Hybrid Robotic/Virtual Pan-Tilt-Zoom Cameras for Autonomous Event Recording

Peter Carr
Disney Research
Pittsburgh, USA
carr@disneyresearch.com

Michael Mistry
University of Birmingham
Birmingham, UK
m.n.mistry@bham.ac.uk

Iain Matthews
Disney Research
Pittsburgh, USA
iainm@disneyresearch.com

ABSTRACT

We present a method to generate aesthetic video from a robotic camera by incorporating a virtual camera operating on a delay, and a hybrid controller which uses feedback from both the robotic and virtual cameras. Our strategy employs a robotic camera to follow a coarse region-of-interest identified by a realtime computer vision system, and then resamples the captured images to synthesize the video that would have been recorded along a smooth, aesthetic camera trajectory. The smooth motion trajectory is obtained by operating the virtual camera on a short delay so that perfect knowledge of immediate future events is known.

Previous autonomous camera installations have employed either robotic cameras or stationary wide-angle cameras with subregion cropping. Robotic cameras track the subject using realtime sensor data, and regulate a smoothness-latency trade-off through control gains. Fixed cameras post-process the data and suffer significant reductions in image resolution when the subject moves freely over a large area.

Our approach provides a solution for broadcasting events from locations where camera operators cannot easily access. We can also offer broadcasters additional actuated camera angles without the overhead of additional human operators. Experiments on our prototype system for college basketball illustrate how our approach better mimics human operators compared to traditional robotic control approaches, while avoiding the loss in resolution that occurs from fixed camera system.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Experimentation, Theory

Keywords

Camera, control, tracking, planning

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502086.

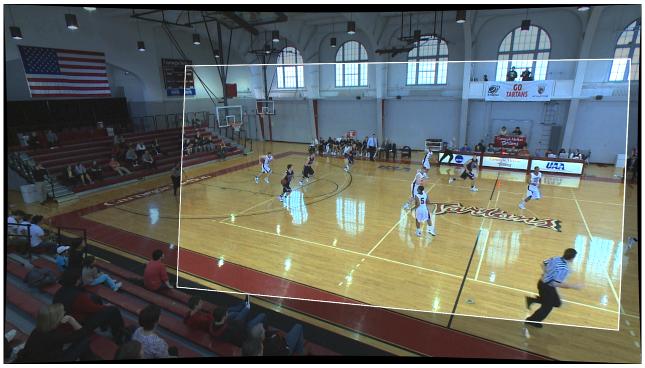

Figure 1: Hybrid Camera. We first use a robotic pan-tilt-zoom camera to follow the centroid of player positions estimated by a realtime person detector. After a short delay, we then resample the image to generate the video frame of a virtual camera (highlighted) which has a smooth camera motion trajectory because the path planning algorithm can take into account how the players will move in the immediate future (see accompanying video for full effect).

1. INTRODUCTION

Robotic cameras which can record live events autonomously have the potential to streamline and improve current broadcast production models. For example, the optimal shooting location may be impractical for a human operator (which arises often in sports) or the cost of hiring a professional camera operator may not justify the benefit of an additional perspective. In these instances, methods to improve or automate the control of robotic cameras would be extremely useful. Automatically planning where the camera should look is a key challenge [14], especially when the source information is noisy sensor data. Fundamentally, the camera must be controlled to ensure it tracks the intended target. Although path planning and camera control are separate tasks, the two are highly coupled: there is no point planning a motion path which the camera is physically unable to follow.

The camera should also move smoothly and purposefully to avoid disorienting the viewer. Although smooth motion can be achieved through low control gains (which limit changes in acceleration), the resulting system will be unable to follow dynamic object trajectories. Instead, the autonomous camera should mimic human behavior and plan a trajectory which balances smooth motion against tracking error. As a result, planning requires anticipating object motion, or as Owens writes: “understanding the intricacies of the event . . . gives camera operators the ability to predict how they should be moving their cameras” [20].

An online, realtime system is preferable because the event can then be broadcast live. Furthermore, if the sensor data can be processed in realtime, a robotic pan-tilt-zoom (PTZ) camera can follow the subject of interest and capture high-resolution images using all pixels from its image sensor. However, to output aesthetic video, the system must also be able to anticipate future object locations so that a smooth trajectory can be planned. Previous online autonomous robotic camera systems have only been employed in environments with limited dynamic motion such as lecture halls and video conference facilities. We handle

Team sports, on the other hand, have highly dynamic object motions. As a result, all implementations to date have employed non-realtime offline resampling approaches. In the resampling framework, one or more high-definition stationary cameras capture live action, and video from a virtual camera is synthesized after the fact by resampling the pixels from the fixed cameras (often by simply cropping a rectangular subregion). As Gleicher and Masanz [11] point out, the offline approach is attractive because complex non-realtime algorithms can be used to plan the camera trajectory, and more importantly, there is no need to accurately anticipate future object motions because the true future motion information is readily available. Similarly, there are no control issues because the virtual camera can move anywhere immediately. The downside of resampling is that only a fraction of the system’s resolution is used in the output. For instance, in a sport like basketball, all players typically occupy only half the court at any given time. Therefore, with a set of fixed cameras covering the entire court, at least half the recorded pixels would never be used in the output video. In addition, it is impossible to gain high-resolution close-up images.

In this work, we propose a hybrid robotic/virtual camera which balances the strengths and weakness of both online and offline approaches: a robotic PTZ camera is equipped with a wide angle lens and tracks sports players in realtime, and a second virtual camera resamples the original source video in realtime, but on a short temporal delay (see Fig. 1). Our proposed solution has several interesting properties:

- The hybrid camera remains an online, realtime system with small latency, but gains the offline benefit of perfect knowledge about future events (up to the duration of the induced temporal delay).
- The synthesized video of the virtual camera exhibits minimal loss in image resolution (unlike offline resampling systems) because the robotic camera follows the action.
- The hybrid camera is a redundantly actuated system, which presents certain control issues.

The amount of delay is a critical design factor which regulates the hybrid nature of the system: zero delay produces a completely online system, and infinite delay produces a completely offline system. A longer delay improves the virtual camera’s ability to plan a good trajectory because it has knowledge about events further into the future. However, the robotic camera must be controlled in realtime to ensure the virtual camera remains within the frame boundary so that the resampling process has sufficient information to synthesize the image that would have been captured had the robotic camera looked where the virtual camera is now looking. Longer delays make the controlling feedback loop more unstable. As a result, our hybrid camera has a suitable delay which balances the benefits of improved planning against the drawbacks of more difficult control.

We demonstrate the effectiveness of our proposed hybrid robotic/virtual camera through a prototype system deployed for broadcasting college basketball games. Our experiments demonstrate how the introduction of a small delay improves the system’s ability to plan a smooth and purposeful path for the virtual camera using a hysteresis filter which appears to anticipate state changes because of knowledge about the immediate future. The robotic camera is controlled to follow a live prediction of where the virtual camera is expected to look, as well as monitoring where the virtual camera is actually looking. As expected, we show how the control stability of the robotic camera decreases as delay increases. Our supplementary video shows how a hybrid camera is able to produce live footage which more closely resembles the work of a human operator.

