EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Panoramic Video from Unstructured Camera Arrays

F. Perazzi[1,2], A. Sorkine-Hornung[2], H. Zimmer[2], P. Kaufmann[2], O. Wang[2], S. Watson[3], M. Gross[1,2]

[1]ETH Zurich [2]Disney Research Zurich [3]Walt Disney Imagineering

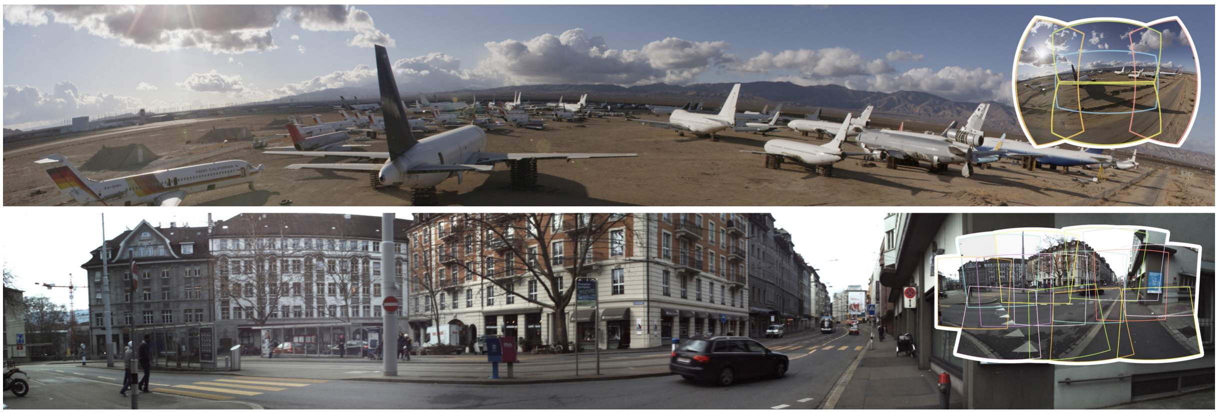

Figure 1: Two panoramas created with our system. Top - (00:05 in the accompanying video): a cropped frame from a 160 megapixel panoramic video generated from five input videos. The overlay on the right shows the full panorama, with the respective individual field of views of the input cameras highlighted by colored frames. Bottom - (01:22): a crop from a 20 megapixel panorama created from a highly unstructured array consisting of 14 cameras.

Abstract

We describe an algorithm for generating panoramic video from unstructured camera arrays. Artifact-free panorama stitching is impeded by parallax between input views. Common strategies such as multi-level blending or minimum energy seams produce seamless results on quasi-static input. However, on video input these approaches introduce noticeable visual artifacts due to lack of global temporal and spatial coherence. In this paper we extend the basic concept of local warping for parallax removal. Firstly, we introduce an error measure with increased sensitivity to stitching artifacts in regions with pronounced structure. Using this measure, our method efficiently finds an optimal ordering of pair-wise warps for robust stitching with minimal parallax artifacts. Weighted extrapolation of warps in non-overlap regions ensures temporal stability, while at the same time avoiding visual discontinuities around transitions between views. Remaining global deformation introduced by the warps is spread over the entire panorama domain using constrained relaxation, while staying as close as possible to the original input views. In combination, these contributions form the first system for spatiotemporally stable panoramic video stitching from unstructured camera array input.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms

1. Introduction

The richness and detail of our surrounding visual world is challenging to capture in a regular photograph or video. The idea of combining content from multiple cameras into a wide field of view panorama therefore is essentially as old as photography and film themselves. Popular contemporary implementations are, for instance, dome-based projections as in planetaria or IMAX cinemas. But while tools for creating panoramic *still* images are available in most consumer cameras and software nowadays, capturing panoramic *video* of comparable quality remains a difficult challenge.

One source of the problem is the fundamental physical limitations of the acquisition hardware. Currently, professional video sensors capture at horizontal resolutions around 4k to 5k, which are insufficient for large scale, wide-angle capture. Moreover, wide-angle optics unavoidably introduce image distortions and imperfections such as blur, resulting in an additional loss of image resolution. Hence, in order to create panoramic video beyond the resolution of a single device, one has to combine or "stitch" multiple views from an array of cameras. Stitching is impaired by parallax between the input views, which can lead to disturbing artifacts such as blur, ghosting, or discontinuities. Array miniaturization or mirrors for minimizing parallax is often not an option, depending on the type of cameras and lenses or the desired configuration of the array. For example, the top panorama in Fig. 1 shows a crop from a 160 megapixel panoramic video captured with five RED Epic-X cameras, with an interaxial distance about 30 cm due to the dimensions of the camera bodies and lenses. The panoramic video on the bottom has been generated from a rig with 14 cameras, as shown in Figure 2. Both datasets feature considerable parallax between the input videos.

