Rich360: Optimized Spherical Representation from Structured Panoramic Camera Arrays

Jungjin Lee[1] Bumki Kim[1] Kyehyun Kim[1] Younghui Kim[2] Junyong Noh[1]

[1]KAIST [2]KAI Co., Inc.

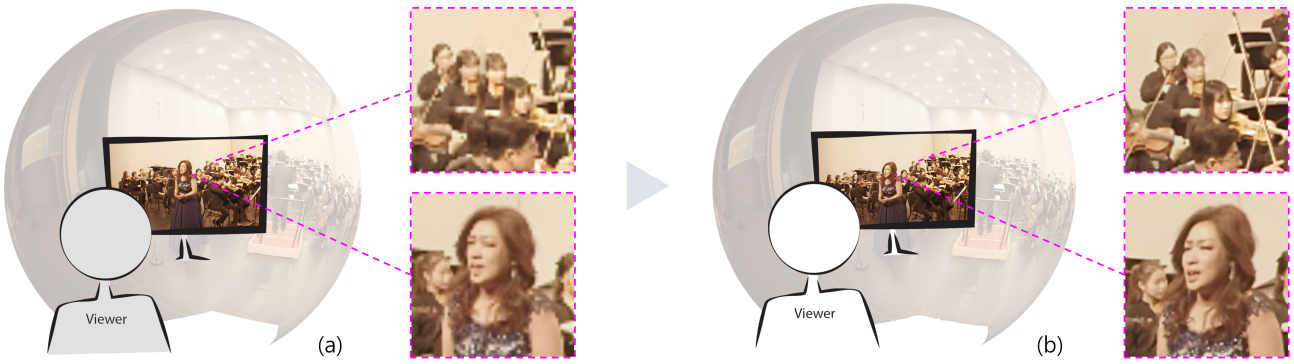


Figure 1: *(a) There are two major issues in 360° panoramic video creation using multiple cameras: parallax in the overlapping regions and loss of richness caused by downsampling. (b) Rich360 handles the two issues with a deformable spherical projection surface and non-uniform ray sampling.*

Abstract

This paper presents Rich360, a novel system for creating and viewing a 360° panoramic video obtained from multiple cameras placed on a structured rig. Rich360 provides an as-rich-as-possible 360° viewing experience by effectively resolving two issues that occur in the existing pipeline. First, a deformable spherical projection surface is utilized to minimize the parallax from multiple cameras. The surface is deformed spatio-temporally according to the depth constraints estimated from the overlapping video regions. This enables fast and efficient parallax-free stitching independent of the number of views. Next, a non-uniform spherical ray sampling is performed. The density of the sampling varies depending on the importance of the image region. Finally, for interactive viewing, the non-uniformly sampled video is mapped onto a uniform viewing sphere using a UV map. This approach can preserve the richness of the input videos when the resolution of the final 360° panoramic video is smaller than the overall resolution of the input videos, which is the case for most 360° panoramic videos. We show various results from Rich360 to demonstrate the richness of the output video and the advancement in the stitching results.

Keywords: video stitching, spherical video, spherical rendering, spherical representation, panoramic video

Concepts: •**Computing methodologies** → **Image and video acquisition; Computational photography;** *Image representations;*

SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925983

1 Introduction

Unlike a typical rectangular video that shows only the front view of a scene, a 360° panoramic video captures omni-directional lights from the surrounding environment. This allows a viewer to interactively look around the scene, possibly providing a strong sense of presence. This potential change of the viewing paradigm arising from the use of 360° panoramic videos has attracted much attention from the industry and the general public. Panoramic video streaming services are now available through companies such as *Youtube* and *Facebook* and head-mounted display devices such as *Samsung GearVR* and *Oculus Rift* that support 360° viewing are starting to be widely deployed. Content creators have begun to produce 360° panoramic videos in order to deliver stories with more visually immersive experiences than previously possible.

One of the most popular methods to produce a professional 360° panoramic video utilizes a structured panoramic rig and multiple wide-angle, small-sized cameras (e.g. *GoPro*) with 2K or higher resolution (Figure 2). Two additional steps follow before obtaining a final 360° panoramic video. First, the videos obtained from the multiple cameras are aligned on a viewing sphere surface. Blending and exposure adjustments are required for smooth transition between adjacent videos. Second, the final spherical video is rendered through uniform sampling of the sphere surface according to the target rectangular resolution (i.e. equirectangular projection). A typical 360° panoramic video player projects the spherical video onto a 3-dimensional sphere and renders a desired scene with a virtual camera placed in the middle of the sphere. The viewer can then rotate the camera to interactively navigate through the video space.

We observed two problems in the existing workflow that can degrade the quality and the richness of the original source obtained from the multiple cameras (Figure 1(a)). First, the parallax between cameras may cause disturbing artifacts such as misalignment and discontinuity [Szeliski 2006]. While use of mirrors can be a remedy to minimize the parallax, a full 360° panoramic video cannot be produced in this manner due to occlusion. Second, the reso-

Figure 2: *Two widely used structured panoramic rigs with six GoPro cameras: Freedom360 and 360Heros.*

lution of the final 360° panoramic video is usually smaller than that of the original source videos combined. For example, a common panoramic rig captures more than 30 million pixels using six or more cameras. However, rendering a 4K resolution image from these sources will end up wasting more than half of the information. Moreover, because the virtual camera displays only a part of the video at any given time, the resolution of the video that the viewer experiences is even lower. Previous research on panoramic video creation has concentrated mostly on the first issue while neglecting the second.

In this paper, we present Rich360; a novel system that provides the viewer with an as-rich-as-possible 360° panoramic video by efficiently handling the aforementioned two issues (Figure 1(b)). The Rich360 pipeline is specifically designed for structured camera arrays. The first step is to perform 3D calibration of the cameras placed on the rig. While previous video stitching approaches typically apply pairwise video warping in an undeformed projected space [Perazzi et al. 2015; Jiang and Gu 2015], Rich360 deforms the projection sphere to minimize parallax artifacts using a small number of 3D points recovered from overlapping image regions. Therefore, fast and efficient stitching is possible with a single optimization regardless of the number of input videos. In the rendering process, we perform an importance-based non-uniform ray sampling to preserve the information of the source as much as possible. Rich360 also allows the content creator to manually assign a higher resolution to desired regions in a keyframing manner. Finally, for playback, the resulting non-uniform 360° panoramic video is mapped onto a uniform sphere using a UV map that contains ray information. We demonstrate that Rich360 delivers much richer visual information compared to that produced by existing methods, through various scenes and resolution tests.

The main contributions of this work can be summarized as follows:

- A novel approach to creating and viewing an as-rich-as-possible 360° panoramic video under a given resolution
- An efficient stitching method that employs a deformable spherical projection surface where calibrated videos are projected with minimal parallax artifacts
- Non-uniform spherical ray sampling that assigns an appropriate resolution according to the importance of image regions

2 Related Work

Parallax Removal Image stitching combines overlapping views from multiple cameras into a wide field of view single panoramic image. This process starts with the geometric alignment of multiple views in a common image space. The mathematical models of the motions between the views have been well established [Szeliski 2006] for the ideal situation where the images are captured from the same center of projection. However, it is difficult to assume such a configuration in the real world. For general use, it is necessary to address the parallax effect caused by the use of multiple cameras. A common strategy is to find feature (or pixel) correspondences in the overlapping region and to warp the images in the common 2D space to match the correspondences. Advanced warping methods for parallax-free image stitching have been introduced [Uyttendaele et al. 2004; Lin et al. 2011; Zaragoza et al. 2014; Chang et al. 2014; Zhang and Liu 2014; Lin et al. 2015; Li et al. 2015]. Another strategy similar to our stitching method utilizes depth information in the overlapping region to synthesize a novel view using image-based rendering [Shum et al. 2008; Chaurasia et al. 2013]. Uyttendaele et al. [2004] compensated for the parallax in the overlapping region by recovering a multiperspective image using a plane sweeping method. Methods generating a seamless light-field panoramic image have also been proposed [Richardt et al. 2013; Birklbauer and Bimber 2014]. However, they require a relatively dense ray space acquired by rotating the camera.

