The Replate

GE CHEN, Hong Kong UST
PEDRO V. SANDER, Hong Kong UST
DIEGO NEHAB, IMPA

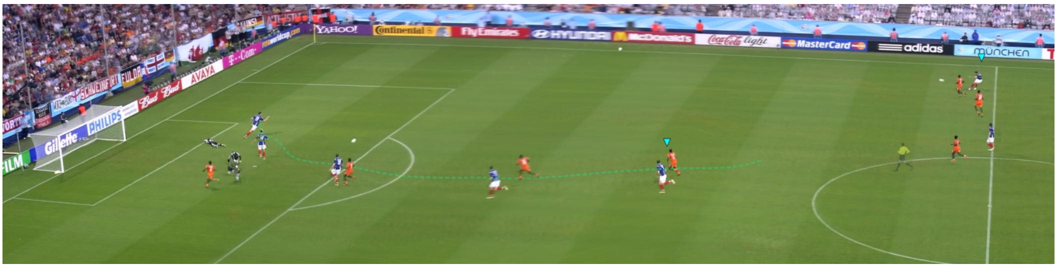


Fig. 1. We use a soccer example to describe the basic idea and features of *replates*. (Note: the picture can never substitute for the video. Please see the supplemental materials.) Multiple instances of key players and the ball can be seen at different stages of the play. The looping animation continuously shows the entire run of the scoring player (in blue) and the nearest defending player (in orange). Other players, such as the passing player on the top right, appear briefly in only one instance with a shorter animation loop. Matching colored markers (▽) help connect the passing player with the instance of the scoring player that receives the pass. A dotted line annotates the path of the scoring player. Additional players that were involved in the play are also included. Even though the play takes 7.3 seconds to complete, the replate video is just 1.3 seconds long. The file is only 130KB, with resolution of 1280×316.

Replates are a new way to experience replays of sporting events. We use computer vision tools, such as camera tracking and background subtraction, to align all frames in the original video with regard to each other. Using this alignment, we create a single *plate* as it would be seen from a wide-angle static camera. Key players and events are then reinserted into this arena using an interactive tool, recovering the big picture that is typically lost due to camera panning and zoom. Replates are played in continuous loop. We allow replate authors to include multiple instances of each player, offset by regular intervals, so viewers can quickly and repeatedly inspect their favorite moments. Authors can also choose from a variety of effects that help them construct a narrative of their favourite plays. The interactive system gives immediate feedback allowing authors to adjust the rendered replates interactively. Replates take a fraction of the size of the original input video, and produce an effect reminiscent of video game reenactments of famous sporting events. The footage, however, is real.

CCS Concepts: • **Human-centered computing → Information visualization**; • **Computing methodologies → Image processing**;

Additional Key Words and Phrases: sports visualization, video generation, interactive video processing

Authors' addresses: Ge Chen, Hong Kong UST, gechen@connect.ust.hk; Pedro V. Sander, Hong Kong UST, psander@cse.ust.hk; Diego Nehab, IMPA, diego@impa.br.

2577-6193/2018/5-ART4 $15.00
https://doi.org/10.1145/3203205

ACM Reference Format:
Ge Chen, Pedro V. Sander, and Diego Nehab. 2018. The Replate. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 4 (May 2018), 14 pages. https://doi.org/10.1145/3203205

1 INTRODUCTION

Every four years, between the months of June and July, over 3 *billion* people around the world cheer for their favorite soccer teams as they compete for the World Cup. The most remarkable moments of each game will be replayed an untold number of times for generations to come, in the same manner as today's soccer fans watch replays of games long past. This, of course, is not something particular about soccer. Fans of any sport frequently watch and exchange replays of their favorite plays. Although each sporting event is now captured from multiple angles by multiple high-definition cameras (perhaps even in 3D), surprisingly little has changed in the way replays are experienced, ever since the advent of recorded video made the replays popular and widespread.

Oftentimes, the viewer wants to focus on particular events: a fantastic dribble, the moment at which the player strikes the ball, or the ball hits the net. Unfortunately, replays typically show an entire continuous play. To see the event of interest multiple times, the viewer must repeatedly rewind and fast-forward the video to the important parts, or cut a piece of the video and play it in a loop. Neither alternative is desirable: the first is tedious, the second destroys the context in which the play took place.

To make matters worse, replays are often available from a single point of view under camera pans and zooms that reflect the intentions of the camera operator. The replay viewer may be motivated by entirely different reasons. Furthermore, replays typically focus too narrowly into a particular event. This often makes it difficult for viewers to *see the big picture.*

For obvious reasons, there is no way to retroactively recover information that has been discarded during editing or that was not captured at all. We *can*, however, take all the information that *is* present in the replay and significantly change the way it is structured and presented to the viewers. Better yet, we can empower authors by providing them with the tools they need to create their own narratives of how the particular play unfolded.

In this work, we introduce *replates*, our attempt at rethinking the way in which replays are experienced. It is based on two main ideas. Given the input video for the play (e.g., the soccer goal in figure 1), we first recreate the global arena (or *plate*) in which all events unfolded. Authors then use our proposed interactive system to create a new video that shows the events inserted in the plate's reference frame. The result is a video that shows the play as seen from a static camera, thereby restoring the big picture that had been lost: the relative positioning of players on the field, the sequence of passes, the trajectory of the ball.

Replates are continuously displayed in a loop. We allow authors to include multiple instances of important animated subjects (e.g., players, balls, referees, the crowd), separated by regular intervals in time. The frequency with which each event occurs is therefore multiplied. Together, these modifications enable the viewer to quickly visualize and inspect any part of the play, at any time, in a global arena. The final effect is reminiscent of video game reenactments of famous sporting events; except the footage is real.

Since replates often include multiple instances of different subjects, unexperienced viewers may at first find it difficult to understand the timeline. We have found that the best way to start enjoying a replate is to identify where the play starts (Authors can help viewers by adding a visual aid marking the event.) Then, follow the play in its entirety at least once, by tracking the subjects in the leading role (the ball, a skater, a car, etc). Having first obtained a global understanding of how the play unfolded, most viewers can then comfortably move from subject to subject, and from instance to instance, without getting lost.

