Video Collections in Panoramic Contexts Supplemental Material

James Tompkin[1] Fabrizio Pece[2] Rajvi Shah[1,3] Shahram Izadi[2,4] Jan Kautz[2] Christian Theobalt[1]
MPI für Informatik[1] University College London[2] IIIT Hyderabad[3] Microsoft Research[4]

INTRODUCTION

This document contains additional information on video alignment, the display applications exploration, and on the user study conducted to evaluate our system. We give details on homography validation, estimating missing homographies, and homography filtering; we report on methods to extended Vidicontexts to work on portable devices and spherical displays; the iMovie and iMovie+pano interfaces used in our experiment are described in more detail; and we include a additional descriptions of the results obtained from our user study presented in the main paper.

FURTHER DETAILS ON VIDEO ALIGNMENT

In this section, we provide further details on the video-to-panorama alignment process.

Homography Validation

If $p = (x, y, w)$ and $p' = (x', y', w')$ are homogeneous coordinates of two corresponding points in a video frame and in the panorama, then they are related by a 3x3 transformation matrix H as given below, where H is called a homography.

$$p' = H.p \tag{1}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \tag{2}$$

Since the expected transformation between the spherically projected video frames and the spherical panorama is only a translation, we perform a conservative outlier rejection and discard homographies which are not projective or have a large skew along any direction. The MATLAB function for homography validation is shown in Figure 1.

```
function valid = validHomography(H)
% Test conditions for invalid homographies.
valid = true;

% Degenerate homographies.
N = 1000;
D = det(H);
if( D < 1/N || D > N )
    valid = false;
end

% Orientation reversing homographies.
A = H(1:2,1:2);
if( det(A) <= 0 )
    valid = false;
end

% Eigenvalue ratio is too large.
maxEigValRatio = 3;
[v w] = eig(A);
evRatio = max(diag(w))/min(diag(w));
if(evRatio > maxEigValRatio)
    valid = false;
end

% Foreshortening factor is too small;
% less than 1/3 along each direction.
if( w(2,2)*w(1,1) < (1/3).^2 )
    valid = false;
end

% Projectivity test.
maxVar = 0.01;
if( H(3,1).^2 + H(3,2).^2 < maxVar*maxVar)
    valid = false;
end
```

Figure 1: MATLAB function to resolve homography validity.

Estimation of Missing Homographies

Due to conservative rejection, it is possible that no good homographies are found for some video frames as the initial spherical projection of the video frame, which uses absolute sensor orientation data, is inaccurate. We explain the method used in our system to estimate these missing homographies using neighboring homographies and relative orientation information. Suppose the homographies for frame k and frame $k+N$ are known and the homographies for the intermediate frames i, where $i \in [k+1, k+N)$, are missing.

We project the corner positions of frame rectangles within panorama space using sensor rotation data. Let us denote projected corners of frame i as TL_i, TR_i, BL_i, and BR_i. Using these corner positions we estimate the angle between neighboring frames after orientation-based projection. We also compute the x and y translation of these projected frames as the difference between the centroids of these corners after projection.

Let us denote the neighboring frames as i and $i+1$, the translation as tx_i and ty_i and the angle between edge vectors e_i and

UIST'13, October 8–11, 2013, St. Andrews, United Kingdom.

http://dx.doi.org/10.1145/2501988.2502013

e_{i+1} as θ_i. We can estimate θ_i using the dot product:

$$e_i = TR_i - TL_i \tag{3}$$

$$\theta_i = \arccos\left(\frac{e_i \cdot e_{i+1}}{\sqrt{\|e_i\|\,\|e_{i+1}\|}}\right) \tag{4}$$

The dot product only gives us a positive θ_i. To discover the sign of the angle, we need to define a normal to the surface N, and use the cross product:

$$\theta_i = \begin{cases} -\theta_i, & \text{if } N \cdot (e_i \times e_{i+1}) < 0. \\ \theta_i, & \text{otherwise.} \end{cases} \tag{5}$$

In our 2D case, $N = [0,0,1]$, with e vectors zero padded in z. More simply, this operation reduces to checking the sign of $(e_i^x e_{i+1}^y) - (e_i^y e_{i+1}^x)$.