Handling parallax for video stitching is more challenging due to moving objects and camera motion. Applying image stitching methods to each frame individually would result in noticeable jittering artifacts over the sequence [Perazzi et al. 2015]. A few attempts to resolve this were recently reported. Zhi and Cooperstock [2012] compute a depth map for the overlapping regions and extrapolate the depth using color segments. For dynamic image stitching, their method synthesizes the panoramic images of foreground (i.e. moving objects) and background layers separately. Perazzi et al. [2015] extended the local warping method based on the optical flow, for parallax removal of unstructured camera arrays. Pairwise warping fields are computed using a weighted warp extrapolation and temporal instability is resolved by a constrained global relaxation step per frame. Jiang and Gu [2015] explicitly formulated the video stitching problem as a spatio-temporal mesh optimization that is built upon the method proposed by Zhang and Liu [2014]. Bundle adjustment for all the frames is required to minimize the accumulated deformations caused by the pairwise warping. Unlike previous approaches that utilize pairwise video warping, Rich360 deforms the projection sphere to compensate for the disparities in the overlapping regions while preserving the spatio-temporal smoothness. Consequently, calibrated videos are projected onto the deformed sphere with minimal parallax artifacts. As the deformation can be formulated as a single linear system regardless of the number of the input videos, our method is faster and more efficient than previous methods.

Spherical Projection Our non-uniform spherical ray sampling optimizes the projection mapping from the deformed 3D sphere to a 2D image plane while considering the importance of image regions. Similarly, there have been previous works that generate optimal projections. Because a sphere is not developable, fitting a panoramic and wide-angle image into a flat image inevitably introduces shape distortions [Zelnik-Manor et al. 2005]. Kopf et al. [2009] introduced locally-adapted projections to reduce panorama distortions (e.g. curving straight lines) for better perception of salient shapes. The cylindrical projection surface is deformed to become planar in user-specified regions to avoid the formation of curved lines in the final projection image. Content-preserving projection [Carroll et al. 2009] maps a wide-angle image defined on the viewing sphere to a flat image with minimal shape distortion, while preserving salient features. These previous studies focused on providing a user with a distortion-free wide-angle image. In contrast, Rich360 is designed for a different purpose of assigning a wider projection area to important regions, resulting in a distorted appearance of the rendered spherical video. This is motivated by the observation that a viewer enjoys a 360° panoramic video in an interactive way through a specifically-designed player. The distorted spherical video is efficiently restored during the play-

back.

Mesh-based Video Processing The core of Rich360 is built on a mesh optimization scheme that has been widely utilized for various video applications. Video stabilization [Liu et al. 2009; Wang et al. 2013; Liu et al. 2013] employs mesh optimization in computing spatially-varying warps from each input frame into a stabilized output frame. Lang et al. [2010] proposed a stereoscopic mesh warping technique to change the disparity range of stereoscopic video. He et al. [2013] fit a panoramic image with an irregular boundary into a rectangle via content-preserving mesh warping considering line structures in the image. For temporally coherent resizing of videos, video retargeting [Wang et al. 2009; Krähenbühl et al. 2009; Wang et al. 2010] deforms a grid mesh placed on the image according to the importance. When the aspect ratio of the target display is different from that of an input video, to avoid visible distortions the shape of the important quads is preserved and less important quads are distorted more. We formulate the non-uniform ray sampling of the deformed sphere as a mesh optimization with a few constraints on the target image space in an effort to preserve the information of the source as much as possible. The resulting mesh is used for interpolating the ray.

3 Rich360 Pipeline

The pipeline of Rich360 consists of four steps: calibration, projection, rendering, and viewing. In this section, we briefly describe the purpose and process of each step. In sections 4 and 5, the mathematical details of the projection step and the rendering step are described, respectively.

Calibration To combine views from multiple cameras, existing stitching methods for images or videos mostly utilize feature-based calibration that can be applied to arbitrary environments where the camera arrangement is typically temporary. The success of these methods heavily depends on the quality of the detected features and the identified correspondences. As most of the rigs for 360° panoramic filming are tightly structured, it is reasonable to perform the calibration in advance and reuse the result as a template. In practice, the commercial video stitching software such as *Kolor's Autopano Video* and *VideoStich Studio* already offers usable templates for the off-the-shelf rigs.

The calibration step estimates the intrinsic and extrinsic parameters of the cameras on a structured rig. We adopt the classical calibration method that utilizes a physical pattern like a checkerboard. With the position of the rig fixed, one moves around holding the checkerboard to capture the pattern viewed from each camera. This process is repeated for various distances and angles for accurate calibration. After the patterns are detected, the traditional lens calibration and the stereo calibration between adjacent cameras are performed with a unified projection model proposed for a wide-angle camera [Mei and Rives 2007]. Additionally, all camera parameters are refined by bundle adjustment [Li et al. 2013]. Please refer to the book by Hartley and Zisserman [2003] and recent studies [Mei and Rives 2007; Li et al. 2013] for the complete technical details on the calibration and bundle adjustment. Once all the cameras on the rig are calibrated, the output parameters can be reused for other scenes captured using the same rig.

Projection In the projection step, the input images are individually projected onto a common projection surface according to the corresponding camera parameters. Generally, the projection surface is defined as a sphere for a full 360° panoramic video. A simple sphere, however, cannot represent depth information, which is crucial for parallax handling. Therefore, Rich360 defines a projection surface with varying depth to minimize the disparities in an overlapping region (Section 4). First, the disparities are computed using feature correspondences that are obtained by applying SIFT [Lowe 2004] and an optical flow method [Brox et al. 2004]. The triangulation between adjacent two cameras then recovers 3D points. Finally, the spherical projection surface is deformed using the 3D points as constraints while ensuring spatio-temporal smoothness. As a result, the images projected on the deformed spherical surface have minimal parallax artifacts.

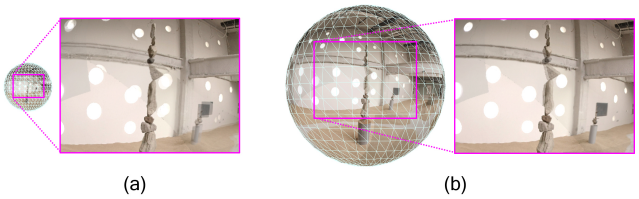


Figure 3: *Projected images onto the sphere with differing radial distances; (a)* $2.1m$ *(b)* $7m$*. The parallax artifacts in the region at the same distance as the radial distance of the sphere are removed.*

Rendering In the rendering step, the stitched spherical images are fitted into a rectangular frame. The traditional equirectangular projection is inefficient when a desired rendering resolution is smaller than the resolution obtained by summing up all of the source videos (i.e. downsampling), as detailed information in the source can be omitted during the sampling. The main strategy of Rich360 is to assign more rays (i.e. pixels) to important regions and fewer rays to less important homogeneous regions to fully exploit the resolution of the source videos. The importance of a region is calculated by the combination of the image gradient, saliency, and face detector. Rich360 also provides a key-frame based mechanism for users who wish to assign the importance manually.