A great side-effect of replates is a *significant size compression*. To see this, imagine a player takes 90 frames to move from the start to the end of the play. Now assume we include 3 copies of this player in the video, each separated by 30 frames. Finally, assume we play the video in a loop. Clearly, we only need to include 30 frames in the final video. Yet, a viewer can track the continuous motion of each instance of the player across the entire duration of the play. *No useful information is lost*, but the new video has been compressed to a third of the original size. Replates are therefore light-weight and can be viewed over the web without requiring lengthy download times. All results we discuss are available in the supplemental materials, and can be viewed with the web-based JavaScript viewer we provide.

Finally, our software enables authors to quickly compose their own replates, and includes several tools that help them guide viewers' attention, tailoring the final experience to their taste:

- **Variable number of instances**: Our system allows multiple copies of each subject (e.g., players) to appear in the replate while keeping them synchronized;
- **Selective slow motion**: Some of the subjects can be shown in slow motion for a part (or the entire extent of) their roles, while others remain at regular speed. The system allows the user to synchronize them to the standard speed of the play at a given point in the timeline;
- **Localized sound using the mouse or by gaze tracking**: The viewer can listen to the sound from different parts of the play by simply moving the mouse cursor over the relevant region of the replate. Alternatively, with the aid of a webcam, gaze tracking can be used;
- **Additional features and annotations**: Subjects can optionally be shown as blurred trails, have superimposed tracking markers, dotted lines indicating their paths, and varying fade-in/fade-out and opacity properties based on the importance of their role at each moment in the play.

2 RELATED WORK

Replates follow a recent trend in computer graphics, whereby computer vision techniques are employed to empower authors to create new experiences from pictures or videos [Thomas 2012]. The main contributions in these works are the novel ways in which the tools are combined, the user interface through which they are exposed to authors, and, most of all, the new experiences that they enable. Below, we discuss the works that are most closely related to replates.

There are many alternatives for summarizing the content in video footage. One approach is to generate a single image that summarizes motion by including the subject at different moments of the action [Assa et al. 2006; Teodosio and Bender 2005]. Video Summagator [Nguyen et al. 2012] uses a different approach: it interprets the video as a 3D volume and uses volume rendering techniques to navigate on it and extract useful still images. Annotations can also help visualize motion [Collomosse et al. 2005; Kim and Essa 2005].

Earlier work generate small videos as well (oftentimes in loops) to depict a scene or moment [Agarwala et al. 2005; Bai et al. 2012; Liao et al. 2013; Schödl et al. 2013; Tompkin et al. 2011]. These methods generally assume some form global stationarity or stochasticity in the subjects, as well as minimal overlap or interference between them. They are not suitable for the replays we target.

Joshi et al. [2012] create an experience that is intermediate between a picture and a video by locally specifying scene elements as static or dynamic. Our authoring system has a workflow that is similar to theirs. Irani et al. [1995, 1996] generate static or dynamic sequences as panorama videos but do not consider the use of multiple instances of each subject. Approaches that include multiple instances of each subject have been used before with the goal of authoring video narratives of multi-scene videos [Correa and Ma 2010], to summarize surveillance footage [Pritch et al. 2008], or to combine highlights of a videoclip into a short trailer [Caspi et al. 2006].

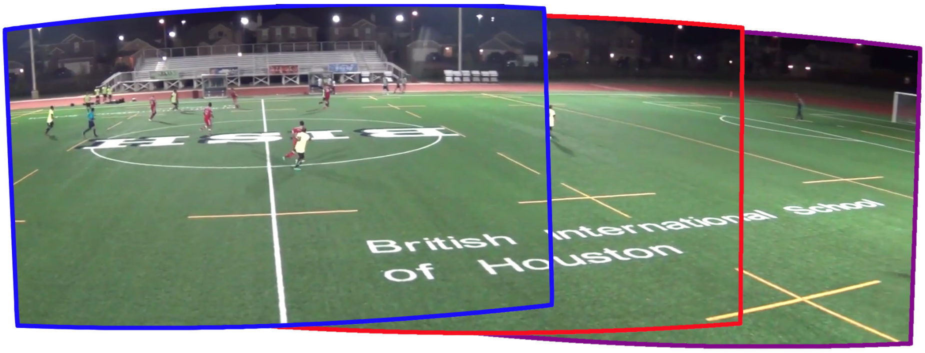


Fig. 2. Three aligned frames used in our background plate generation algorithms. Source images: Houston Sports game filming service (https://www.youtube.com/watch?v=Ix4mV3vEJzg)

The experience we seek if of an entirely different nature. Replates depict events *as they actually happened*. The artistic freedom enabled by the special effects we offer—selective slow-motion, localized sound, and annotation extensions—are only used to emphasize real events, or to guide viewer attention towards them. Being faithful to the input replay requires the careful synchronization of all subject instances, so each of them can be seen over the entire length of the replay, merging seamlessly when the output videos are looped.

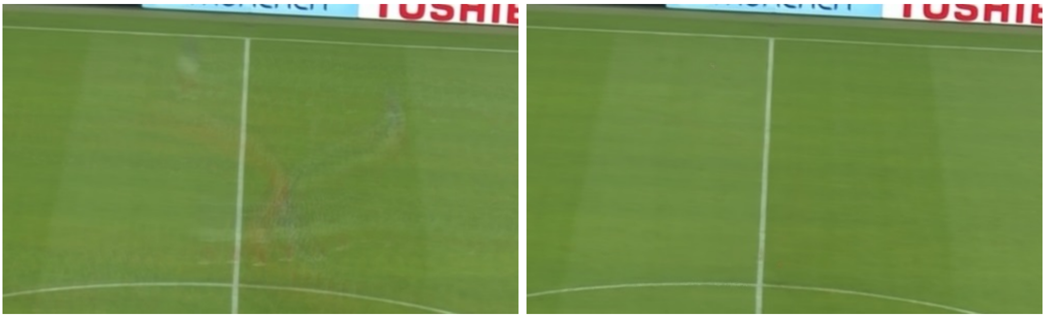
Companies as Vizrt [2014] and SportsVision [2014] offers several sophisticated visual effects to overlay information and annotations on replays. These are orthogonal to our work and could be used in conjunction with, or replacing the annotations we offer as a proof of concept.