We use the signed θ_i, tx, and ty to compute the corresponding affine transform between every pair of neighboring frames:

$$H_{\theta_i} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & tx_i \\ sin(\theta_i) & cos(\theta_i) & ty_i \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The homography between frame $k+j$ and the panorama can be then estimated as the cumulative multiplication of the latest known homography matrix H_k and the estimated affine projection matrices H_{θ_i}, where $i \in [k+1, k+j]$.

$$H_{k+j} = H_k \cdot H_{\theta_{k+1}} \cdot \ldots \cdot H_{\theta_{k+j}} \tag{7}$$

Temporal Filtering

Since the homography for each frame is estimated independently, some temporal jitter is observed due to small but independent alignment errors. We bilaterally filter over time the frame corner positions to reduce temporal jitter. We modulate the contribution of each filter window position (temporal weight) by the image-space Euclidean distance from the center window position (range weight).

Let us denote the four corner points for frame i as P_i^k and the filtered corner locations as Q_i^k, where $k \in [1,4]$. Let us denote the centroid of the four corner points for frame i as P_i^c. The filtered locations are estimated using a bilateral filter:

$$Q_j^k = \frac{\sum_{i=j-T}^{j+T} W_t(i) \cdot W_s(P_i^c, P_j^c) \cdot P_i^k}{\sum_{i=-T}^{T} W_t(i) \cdot W_s(P_i^c, P_j^c)} \tag{8}$$

$$W_t(i) = e^{\frac{-i^2}{\sigma_t^2}} \tag{9}$$

$$W_s(P_i, P_j) = e^{\frac{-\|P_i - P_j\|^2}{\sigma_s^2}} \tag{10}$$

Here, the temporal filter W_t is the domain filter which ensures temporal smoothing. The spatial filter W_s is the range filter, which ensures that when the frame position difference is large, i.e., due to a sudden change in camera position or erroneous homography, it is not propagated through smoothing. We use the temporally filtered corner points Q_i^k to estimate inter-frame homographies, and we multiply these with the original homographies to produce a temporally smooth series of transformations.

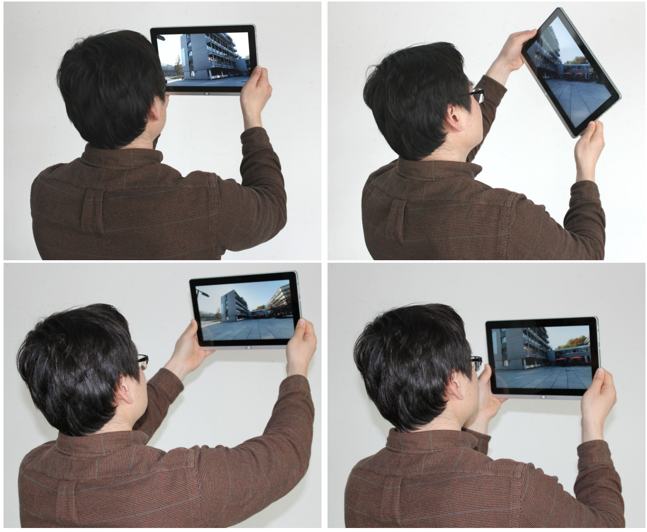


Figure 2: *Top:* The tablet interface is free to rotate along all axes in space to provide a virtual window. *Bottom:* Front camera face tracking provides zoom control.

EXPLORING DISPLAY APPLICATIONS

The video-collection+context representation naturally fits display and interaction devices beyond desktop environments. We extend Vidicontexts to work on portable devices, such as tablets, and spherical displays. While our desktop interface shows either a perspective projection or an equirectangular projection, this exploration of display applications maps the panorama to both virtual and real spatially-located spheres. We demonstrate our system running on a flat display, a tablet, and a spherical display in our supplemental video.

Tablet Interface

Mobile devices can naturally respect the geometry of 'inside→out' video collections, with a display that can respond to rotation and head movements. Similarly to [2], our tablet interface performs perspective projection camera control through the device's orientation sensors, allowing the user to physically rotate the device to navigate the context (Fig. 2). In this way, the real proxy geometry of the scene is maintained as the user explores with a virtual window. A simple button press locks the orientation rotation and returns control of the virtual camera to touch. Additionally, we use the front-facing camera and an off-the-shelf face tracker to provide zoom control: as the user moves their head closer to and farther from the tablet, the view zooms in and out. Our video shows user interaction with Vidicontexts running on an Acer W700 tablet. This scheme also fits naturally to head-mounted displays (HMDs), which will increase in relevance with upcoming low-cost HMDs such as the Oculus Rift.