We formulated this non-uniform sampling as an optimization problem that projects the mesh obtained in the projection step onto the target image space that is defined using the spherical coordinates system. First, the target width and the height of a quad face are calculated according to the importance of the pixels within the quad. A quad face with higher importance occupies a bigger area in the target image space to have more pixels assigned for sampling. The final ray mesh is obtained by optimizing the vertex positions in the image space, which satisfies a few constraints including the target size. In this process, the degree of non-uniformness can be controlled through a user parameter. Each pixel in the final video is determined by sampling the pixel of each video projected onto the deformed surface, along the linearly interpolated ray from the surrounding vertices using the spherical coordinates. Finally, we perform blending and exposure adjustment using existing methods [Szeliski 2006] to create a seamless resulting image. The final ray mesh sequence is converted into UV space and stored with a header containing the mesh resolution to be used in the viewing step.

Viewing For interactive viewing, a Rich360 player maps the non-uniformly sampled video onto the viewing sphere with minimal overhead using the ray mesh sequence. First, a viewing sphere is constructed with resolution equal to that of the final ray mesh. Each video frame is applied to the sphere as a texture map. The UV position of each vertex in the viewing sphere is determined through the corresponding UV information in the ray mesh sequence. The rest of the process remains the same as the existing workflow. A virtual camera is created in the center of the sphere, which can be rotated interactively by the user.

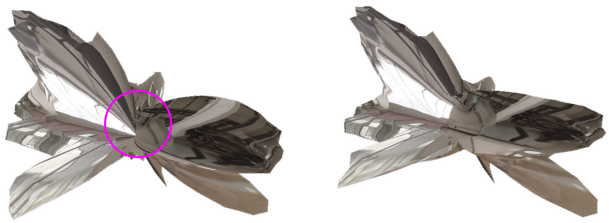


Figure 4: *(a) A resulting mesh from the optimization without use of Equation 3. The magenta circle shows the problematic vertices that have negative radial distances. (b) The same mesh generated using Equation 3 that minimizes the first partial derivatives.*

4 Deformable Spherical Projection Surface

We represent a projection sphere as an $m \times n$ mesh $M = (V, F, Q)$. V denotes a set of vertices $V = \{v_1, \ldots, v_{m \times n}\}$ defined in the spherical coordinate system, where $v_i = (\theta_i, \phi_i, r)$. F and Q denote a set of triangle faces and quad faces, respectively. The quad faces are used in Section 5. Uniform sampling of m horizontal angles ($0 \leq \theta \leq 2\pi$) and n vertical angles ($0 \leq \phi \leq \pi$) produces a sphere mesh. A simple sphere has a single radial distance value r from the origin for all the vertices. However, r can play a role as a zero parallax distance, as shown in Figure 3. Therefore, our goal in the projection step is to generate the optimized projection surface M^t that has locally-varying radial distances r_i^t at frame t to minimize the parallax in the overlapping regions between adjacent images.

A set of 3D points P^t that will constrain the mesh in the optimization step is computed using features in the overlapping regions. We first extract the sparse feature points from each image using SIFT [Lowe 2004]. Feature correspondences are established between two adjacent images to compute disparities. However, the sparse feature matching process may fail to capture the disparities in textureless regions. Therefore, similar to previous stereoscopic image processing methods [Lang et al. 2010], we obtain additional disparities from downsampled dense correspondences that are estimated using an optical flow method [Brox et al. 2004]. Through our experiments, we found that the reliability and the accuracy of both methods decrease when they are applied to the original images, because, first, a wide-angle camera suffers from severe lens distortions in the areas close to frame borders and, second, the orientations of the cameras on the rig are very different. As a remedy, both methods are applied to the spherical images projected onto the default sphere ($r = 5000mm$) similar to Perazzi et al. [2015]. A simple linear triangulation [Hartley and Zisserman 2003] recovers $p_k^t \in P^t$ from the disparities and then p_k^t is converted to the spherical coordinates.

The energy function that measures the difference between r_i^t and p_k^t can be defined as

$$E_p = \sum_k \| \sum_{i \in f(p_k^t)} \lambda_i^t r_i^t - r(p_k^t) \|^2, \tag{1}$$

where $f(p_k^t)$ is a set of indices to vertices comprising a face containing p_k^t in the 2D polar coordinate space (θ, ϕ). λ_i^t is the barycentric weights for (θ, ϕ) of p_k^t with respect to those of the surrounding vertices. Equation 1 enforces the linear combination of surrounding vertices r_i^t to match $r(p_k^t)$ that denotes the radial distance of p_k^t.

To encourage r_i^t to vary smoothly in the resulting mesh, we add a spatial smoothness term as follows.

$$E_{s1} = \sum_i \| r_i^t - \frac{1}{n_i} \sum_{j \in N(i)} r_j^t \|^2, \tag{2}$$

where $N(i)$ denotes the indices of the 4-connected neighbor vertices of v_i^t and n_i is the cardinality of $N(i)$. E_{s1} describes a Laplacian smoothing operation that has been successfully utilized in the field of differential geometry processing [Vollmer et al. 1999]. This term smoothly interpolates r_i^t in the non-overlapping regions by minimizing the second partial derivatives of r_i^t. However, if the gradient of r_i^t in the overlapping regions is relatively large, minimizing E_p and E_{s1} may lead to unstable results that have either negative or very large values.

Therefore, we regularize the vertices that are not constrained by E_p by minimizing the approximate first partial derivatives. The energy functions are as follows.

$$\begin{aligned} E_{dx} &= \sum_{(i,j) \in \bar{\Omega}} \| \tfrac{1}{2} r^t(i+1, j) - \tfrac{1}{2} r^t(i-1, j) \|^2, \\ E_{dy} &= \sum_{(i,j) \in \bar{\Omega}} \| \tfrac{1}{2} r^t(i, j+1) - \tfrac{1}{2} r^t(i, j-1) \|^2. \end{aligned} \tag{3}$$

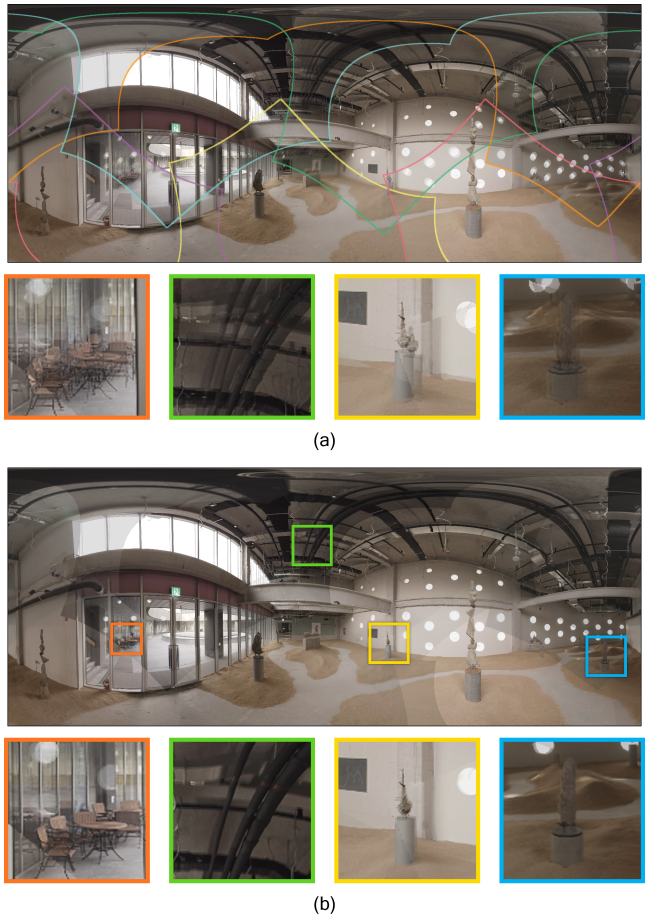
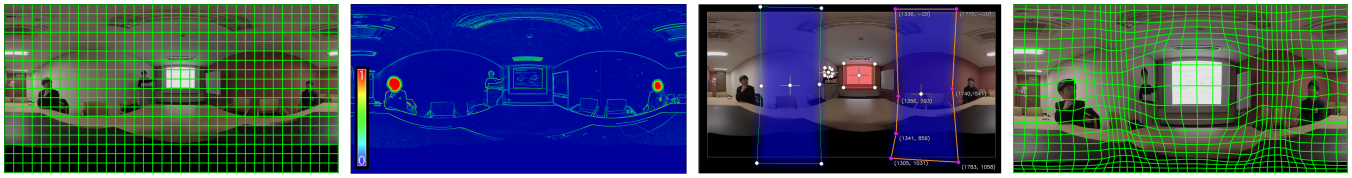