3 PREPROCESSING

We now describe the preprocessing steps that precede the creation of a new replate. The first step of our approach is to stitch the input video frames into a consistent background plate.

Video frame registration. There is a vast number of techniques available in literature to align images or video frames, of which many are suitable for our purpose. We have found that using the automatic panorama stitching pipeline of Brown and Lowe [2007] with some modifications, mostly to remove the moving subjects, was easy to use and worked very well for this application. For feature extraction and matching, we use SURF features due to their high efficiency and quality [Bay et al. 2008]. Due to the sequential nature as well as high temporal coherence and overlap of our video frames, we only consider the two nearest neighbors for each feature during matching. Additionally, instead of selecting the image pairs with the most feature matches among all video candidates, we only consider the n frames immediately before or after the current frame (we have found that setting $n = 5$ works well in our experiments). Note that this simply provides a speed optimization and we still verify the matches using the probabilistic model from Brown and Lowe [2007]. Figure 2 shows an example of three aligned frames overlaid over an empty background plate. We use all the frames to generate a *frame-aligned video*, where all the frames are transformed based on the resulting alignment and superimposed over a blank background.

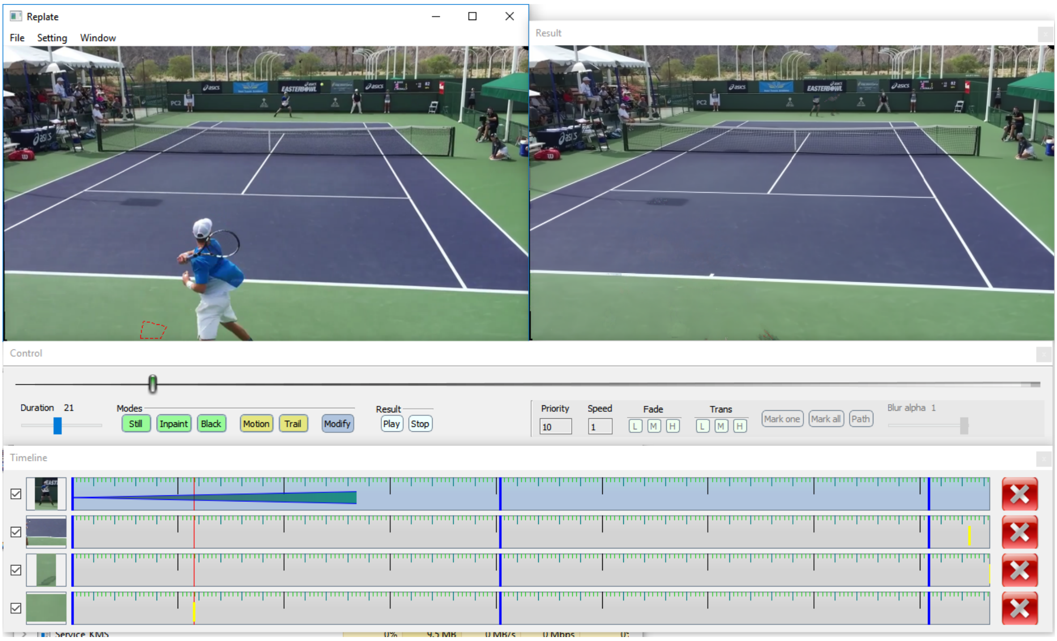
Generating the background plate. If we simply stitch the aligned frames, some of the moving subjects often get blended on the background plate (see figure 3a). To address this, we need to mask out all of the moving objects. To do so, we perform foreground detection by applying the ViBe algorithm [Barnich and Van Droogenbroeck 2011] on the frame-aligned video (for better results, we bootstrap ViBe's cache using the average color value of each pixel location throughout the frame-aligned video). Pixels that ViBe labels as foreground are then mapped to their original image



(a) *plate without masking*

(b) *plate with masking*

Fig. 3. Masking out moving subjects prevents them from being blended onto the background plate.



Duration 21

Priority 10

Speed 1

Blur alpha 1

Fig. 4. The user interface of our system consists of four main panes: Input on the top left, result on the top right, control in the middle, and timeline on the bottom. Source image: ZooTennis.com (https://www.youtube.com/watch?v=8iS02EABcN4)

frames and masked out (i.e., removed from consideration for the background plate). Additionally, the replate author can optionally select static regions of the video to be masked out (e.g., the region where a digital scoreboard is overlaid by a TV broadcaster). Once these masks are determined, we stitch the unmasked regions of the remaining images by feathering, resulting in a cleaner background (see figure 3b).

Optical flow for tracking. Next we compute dense optical flow for all pixels on the frame-aligned video [Manao et al. 2011]. The optical flow information will soon be used for object tracking.

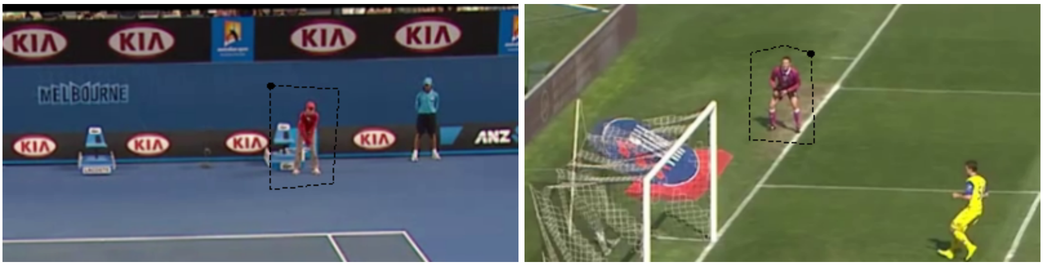


Fig. 5. Selection of stills for the *Tennis1* and *Soccer1* replates.

