Experiencing the Ball's POV for Ballistic Sports

Kodai Horita,
Hideki Sasaki, Hideki Koike
University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
horita,sasaki,koike@vogue.is.uec.ac.jp

Kris M.Kitani
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA. USA
kkitani@cs.cmu.edu

ABSTRACT

We place a small wireless camera inside an American football to capture the ball's point-of-view during flight to augment a spectator's experience of the game of football. To this end, we propose a robust video synthesis algorithm that leverages the unique constraints of fast spinning cameras to obtain a stabilized bird's eye point-of-view video clip. Our algorithm uses a coarse-to-fine image homography computation technique to progressively register images. We then optimize an energy function defined over pixel-wise color similarity and distance to image borders, to find optimal image seams to create panoramic composite images. Our results show that we can generate realistic videos from a camera spinning at speeds of up to 600 RPM.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Miscellaneous—*Artificial, Augmented, and Virtual realities.*

Keywords

Digital Sports, BallCam, Video Synthesis, Image Stitching.

1. INTRODUCTION

The visual component of experiencing sports through video is a critical aspect of experiencing the game. In many sports, videos of the game are taken from a third person point-of-view (POV). The third POV is effective for broadcasting many of the global aspects of sports (e.g., large-scale team dynamics, excitement and response of the crowd). However, there is an entirely hidden experience of the game that is only accessible to the players on the field – the perspective of the players and the ball. In an effort to bring the audience 'closer' to the game, technologies such as the SkyCam (cable-suspended camera system) offer a perspective that more closely resembles the visual experience of being on the field. In this work, we desire to extend the frontiers of the spectator experience, by capturing the excitement of being 'on the field' through the ball's first-person POV.

Figure 1: American Football from a Ball's POV

While the idea of putting a camera in a spinning ball is certainly not new [1], recent advances in technology now provide a strong foundation on which such a system can be realized. The framework proposed in this work is made possible by key technological advances in wearable camera devices and computer vision. Wearable cameras such as the GoPro Hero can now capture high-quality images at a frame rate of up to 240 FPS, at a price accessible to the general public. These wearable cameras perform well under significant camera motion and are used for sports such as Formula 1, surfing and sky-diving. In the field of computer vision, the design of robust local visual feature for cross image matching [8] and physics-based energy minimization techniques [2] applied to image stitching now allow everyday users to create wide angle panoramic images with a cellphone camera [3].

Based on these underlying technologies, we will show that we can generate a new viewing experience from a ball's POV using an embedded ball-camera system (Figure 1). In particular, we capture high frame-rate images with a wireless camera system and use image compositing techniques to seamlessly blend images to create a set of wide-angle images.

Using these images, we then generate a downward looking video (a sequence of images) by interpolating the camera motion between frames. The result is a highly dynamic downward-looking video which gives the viewer a sense of flying with the ball.

In this work, we improve on the initial prototype of Kitani *et al.* [6] in three ways: (1) removal of rolling-shutter distortion, (2) a robust coarse-to-fine image homography computation technique, and (3) an energy-based image stitching technique to handle more complex image compositing. Previously, H. Mori *et al.* [9] developed a wired ball-camera system using multiple cameras to generate a stabilized video. Their system used an optical flow-based approach to estimate camera rotation parameters. Kuwa *et al.* [7] generated a static wide-range image using a throwable camera. The throwable camera system created a ribbon image by connecting images taken by a single camera. Pfeil *et al.* [11] introduced a system to capture a single static spherical panorama by triggering an array of cameras with an accelerometer. Their system detects the highest point of the ball's trajectory to trigger the shutters. After downloading the picture from the ball, they generate a panoramic image from the ball's POV. In contrast to past systems, our system generates a dynamic wide-angle video with a single camera and is able to deal with extremely fast camera motion.

2. SYNTHESIZING THE BALL'S POV

Since a typical football rotates at roughly 600 RPM while in flight, the high-speed motion of the camera poses several significant challenges that make the task of extracting a stabilized downward looking video very difficult. In this section, we explain how, under certain domain assumption and by leveraging the unique characteristics of camera images recorded under high-speed rotation, we can create plausible ball POV videos in the context of American football. In particular, we outline a method for removing camera distortion (Section 2.1), view expansion (Section 2.2), and video motion interpolation (Section 2.3).