Spherical Interface

In this example, our context is displayed on a physical sphere[1] in tandem with complementary controller interfaces. Multi-

[1] Global Imagination's Magic Planet.

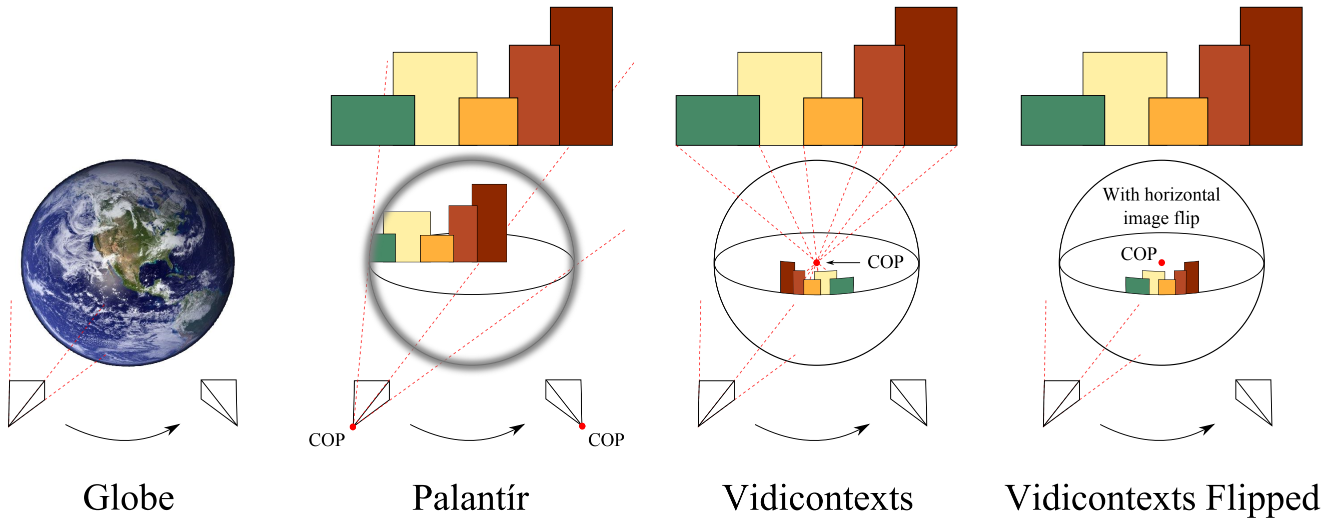


Figure 3: *Left:* World globe - if the user changes their viewpoint, then they will reveal content located on the far side of the sphere. *Centre Left:* Palantìr - changing viewpoint reveals different areas of the projected space similarly to what happens when moving past a window. *Center-Left:* Vidicontexts - the projected world appears flipped left to right, and when moving to the right, the world to the left is revealed. *Center-Right:* Vidicontexts with Flip: if we horizontally flip the image, when walking to the right, the world to the right is revealed.

ple users are able to walk around the display to inspect different areas of the context and physically track videos as they move. Users can also control the system through a joypad or a tablet device, though a touch interface is also possible [1]. With the tablet complement, our existing flat display interface acts as a proxy controller, and any view changes on the tablet are reflected on the sphere. With the joypad, users control a cursor on the spherical display, with modifications to exploit the specific controls, e.g., manipulating time in videos using the analog triggers.

While mobile devices and HMDs naturally respect the geometry of ‘inside→out’ video collections, the mapping to a spherical display requires more thought. Users observing a spherical display typically expect it to behave either as a) a world in miniature, such as a globe, or b) a magical seeing stone, or *palantìr*[2], which acts as a portal to another place or world. However, the Vidicontexts case is neither of these, as we explain here and in Figure 3:

Globe: Content on the globe is mapped directly to the spherical display. If the user changes their viewpoint by walking around the spherical display, then they will reveal content located on the far side of the sphere (Figure 3, left). Moving to the right reveals content on the globe farther to the right — the motion/content is consistent with the world in miniature.