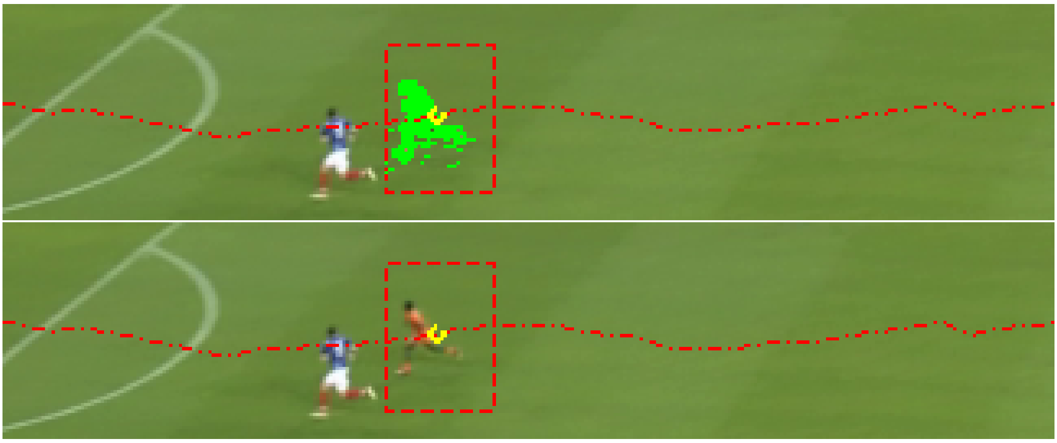


Fig. 6. Tracking a player in the *Soccer3* example. The region of interest and the tracked path are shown in dotted red lines. On the bottom row foreground pixels are highlighted in green.

4 INTERACTIVE REPLATE DESIGN

Once the video has been processed into a single static plate, we are ready to design the replate. We next describe the details of the user interface and implementation details of the authoring system we developed for this purpose.

The user interface consists of four main panes (figure 4). The input pane shows the original video registered into the background plate. The result pane continuously loops a preview of the replate being authored. The timeline pane shows the timeline for each selected subject in the video. It is used to specify when and how subjects appear and disapear on the replate. It is parameterized by time over the input video. Finally, the control pane is used to perform seeking over the input video, add subjects to the timeline, and specify settings of each subject instance that was added to the timeline. Next, we present a detailed description of modes and settings.

4.1 Modes

Each subject added to the replate has a specified *effect mode* that determine how subjects are displayed. We describe them below.

Still. Stills allow the author to place a static image region from one of the video frames into the replate. For example, we may want the referees to stay still during the action since their movement often irrelevant and distracting (see figure 5). When a still region is selected from the input pane, a timeline entry is added with a vertical yellow bar indicating the frame from the input video where

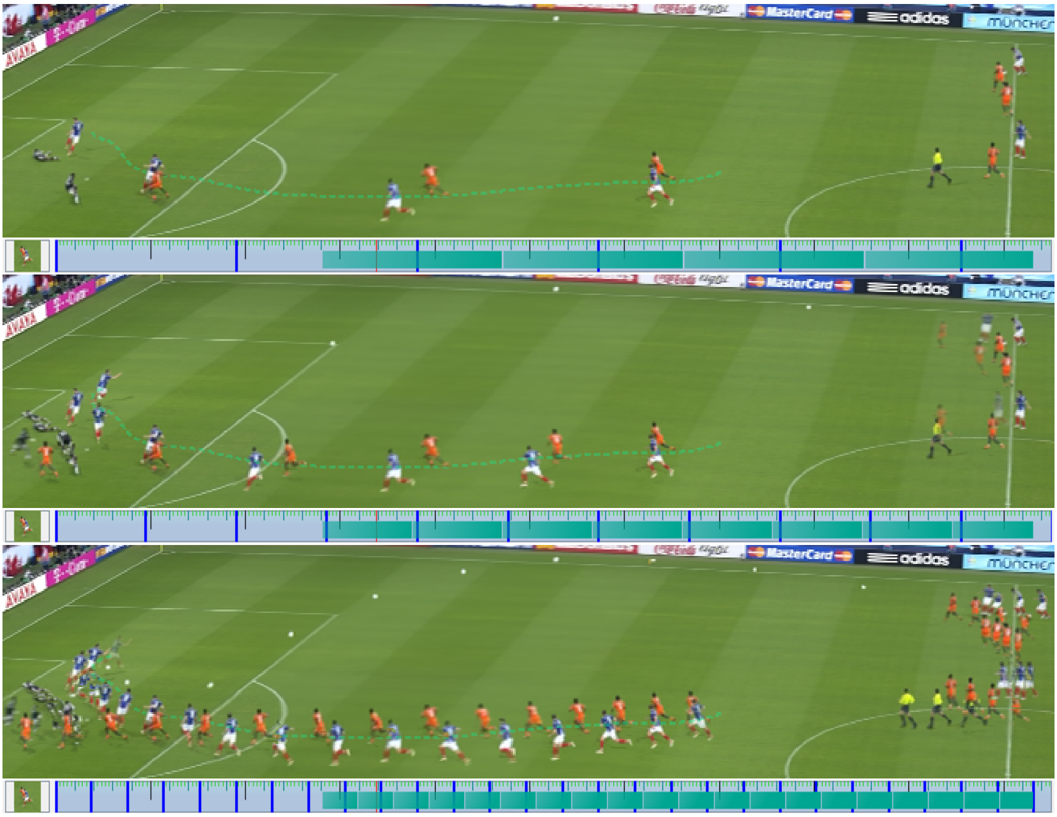


Fig. 7. The *Soccer3* replate with different final duration settings.

this still was taken from (e.g., the top timeline entry in figure 4). When generating the final video, we use feathering to blend the still into the background.