2.1 Removing Camera Distortion

The images taken by our spinning BallCam are significantly distorted due to the rolling-shutter effect (the camera motion is faster than the CMOS camera array read-in time). In this work, we introduce a rolling-shutter distortion removal step to overcome the image deformation which was not accounted for in [6]. As depicted in Figure 2, notice the severe image deformation before removing distortion, *e.g.*, the lines on the football field are curved. However, after removing the lens and rolling-shutter distortion, we observe that the deviation of the lines have decreased.

We begin by removing the barrel distortion of the lens and the rolling shutter distortion caused by the CMOS sensor acquisition latency. If a camera has no lens distortion, an image point distorted by the rolling-shutter effect $\boldsymbol{x}_d$ and its rectified position $\boldsymbol{x}_r$ are related by the following equation [5]:

$$\boldsymbol{x}_r = KR^\top(t)K^{-1}\boldsymbol{x}_d \tag{1}$$

where K is the calibration matrix of the camera and $R(t)$ is the camera rotation. Since the camera is moving, the rotation depends on time t, and t is different for each image row.

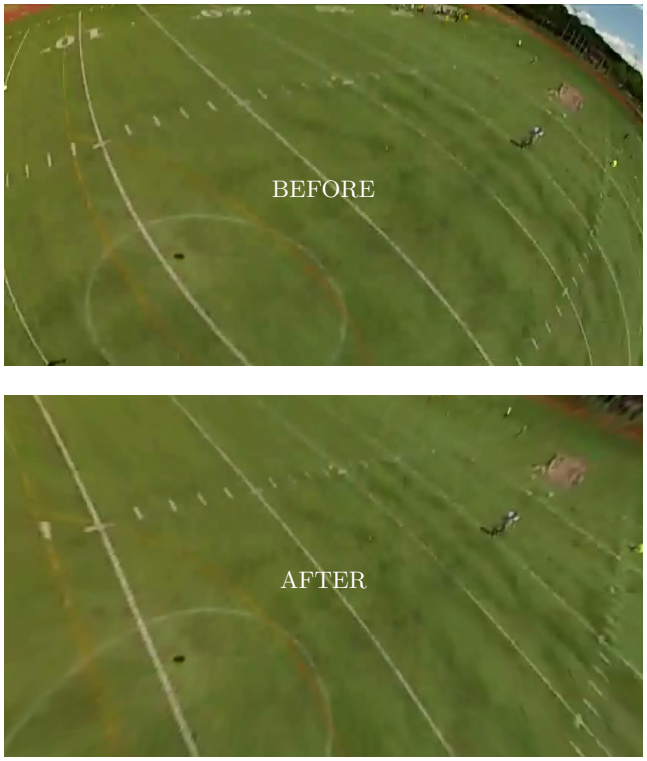


Figure 2: Result of rolling-shutter and lens distortion removal.

In order to correct the lens distortion and the rolling-shutter distortion simultaneously, we extend (1) as:

$$\boldsymbol{x}_r = KR^\top(t)\mathcal{P}^{-1}(\boldsymbol{x}_d) \tag{2}$$

where $\mathcal{P}^{-1}$ represents the back-projection from the image plane to 3D space. Note that this back-projection function takes the lens distortion into account. We use a simple radial distortion model in our implementation [4].

To perform image rectification, we need to compute the distorted image point $\boldsymbol{x}_d$ for each rectified image pixel position $\boldsymbol{x}_r$. We use the Gauss-Newton method to solve (2) for $\boldsymbol{x}_d$ and create a lookup table for image rectification. This result of this procedure is illustrated in Figure 2.

2.2 View Expansion

Following [6] we use the mean intensity of the images to first generate a sequence of images that share a similar viewing angle. However, simply interpolating between sub-sampled image results in a very shaky video since the axis of rotation is not perfectly orthogonal to the camera axis and the camera rotation is not in sync with the camera frame-rate. Before we proceed to synthesize novel views between images to temporally up-sample the video, we would like to first expand the field-of-view for each image frame. This is needed because there is often large displacements between subsampled frames and we must ensure we can interpolate

