a moving camera, motion blur is often reduced because the camera typically tracks the projectile. The analysis, however, requires extrinsic camera calibration which is the primary source of error. In our simulations and experiments, however, this error is relatively small – small enough that the same error threshold can be used for the analysis of video from both a static and moving camera.

Although somewhat narrowly applicable, this forensic tool adds to a growing set of techniques for authenticating images and video. One of the advantages of geometric techniques over techniques based on low-level image statistics is that the modeling and estimation of geometry is less sensitive to resolution and compression that can easily confound the statistical analysis of images and video.

The analysis of projectiles in video also has applications in computer vision and robotics. Unlike previous approaches, our analysis does not require a stereo camera pair, explicit camera calibration, or any other information about the scene geometry. Further, our analysis is applicable for a stationary or moving camera with unknown camera motion.

As with any forensic technique, it is important to consider the ease with which counter-measures can circumvent our forensic analysis of ballistic motion. Because our analysis considers the three-dimensional trajectory of a projectile, a two-dimensional image-based manipulation is unlikely to correctly generate a physically plausible three-dimensional trajectory. This is made even more unlikely when the video is filmed with a moving camera, because there is no frame of reference from which a forger can specify the overall trajectory. A determined forger could, of course, compute a desired 3-D parabolic trajectory, project it under the appropriate homography, and then alter a projectile's path accordingly. However, such a trajectory may not always exist for scenarios that a forger wishes to depict. In particular, constraints such as combining an altered trajectory with an authentic initial segment of motion or connecting specific targets in a scene, can create situations where no physically possible ballistic motion would meet the forger's requirements. Furthermore, for a moving camera the forger would have to first align all of the frames into a common coordinate system and project edits back into the original video sequence. This counter-measure, while not impossible, is currently outside of the capabilities of available video-editing software.

While we have focused only on ballistic motions, the basic framework presented here could be applied to analyzing the motion of cars, planes, rockets, or other moving objects. This would, of course, require more sophisticated models of motion to account for the particular type of accelerations experienced by these objects.

Acknowledgments

The authors thank Marty Banks for his helpful comments and for inspiring this work by forwarding the video "megawoosh" (*http://www.youtube.com/watch?v=E2D6MJzva7E*). The authors also thank Eric Kee for his basketball skills, and helpful comments on this manuscript.

References

[1] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *ACM Multimedia and Security Workshop*, 2006, pp. 37–47.

[2] W. Chen and Y. Q. Shi, "Detection of double MPEG video compression using first digits statistics," in *International Workshop on Digital Watermarking*, 2008.

[3] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *ACM Multimedia and Security Workshop*, 2009, pp. 39–48.

[4] ——, "Exposing digital forgeries in interlaced and de-interlaced video," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 438–449, 2007.

[5] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini, "Detection of malevolent changes in digital video for forensic applications," in *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, E. J. Delp and P. W. Wong, Eds., vol. 6505, 2007.

[6] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *IEEE Workshop on Multimedia Signal Processing*, 2008, pp. 170–174.

[7] W. Van Houten and Z. J. Geradts, "Source video camera identification for multiply compressed videos originating from YouTube," *Digital Investigation*, vol. 6, no. 1-2, pp. 48–60, 2009.

[8] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *ACM Multimedia and Security Workshop*, 2007, pp. 35–42.

[9] S. Bayram, H. T. Sencar, and N. Memon, "Video copy detection based on source device characteristics: A complementary approach to content-based methods," in *ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 435–442.

[10] K. Nishiwaki, A. Konno, K. Nagashima, M. Inaba, , and H. Inoue, "The humanoid saika that catches a thrown ball," in *IEEE International Workshop on Robot and Human Communication*, 1997, pp. 94–99.

[11] M. Riley and C. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, no. 1, pp. 119–128, 2002.

[12] J. Ren, J. Orwell, G. Jones, and M. Xu, "A general framework for 3D soccer ball estimation and tracking," in *IEEE International Conference on Image Processing*, 2004, pp. 1935–1938.