Figure 5: *(a) The equirectangular projection image from the simple sphere ($r = 3000mm$ to remove the parallax of the front stone tower). The color frames represent the field of views of the cameras. (b) The equirectangular projection image from the deformed spherical projection surfaces. The color boxes show close-ups of the overlapping regions at different depths.*



(a) (b) (c) (d)

Figure 6: *(a) The initial arrangement of M^t on the target image. (b) The average importance map of S_x and S_y. The map is color coded from blue (0) to red (1). (c) The user provided polygonal masks where the color indicates the level of importance. (d) The rendered image overlaid with the resulting ray mesh.*

$r^t(i,j)$ represents the radial distance of $v^t_{i\times m+j}$ at the i^{th} row and the j^{th} column of M^t. $\bar{\Omega}$ denotes a set of vertices that are not constrained by P^t. E_{dx} and E_{dy} approximate the first partial derivatives in the horizontal and the vertical direction, respectively. Figure 4 shows the effect of applying Equation 3 to the resulting mesh.

The following energy function takes temporal smoothness into account.

$$E_{t1} = \sum_i \|r^t_i - \frac{1}{2t_w} \sum_{j\in T(i,t)} r_j\|^2, \tag{4}$$

where t_w is the temporal window size and $T(i,t)$ denotes a set of indices to the temporal neighbor vertices of r^t_i from frame $t-t_w$ to $t+t_w$. t_w is set to 3 for all of our experiments. E_{t1} encourages the radial distance of v^t_i to vary smoothly in time.

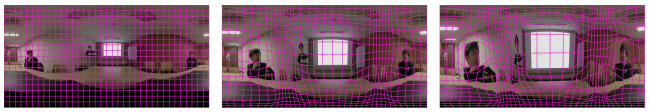
Our final optimization can be expressed as a linear combination of Equation 1 to 4:

$$\arg\min_{\{r^t_i\}_{t,i}} \sum_t \alpha_p E_p + E_{s1} + E_{dx} + E_{dy} + E_{t1}, \tag{5}$$

where α_p is the weight for E_p, which determines the influence of the constraining 3D points over the projection surface. We set $\alpha_p = 2$ for all of our experiments. Solving the linear system of Equation 5 produces the deformed spherical projection surfaces at each frame where the images are projected with minimal parallax artifacts (Figure 5(b)), while preserving the spatio-temporal smoothness. Because our optimization scheme computes the meshes for multiple videos simultaneously at every frame, a long video may require a huge amount of memory to process. We take an approach similar to Wang et al. [2009] to improve the scalability of our method. A long video is divided into short clips (20 frames in our experiments) and the optimization for each clip is solved sequentially. To achieve a smooth transition between two consecutive clips, the clips are overlapped with t_w frames. The meshes for the first t_w frames of each clip are strongly constrained to follow the results from the last t_w frames of the previous clip.

5 Non-uniform Ray Sampling

Given a target resolution $I_{width} \times I_{height}$, the rendering step first assigns a ray that is defined as a pair of longitude and latitude (θ, ϕ), to each pixel in the target image. Sampling from the projected original image at the position corresponding to the ray determines the color of the pixel. Let v^t_i of M^t be augmented with the 2D image coordinates (x^t_i, y^t_i). The initial arrangement of the vertices with a constant interval on the target image (i.e. regular grid) is equivalent to the equirectangular projections (Figure 6(a)). The ray for each pixel can be computed by linearly interpolating (θ_i, ϕ_i) of the surrounding vertices. The number of samples between the neighboring vertices depends on the length of their interval in the image coordinates. Therefore, our goal is to widen the image area occupied



$Nc = 0$ $Nc = 0.5$ $Nc = 1$

Figure 7: *The influence of Nc on the resulting ray mesh. $Nc = 0$ produces the equirectangular projections.*

by d^{th} quad face q^t_d of M^t if the area contains important information (Figure 6(d)). We formulate this as an optimization problem of (x^t_i, y^t_i) in the target image space.

Similar to previous content-aware methods [Wang et al. 2009; Krähenbühl et al. 2009], we estimate the importance of each pixel by combining a range of measures in an effort to reflect comprehensive information from low level features (e.g. image gradient) to high level features (e.g. human face). We use two importance maps S_x and S_y accounting for x and y intervals of the sampling. The two importance maps build on two normalized gradient maps $[0,1]$ obtained by performing the Sobel kernels for x and y directions on the initial equirectangular projection image, indicating the structural details. Next, the normalized saliency map $[0,1]$ considering the attractiveness of a region [Yildirim and Süsstrunk 2015] is multiplied to each importance map to cull out trivial and repeated structural textures. Finally, as a human face is usually one of the most important objects, we assign a high importance value to the face regions in S_x and S_y detected by existing methods [Viola and Jones 2001].

The automatic importance detectors may not be perfect in every circumstance. Therefore, Rich360 provides a user with an interactive tool with which the user can mark low or high importance to the desired regions, as shown in Figure 6(c). To reduce per-frame manual intervention, the user provided polygonal masks on sparse key frames are propagated across the sequence using linear interpolation. The user masks are applied to both of the importance maps. Finally, the importance sx^t_d and sy^t_d of a quad face q^t_d is defined as the average importance of interior pixels in S_x and S_y at each frame, respectively.