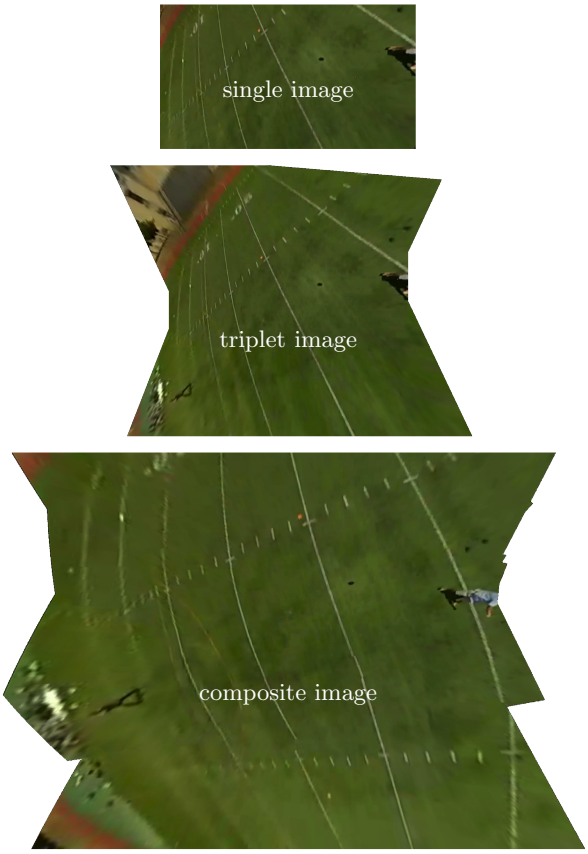


Figure 3: View expansion.

between images without 'holes' in the image frames. We do this by first generating small composite images using sets of 3 temporally neighboring images. Next, we further expand the triplet image by using neighboring triplet images from adjacent rotation cycles (Figure 3).

2.2.1 *Coarse-to-Fine Homography Estimation*

To generate composite images, it is necessary to compute the image transformation between images by computing their homographies. However, since the football field has many repetitive patterns and the image distortion makes feature matching very challenging, we introduce a robust coarse-to-fine homography estimation technique to ensure reliable image registration. In the coarse step, we assume an affine motion model that only allows for translational motion T, by solving the following linear equation (4),

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3}$$

where

$$A = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}, \tag{4}$$

and x and y are the feature points in current frame image, x' and y' are the point of feature point in the next frame.

We can solve for A via linear regression and then proceed

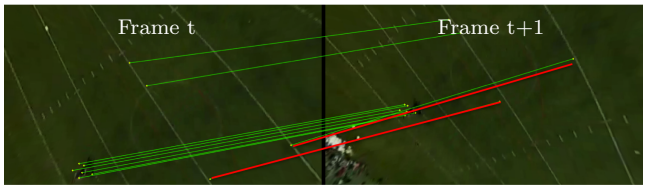


Figure 4: Bad point correspondences (marked in red) identified via a coarse-to-fine motion estimation