2. RELATED WORK

Autonomous Cameras

Previous work in autonomous camera systems for sports production [1,3,4,7] has employed a common framework: one or more high-resolution fixed cameras capture the game, and features such as player and ball locations are extracted offline. The output broadcast is then generated as a post process by determining the optimal subregion of the appropriate fixed camera at each time instant. There has been significant variety in how the optimal subregion is determined at each time instant. Daigo and Ozawa [7] augment player features with audience gaze angles. Images of the three fixed cameras are stitched together using cylindrical projection and a rotational offset based on player and audience gaze features. Ariki *et al.* [1] considered three different shot sizes depending on the estimated game situations. A smooth path was achieved using a Schmitt trigger which only put the camera in motion when the ball neared the edge of the frame. Chen and De Vleeschouwer [3] generated a virtual camera trajectory for basketball using an MRF chain to balance smoothness against deviating from the optimal virtual camera state at each time instant.

In addition to sports, autonomous camera systems have been deployed in lecture halls, video conferences, and television production stages. In these situations, the motion of subjects is significantly less dynamic than team sports, and a range of camera solutions has been employed. Pinhanez and Bobick [21] demonstrated how a user-supervised autonomous camera system could automatically frame shots for a cooking show. Various vision algorithms were deployed depending on the type of shot as requested by the human director. Yokoi and Fujiyoshi [24] used a fixed 1080i camera to record a lecturer. A cropping window was computed from frame differencing, and the authors investigated both bilateral filtering and human specified control points for a learned acceleration model to smooth the noisy input signal. Sun *et al.* [23] controlled a virtual camera to record a lecturer. The motion of the virtual camera was regulated using a Kalman filter augmented with a three state rule-based post filtering technique to prefer a stationary cameras unless the lecturer was moving significantly. Zhang *et al.* [25] use a fixed camera to estimate a saliency map of the video conference room and computed an optimal cropping window which balanced a loss of information from aperture and resolution effects. Instead of cropping from the wide-angle camera, the desired subregion is used to control a robotic PTZ camera.

Our work is most similar to Zhang *et al.* [26] who also used a hybrid robotic/virtual camera. The robotic camera tracked a lecturer and moved as necessary to keep the subject in the center of the image. A subregion of the image was then cropped to compensate for motor control errors. There was no delay between the virtual and robotic cameras. In effect, the virtual camera was to used to achieve ideal perfect control by compensating for any discrepancy between the plan for where the physical camera was supposed to look, and where the physical camera was actually looking. In a more complex environment, such as basketball, accurate prediction of object motion is necessary for generating aesthetic video. Instead, we operate a virtual camera on a delay to avoid the need to anticipate player movements.

Path Planning and Control

Determining where the cameras should look is a key component of any autonomous system. Additionally, the planned trajectory must be smooth, which means the process to decide where the camera should look at any given time instant must take into account where the camera should be looking during a temporal window which spans both before and after the current time instant.

Camera planning continues to be a popular topic in computer graphics (see the recent survey by Christie *et al.* [5]). However, computer graphics algorithms rarely consider incomplete and noisy data generated from computer vision and other sensing modalities.

The tasks of moving a physical camera to keep an object of interest within the field of view is referred to as *visual servoing* in the robotics literature. Stanciu and Oh [22] employed a proportion-only feedback control algorithm to adjust the pan-tilt angle of a camera mounted on the end of a human operated boom to keep a target object in the center of the camera image. When multiple targets are tracked, control algorithms often monitor features derived from the point set, such as mean and standard deviation. Farag and Abdel-Hakim [9] use proportion-only control to position the centroid of detected image features near the centers of the images of a stereo camera pair. Gans *et al.* [10] use task-priority kinematic control to keep a set of interest points within the camera field of view. They showed how the mean and variance are independent objectives: pan-tilt values are modified to keep the mean near the center of the image, and zoom is regulated to keep the standard deviation within the image boundary.

Two recent works in machine learning and computer vision have examined the problem of determining where cameras should look based on player motions. Dearden *et al.* [8] used a K nearest neighbor classifier to learn the relationship between features (such as player position) and the PTZ state of cameras operated by professionals. Kim *et al.* [17] track individual players using a particle filter and extrapolate a global motion vector field on the ground plane using Gaussian process regression. The authors show how convergence regions in the vector field correlate with actual broadcast camera movements.

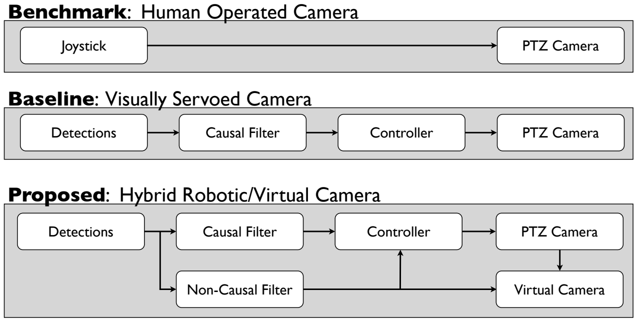

Figure 2: Camera Operation Techniques. We compare our proposed hybrid robotic/virtual camera operation technique to an autonomous camera operated using visual servoing. Our objective is to mimic human operation, so we express the performance of both the baseline and proposed autonomous systems relative to the benchmark of a human operated camera.

Finally, both robotics and computer vision have examined the problem of planning smooth trajectories for cameras. Nieuwenhuisen and Overmans [19] use a probabilistic roadmap to generate an initial estimate of linear segments which link the current camera state to the desired future camera state. The path is refined by fitting circular arcs between segments and computing a smooth velocity plan which depends on path curvature. Gleicher and Liu [12] developed a video stabilization technique by estimating the trajectory of a hand held camera using inter-frame homographies, and identifying segments of constant velocity linked together with ease in/out curves. Grundmann *et al.* [13] refine a noisy trajectory using a linear program which generates a trajectory preferring constant postion or constant velocity segments.

3. EXPERIMENTAL DESIGN

In this work, we investigate two critical aspects of robotic camera operation: path planning and motor control. A planning algorithm generates a saliency signal $\mathbf{s}(t) = [\hat{\phi}, \hat{\theta}, \dot{\hat{\phi}}, \dot{\hat{\theta}}]^\intercal$ specifying a desired instantaneous pan-tilt position $(\hat{\phi}, \hat{\theta})$ and velocity $(\dot{\hat{\phi}}, \dot{\hat{\theta}})$ for the camera at every time instant. The control algorithm regulates the speed of the pan-tilt motors so that the camera follows the planned state space trajectory $\mathbf{s}(t)$ as best as possible.