The target width qw^t_d and the height qh^t_d of q^t_d are obtained by sx^t_d and sy^t_d as follows.

$$\begin{cases} qw^t_d = \frac{(sx^t_d)^{Nc}}{\sum_{j\in row(d)} (sx^t_j)^{Nc}} \times I_{width} \\ qh^t_d = \frac{(sy^t_d)^{Nc}}{\sum_{j\in col(d)} (sy^t_j)^{Nc}} \times I_{height} \end{cases}, \tag{6}$$

where $row(d)$ and $col(d)$ denote a set of the indices of the quads that belong to the same row and column as q_d, respectively. Equation 6 computes qw^t_d and qh^t_d according to the ratio of sx^t_d and sy^t_d to the sum of the importances in the same row and column,

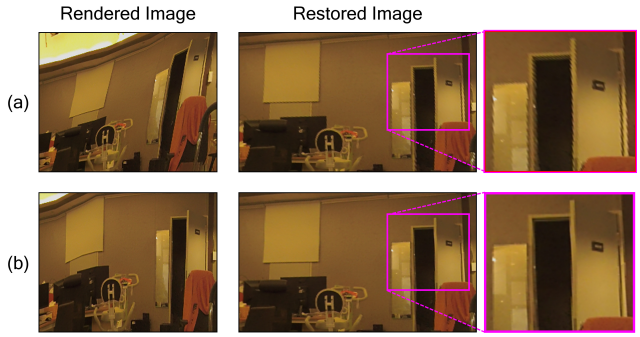


Figure 8: *Rich360 player restores the original shape of the non-uniformly rendered spherical image using the accompanying ray mesh. This process can lead to jagging artifacts on the strong horizontal and vertical lines that are skewed in the rendered image (a). Incorporating Equation 9 with line detection removes these artifacts (b).*

respectively. Nc is a user parameter that determines the degree of non-uniformness. As Nc becomes greater, accordingly stronger contrast of the resolution between the quads is obtained, as shown in Figure 7.

In the optimization, the width and the height of each quad are constrained to be qw_d^t and qh_d^t from Equation 6.

$$E_g = \sum_d \sum_{\{i,j\}\in hE(q_d^t)} \|(x_j^t - x_i^t) - qw_d^t\|^2 + \sum_d \sum_{\{i,j\}\in vE(q_d^t)} \|(y_j^t - y_i^t) - qh_d^t\|^2, \quad (7)$$

where $hE(q_d^t)$ and $vE(q_d^t)$ denote a set of directed edges (i.e. $i \rightarrow j$) of q_d^t along the horizontal and vertical directions, respectively.

To smoothen the variations of the resolution across the quads and avoid the face flipping problem, we add a similarity transformation term [Igarashi et al. 2005] that has been widely employed as a metric for local shape distortions.

$$E_{s2} = \sum_f \beta_f^t \|v_{f1} - (v_{f2} + u(v_{f3} - v_{f2}) + vR_{90}(v_{f3} - v_{f2}))\|^2, \quad (8)$$

where (v_{f1}, v_{f2}, v_{f3}) denotes the vertices comprising a triangle face in the image space and R_{90} is the 90 rotation matrix, $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. The local coordinates u and v are computed from the initial uniform mesh. β_f^t is the weight for the similarity transformation term of each face. The detailed derivation of this term can be found in a paper by Igarashi et al. [2005]. This term measures the deviation of a triangle under a similarity transformation. Consequently, each quad is encouraged to be shaped into a rectangle. To completely dispose of the face flipping problem, we take an iterative approach. Initially, we set $\beta_f^t = 1$ for all of the faces. After solving Equation 11, flipped faces are identified by examining the face orientation. The corresponding weights are then set to a very large number ($\beta_f^t = 1000$ in our case). Equation 11 is solved again with the updated weights to prevent the face flipping by strongly enforcing the shape of the corresponding faces to be a proper rectangle. This process is repeated until no face flipping is detected.

Unlike previous mesh parameterization methods that minimize the distortions of the resulting image [Kopf et al. 2009; Carroll et al. 2009], Rich360 intentionally distorts the resulting spherical image to allocate more sample pixels in important regions. The distorted spherical video is restored back during the playback in the viewing step by utilizing the resulting mesh. As a result, jagging artifacts can arise from the distortions across strong horizontal and vertical lines. Figure 8 shows an example of this problem. In the case of the vertical line, the skew distortion in the horizontal direction generates aliasing along the line when rendered. A similar explanation is true for the horizontal line. This aliasing becomes apparent when the spherical image is restored to the original shape.

To avoid this artifact, we minimize the skew distortion of the quad face containing a strong horizontal or vertical line. Line segments are first detected by the method proposed by Giol et al. [2008]. Diagonal lines are filtered out and the remaining segments are labeled as either a horizontal or vertical line; $hLine$ and $vLine$, respectively. All quads q_d^t containing the line segments minimize the following energy function in the optimization.

$$E_k = \sum_d \sum_{\{i,j\}\in hE(hLine\in q_d^t)} \|y_j^t - y_i^t\|^2 + \sum_d \sum_{\{i,j\}\in vE(vLine\in q_d^t)} \|x_j^t - x_i^t\|^2 \quad (9)$$

Equation 9 prevents the skew distortion of the mesh edge according to the line direction within the quad. Figure 8 illustrates the effect of this energy term.

To ensure efficient video compression and prevent rapid resolution changes, a temporal smoothness term is added to the optimization in a similar manner to that described in Section 4.

$$E_{t2} = \sum_i \|x_i^t - \frac{1}{2t_w} \sum_{j\in T(i,t)} x_j\|^2 + \|y_i^t - \frac{1}{2t_w} \sum_{j\in T(i,t)} y_j\|^2 \quad (10)$$

Similar to previous importance-based methods [Krähenbühl et al. 2009], we apply temporal box filtering to the per-frame importance map sx_d^t and sy_d^t with the temporal window $[t, t + t_w]$ to take future events such as moving objects into account. This simple filtering achieves a smoother mesh appearance over the sequence, as the importance of the regions largely affects the determination of the shape of a mesh at each frame.

Finally, we minimize the sum of the above energies subject to the boundary constraints preserving a rectangular frame:

$$\underset{\{x_i^t, y_i^t\}_{t,i}}{\arg\min} \sum_t E_g + \alpha_s E_{s2} + \alpha_k E_k + E_{t2}$$

subject to

$$x_i^t = \begin{cases} 0 & \text{if } v_i^t \text{ is on the left boundary} \\ I_{width} & \text{if } v_i^t \text{ is on the right boundary} \end{cases}$$
$$y_i^t = \begin{cases} 0 & \text{if } v_i^t \text{ is on the top boundary} \\ I_{height} & \text{if } v_i^t \text{ is on the bottom boundary} \end{cases} . \quad (11)$$

The weight α_s for the spatial smoothness is set to 0.5. α_k is the weight for minimizing the skew distortion described by Equation 9. We set $\alpha_k = 10$ for all of our experiments to avoid jagged lines. Solving Equation 11 for all the frames in the same manner as in Section 4 produces the temporally coherent ray meshes used for the rendering. The image coordinates (x_i, y_i) of the resulting meshes are normalized by I_{width} and I_{height} and then stored in a separate file. Along with the rendered non-uniform 360° panoramic video, the corresponding mesh file is fed into the player in the viewing step. The 360° panoramic video is undistorted and mapped onto

Figure 9: *Comparison of stitching results from Rich360, GCW, and STCPW.*

a uniform viewing sphere by modifying the UV coordinates of the sphere according to the given mesh at every frame.

The file size of the mesh sequence depends on the number of vertices. This paper does not address mesh compression as Rich360 is not specifically intended for video streaming. However, for real-time transmission of the meshes over the network, reduction of the bit rate is an important consideration. As the resulting meshes from Rich360 are temporally coherent, a dynamic mesh compression method using a motion prediction scheme [Collet et al. 2015] can achieve a streamable bit rate for the mesh size used in our experiments.

6 Results

Rich360 is implemented with C++. The *Intel* Math Kernel Library(MKL) is used for sparse solvers in the optimization process. All of the experiments were performed on a PC with an *Intel Core i7-5930K* 3.5Ghz CPU, 32 GB memory, and *NVidia GeForce GTX Titan X* graphics chipset. A wide variety of datasets were captured with two popular panoramic capture rigs (*Freedom360 Mount*, *360Heros Pro6L*) that have six *GoPro Hero 4 Black* cameras, each capturing 2.7K (2704 x 2028) video at 30fps. For all of our experiments, we used the same mesh size of 181 x 91 (16471 vertices), which was determined empirically.