Motion. This is the mode for moving subjects, such as players and the ball. The author selects the subject and the desired range of frames during which the subject should appear. For example, the scoring player in the *Soccer3* example from figure 1 appears from the time he starts running until after the goal is scored. The subject appears in the timeline as a green bar representing the chosen range of frames from the input video. Note that the range may later be changed by dragging the edges of the bar. As with stills, feathering is used for compositing.

Our interface provides an object tracking tool to help user quickly select a rectangular interest region around the object and track it through the entire time range (see example in figure 6). Tracking of the interest region is accomplished by using the dense optical flow computed during preprocessing and Kalman filtering to further improve the estimate. Additionally, the interface allows for simple and fast manual editing to modify the tracked path for instances where automatic tracking based on optical flow is not appropriate (e.g., occlusion, fast moving balls, and blurred regions due to abrupt camera motions).

Trail. This mode operates just like motion, but instead of showing the moving subject, it blends faded-out versions of the subject along the its path in the replate. The result is blurred trail or streak effect (e.g., the path of the server in the *Tennis1trail* replate). The author can optionally apply Laplacian smoothing to the path of the subject to achieve a more pleasant rendering.

Inpaint and Black. Oftentimes the author wants to remove an undesirable region from the final replate. We provide two different options: Inpainting the selected region of the final replate using the algorithm by Telea [2004] (e.g., the removed digital scoreboard from the television transmission in figure 4), or filling the region in black (e.g., part of the boundary of the *Skating1* and *Racing1* examples of figure 9). While we only experimented with these two methods, other techniques such as texture synthesis could be easily plugged into our framework.

4.2 Setting the duration

A key step in the creation of a replate is setting its final duration. In order to enable viewers to see any part of the play at any time, replates often last only about 1–2 seconds (which are played in a loop, of course). The shorter the duration, the more instances of each subject will be included. This can cause too much overlap and clutter. Thus, the author has a balance between the two extremes. This sweet spot is specific to each play. Figure 7 shows an example replate with different duration settings. In each case, the timeline for the scoring player appears below the replate. Note that the example at the top is very sparse with too much time between events. In contrast, the example at the bottom is too cluttered. The example on the middle provides a more reasonable tradeoff for this play. It is a matter of taste. (See the accompanying video for a UI demonstration.)

In the timelines shown in figure 7, the space between vertical blue lines represents the selected replate duration. These lines are updated dynamically when the user modifies the duration. The number of instances of a subject that are visible in the replate at any given time also depends on the length of the subject's role in the play (depicted by the length of the green bar). Furthermore, authors can also change the number of instances of a subject temporally along the replate, by fading it in and out at the beginning and end of its role (i.e., beginning and end of the green bar in the timeline). In the example of the figure, the player on the top-right who passes the ball has fewer instances than the player who makes the run and scores the goal.

4.3 Settings

We now describe the different settings that can be applied to each subject or timeline entry.

Slow motion. Our system allows subjects to appear at different speeds during the replate. Although this may sound counterintuitive at first, it is in fact a powerful tool for highlighting a particular aspect of the play (e.g., slow motion to show how a player strikes a soccer ball), as well as realism (e.g., regular speed to emphasize the power of the kick). The trick is to synchronize the events happening at different speeds so they meet at a relevant moment (e.g., the time when foot hits ball). The *Soccer6slow* replate in the supplemental material shows a demonstration of this effect. The interface allow the author to specify a speed factor on each timeline entry, as well as the synchronization frame between them.

The same subject can also be shown at different speeds during the play. To achieve this, we create two different timeline entries for the subject, each representing adjacent time segments in the input video, and synchronize them at the transition frame between them. The *Tennis1* replate shows an example where only the player's serve is shown in slow motion.

The result pane previews the slow motion by simple frame duplication. When saving the final replate we use optical flow to create smoother interpolated animations [Manao et al. 2011].

Markers. An upside-down triangle marker (▽) can be automatically added above a subject as it moves along in the replate (figure 8). If there are multiple instances of the subject, each instance is given a different color. The colors are consistently assigned across different subjects based on their temporal position in the timeline. An example of this is shown in the *Soccer3marks* replate. The color of the marker shown above an instance of the passing player matches the color of the marker

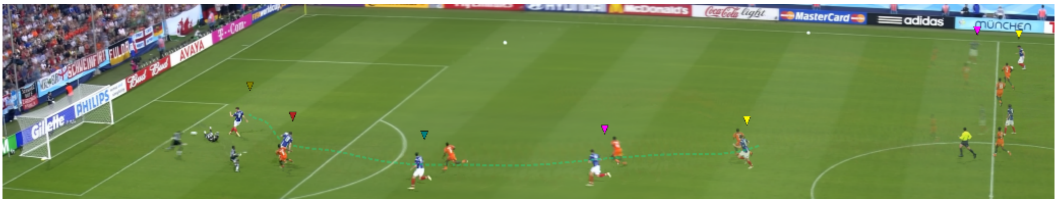


Fig. 8. The overlaid markers help the viewer in matching the passing players at the top-right, with the receiving players in at the bottom-center.

shown above the instance of the running player that will receive that pass. This helps the viewer more easily track which instance of the player will be receiving the pass. We also allow the marker to be displayed over only of the instances (figure 1). This helps prevent clutter while still allowing easier tracking in some of the instances. See the *Soccer3mark* replate in the supplemental material for a demonstration.

In our current implementation, markers are baked in the output replay video frames. Since each instance of the subject uses a different marker, we lose the desirable video compression side-effect of replates. It is a simple engineering problem to design a viewer that overlays these markers instead, thereby restoring the compression.

Path lines. Our interface allows the path of any subject to be overlaid with a dotted line. This is a less obtrusive alternative to rendering subjects using the trail mode. It is useful to show important aspects of the play, such as the path of the ball in multiple soccer examples found in the supplemental material. As with markers, Laplacian smoothing is applied to path lines.

Fade-in/fade-out. Subjects can be faded in and out at different speeds at the discretion of the author. Our current interface allows three fading speed settings (5, 10, and 15 frames). Different replates in the supplemental material use different fading speed settings.