To gauge the success of our proposed hybrid robotic/virtual camera solution, our experimental set-up consist of three different camera operation techniques (see Fig. 2):

Benchmark Our objective is to mimic a human operator. Therefore, we measure the performance of a human operator using a joystick with direct control over the pan-tilt motor speeds.

Baseline We use a standard visual servoing implementation as our baseline: a realtime, zero-latency moving average causal filter generates a saliency signal from noisy player detections. An instantaneous position error proportion-only controller is used to regulate the pan-tilt motor speeds.

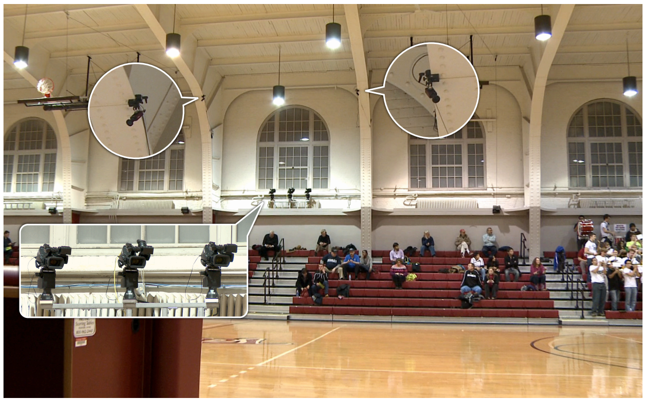

Figure 3: Experimental Setup. Two Allied Vision GX 1920C machine vision cameras mounted near the ceiling of the gym are used to detect and track the basketball players (see Fig. 4). Three Sony EX3 cameras situated behind the spectator seating at center court capture the broadcast video. Each camera is mounted on a FLIR D48-E robotic pan-tilt head. The motion of each camera is determined from either a human operated joystick, or an autonomically generated plan based on the realtime analysis of the player positions (see Fig. 5).

Proposed Our proposed solution employs a zero-latency causal filter for the robotic camera, and an M second delay inducing non-causal filter for the virtual camera. The position errors of both the robotic and virtual cameras are used to regulate the speeds of the pan-tilt motors or the robotic camera.

Because each technique regulates the pan-tilt motor speeds in a different way, our experimental setup (see Fig. 3) includes three Sony EX3 cameras mounted on FLIR D48-E robotic pan-tilt units. As a result, we are able to compare the three operation techniques on the same live data. The benchmark camera requires a human operator, whereas the baseline and proposed autonomous techniques both rely on a realtime computer vision system to detect the (x, y) locations of basketball players. Therefore, our setup includes two Allied Vision GX 1920C machine vision cameras mounted near the ceiling. We use the method of Carr *et al.* [2] to detect players at 25fps with one frame latency (see Fig. 4).

Changes in tilt are highly unaesthetic, so we employ a constant tilt angle for all three Sony EX3 broadcast cameras. For simplicity, the remainder of the paper only refers to the planning and control aspects of each camera's instantaneous pan position and velocity, but the theory and techniques which will be discussed shortly apply equally well to tilt. Similarly, we only consider the X coordinate of locations on the basketball court (corresponding to the lengthwise axis).

We first describe how a path is planned in each of the three scenarios, and examine how planning improves with increased delay. We then discuss the details of the resampling algorithm used to synthesize the images of the virtual camera. Finally, we detail how the robotic pan-tilt motors are controlled for each operation technique, and demonstrate how increased delay reduces the controllability of the hybrid camera.

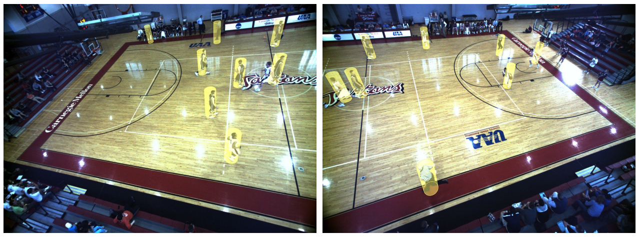

Figure 4: Player Detections. Example frames from the machine vision cameras and corresponding detections.

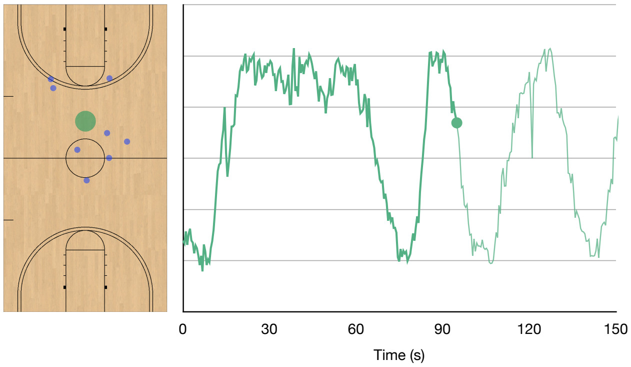

Figure 5: Centroid of Players. The vision system outputs a set of player locations (blue circles) for a given time instant. The centroid (green circle) approximates the appropriate fixation point. The trace of the centroid $\mathbf{r}(t)$ over time is shown on the right, where the current time is ~ 90 seconds.

4. PATH PLANNING

In this work, planning refers to the problem of generating the saliency signal $\mathbf{s}(t)$ from the detected player positions output by the vision system. It is equivalent to framing a shot in cinematography. The first task is to identify the subject(s) of interest, which in basketball corresponds to the players, ball and nets. Afterwards, the rules of shot composition [14, 16, 20] then dictate how the camera should be operated such that objects of interest appear in salient regions of the image (see Fig. 6). It is important to compose the shot such that the viewer can draw the correct interpretation. In basketball, the shot should be framed based on the current and anticipated player positions.

Perfect tracking of the players and ball is not yet possible in either online or offline approaches. Therefore, the autonomous understanding of the scene in realtime includes missed players and false detections. As a result, we employ assumptions common in visual servoing [9, 10] and approximate the camera fixation point as the centroid $\mathbf{r}(t) = [X, \dot{X}]$ of detected player positions.

PTZ Projection

The vision system outputs a fixation point $\mathbf{r}(t)$ in the world coordinate system, but the path planning algorithm must generate a trajectory $\mathbf{s}(t) = [\hat{\phi}, \hat{\dot{\phi}}]^\mathsf{T}$ in each camera's pan-tilt coordinate system. Before describing how an suitable camera path can be extracted from $\mathbf{r}(t)$, we briefly describe how a target location in the world coordinate system is converted into the pan-tilt coordinate system of a PTZ camera.