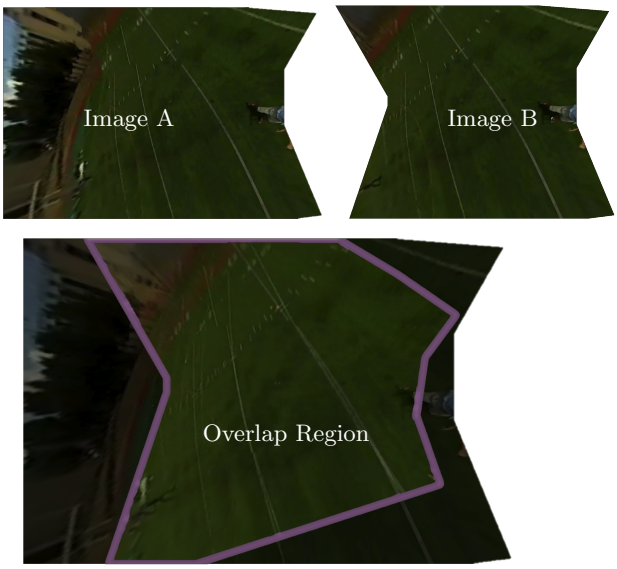


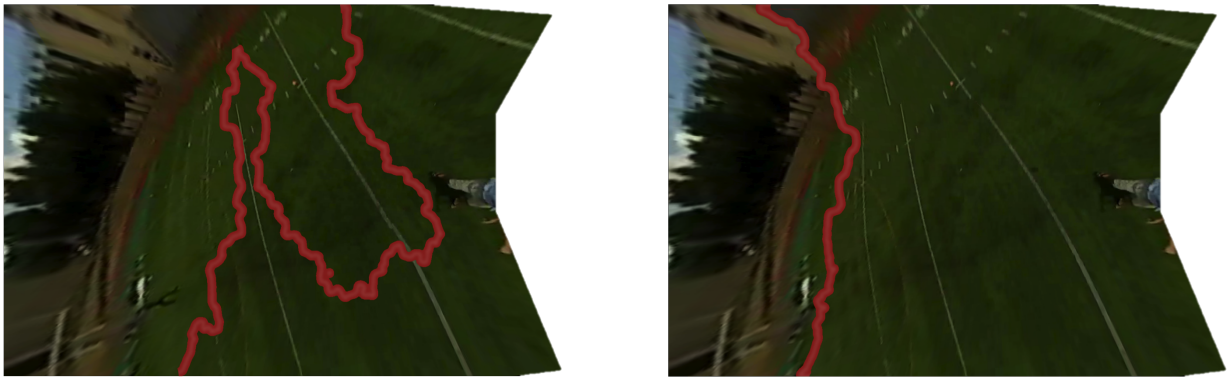
Figure 5: Overlap region from merging images.

to remove outlier points which have high reprojection error. With the remaining points we use RANSAC to estimate a full homography matrix H to account for changes in perspective and scaling.

2.2.2 *Image Stitching with Graphcuts*

When we merge two images together to create a larger composite image we must be careful about how we blend the two images together to avoid various types of image noise. In our method we overlap two images and find the optimal seam at which we can merge the two images. Figure 5 illustrates the overlap region caused by stacking two images. Our goal is to compute an optimal seam through this overlap area to merge the two images.

Simply blending the composite images using a homography causes significant image noise since the world is not static and planar. In contrast to [6] and [10] which only finds a single seam, we pose the image merging problem as a graph cutting problem that allows for multiple seams to be computed. Formally, we use the pairwise color continuity cost and inverse distance to image boundary cost as our binary and unary potentials, respectively, to define our cost function.



Color cost only

Color and distance cost

Figure 6: Optimal seam computed with the distance cost preserves spatial continuity.

The pairwise color continuity cost between two neighboring pixels i and j, is computed as the color difference between neighboring foreground pixels $\boldsymbol{f}(\cdot)$ and background pixels $\boldsymbol{b}(\cdot)$,

$$C_{\text{color}}(i,j) = \frac{1}{N}\left(||\boldsymbol{f}(i) - \boldsymbol{b}(i)||^2 + ||\boldsymbol{f}(j) - \boldsymbol{b}(j)||^2\right), \quad (5)$$

where $\boldsymbol{f}(\cdot)$ and $\boldsymbol{b}(\cdot)$ are 3 dimensional vectors of RGB values and the indices i and j denote neighboring pixels. The normalization factor N ensures that the value is between 0 and 1.

While the pairwise color continuity cost does finds the lowest cost seam, it does not always preserve spatial continuity in the center of the overlapped region. This effect is illustrated in Figure 6. The spatial continuity of the center of the composite image is extremely important for the viewing experience as it contains the most important information about the game. In our experience, bad image compositing in the center of the image is much more distracting that errors at the perimeter of the image. Therefore, in order to preserve spatial continuity of the center of the image we also introduce a unary cost term that quantifies the inverse distance to image boarders to increase the cost of seams that cross the center of the overlapped region,

$$C_{\text{dist}}(i) = \min_{\boldsymbol{b}\in B} ||\boldsymbol{b} - \boldsymbol{x}(i)||_{L2}^{-1}, \quad (6)$$

where b is the coordinate of the nearest boundary location of all the boundary points B and $\boldsymbol{x}(i)$ is the coordinate of i. This cost term has the effect of generating seams that are near the image overlap boundaries.

2.3 Video Motion Interpolation

To generate a virtual camera path, we compute a series of image warps based on the computed homographies between composite images. From a homography we can compute a per-pixel mapping, by applying the homography as follows,

$$z = h_{20} \times x + h_{21} \times y + h_{22}, \quad (7)$$

$$x' = (h_{00} \times x + h_{01} \times y + h_{02})/z, \quad (8)$$

$$y' = (h_{10} \times x + h_{11} \times y + h_{12})/z, \quad (9)$$

where z is the normalization constant, x' and y' are the transformed points of the x and y and h_{ij} is an element of the homography matrix,