Opacity. Subjects can be drawn with different opacity settings. Lead players are normally opaque, while supporting subjects can be deemphasized with a lower opacity. This helps reduce excessive clutter on the replate. The interface provides three opacity settings. The replates in the supplemental material showcase these differences.

Z-order. Occlusion relations of different subjects can be specified. Our authoring tool associates each timeline entry to a Z-value between 0 and 80. During composition, a *greater-than* Z-test is used to select the order. We have preset depth values associated to the following tags: black or inpaint (70), motion (50), trail (20), and still (10). The author can then fine-tune the relative ordering by changing the Z-value of each timeline entry.

Z-subject. When multiple instances of the same subject are added to the timeline, it is possible for them to overlap one another. In that case, the author has the option to specify whether the rendering of instances from a later part of a play should appear in front or behind earlier instances.

4.4 Sound

Since the replate compresses the video into a short 1-2 second looping animation, it is not possible or even meaningful to include the original audio of the play. However, we can include snipets of the audio that have the duration of the replate and selectively play them based on viewer demand. For example, in the *Bowling1* replate, a viewer may want to hear the sound of the ball striking the pins. Our video player will play that looping sound continously whenever the mouse cursor is placed over the region of the replate where the strike occurs. The viewer can hear the sound of the

Table 1. Statistics for all of the replates in the supplemental material.

	Input			*Processing times (s)*		*Replate*		
Name	Resolution	Duration (s)	File size (MB)	Preproc	Postproc	Resolution	Duration (s)†	File size (MB)†
Soccer1A	1272 × 720	6.0	1.88	124	38	1280 × 228	1.3 (4.6 ×)	0.10 (19.5 ×)
Soccer1B	1280 × 720	3.7	1.05	64	34	1280 × 408	1.3 (2.8 ×)	0.14 (7.7 ×)
Soccer2	1264 × 720	8.9	2.76	227	79	1280 × 236	1.5 (5.9 ×)	0.11 (26.1 ×)
Soccer3	1264 × 720	7.3	3.17	197	52	1280 × 316	1.3 (5.6 ×)	0.13 (25.4 ×)
Soccer4	1276 × 712	10.8	4.17	237	63	1280 × 296	1.8 (6.0 ×)	0.11 (39.1 ×)
Soccer5	1220 × 720	4.5	2.17	83	38	1280 × 540	1.6 (2.8 ×)	0.19 (11.7 ×)
Soccer6	1264 × 720	4.6	2.71	81	27	792 × 552	1.8 (2.6 ×)	0.12 (22.1 ×)
Soccer6slow	1264 × 720	4.6	2.71	81	42	792 × 552	1.8 (2.6 ×)	0.12 (23.0 ×)
Soccer7	1268 × 712	3.3	1.17	62	25	1280 × 476	1.0 (3.3 ×)	0.11 (10.3 ×)
Soccer8	1272 × 720	3.0	1.30	64	24	1280 × 672	1.3 (2.3 ×)	0.12 (10.8 ×)
Soccer9	1268 × 720	6.4	2.23	112	36	1280 × 496	1.6 (4.0 ×)	0.14 (15.7 ×)
Tennis1	1280 × 720	6.2	0.90	134	25	860 × 472	2.5 (2.5 ×)	0.14 (6.2 ×)
Tennis1trail	1280 × 720	6.2	0.90	134	26	860 × 472	2.5 (2.5 ×)	0.14 (6.4 ×)
Tennis1slow	1280 × 720	6.2	0.90	134	32	860 × 472	2.5 (2.5 ×)	0.15 (6.0 ×)
Bowling1	1280 × 720	12.4	1.77	259	61	700 × 388	3.3 (3.8 ×)	0.15 (12.2 ×)
Racing1	615 × 466	8.7	0.88	114	74	1280 × 216	2.5 (3.5 ×)	0.07 (13.3 ×)
Skating1	1264 × 708	3.9	1.40	156	108	1280 × 268	1.3 (3.0 ×)	0.11 (12.7 ×)
Skating2	1264 × 720	6.6	1.61	133	43	1280 × 352	1.3 (5.1 ×)	0.14 (11.7 ×)

†Size reduction factor relative to input shown in parenthesis.

ball striking the floor after the throw by moving the mouse cursor over the bowling player. Our system allows the author to configure these *sound regions*, designating specific sound snipets to each of them. A default sound can also be played if the mouse cursor is outside all of those regions.

Gaze tracking. Finally, we can use off-the-shelf gaze tracking software to determine the region of the screen the viewer is currently looking at. We can use this information to drive the selection of the sound to be played. This provides a very intuitive way to visualize a play, since *the users only repeatedly hear the sound from the part of the play they are currently looking at.* The accompanying supplemental material shows an example of a head tracking session.

5 RESULTS

Table 1 shows statistics of the input videos and our resulting replates. Note that for all examples, preprocessing and postprocessing take at most a few minutes. Postprocessing includes the time for blending and slow motion computation, when needed. To achieve higher quality, postprocessing is first performed at a very high resolution and then downsampled (by approximately 2-3×) to a resolution that is suitable for most displays. The editing time for these examples is between 1-10 minutes, depending on the complexity of the replate. Since replates comprise of a very short animation loop, they have a very short video duration (a factor of 2-6× shorter than the original video). Furthermore, since they use a static background plate, they compress extremely well, most of them being around 100KB in size. As mentioned earlier, if subjects are anotated with colored markers baked into the video itself, its duration must be extended. However, this can be avoided by adding functionality to overlap the markers in the video player.

To clearly see the benefits of using replates, we refer the reader to the supplemental material. It includes all of the input videos and resulting replates, along with a description of what to look for when viewing each example. Sound and gaze tracking results are also shown. Nonetheless, we also added some snapshots of many of those replates in figure 9 to give a general idea of the different types of action plays that are suitable for this technique.

Limitations. Replates are not suitable to visualize all kinds of action plays. In many situations where the play is extremely complex, converting it into small animation loops adds too much clutter and overdraw, making it difficult to clearly follow the action. This is the case for the *Soccer9* example, where there is a complex string of passes and player motions.