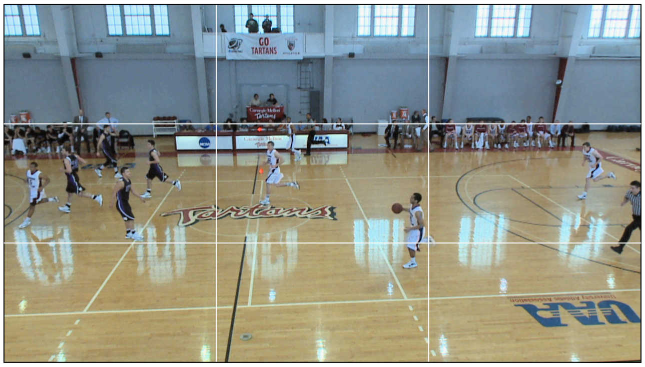

Figure 6: Rule of Thirds. Camera operators frame shots such that important objects fall on imaginary lines which divide the image into thirds. Objects in motion should have 'lead room' to illustrate where the object is going.

The PTZ cameras are calibrated using point correspondences between the ground plane and image plane (and assuming square pixels) [15]. The 3×4 projection matrix $\mathtt{P}$ maps a homogeneous 3D world point $\mathbf{X} = [X, Y, Z, 1]^{\mathsf{T}}$ to a homogeneous 2D image point $\mathbf{x} = [u, v, w]^{\mathsf{T}}$ via

$$\mathbf{x} = \mathtt{P}\mathbf{X}. \tag{1}$$

The projection matrix factors into matrices representing projective $\mathtt{K}$ (3×3), rotation $\mathtt{R}$ (3×3) and position parameters $\mathbf{C}$ (3×1)

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I}| - \mathbf{C}]. \tag{2}$$

The intrinsic matrix $\mathtt{K}$ contains the focal length f and principal point (u_0, v_0)

$$\mathtt{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

The rotation matrix of each Sony EX3 camera changes as its corresponding robotic head moves. As a result, we factor $\mathtt{R}$ into two rotation matrices $\mathtt{Q}$ and $\mathtt{S}$ such that

$$\mathtt{R} \overset{\text{def}}{=} \mathtt{Q}\mathtt{S}. \tag{4}$$

The rotation matrix $\mathtt{S}$ represents the rotational aspects of the transformation from the world coordinate system to the home (0,0) pan-tilt position and remains fixed. The matrix $\mathtt{Q}$ is the 3D rotation for the current pan-tilt position (ϕ, θ) relative to the coordinate system of the pan-tilt motor axes and must be recomputed whenever the robotic head moves

$$\mathtt{Q}(\phi, \theta) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ -\sin(\phi)\sin(\theta) & \cos(\theta) & -\cos(\phi)\sin(\theta) \\ \sin(\phi)\cos(\theta) & \sin(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}. \tag{5}$$

Oracle

The purpose of the path planning algorithm is to generate a saliency signal $\mathbf{s}(t)$ defined in each camera's pan-tilt coordinate system based on the centroid $\mathbf{r}(t)$ of player locations as observed by the vision system. We begin by evaluating the assumption that the centroid of player positions is a reasonable approximation for the true fixation point.

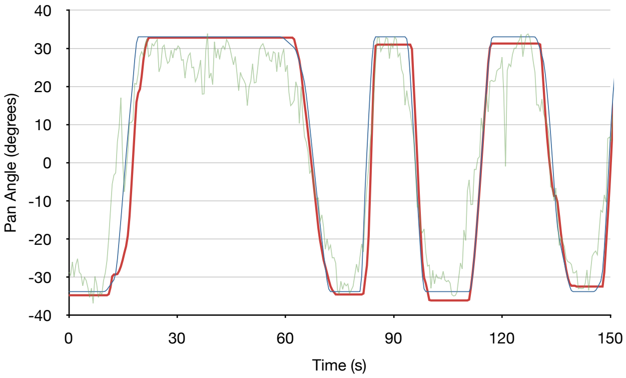

Figure 7: Human and Oracle Signals. The human operated joystick (red) reasonably tracks the oracle signal (blue), although it lags on occasion and misses the optimal pan angle at each end of the court. The raw player detections centroid (green) also follows the oracle signal, but with substantially more noise, and tends to lead and lag the oracle signal more than the human controlled camera.

We generate a ground truth *oracle* saliency signal $\mathbf{s}^{\star}(t)$ for the first half of a basketball game by manually identifying key pan positions every second which generate well composed shots. For example, the player with the ball is placed on the appropriate third so that the majority of the image shows where the ball may go next, while at the same time, the amount of visible court area is maximized.

Over the same period of the game, the human operated camera reasonably traces the oracle signal (see Fig. 7). In addition, the centroid of player positions also tracks the oracle signal reasonably well, although it does lag or lead significantly at times. The centroid is not a good approximation in certain basketball situations such as fast breaks, or slow approaches by the point guard (when the defending team falls back to half court defense). However, it is a reasonably good fixation point the majority of the time.

Error Measure

Good camera operation should follow the target of interest with smooth, purposeful motion. Therefore, when comparing a saliency signal $\mathbf{s}(t)$ to the oracle $\mathbf{s}^{\star}(t)$ signal we evaluate discrepancies between position (for accurate tracking), velocity (for smooth motion) and changes in direction (for purposeful motion). We arbitrarily assign equal importance to these three factors, although one could tailor the weighting for a specific preference. Additionally, we normalize the measured discrepancy with respect to the oracle based on the performance of the human operated camera.

Causal Filtering

Online systems require causal filters which only depend on current and previous values. Our benchmark implementation of visual servoing uses a moving average filter to smooth out the noise in the pan angles from the noisy centroid of player positions. The filter buffers the pan angles previously computed over the last N seconds and outputs the average of the buffered values (see Fig. 8). As N increases, the filtered signal $\mathbf{s}_N(t)$ becomes more smooth, and changes in direction are reduced. However, the filter performance saturates for large N because any gain in smooth and purposeful motion is offset by tracking errors arising from significant filter lag.

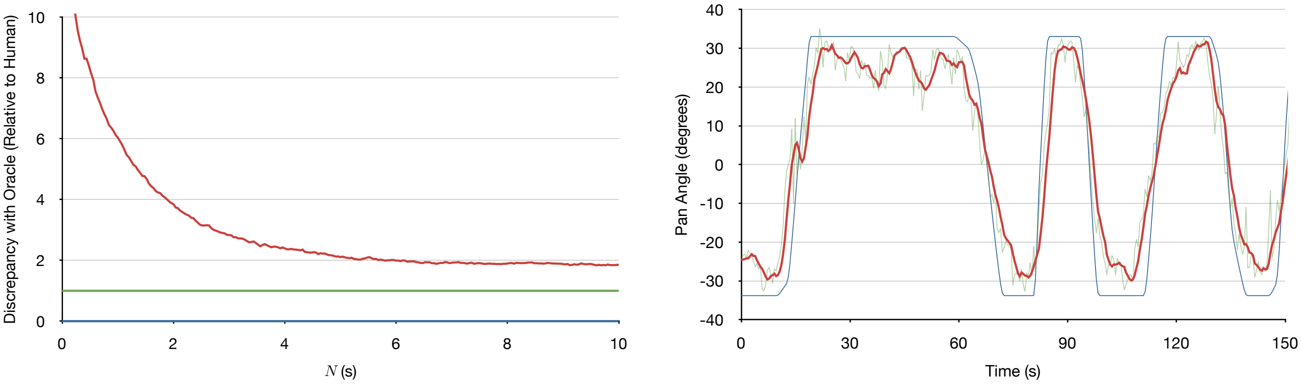

Figure 8: Causal Filter. Left: The performance of moving average causal filter (red) as a function of buffer size is plotted relative to the human operated camera (green) and oracle (blue). A larger buffer size produces smoother and more purposeful camera motion, but struggles to track a rapidly moving target, which is why performance saturates at roughly 2×. **Right:** The filter for $N = 2.0$s produces a slightly smother output than the noisy input signal (green) without introducing significant lag with respect to the oracle (blue).