[13] T. Kim, Y. Seo, and K.-S. Hong, "Physics-based 3D position analysis of a soccer ball from monocular image sequences," in *IEEE International Conference on Computer Vision*, 1998, pp. 721–726.

[14] I. Reid and A. North, "3D trajectories from a single viewpoint using shadows," in *British Machine Vision Conference*, 1998, pp. 721–726.

[15] J. Assfalg, C. M. Bertini, A. Bimbo, and W. Nunziati, "Semantic annotation of soccer videos: Automatic highlights identification," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 285–305, 2003.

[16] Y. Liu, D. Liang, Q. Huang, and W. Gao, "Extracting 3D information from broadcast soccer video," *Image and Vision Computing*, vol. 24, no. 10, pp. 1146–1162, 2006.

[17] E. Ribnikc, S. Atev, and N. Papanikolopoulos, "Estimating 3D positions and velocities of projectiles from monocular views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 938–944, 2009.

[18] J. S. Watson, M. S. Banks, C. von Hofsten, and C. S. Royden, "Gravity as a monocular cue for perception of absolute distance and/or absolute size," *Perception*, vol. 21, pp. 69–76, 1992.

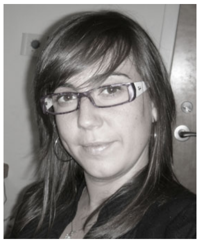
[19] H. Hecht, M. K. Kaiser, and M. S. Banks, "Gravitational acceleration as a cue for absolute size and distance?" *Perception & Psychophysics*, vol. 58, no. 7, pp. 1066–1075, 1996.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[21] T. Thormählen and H. Broszio, "Voodoo camera tracker: A tool for the integration of virtual and real scenes," http://www.digilab.uni-hannover.de/docs/manual.html, retrieved March 2011.

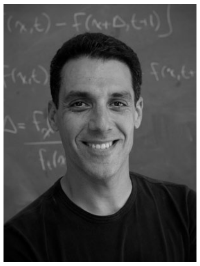
[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, pp. 91–110, 2004.

[23] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, 1972.



Valentina Conotter received her B.S. degree in Telecommunication Engineering from the Faculty of Engineering at the University of Trento in 2004, her M.S. degree in Telecommunication Engineering from the University of Trento in 2007, and her Ph.D. in Telecommunication Engineering from the ICT International Doctorate School, within the Dept. of Engineering and Computer Science (DISI) at the University of Trento in 2011. Between 2009-2010, she was a visiting student in the Image Science Group at Dartmouth College. Her research is mainly focused on active and passive multimedia forensics, including digital watermarking and digital image and video forensics.

James F. O'Brien is a Professor of Computer Science at the University of California, Berkeley. His primary area of interest is Computer Animation, with an emphasis on generating realistic motion using physically based simulation and motion capture techniques. He has authored numerous papers on these topics. In addition to his research pursuits, Prof. O'Brien has worked with several game companies on integrating advanced simulation physics into game engines, and his methods for destruction modeling were recently used in the film Avatar. He received his doctorate from the Georgia Institute of Technology in 2000, the same year he joined the Faculty at U.C. Berkeley. Professor O'Brien is a Sloan Fellow and ACM Distinguished Scientist, Technology Review selected him as one of their TR-100, and he has been awarded research grants from the Okawa and Hellman Foundations. He is currently serving as ACM SIGGRAPH Director at Large.



Hany Farid received his undergraduate degree in Computer Science and Applied Mathematics from the University of Rochester in 1989. He received his Ph.D. in Computer Science from the University of Pennsylvania in 1997. Following a two year post-doctoral position in Brain and Cognitive Sciences at MIT, he joined the faculty at Dartmouth College in 1999, where he is currently a professor of computer science. Hany is the recipient of an NSF CAREER award, a Sloan Fellowship and a Guggenheim Fellowship.