Oftentimes, during a replay the camera is moving extremely fast or even eratically. This gives rise to two problems. First, it may be too challenging to register the frames and create the background plate, and second, it may cause some subjects to appear completely blurred. The later is not noticeable on the original replay due to the fast moving camera, but it is clearly evident over the static background plate. This can be seen in a few of our replates, including when Pelé comes into frame to head the ball in *Soccer5*, and more aggressively in the players and specially the goalkeeper of the problematic *Soccer9* replate. In some situations, we can simply omit that portion of the subject's animation by shortening its role in the timeline. Ideally, if a sporting event is being videotaped with the objective of generating replates, a wide angle high-resolution (e.g., Ultra HD) static camera can be placed facing where most of the interesting action occurs. This avoids all problems related to registration.

5.1 User study

In order to evaluate viewers' reactions to replates, we conducted a small user study. Participants were 29 predominantly male graduate and undergraduate student volunteers.

The first part of the study considers the hypotheses that replates improve viewers' global understanding of a play and enable quick access to events of interest. Participants were exposed, in randomized order, to the original footage for a soccer replay, as well as a replate based on the same footage. They were then asked to rate their agreement with the following statements concerning each stimulus, in a scale between 1 and 5:

- *I can quickly inspect any given part of the play*

Q0. Replate;
Q1. Standard replay.

- *I have a clear global understanding of the entire play*

Q2. Replate;
Q3. Standard replay.

The second part of the study evaluates the selective slow-motion effect. Participants were exposed to three different stimuli, in randomized order, based on a soccer replate involving a pass and a kick that leads to a goal: A real-time replate, a slow-motion replate, and a selective slow-motion replate in which only the kick happens in slow-motion. Subjects were then asked to rate their agreement with the following statements concerning each stimulus, in a scale between 1 and 5:

- *I can see the kick in detail*

Q4. Slow-motion replate;
Q5. Real-time replate;
Q6. Selective slow-motion replate.

- *This video takes too long*

Q7. Slow-motion replate;
Q8. Real-time replate;
Q9. Selective slow-motion replate.

Results are given in tables 2–4. Self-reported ability to inspect any part of the play and grasp the big picture are higher for replate than for a traditional replay ($p = 0.003$ and $p < 0.001$, respectively). Selective slow-motion is better at allowing users to clearly see the kick than a real-time replate ($p = 0.046$), while at the same time keeping the viewers' interest better than the slow-motion replay

Table 2. Results of user study.

Q	μ	σ	Q	μ	σ
Q0	4.14	0.93	Q5	3.75	0.75
Q1	3.14	1.21	Q6	4.14	0.80
Q2	4.36	0.73	Q7	4.29	0.71
Q3	2.86	1.04	Q8	1.82	0.77
Q4	4.57	0.63	Q9	2.21	0.92

Table 3. Paired samples test.

Pair	μ	t	p
Q0 – Q1	1.00	3.29	0.003
Q2 – Q3	1.50	5.37	0.000

Table 4. Within-subjects contrasts.

Pair	μ	F	p
Q6 – Q4	-0.43	6.66	0.016
Q6 – Q5	0.39	4.37	0.046
Q7 – Q9	2.07	148.41	0.000
Q8 – Q9	-0.39	5.14	0.032

($p < 0.001$). The remaining (negative) results were expected, since the slow-motion replay includes at least as much detail as the selective slow-motion replate, and the real-time replate is as long as the selective slow-motion replate.

6 CONCLUSION

This paper introduces the replate, a new approach to visualize sports action plays. It presents a processing pipeline and interactive editing tool to generate replates using videotaped footage from a variety of sporting events. By placing all of the important subjects in short looping animations over a static background, replates allow viewers to quickly analyze and inspect any moment of the play. We present several new features that allow the author to better convey the key aspects of the play, including selective slow motion, variable number of subject instances, localized sound, and annotations. Due to their effectiveness and small duration and memory footprint, we hope that replates will become a popular means of showing action plays in different platforms, particularly in web news outlets and on television (e.g., for analysis during intermissions of sporting events).

ACKNOWLEDGMENTS

This work was partly supported by Hong Kong grants GRF16208814 and HKUST-FSGRF16EG08, and by research grants from CNPq (PQ) and FAPERJ (JCNE).

REFERENCES

Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. 2005. Panoramic Video Textures. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)* (2005), 821–827.

Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. 2006. Action Synopsis: Pose Selection and Illustration. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)* (2006), 667–676.

Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. 2012. Selectively De-animating Video. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012)* (2012), 66:1–66:10.

O. Barnich and M. Van Droogenbroeck. 2011. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *Trans. Img. Proc.* 20, 6 (June 2011), 1709–1724.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* 110, 3 (June 2008), 346–359.

Matthew Brown and David G. Lowe. 2007. Automatic Panoramic Image Stitching Using Invariant Features. *Int. J. Comput. Vision* 74, 1 (Aug. 2007), 59–73.