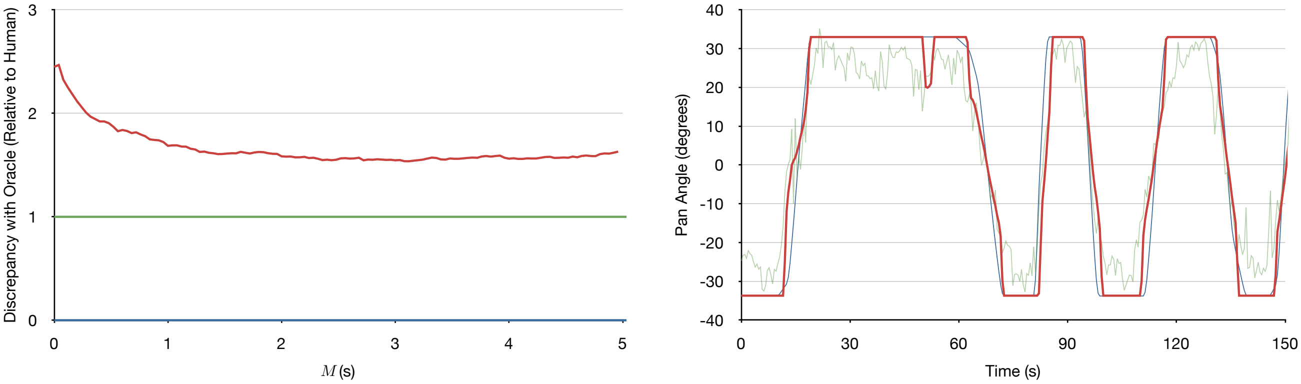

Figure 9: Non-Causal Filter. Left: The performance of the L1 trend non-causal filter (red) increases with a larger induced delay M. In effect, the filter is able to look further into the future before determining whether the camera should maintain its current trajectory or transition to a different trajectory. **Right:** The filter for $M = 2.0$s generates a reasonable approximation of the oracle signal (blue). For clarity, the filter output has been shifted M seconds into the future in order to align temporally with the raw input (green) and oracle signals.

Non-Causal Filtering

Offline and delayed online systems can use non-causal filters which consider previous, current and future values to output a filtered response at the current time. We buffer $N = 10.0$s into the past and M seconds into the future. We begin by approximating the buffered signal as a series of constant velocity segments using L1 trend filtering [18]. The filtered pan positions at the current time and up to four seconds into the future are examined by a Schmitt trigger to produce a hysteresis effect (moving the camera when tracking error is large, and stopping the camera when tracking error is small). If the Schmitt trigger anticipates a change in state, the filter applies a fourth-order polynomial to ease-in/out from the current position/velocity to the anticipated position/velocity [6].

We evaluate the non-causal filter's performance for different values of M (see Fig. 9). The hysteresis aspect of the filter produces a dramatic improvement in purposeful camera movement. As expected, the filter exhibits diminishing returns for larger and larger values of M because distant future events should not significantly impact the decision of how the camera should be moved at the current time instant. Unlike the causal filter, the non-causal filter does not introduce any lag dependent on N because the L1 trend filter fits constant velocity segments to the raw signal (and knowledge about future events prevents the constant velocity model from overshooting).

Although the non-causal filter produces a trajectory that is similar to the oracle, the filtering process is unable to remove the systematic errors associated with the assumption that the centroid of player positions is an adequate proxy for the correct camera fixation point. Quite often the centroid leads the oracle signal when easing out of a stationary state (since the defending team often falls back to half court defense). To overcome these systematic measurement errors, more complex feature extraction and planning algorithms would be needed.

5. SPHERICAL RESAMPLING

The non-causal filter generates a trajectory $\mathbf{s}_{N,M}(t)$ for the virtual camera operating on a delay of M seconds. The video of the virtual camera must be synthesized from the video of the robotic camera. The virtual camera is operating on a delay of M seconds, and we denote its current time t. To generate the current virtual image, we must resample the image $\mathbf{I}(t - M)$ captured by the robotic camera M

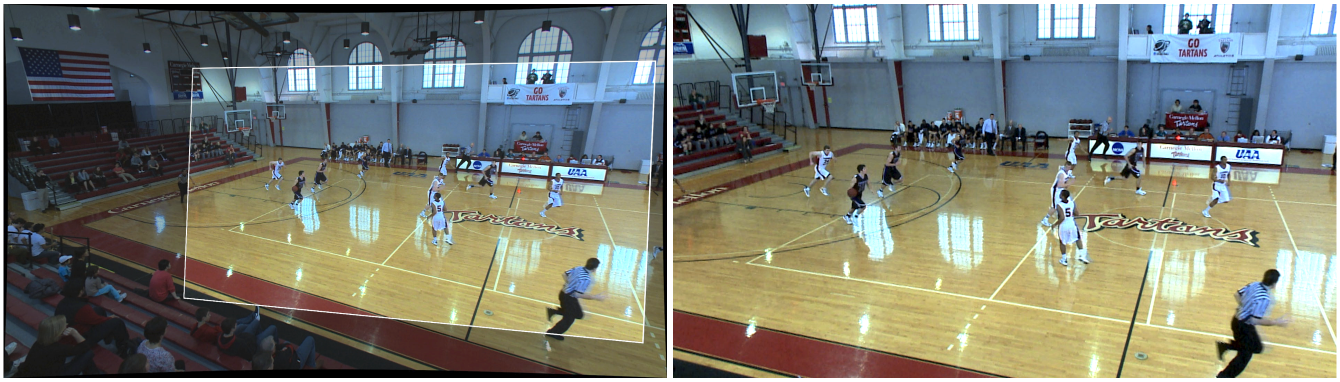

Figure 10: Spherical Resampling. The image captured by the robotic camera (left) at pan $\phi_{\mathsf{robotic}}(t-M)$ is resampled to synthesize the image of the virtual camera (right) oriented at $\hat{\phi}_{\mathsf{virtual}}(t)$. Here, the virtual camera is looking further to the right than where the robotic camera had looked, which is why only pixels from the right side of the image are used for resampling.

seconds ago. At that time, the robotic camera was at pan position $\phi_{\mathsf{robotic}}(t-M)$. The non-causal filter will generate a target pan angle $\hat{\phi}_{\mathsf{virtual}}(t)$ for the virtual camera, which should be similar to both the planned $\hat{\phi}_{\mathsf{robotic}}(t-M)$ and actual $\phi_{\mathsf{robotic}}(t-M)$ robotic camera positions.