Parallax Removal

Figure 9 shows a comparison of stitching results on various scenes from our method and two state-of-the-art methods: global coherent warping (GCW) proposed by Perazzi et al. [2015] and spatio-temporal content-preserving warping (STCPW) proposed by Jiang and Gu [2015]. Note that we reimplemented both GCW and STCPW because the original codes or the executable files are not publicly available. The first column shows the initial spherical pro-

Figure 10: *Comparison with the dataset provided by the authors of GCW. (Top) The resulting frames produced by GCW that were captured from the accompanying video of the paper. (Bottom) The resulting frames from Rich360*[1]*. Compared to GCW, Rich360 handles temporal-jittering in optical flows better as seen on the wall of the front building.*

[1]Rich360 is applicable to an irregular camera rig if the calibration data are available. We used *Agisoft PhotoScan* to estimate 3D camera parameters.

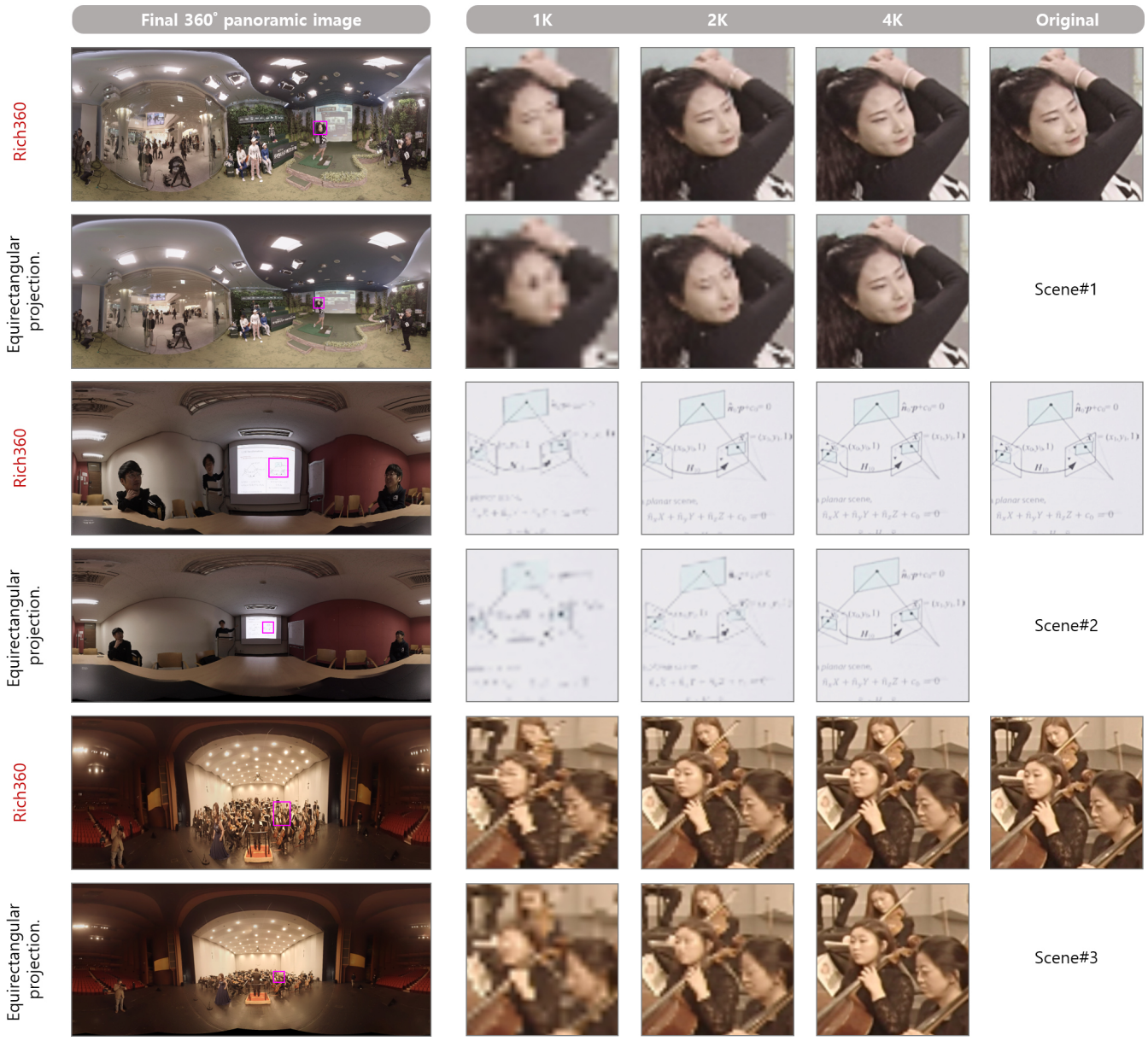


Figure 11: *Comparisons of rendering results from Rich360 and from the equirectangular projection. Rich360 preserves richness in important regions. The face features are recognizable in 1K resolution and the equation on the screen is legible in the 2K resolution result from Rich360, whereas the quality of the result from the equirectangular projection degrades rapidly with a decrease in the rendering resolution.*

jection image that was used for all three methods. The same optical flow and SIFT feature matching results that were utilized for Rich360 were employed for GCW and STCPW. Blending is omitted and parts of the entire stitching results are shown to clearly demonstrate the parallax artifacts. The complete results can be found in the accompanying video.

The first row shows a scene where a moving camera captures static objects. The outdoor scene contains the objects that are located far from the cameras where the parallax artifacts can be removed well by all of the three methods. The quality of the result from Rich360 surpasses those of the other results in the second scene where close objects captured by moving cameras are present. The magenta boxes in the second scene show the stitching result of the pillar and the wall over two consecutive frames. The result from GCW is highly dependent on the performance of the optical flow because the calculation is done per frame without explicit formulation of a temporal term. Therefore, as the authors of GCW pointed out in their paper, when the optical flow has temporal jittering, the stitching result suffers from the same artifact (the top row in Figure 10). STCPW added temporal terms for mesh warping. Because it relies on a sparse set of feature correspondences computed by SIFT, however, one or two erroneous matches can adversely influence the result. This happens often when there is a repeated pattern as can be seen at the patterned wall of the second scene. Rich360 handles the two cases better by utilizing sufficient matching results from the optical flow and feature detectors to reduce the effect from noise, and by guaranteeing temporal smoothness through mesh optimization. The third scene shows a challenging case where moving objects are included. Both of the results from GCW and STCPW

	Rich360	GCW	STCPW
Computation time	29.85	251.382	44.755

Table 1: *Average computation time per frame in seconds for the stitching step.*

show parallax at people's feet whereas results from Rich360 show great improvement. Unlike Rich360 and STCPW, which utilize a mesh, the per pixel warping method of GCW has an advantage when stitching a scene that contains adjacent objects with a large depth difference. In the last scene, to reduce the parallax of the objects in the back, the results from Rich360 and STCPW show parallax around the man's silhouette, whereas the result from GCW shows successful parallax removal.

Table 1 shows the average computation time for each method when six 2.7K images are stitched into a single 2K image. GCW achieved higher per pixel accuracy but required a long computation time. For 1K resolution optical flow, GCW required over 4 minutes per frame. STCPW required about 44 seconds with 30×15 mesh. Rich360 achieved the fastest per frame computation time because it solves only a single linear system, while showing stable results with comparable quality.