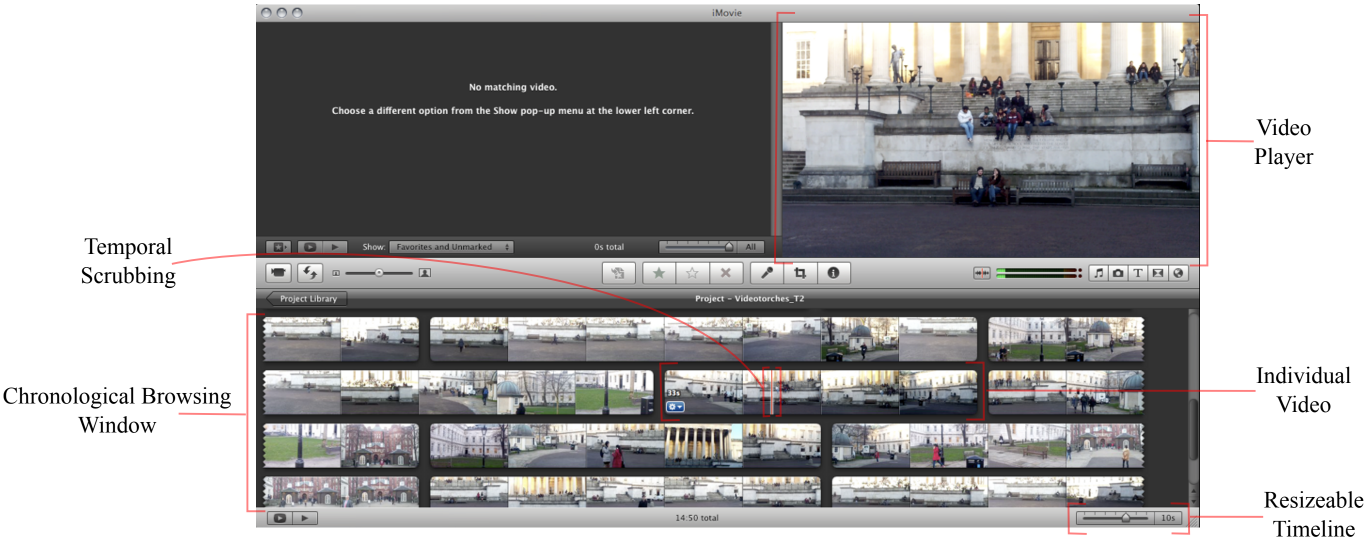
Palantír: The sphere is a portal to a different place. Changing viewpoint reveals different areas of the space through the portal via parallax, similar to what happens when moving past a window (Figure 3, center-left). The sphere boundary as seen from the viewer separates the two places, and the world is projected ‘through’ the sphere to the eyes of the user. Thus, simulating a palantír with a spherical display and correctly rendering the panoramic context requires knowledge of the user’s eye position. This could be discovered with an external head-tracking system, and this would limit the display to a single user.

Vidicontexts: The world to be viewed is projected onto the surface of a sphere, with center of projection at the center of the sphere. This is the creation of the panoramic context by photography. When the context is viewed with a tablet or HMD, the viewer is effectively in the center of the sphere. However, when we map this to the surface of a spherical display, we are now observing the world from *outside* — we have turned the world in on itself. There are two options for this projection:

1. **Flipped:** (Figure 3, center-right) The world is projected onto the sphere. To maintain viewing directions, the world is projected onto the back of the sphere, that is, the sphere-ray intersection points which are farthest from the world when projected through the center of the sphere. When this projection is viewed from outside the sphere, the world appears flipped left to right. As the user walks around the sphere, the world is revealed as per the palantír case, where movement to the right reveals the world to the left. However, the whole world is horizontally flipped.
2. **No flip/bad parallax:** (Figure 3, right) If we horizontally flip the image to try to correct this problem, the world appears correct from a single viewpoint. However, now, when walking to the right, the world to the right is revealed rather than the expected parallax effect of the world to the left being revealed.

Without head tracking, it is impossible to reconcile these two problems as we are viewing the world inverted. Either

[2] From *The Lord of the Rings* literary saga, by J. R. R. Tolkien.



No matching video.

Choose a different option from the Show pop-up menu at the lower left corner.

Figure 4: The iMovie interface used in our user study.

the world is horizontally flipped and movement around the spherical display is correct, or the world is not flipped and movement is inverted. The influence on users of these effects is not straightforward to understand or quantify. Future work should experimentally investigate the three options presented to try to estimate the impact on users perception and performance of these projection methods for spherically displaying ‘inside→out’ video collections.