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}. \quad (10)$$

The forward mapping $M_f(x,y)$ describes how a single pixel in frame t can be forward mapped to a pixel in frame $t+1$. We can generate an arbitrary view between these two frames using linear interpolation between an identity mapping $M_I(x,y)$ and the forward map M_f or backward mapping M_b, for frame t and $t+1$, respectively,

$$F(x,y) = (1-\alpha)M_I(x,y) + \alpha M_f(x,y), \quad (11)$$

$$B(x,y) = (\alpha)M_I(x,y) + (1-\alpha)M_b(x,y). \quad (12)$$

The new mapping F is indexed by the parameter $0 \leq \alpha \leq 1$ and contains the mapping from the image at time t to $t+1$. Likewise, B contains the mapping from image $t+1$ back to image t. For example, when the weight α equal 0, $F(x,y)$ is exactly $M_I(x,y)$ (image t remains unchanged) and $B(x,y)$ is exactly $M_b(x,y)$ (image $t+1$ has been projected to the coordinate frame of image t). By gradually increasing the value of α we can generate a synthesized video of the motion between image t and image $t+1$.

However, simply applying linear interpolation between image homographies (first-order interpolation), introduces a undesirable high-frequency motion component as an artifact of aliasing (i.e. the synthesized view sways from side to side). This is mainly caused by the fact that the balls axis of rotation is not exactly aligned with the major axis of the ball. In order to account for this camera motion noise, we use second-order interpolation by interpolating between half maps.

We generate half maps by forward warping images to the half way point (*i.e.*, $\alpha = 0.5$) and we then recompute all of the homography between these intermediate images. Then we use linear interpolation to warp between two successive intermediate images. This has the effect of minimizing the swaying motion introduced by off-axis ball rotation.

3. RESULTS AND DISCUSSION

Figure 7 shows several the results generated by our ball's POV system. Notice that the image distortion has been reduced and the image stitching has joined to together neighboring images to create a very wide-angle image. However, the results are best viewed as video which can be found on the author's website.

We have shown that we can remove much of the image distortion caused by the imaging conditions. However, close examination of the images will show remaining evidence of image distortion (lines are still slightly curved) and motion blur. Since our model of rolling-shutter distortion assumes no distortion over image rows, this assumption is actually being violated for very fast spinning cameras and will require a more expressive distortion model for high-quality rectification. As for the motion blur, this can be addressed with a faster camera sensor or techniques such as motion deblurring and image super-resolution.

Our current framework also assumes that we roughly know a ball's axis of rotation, which is only true in the case of an American football thrown with a clean spiral. This is obviously not the case for other types of ballistic sports. Future work will focus on developing techniques that can automatically infer the ball's rotational axis through visual motion estimation or additional motion sensors.

4. POTENTIAL APPLICATIONS

In the current context of American football we believe that our approach can be used to augment the current viewing paradigm by providing a ball's POV in highlight videos of pass completions. Our BallCam will give the audience access to the up-close battle that ensues as defenders try to grab the ball away from the receiver.

Our system can also be combined with other image-based technology such as tracking and face detection to generate human-centric videos focused on a particular player. We also envision potential interactions with other cameras that are already available on the playing field. For example, by orchestrating the movement of the ballCam, cable-suspended cameras and wide-area camera, we can also stitch images and videos across different different cameras, providing the ability to zoom-in and zoom-out on a very dynamic scale (similar to first-person POV video games).

5. CONCLUSION

We have proposed a robust method for generating novel ball's POV video sequences from a spinning camera. Our system shows a new type of spectator technology that can be applied to ball-based sports. We believe that our prototype system is a strong proof-of-concept that shows that embedded ball-camera systems have the potential to change the viewing paradigm of ballistic sports.

6. REFERENCES

[1] Camera inside a football films view from air. Popular Mechanics, December 1938.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[3] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

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

[5] J. Hedborg, E. Ringaby, P.-E. Forssén, and M. Felsberg. Structure and motion estimation from rolling shutter video. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 17–23, 2011.

[6] K. Kitani, K. Horita, and H. Koike. Ballcam!: dynamic view synthesis from spinning cameras. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 87–88, 2012.

[7] T. Kuwa, Y. Watanabe, T. Komuro, and M. Ishikawa. Wide range image sensing using a thrown-up camera. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 878–883, 2010.

[8] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[9] H. Mori, D. Sekiguchi, S. Kuwashima, M. Inami, and F. Matsuno. Motionsphere. In *ACM SIGGRAPH 2005 Emerging technologies*, page 15, 2005.