Similar to [7, 11, 12], we synthesize the video of virtual PTZ camera via a projective warping (see Fig. 10). In this situation, the mapping from virtual image plane to the robotic image plane is governed by a purely rotational homography [15]

$$\mathtt{H}_{\text{real}\rightarrow\text{virt}} = \mathtt{K}_{\mathsf{virtual}}\mathtt{R}_{\mathsf{virtual}}\mathtt{R}^{-1}_{\mathsf{robotic}}\mathtt{K}^{-1}_{\mathsf{robotic}} \tag{6}$$

which can be obtained by substituting $\hat{\phi}_{\mathsf{virtual}}(t)$ and $\phi_{\mathsf{robotic}}(t-M)$ into (5). If the video of the virtual PTZ is synthesized by cropping a rectangular subregion from a stationary camera, the resulting video no longer has its optical axis near the center of the image which makes the camera appear as though it is translating left/right on a track or up/down on pedestal.

It is important to note that the size of the image generated by (6) remains constant regardless of any change in focal length between the robotic and virtual cameras. However, as the focal length of the virtual camera increases, the number of pixels sampled from the robotic camera decreases; lowering the effective resolution of the virtual camera. Additionally, we currently assume there is no significant motion blur in the images captured by the robotic camera, and do not render blur effects into the synthesized images of the virtual camera based on its virtual motion.

If there is a large discrepancy between the virtual camera's planned position $\hat{\phi}_{\mathsf{virtual}}(t)$ and the actual state $\phi_{\mathsf{robotic}}(t-M)$ of the robotic camera, it is entirely possible that (6) will map all or part of the virtual camera's image beyond the boundary of the robotic camera's image. In this situation, the system can clamp the planned state of the virtual camera to ensure it remains within the field of view of the robotic camera, or render empty/black regions in the virtual camera image. Neither solution is ideal. Clamping the virtual camera motion will induce jitter into the smooth trajectory, and black regions rendered in the images of the virtual camera are unaesthetic. In the next section, we mitigate these non-ideal boundary effects by controlling the pan-tilt motors to balance the deviation of the virtual camera from the center of the robotic camera's image against the deviation of the robotic camera from its planned state space trajectory $\mathbf{s}_N(t)$.

6. CAMERA CONTROL

With plans for the robotic and virtual cameras established, the final task is to operate the robotic camera such that the virtual camera is able to follow its planned trajectory as much as possible (since the broadcast video originates from the virtual camera, not the robotic camera). We first discuss the control aspects of each camera individually, and then address the issues of controlling both cameras in a holistic fashion.

Robotic Control

Each FLIR pan-tilt unit is operated with a 30Hz control loop that gets the current pan-tilt positions $(\phi_{\mathsf{robotic}}, \theta_{\mathsf{robotic}})$ and sets the pan-tilt velocities $(\dot{\phi}_{\mathsf{robotic}}, \dot{\theta}_{\mathsf{robotic}})$. In traditional visual servoing, the task of getting the robotic camera to follow the trajectory $\mathbf{s}_{\mathsf{robotic}}(t)$ generated by the causal filter is regulated based on the residual $e_{\mathsf{robotic}}(t)$ between the actual pan position and the desired position

$$e_{\mathsf{robotic}}(t) = \hat{\phi}_{\mathsf{robotic}}(t) - \phi_{\mathsf{robotic}}(t). \tag{7}$$

We employ proportion-only feedback

$$\dot{\phi}_{\mathsf{robotic}}(t) = \dot{\hat{\phi}}_{\mathsf{robotic}}(t) + \kappa\, e_{\mathsf{robotic}}(t). \tag{8}$$

Virtual Camera

The virtual camera represents an infinitely fast, massless system which compensates for the deviation between the current planned position generated by the non-causal filter, and where the robotic camera was looking M seconds ago

$$e_{\mathsf{virtual}}(t) \overset{\text{def}}{=} \hat{\phi}_{\mathsf{virtual}}(t) - \phi_{\mathsf{robotic}}(t-M). \tag{9}$$

If the deviation is large, it is entirely possible the virtual camera will move beyond the boundary of the image captured by the robotic camera M seconds ago. One possibility is to restrict $|e_{\mathsf{virtual}}(t)| \leq \Delta\phi$ by modifying $\hat{\phi}_{\mathsf{virtual}}(t)$. Another possibility is to increase the zoom of the virtual camera. In our experiments, we define the focal length of the virtual camera to be equal to that of a Sony EX3 camera equipped with a standard lens (so that we can make side-by-side comparisons between the video synthesized by the virtual camera and that captured by a visual servoed camera).

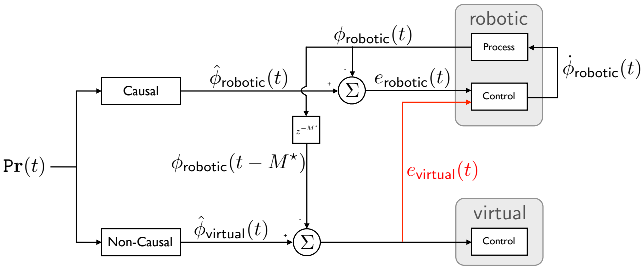
robotic
Process
Control
virtual
Control
Causal
Non-Causal
$\mathbf{Pr}(t)$
$\phi_{\text{robotic}}(t)$
$\hat{\phi}_{\text{robotic}}(t)$
$e_{\text{robotic}}(t)$
$\dot{\phi}_{\text{robotic}}(t)$
$z^{-M^{\star}}$
$\phi_{\text{robotic}}(t - M^{\star})$
$e_{\text{virtual}}(t)$
$\hat{\phi}_{\text{virtual}}(t)$
Σ
Σ
+
-
+
-

Figure 11: We control the hybrid camera by modifying the control of the robotic camera to consider not only the discrepancy $e_{\text{robotic}}(t)$ of where the robotic camera should look (determined by the output of the causal filter) but also the discrepancy $e_{\text{virtual}}(t)$ (highlighted in red) of where the virtual camera should look (determined by the non-causal filter). The robotic camera's feedback control loop contains delayed values which decreases the stability of the system as the delay grows in magnitude.

The change in focal length between the virtual and robotic cameras is equivalent to a 1.5× zoom factor applied to the wide angle lens of the robotic camera. The robotic camera has a horizontal field of view of $\sim 80°$, and the virtual camera $\sim 60°$. As a result, the virtual camera can deviate $\pm\Delta\phi = 10°$ before hitting the frame boundary. If the focal length of the virtual camera is increased, the deviation $\Delta\phi$ would increase as well, with exact values determined by (6).