Augmented Reality

Our representation is also useful in augmented reality applications where the goal is to compare videos in situ using the real world as a context. This situation might occur as a curated experience at a cultural heritage site, or as a virtual tourism application where participants are GPS guided around a city and stand in hotspots to compare videos of past events with the current situation. GPS and orientation data are often sufficient for rough registration with the environment and, with this, in our example the user sees a protest in video that no longer exists in the real environment (Fig. 8, *right*, main paper). If a vision-based registration between mobile device and environment is required, with our representation the back-facing camera image need only be registered with the panorama once in real-time for all videos in the collection to be registered. In this case, the camera image would replace the panorama in our interface, though we leave this fine registration for future work.

USER STUDY

This section introduces the iMovie and iMovie+pano interfaces in more detail for readers who are not familiar with them. We also present a full description of the analysis performed on the user study results.

iMovie Interface

To evaluate Vidicontexts, we decided to compare our system with *iMovie* as a baseline, and against iMovie with the panoramic context image available for reference (*iMovie+pano* henceforth). iMovie (Figure 4) is consumer software typically used for non-linear video editing and, as its intended users are novices, it presents an intuitive interface. As part of this interface for novices, it includes tools for browsing video collections and finding video content with which to edit. For our experiment, we ignore all of the editing features of iMovie and use only the intuitive video browsing tools. These tools are all accessible from the main window: 1) a chronological browsing area that displays videos as thumbnails and lets user skim through a video collection using hover scrubbing, 2) a resizeable timeline that can expand and contract the unit of time that each video thumbnail represents, and 3) a large panel used for video playback.

Initially, each video in the collection is presented as a single thumbnail and placed in chronological order in the browser. The user can expand the video into multiple thumbnails by using the timeline: coarser expansion values increase the time represented by each video thumbnail and so provides a collection overview, while finer values show more of the video as thumbnails and allow more time instances to be visible at once. Once a desirable video is found, the user can either select and play an entire video, or can hover the mouse over the thumbnails to scrub though that video section. iMovie offers additional functionalities for video editing, such as voice recording or video cutting, which we did not use in our study.

In the iMovie+pano condition, users could also view a panoramic context for reference. The panorama of the scene was displayed at the same resolution as the one employed in Vidicontexts and in a separate window, and it was left to the user how they arranged their desktop space. All our users kept both iMovie and the panorama as full-screen windows and switched between having iMovie and the image viewer in the foreground. Most of our users switched back and forth throughout the tasks to view the reference image. Only a few users employed a different strategy: they viewed the context panorama once at the beginning of the task to obtain an idea of the surrounding space, and then focused only on the iMovie interface.

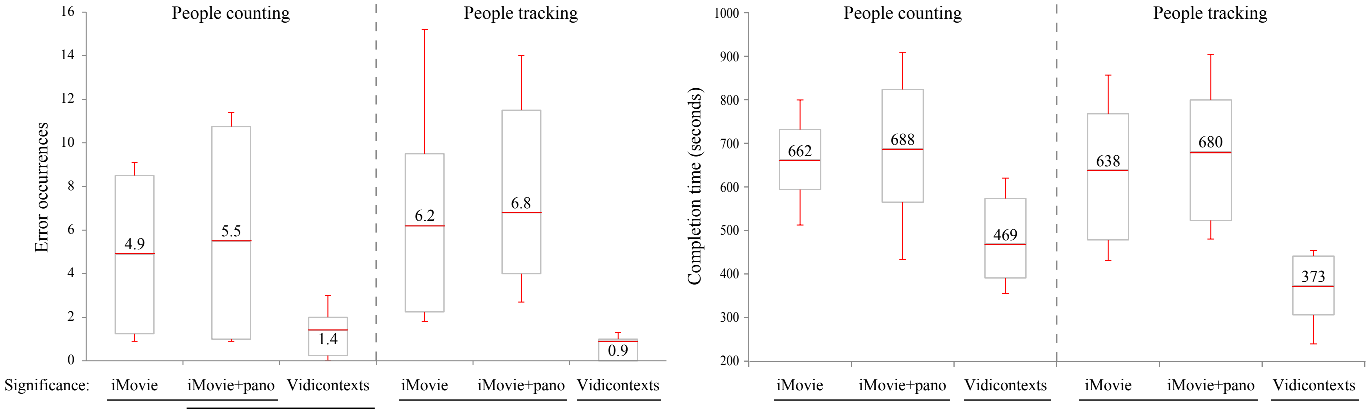


Figure 5: Box and whisker plots for each condition in both tasks. *Left:* Error occurrences. *Right:* Time to complete occurrences. First and third quartiles are reported in the upper and lower boxes, respectively. Any conditions whose name are underlined are considered statistically similar. *This is a duplicate of Fig. 7 from the main paper.*