[10] T. Ozawa, K. M. Kitani, and H. Koike. Human-centric panoramic imaging stitching. In *Proceedings of the 3rd Augmented Human International Conference*, page 20, 2012.

[11] J. Pfeil, K. Hildebrand, C. Gremzow, B. Bickel, and M. Alexa. Throwable panoramic ball camera. In *SIGGRAPH Asia 2011 Emerging Technologies*, page 4, 2011.

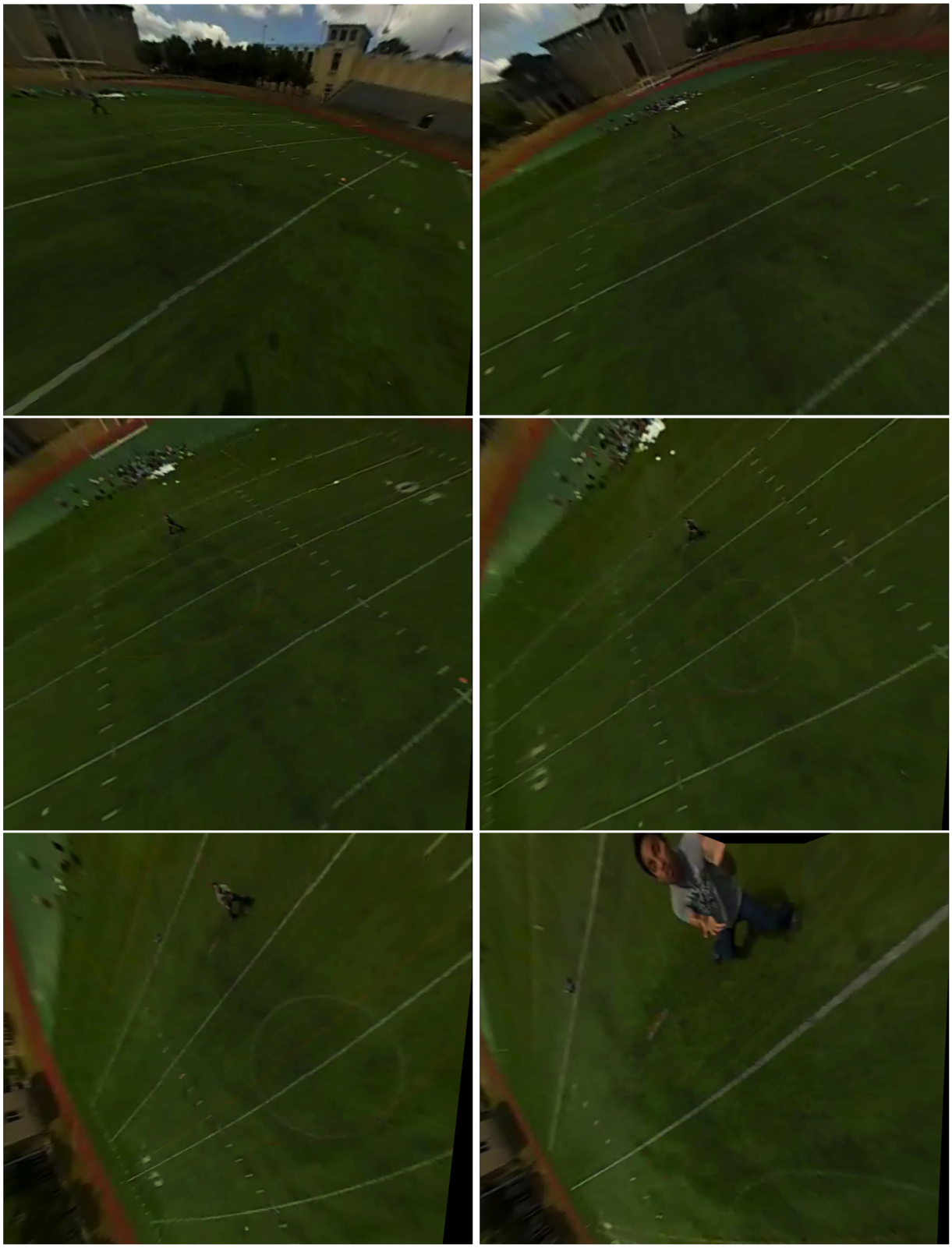


Figure 7: Sample images from the video sequence