Arbitrarily restricting the virtual camera to remain within the image captured by the robotic camera may create an erratic motion path. Ideally, the non-causal filter should take deviation limits into account when planning the trajectory. However, even if a maximum deviation $\Delta\phi$ were enforced in the non-causal filter, the system would not be correcting the error in the robotic camera, and the virtual camera may drift further from the planned trajectory. In this work, we investigate an alternative solution: as the virtual camera deviates from the center of the captured image, we incorporate an addition term into the robotic camera's velocity controller (8) to induce motion which pushes the virtual camera back to the center of the image (or equivalently pulls the robotic camera to follow the plan of the virtual camera).

Hybrid Camera

We control the robotic and virtual cameras in a holistic fashion by defining the deviation of the hybrid camera as a linear combination of the robotic and virtual camera deviations

$$e_{\text{hybrid}}(t) = e_{\text{robotic}}(t) + \gamma\, e_{\text{virtual}}(t) \tag{10}$$

The robotic deviation indicates how the robotic camera should move to follow $\hat{\phi}_{\text{robotic}}(t)$. The virtual deviation indicates how the robotic camera should move to return the virtual camera to the center of the image.

$$\begin{aligned} &= \hat{\phi}_{\text{robotic}}(t) - \phi_{\text{robotic}}(t) + \\ &\qquad \gamma\hat{\phi}_{\text{virtual}}(t) - \gamma\phi_{\text{robotic}}(t - M) \end{aligned} \tag{11}$$

Substituting $e_{\text{hybrid}}(t)$ for $e_{\text{robotic}}(t)$ in (8) results in a proportional feedback control

$$\dot{\phi}_{\text{robotic}}(t) = \dot{\hat{\phi}}_{\text{robotic}}(t) + \kappa\, e_{\text{hybrid}}(t) \tag{12}$$

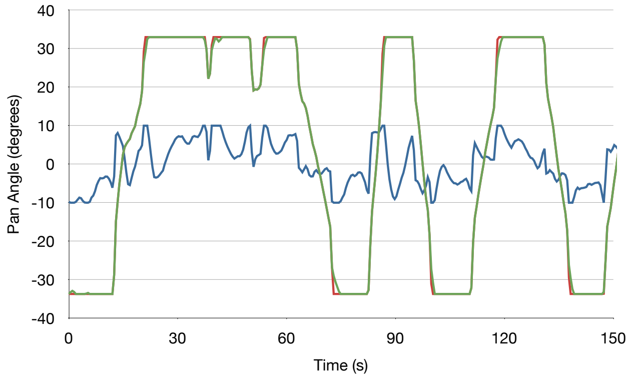

Figure 12: Independent Control. In the above scenario, the virtual camera operates on a 1.0 second delay and is restricted to remain within the image plane of the robotic camera. There is no feedback between the virtual camera and the robotic camera (i.e. the signal $e_{\text{virtual}}(t)$ in Fig. 11 is not sent to the robotic controller). As a result, the relative position (blue) of the virtual camera with respect to robotic camera fluctuates between ±10°. When the virtual camera hits either of its pan limits, there is a deviation between where the virtual camera is looking (green) and where the non-causal filter had planned for it to look (red).

which balances how well the robotic and virtual cameras follow their plans while encouraging the robotic camera to deviate from its plan to ensure the virtual camera does not go outside the image boundary. The challenge with the proposed formulation is that the hybrid feedback includes a delayed term $\gamma\phi_{\text{robotic}}(t - M)$ which decreases the stability of the system (see Fig. 11).

Stability Analysis

We evaluate the hybrid camera's ability to follow the plan generated by the non-causal filter for a variety of values of γ and M (the two parameters introduced into our proposed control loop). We limit the virtual camera's movement to remain within the field of view of the robotic camera — i.e. ϕ_{virtual} must be within $\pm 10°$ of ϕ_{robotic}.

We consider the case of $\gamma = 0$ as our baseline because there is no delayed feedback from the virtual camera in this situation. Effectively, the robotic and virtual cameras are controlled independently. The performance of this configuration for $M = 1.0$s is shown in Figure 12. The relative pan angle (blue) of the virtual camera with respect to the robotic camera illustrates the situations when the virtual camera reaches the frame boundary of the robotic camera (for example, saturating at $+10°$ at roughly 20 seconds). When this occurs, the tracking performance deteriorates and the virtual camera (green) is no longer able to follow the desired non-causal plan (red). Over this 150s sequence, the virtual camera was at the frame boundary 10.5% of the time, which resulted in an average per-frame control error (the deviation from the non-causal filter output) of 0.21° per frame. On average, the virtual camera was looking 5.9° from where the robotic camera had looked M seconds ago.

Next, we examine the situation when delayed feedback from the virtual camera is incorporated into the control loop. The hybrid camera's performance when $\gamma = 0.5$ and $M = 1.0$s is shown in Figure 13. In this configuration, the

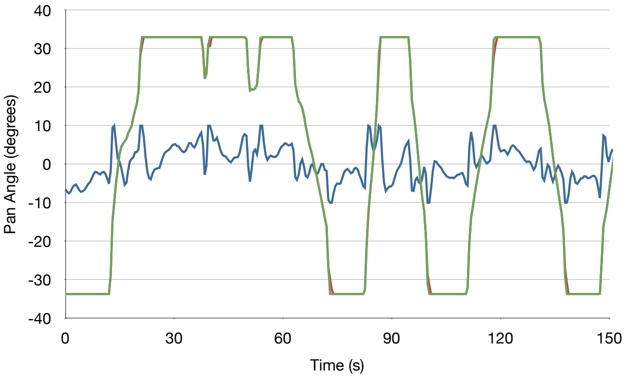

Figure 13: Stable Hybrid Control. The simulation now incorporates a delay $M = 1.0$s, and the relative position of the virtual camera $e_{\mathsf{virtual}}(t)$ is incorporated into the robotic camera's control algorithm. As the virtual camera moves away from the center of the robotic camera's image, the robotic controller will adjust the robotic camera's speed to drive the virtual camera back to the image center. Similarly, the resulting path of the hybrid camera (green) more closely follows its plan (red).

virtual camera is able to track the plan generated by the non-causal filter quite well. Over the same 150s sequence, the virtual camera was only at the frame boundary 5.1% of the time (roughly a 2× improvement over independent control). Similarly, the average per-frame deviation was reduced to 0.11°. Furthermore, the hybrid controller was able to balance the errors of both cameras so that the virtual camera remained closer to the center of the robotic camera's image at all times: the average discrepancy in pan angles was reduced to 4.6°. Qualitatively, hybrid control has increased the frequency of the virtual camera's relative pan angle (blue), but has reduced its overall energy.