Results — Tasks

Figure 5 is duplicated from the main paper for reference. It shows box and whisker plots for the completion time, the number of errors committed, and the significance for each task and condition combination. We computed Analysis of Variance (ANOVA) using SPSS with the system used as the single factor and completion time/counting error as the dependent variable, with post-hoc Games-Howell tests for pairwise significance tests ($\alpha < 0.05$). There was no significant difference between the iMovie and the iMovie+pano cases across all our experiments.

For the *people counting* task, a non-significant main effect of the system used was found ($F_{(2,1)} = 3.09$, $p = 0.06$), with fewer errors for Vidicontexts (*Mean*, $M = 1.4$) and iMovie+pano ($M = 5.5$) than for iMovie ($M = 4.9$). Post-hoc analysis revealed non-significant differences between Vidicontexts and iMovie+pano ($p = 0.107$), and significant differences between our system and iMovie ($p = 0.04$). A main effect was not found between iMovie and iMovie+pano ($p = 0.958$).

For completion time, the system used was a significant factor ($F_{(2,1)} = 5.60$, $p = 0.009$), with less time for Vidicontexts ($M = 469$ sec.) and iMovie ($M = 662$ sec.) than for iMovie+pano ($M = 688$ sec.). Post-hoc analysis revealed a significant difference between Vidicontexts and both iMovie ($p = 0.017$) and iMovie+pano ($p = 0.023$), and non-significant differences between iMovie and iMovie+pano ($p = 0.916$).

For *people tracking*, the system used was a significant main effect ($F_{(2,1)} = 5.08$, $p = 0.013$), with fewer errors for Vidicontexts ($M = 0.9$) and iMovie ($M = 6.2$) than for iMovie+pano ($M = 6.8$). Post-hoc analysis revealed significant differences between Vidicontexts and iMovie ($p = 0.049$), as well as between Vidicontexts and iMovie+pano ($p = 0.012$). No main effect was found between iMovie and iMovie+pano ($p = 0.968$).

For completion time, the system used was a significant factor ($F_{(2,1)} = 7.16$, $p = 0.003$). Vidicontexts obtained the lowest mean time ($M = 373$ sec.), followed by iMovie ($M = 638$ sec.) and iMovie+pano ($M = 680$ sec.). Post-hoc analysis revealed a significant difference between Vidicontexts and both iMovie ($p = 0.014$) and iMovie+pano ($p = 0.005$), but non-significant differences between iMovie and iMovie+pano ($p = 0.898$).

Results — Questionnaires

For the usability questionnaire, only our system scored above average ($SUS = 77.5$), followed by the iMovie+pano ($SUS = 62.75$) and iMovie ($SUS = 59.5$) conditions. Following the SUS classification technique of Lewis et al. [3] (letter-grade ranks varying from A to F), Vidicontexts is a Rank B system, while both iMovie and iMovie+pano mode are Rank C systems. Rank A systems have many promoters, who will definitely use and recommend the product. Rank B systems have a fair number of promoters, who are likely to use and promote the product. All other ranks will only have detractors.

For the task-related questionnaire, both iMovie and iMovie+pano conditions performed poorly, with mean scores of $M = 2.375$ and $M = 2.62$ respectively. Our system scored higher on this questionnaire, with a mean of $M = 3.86$. There is a significant main effect ($p = 0.001$), and post-hoc analysis reveals significant differences between Vidicontexts and iMovie ($p = 0.001$), as well as between our system and iMovie+pano ($p = 0.003$).

REFERENCES

1. Benko, H., Wilson, A. D., and Balakrishnan, R. Sphere: multi-touch interactions on a spherical display. In *Proc. the 21st annual ACM symposium on User interface software and technology*, ACM (New York, NY, USA, 2008), 77–86.

2. Joshi, N., Kar, A., and Cohen, M. Looking at You: Fused Gyro and Face Tracking for Viewing Large Imagery on Mobile Devices. In *Proc. SIGCHI '12*, ACM (New York, NY, USA, 2012), 2211–2220.

3. Lewis, J., and Sauro, J. The Factor Structure of the System Usability Scale. In *Proc. Human Centered Design*, Springer (2009), 94–103.