The comparisons of stitching results can be summarized as follows. GCW can align the images with per-pixel accuracy by utilizing an optical flow. On the other hand, Rich360 based on a mesh optimization scheme is less affected by erroneous flows at the cost of losing pixel level accuracy. Apart from this well known trade-off between the schemes based on optical flow or mesh optimization, our novel projection-based method has several advantages in 360° video stitching. 1) Rich360 produces temporally more stable results in general compared to those yielded by previous methods, as shown in Figure 9 and 10. 2) Results of a comparable quality can be obtained at a lower computational cost. 3) Unlike GCW and STCPW, Rich360 does not require additional constraints to align the displacements of the leftmost and rightmost columns of the resulting 360° image that are adjacent in the viewing sphere.

Non-uniform Spherical Ray Sampling

Figure 11 demonstrates the effectiveness of our non-uniform spherical ray sampling method. The left column shows a comparison between the results from Rich360 ($Nc = 0.5$) and the results from the equirectangular projection. The importance maps for scene 1 and scene 3 were generated automatically. For the second scene, the user masks shown in Figure 6(c) were utilized. Note that the results from Rich360 are distorted according to the importance. For example, in the first scene, the person inside the magenta square is enlarged whereas the homogeneous ceiling region is reduced. Similarly, the presentation screen in the second scene is enlarged. The small images in the right column show close-up views of the magenta square regions. The images were captured in the viewing step after rendering the 360° panoramic video in 1K, 2K, and 4K resolution. The resolution of the viewing screen was Full HD. The right-most images were captured after re-projecting the corresponding regions in the original videos. With 1K equirectangular projection, the expressions on the faces cannot be identified in the first scene and the third scene. However, the features of the face are recognizable in the results from Rich360. In the second scene, the readability of the content of the screen is greatly increased with a simple user input. Even in 2K and 4K results, the quality improvement is prominent in all scenes. The quality of the 4K results from Rich360 is almost the same as that of the original videos.

For a more quantitative analysis, we calculated the root mean square error (RMSE) with the original videos. Figure12 shows the results. After rendering in all 6 directions (front, right, left, back, top, bottom) with a virtual camera that has a 90 viewing angle, the average RMSE with the original videos was calculated. For a precise comparison, we discarded the overlapping regions and used a simple sphere in the projection step. Rich360 shows superior results in all of the comparisons. The difference becomes greater with lower resolutions. To watch the captured video from the actual interactive viewing step, please refer to the accompanying video.

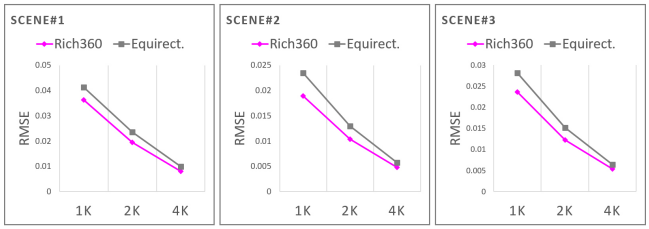


Figure 12: *Quantitative RMSE comparison for rendering results from Rich360 and the equirectangular projection. The scenes used for the comparison are shown in Figure 11. Rich360 produces superior results to those from the equirectangular projection. This advantage is prominent as the rendering resolution decreases.*

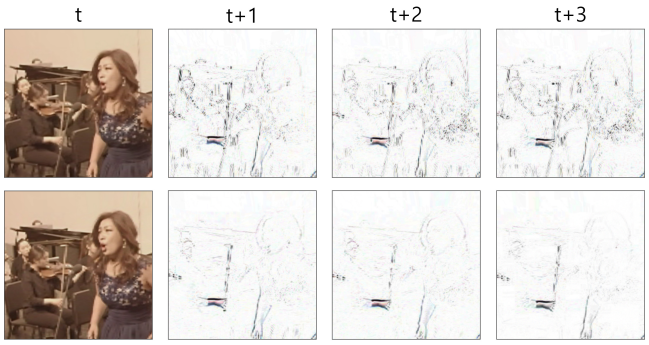


Figure 13: *Comparison between two results over four consecutive frames obtained with (bottom) and without (top) the temporal smoothness term, E_{t2}. The three right columns represent difference images between the current frame and the previous frame (images are inverted and enhanced for visibility).*

Figure 13 illustrates the effect of E_{t2} (Equation 10) in non-uniform sampling. In the viewing step, the resulting video from the optimization without use of E_{t2} (top row) exhibits flickering artifacts in the vicinity of object edges due to rapid resolution changes. In contrast, incorporating E_{t2} (bottom row) effectively removes these artifacts. The video compression of the rendered non-uniform 360° sequence can also benefit from the temporal smoothness in terms of storage efficiency. For example, the video file size of the third scene in Figure 11 is reduced from approximately 53MB to 42MB when 300 frames of 2K resolution images are compressed using a Xvid codec.

7 Limitations and Future work

Rich360 transforms a video stitching task, which requires heavy computation, into a simple and efficient mesh deformation problem based on precisely calibrated data. Therefore, the accuracy of the camera calibration affects the final video stitching results. The magenta circle in Figure 14(a) shows an example of a stitching based on an inaccurate camera calibration. A combination of existing methods with Rich360 to overcome this weakness while maintaining the efficiency would be an interesting future research

(a) (b)

Figure 14: *Limitations of Rich360. (a) the stitching result with erroneous calibration data. Depth variation of the projection surface cannot compensate for inaccurate calibration data. (b) tearing artifacts caused by overly sparse sampling.*

topic. For example, after applying our method, inaccurately calibrated regions can be detected and existing video warping can be applied for improvement. Solving the camera calibration refinement and the deformation of the projection surface simultaneously is also one of our future research directions.

Our experiments show that pixels can be sampled efficiently according to the importance of the region with non-uniform spherical ray sampling. However, a few limitations exist. Our method takes the importance value into account, but does not consider the available source resolution in calculating the target size of a quad face. Therefore, when the difference between the target rendering resolution and the overall resolution of the input videos is small, an oversampling problem may occur. As a simple remedy, a user can lower the user parameter Nc. Noticeable tearing artifacts can be formed at a less important region if the target size of the region becomes too small (the magenta circle in Figure 14(b)). An automatic decision of the degree of non-uniformness and the minimum region size considering the source resolution and visual perception can be an interesting future work.

Rich360 can be adopted for various applications through follow-up research. For a stereoscopic 360° video, our deformable spherical projection surface can be extended by considering the disparities to ensure a plausible and comfortable depth perception. Real-time non-uniform sampling according to the viewing direction of the viewer would provide a richer viewing experience. Although our non-uniform ray sampling method is based on a projection sphere, projection using a cube map is also viable (e.g. *Facebook's Transform*). Compared with the equirectangular projection, the cube map contains pixels more efficiently without stretching the areas near poles. This leads to a decrease of the bit rate and the storage, respectively. However, the available method based on the cube map still samples viewing angles uniformly regardless of the importance of the region. Combining our non-uniform sampling method with a cube map would be an interesting future research topic.