However, if the feedback gain γ is too high, or the delay M becomes too large, the resulting control loop may be unstable. Figure 14 shows how a hybrid controller configured with $\gamma = 0.75$ and $M = 2.0$s begins to exhibit signs of an unstable system: the virtual camera begins to oscillate rapidly from one side of the robotic camera's frame to the other. Similarly, the robotic camera also oscillates because the controller is attempting to regulate the robotic camera's pan angle to keep the virtual camera in the center of the frame. For the same 150s sequence (although the non-causal plan is slightly different since the value of M has changed), the virtual camera now spends 12.6% of its time at the frame boundary, which results in an average per-frame tracking error of 0.30°. Furthermore, at 90s, the impact of unstable control becomes evident: the magnitude of the robotic camera's oscillation about a constant pan angle is so large, the virtual camera is unable to fully compensate, and the virtual camera exceeds the soft 33° pan limit.

7. SUMMARY

Autonomous camera systems have traditionally been implemented with either robotic cameras or virtual cameras resampled from fixed cameras. The offline resampling process can generate smooth aesthetic trajectories because the algorithm has full information about all future events. However, because the physical cameras do not move, the resolution of the system is limited. In addition, it is impossible to broadcast a live event. Robotic cameras, on the other hand, generate plans in realtime with no information about future events. Aesthetics aspects are often neglected because it is extremely difficult to track and accurately anticipate object motions in realtime and zero latency. In this work, we propose a hybrid system which operates a virtual camera on a delay (so that it can plan smooth purposeful motion based on perfect knowledge about immediate future events) and controls a robotic camera to follow the live action (which maximizes the resolution of the synthesized broadcast video). As we have shown, the amount of delay is a crucial design parameter. Longer delay improves the path planning process, but makes stable robotic control more difficult.

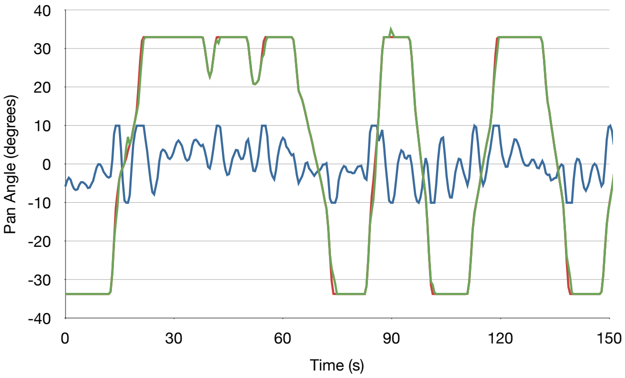

Figure 14: Unstable Hybrid Control. When $\kappa = 0.75$ and $M = 2.0$s, we see the adverse effects of incorporating too much gain and delay in the feedback signal. The robotic camera oscillates around its target trajectory, and the virtual camera oscillates from one frame boundary to the other. As a result, the virtual camera exceeds the 33° pan limit at $\sim$ 90s.

We employ hysteresis to generate purposeful camera motion; it is better to maintain the camera's current trajectory than to deviate slightly to make a small improvement in shot composition. Incorporating hysteresis into a zero latency system is difficult because switching from one state to another may generate a discontinuous plan or introduce temporary latency while the system transitions between states. However, by operating a virtual camera operating on a delay, our non-causal hysteresis filter is able to anticipate state changes based on actual future information.

If the operating delay is sufficiently short, controlling the robotic camera to keep the virtual camera within its field of view is advantageous. However, if the delay becomes too large, the system may become unstable. In this situation, the robotic camera should be controlled in a traditional visual servoing manner with no delayed feedback; the video can still be resampled with an independently controlled virtual camera operating on a delay. Although our example prototype system is geared towards autonomous cameras, the idea of a hybrid robotic/virtual camera can apply to robotic cameras with remote human operators.

8. REFERENCES

[1] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital

camera work based on situation recognition. In *International Symposium on Multimedia*, 2006.

[2] P. Carr, Y. Sheikh, and I. Matthews. Monocular object detection using 3D geometric primitives. In *ECCV*, 2012.

[3] F. Chen and C. D. Vleeschouwer. Personalized production of basketball videos from multi-sensored data under limited display resolution. *CVIU*, 114(6), 2010.

[4] K. Choi, S. W. Lee, and Y. Seo. Automatic broadcast video generation for ball sports from multiple views. In *International Workshop on Advanced Image Technology*, 2009.

[5] M. Christie, P. Olivier, and J.-M. Normand. Camera control in computer graphics. *Computer Graphics Forum*, 27(8):2197–2218, 2008.

[6] J. Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, third edition, 2004.

[7] S. Daigo and S. Ozawa. Automatic pan control system for broadcasting ball games based on audience's face direction. In *ACM Multimedia*, 2004.

[8] A. Dearden, T. Demiris, and O. Grau. Learning models of camera control for imitation in football matches. In *International Symposium on Imitation in Animals and Artifacts*, 2007.

[9] A. Farag and A. Abdel-Hakim. Virtual forces for camera planning in smart video systems. In *WACV*, 2005.

[10] N. R. Gans, G. Hu, and W. E. Dixon. Keeping multiple objects in the field of view of a single ptz camera. In *American Control Conference*, 2009.

[11] M. Gleicher and J. Masanz. Towards virtual videography. In *ACM Multimedia*, 2000.

[12] M. L. Gleicher and F. Liu. Re-cinematography: improving the camera dynamics of casual video. In *ACM Multimedia*, 2007.

[13] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2012.

[14] M. Haigh-Hutchinson. *Real-Time Cameras*. Morgan Kaufmann, 2009.

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

[16] S. Katz. *Film Directing Shot by Shot*. Michael Wiese Productions, 1991.

[17] K. Kim, D. Lee, and I. Essa. Detecting regions of interest in dynamic scenes with camera motions. In *CVPR*, 2012.

[18] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky. L1 trend filtering. *SIAM Review*, 51(2), 2009.

[19] D. Nieuwenhuisen and M. Overmars. Motion planning for camera movements. In *ICRA*, 2004.

[20] J. Owens. *Television Sports Production*. Focal Press, 2007.

[21] C. Pinhanez and A. Bobick. Intelligent studios: Using computer vision to control TV cameras. In *IJCAI Workshop on Entertainment and AI*, 1995.

[22] R. Stanciu and P. Oh. Designing visually servoed tracking to augment camera teleoperators. In *IROS*, 2002.

[23] X. Sun, J. Foote, D. Kimber, and B. S. Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. *IEEE Transactions on Multimedia*, 7(5), 2005.

[24] T. Yokoi and H. Fujiyoshi. Virtual camerawork for generating lecture video from high resolution images. In *ICME*, 2010.

[25] C. Zhang, Z. Liu, Z. Zhang, and Q. Zhao. Semantic saliency driven camera control for personal remote collaboration. In *International Workshop on Multimedia Signal Processing*, 2008.

[26] C. Zhang, Y. Rui, J. Crawford, and L.-W. He. An automated end-to-end lecture capture and broadcasting system. *ACM Transactions on Multimedia Computing, Communications and Applications*, 4(1), 2008.