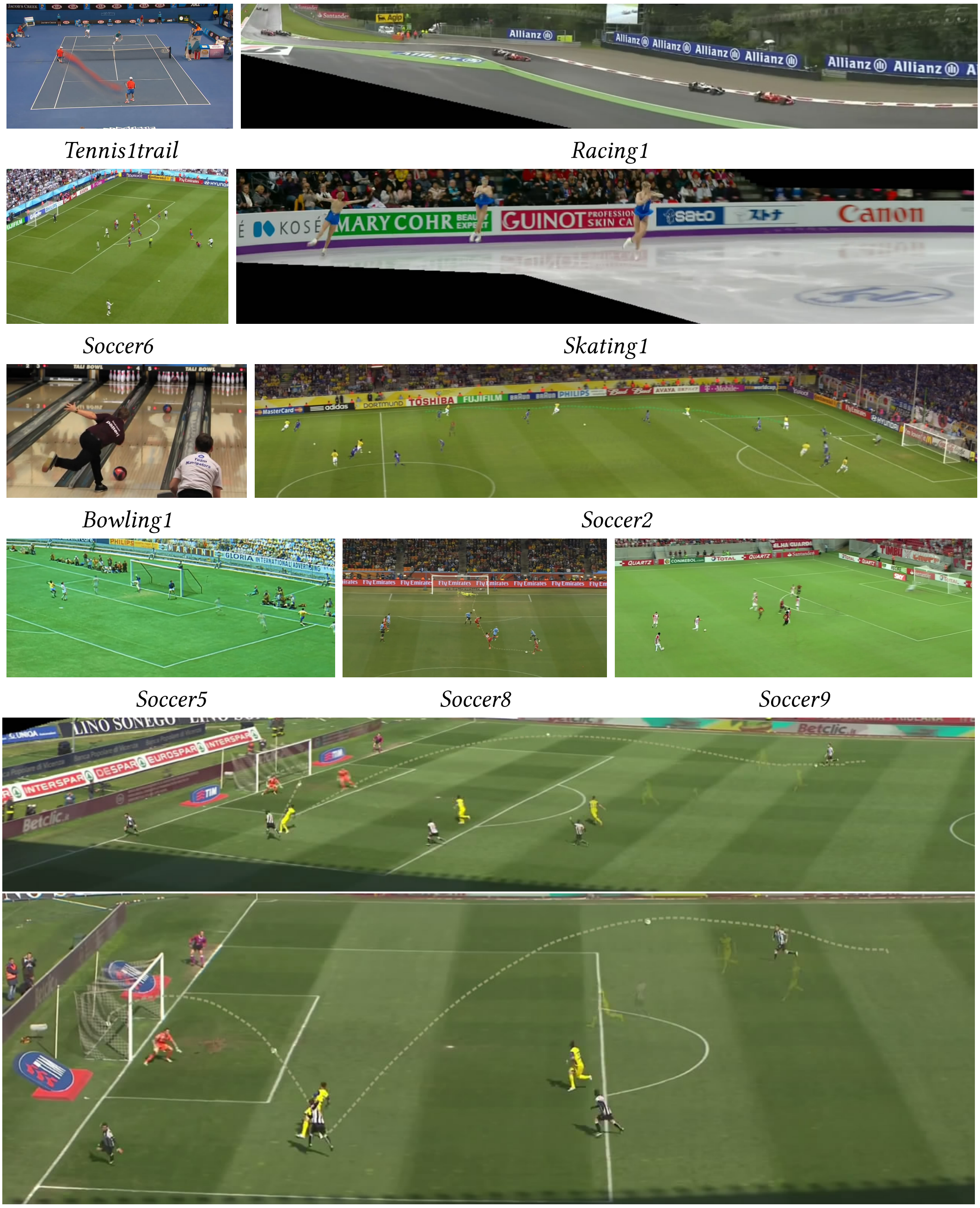
Yaron Caspi, Anat Axelrod, Yasuyuki Matsushita, and Alon Gamliel. 2006. Dynamic Stills and Clip Trailers. *The Visual Computer* 22, 9 (2006), 642–652.

J. P. Collomosse, D. Rowntree, and P. M. Hall. 2005. Rendering Cartoon-style Motion Cues in Post-production Video. *Graph. Models* 67, 6 (Nov. 2005), 549–564.

Carlos D. Correa and Kwan-Liu Ma. 2010. Dynamic Video Narratives. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2010)* (2010), 88:1–88:9.

M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. 1995. Mosaic Based Representations of Video Sequences and Their Applications. *Signal Processing: Image Communication, special issue on Coding Techniques for Low Bit-rate Video* (1995).

M. Irani, S. Hsu, and P. Anandan. 1996. Efficient Representations of Video Sequences and Their Applications. *Signal Processing: Image Communication, special issue on Image and Video Semantics: Processing, Analysis, and Application* (1996).



Tennis1trail

Racing1

Soccer6

Skating1

Bowling1

Soccer2

Soccer5

Soccer8

Soccer9

Soccer1 (two angles: A & B)

Fig. 9. Snapshots of several of the replates available in the supplemental material.

Neel Joshi, Sisil Mehta, Steven Drucker, Eric Stollnitz, Hugues Hoppe, Matt Uyttendaele, and Michael Cohen. 2012. Cliplets: Juxtaposing Still and Dynamic Imagery. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. 251–260.

B. Kim and I. Essa. 2005. Video-based Nonphotorealistic and Expressive Illustration of Motion. In *Proceedings of the Computer Graphics International 2005 (CGI '05)*. 32–35.

Zicheng Liao, Neel Joshi, and Hugues Hoppe. 2013. Automated Video Looping with Progressive Dynamism. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2013)* (2013), 77:1–77:10.

Manao, Fizick, Tsp, TSchniede, and SEt. 2004–2011. MVTools. http://avisynth.org.ru/mvtools/mvtools2.html. (2004–2011).

Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2012. Video Summagator: An Interface for Video Summarization and Navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. 647–650.

Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. 2008. Nonchronological Video Synopsis and Indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 11 (Nov. 2008).

Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. 2013. Video Textures. In *Proceedings of ACM SIGGRAPH 2013*. 489–498.

SportsVision. 2014. http://www.sportvision.com. (2014).

Alexandru Telea. 2004. An Image Inpainting Technique Based on the Fast Marching Method. *J. Graphics, GPU, & Game Tools* 9, 1 (2004), 23–34.

Laura Teodosio and Walter Bender. 2005. Salient Stills. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (Feb. 2005), 16–36.

Graham Thomas. 2012. *Sports TV Applications of Computer Vision*. Technical Report. British Broadcasting Corporation.

James Tompkin, Fabrizio Pece, Kartic Subr, and Jan Kautz. 2011. Towards Moment Imagery: Automatic Cinemagraphs. In *Proceedings of the 2011 Conference for Visual Media Production (CVMP '11)*. 87–93.

Vizrt. 2014. http://www.vizrt.com/products/viz_libero/. (2014).