8 Conclusion

In this paper, we presented Rich360, a novel system for creating and viewing an as-rich-as-possible 360° panoramic video from structured camera arrays. We introduced two novel approaches to resolve the two issues that arise in the existing pipeline. First, a novel stitching method is introduced. Instead of performing pairwise stitching of input videos, a deformable spherical projection surface is employed to project the input videos with minimal parallax artifacts. The projection surface is deformed according to the 3D points recovered from the overlapping regions of the input videos. This approach abstracts the stitching problem into a single energy minimization function regardless of the number of input videos while effectively minimizing the disparities in the overlapping regions. Next, a non-uniform spherical ray sampling method is introduced. A dense sampling is performed in the important regions while a sparse sampling is performed in the less important regions. The richness of the input videos is preserved with this method even if the resolution of the final panoramic image is smaller than the overall resolution of the input videos. Although the sampling is non-uniform, the final viewing process restores the original content of the video with little overhead. The stitching and rendering results were compared with those from existing methods. Rich360 shows higher temporal stability and robustness for the scenes that have high depth variation and moving objects. Also, the computation time is faster than that of existing methods while the quality of the results is comparable. The rendering quality shows superiority over equirectangular projection and the richness of the input videos is preserved even when the rendering resolution is small.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments and Federico Perazzi and Dr. Alexander Sorkine-Hornung for kindly providing the data. We are also grateful to Seunghoon Cha for helpful implementation tips, Kyungwon Gil of VIRNECT for lending a rig, and Sangwoo Lee, Byungkuk Choi, Roger Blanco i Ribera, and Minju Kim for discussions and help. This work was supported by the ICT R&D program of MSIP/IITP (R0101-15-284, Multicamera-based Autostereosopic 3D Acquisition System and Content Production R&D).

References

BIRKLBAUER, C., AND BIMBER, O. 2014. Panorama light-field imaging. *Computer Graphics Forum 33*, 2, 43–52.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*. 25–36.

CARROLL, R., AGRAWALA, M., AND AGARWALA, A. 2009. Optimizing content-preserving projections for wide-angle images. *ACM Transactions on Graphics (TOG) 28*, 3, 43.

CARROLL, R., AGARWALA, A., AND AGRAWALA, M. 2010. Image warps for artistic perspective manipulation. *ACM Transactions on Graphics (TOG) 29*, 4, 127.

CHANG, C.-H., HU, M.-C., CHENG, W.-H., AND CHUANG, Y.-Y. 2013. Rectangling stereographic projection for wide-angle image visualization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2824–2831.

CHANG, C.-H., SATO, Y., AND CHUANG, Y.-Y. 2014. Shape-preserving half-projective warps for image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3254–3261.

CHAURASIA, G., DUCHENE, S., SORKINE-HORNUNG, O., AND DRETTAKIS, G. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG) 32*, 3, 30.

COLLET, A., CHUANG, M., SWEENEY, P., GILLETT, D., EVSEEV, D., CALABRESE, D., HOPPE, H., KIRK, A., AND SULLIVAN, S. 2015. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG) 34*, 4 (July), 69:1–69:13.

HARTLEY, R., AND ZISSERMAN, A. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

HE, K., CHANG, H., AND SUN, J. 2013. Rectangling panoramic images via warping. *ACM Transactions on Graphics (TOG) 32*, 4, 79.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG) 24*, 3, 1134–1141.

JIANG, W., AND GU, J. 2015. Video stitching with spatial-temporal content-preserving warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 42–48.

KOPF, J., LISCHINSKI, D., DEUSSEN, O., COHEN-OR, D., AND COHEN, M. 2009. Locally adapted projections to reduce panorama distortions. *Computer Graphics Forum 28*, 4, 1083–1089.

KRÄHENBÜHL, P., LANG, M., HORNUNG, A., AND GROSS, M. 2009. A system for retargeting of streaming video. *ACM Transactions on Graphics (TOG) 28*, 5, 126.

LANG, M., HORNUNG, A., WANG, O., POULAKOS, S., SMOLIC, A., AND GROSS, M. 2010. Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG) 29*, 4, 75.

LI, B., HENG, L., KOSER, K., AND POLLEFEYS, M. 2013. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1301–1307.

LI, S., YUAN, L., SUN, J., AND QUAN, L. 2015. Dual-feature warping-based motion model estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4283–4291.

LIN, W.-Y., LIU, S., MATSUSHITA, Y., NG, T.-T., AND CHEONG, L.-F. 2011. Smoothly varying affine stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 345–352.

LIN, C.-C., PANKANTI, S. U., RAMAMURTHY, K. N., AND ARAVKIN, A. Y. 2015. Adaptive as-natural-as-possible image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1155–1163.

LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics (TOG) 28*, 3, 44.

LIU, S., YUAN, L., TAN, P., AND SUN, J. 2013. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG) 32*, 4, 78.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2, 91–110.

MEI, C., AND RIVES, P. 2007. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, 3945–3950.

PANOZZO, D., WEBER, O., AND SORKINE, O. 2012. Robust image retargeting via axis-aligned deformation. *Computer Graphics Forum 31*, 2pt1, 229–236.

PERAZZI, F., SORKINE-HORNUNG, A., ZIMMER, H., KAUFMANN, P., WANG, O., WATSON, S., AND GROSS, M. 2015. Panoramic video from unstructured camera arrays. *Computer Graphics Forum 34*, 2, 57–68.

RICHARDT, C., PRITCH, Y., ZIMMER, H., AND SORKINE-HORNUNG, A. 2013. Megastereo: Constructing high-resolution stereo panoramas. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1256–1263.

SHUM, H.-Y., CHAN, S.-C., AND KANG, S. B. 2008. *Image-based rendering*. Springer Science & Business Media.

SZELISKI, R. 2006. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision 2*, 1, 1–104.

UYTTENDAELE, M., CRIMINISI, A., KANG, S. B., WINDER, S., SZELISKI, R., AND HARTLEY, R. 2004. Image-based interactive exploration of real-world environments. *IEEE Computer Graphics and Applications 24*, 3, 52–63.

VIOLA, P., AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, I–511.

VOLLMER, J., MENCL, R., AND MUELLER, H. 1999. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum 18*, 3, 131–138.

VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., AND RANDALL, G. 2008. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 4, 722–732.

WANG, Y.-S., FU, H., SORKINE, O., LEE, T.-Y., AND SEIDEL, H.-P. 2009. Motion-aware temporal coherence for video resizing. *ACM Transactions on Graphics (TOG) 28*, 5, 127.

WANG, Y.-S., LIN, H.-C., SORKINE, O., AND LEE, T.-Y. 2010. Motion-based video retargeting with optimized crop-and-warp. *ACM Transactions on Graphics (TOG) 29*, 4, 90.

WANG, Y.-S., LIU, F., HSU, P.-S., AND LEE, T.-Y. 2013. Spatially and temporally optimized video stabilization. *IEEE Transactions on Visualization and Computer Graphics 19*, 8, 1354–1361.

XU, W., AND MULLIGAN, J. 2013. Panoramic video stitching from commodity hdtv cameras. *Multimedia systems 19*, 5, 407–426.

YILDIRIM, G., AND SÜSSTRUNK, S. 2015. Fasa: fast, accurate, and size-aware salient object detection. In *Computer Vision–ACCV 2014*. 514–528.

ZARAGOZA, J., CHIN, T.-J., TRAN, Q.-H., BROWN, M. S., AND SUTER, D. 2014. As-projective-as-possible image stitching with moving dlt. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 7, 1285–1298.

ZELNIK-MANOR, L., PETERS, G., AND PERONA, P. 2005. Squaring the circle in panoramas. In *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 1292–1299.

ZHANG, F., AND LIU, F. 2014. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3262–3269.

ZHI, Q., AND COOPERSTOCK, J. R. 2012. Toward dynamic image mosaic generation with robustness to parallax. *IEEE Transactions on Image Processing 21*, 1, 366–378.