Various methods for stitching images have been developed in the recent years, including multi-level blending [BA83], content-adaptive seams [EF01] and efficient interactive tools [STP12]. For video containing significant scene motion, however, these strategies designed mostly for still images are not optimal for two reasons. Firstly, processing individual frames independently, without any notion of spatiotemporal coherence, results in noticeable deformations. Secondly, these previous methods are not straightforward to extend with standard strategies for enforcing temporal coherence. Processing multiple frames or even all input videos at once is computationally infeasible due to the sheer amount of data, in particular for high resolution input and large arrays as in Fig. 1. We build on the concept of local image warping [SS00] and extend it to unstructured arrays of video cameras.

Contributions. We present an algorithm based on three key observations. Firstly, for the analysis of parallax errors we found existing image comparison techniques to be not sufficiently robust or targeted towards the specific stitching artifacts we observed. Hence, we introduce a patch-based error metric defined on image gradients, which we designed to be especially sensitive to parallax errors in highly structured image regions, and which ensures visual similarity of content between the input videos and the output panorama.

Secondly, the ability to compensate parallax errors between views depends on the spatial configuration of the individual field of views and the scene content, and the order that the parallax is compensated between images. We therefore propose a method that first analyzes these properties, and then computes an optimized ordering of pair-wise image warps, which results in improved quality of the parallax removal. Our procedure remains efficient even for large numbers of input cameras, where brute-force approaches for finding an optimal warp order would be infeasible.

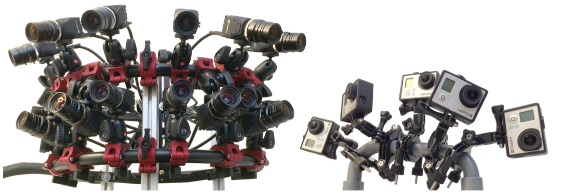

Figure 2: Two of the camera arrays we constructed for capturing the panoramic videos shown in this paper. Left: 14 machine vision cameras. Right: A small rig built using five GoPro cameras. Note that our panoramas have been captured without particularly accurate placement of cameras in order to demonstrate the flexibility of our approach.

Finally, local image warping accumulates globally, leading to significant spatial deformations of the panorama. Since these deformations are dependent on the per-frame scene content, they change for every output frame and hence result in noticeable temporal jitter. We resolve these global deformations and temporal instability by a weighted warp extrapolation from overlap to non-overlap image regions, and a final constrained relaxation step of each full panoramic output frame to a reference projection.

We demonstrate panoramic video results captured with different types of cameras on arrays with up to 14 cameras, allowing the generation of panoramic video in the order of tens to over a hundred megapixels.

2. Related Work

We review the most relevant works here, and refer to Szeliski [Sze06] for an extensive survey.

Parallax-free input. One class of methods focuses on creating a single panoramic image from a set of overlapping images, under the assumption that all images are captured from the same or similar center of projection and hence are basically parallax-free. Such a configuration can be achieved by carefully rotating a camera, e.g. using a tripod [KUDC07]. After estimating the relative camera poses, a panorama can be generated by projecting all images onto a common surface. Smaller errors caused, e.g., by imperfect alignment of projection centers, objects moving during image acquisition, or lens distortion can be removed using image blending [BA83,LZPW04] content-adaptive seam computation [EF01], or efficient, user-assisted combinations thereof [STP12]. Depending on the field of view, a suitable projection has to be chosen in order to minimize undesirable distortion, e.g., of straight lines [KLD*09, HCS13]. While these methods enable the creation of static panoramas up to gigapixel resolution [KUDC07], they are not designed to produce panoramic video and cannot correct for larger parallax errors when camera projection centers do not align.

Handling parallax. When capturing hand-held panoramas or using an array of cameras, parallax has to be accounted for. A common strategy, which is also the basis for our work, is to warp the images in 2D space [SS00, KSU04, JT08] using image correspondences computed as depth maps or optical flow fields in the overlapping image regions. In a general multi-view scenario with more than two input views, this requires combining all correspondence fields [EdDM*08]. We show in Section 3.2 that existing solutions such as averaging of correspondence fields [SS00] are sensitive to wrong correspondence estimates, which occur frequently in real-world images. Stereo-based approaches are inherently sensitive to rolling-shutter artifacts or unsynchronized images. Lin et al. [LLM*11] describe a method based on smoothly varying affine stitching which is, however, computationally expensive and not straightforward to extend to high resolution input or video. Zaragoza et al. [ZCT*14] extend this approach and combine a series of local homographies to reduce parallax while preserving the projectivity of the transformation. For specific scenes and camera motion such as long panoramas of buildings along a street [AAC*06,KCSC10], parallax errors can be reduced by using approximate geometric proxies and seam-based stitching. Stereo panorama techniques [PRRAZ00] intentionally capture parallax to enable stereoscopic viewing, but require a dense sampling and significant overlap of views. Recently, multi-perspective scene collages [NZN07, ZMP07] showed interesting artistic results by aligning and pasting images on top of each other. Both works propose strategies to find optimal orderings for combining images, but their respective solutions are computationally too expensive for processing video and not designed towards seamless parallax removal of dynamic content. Inspired by recent advances in video stabilization [LYTS13], a recent state-of-the-art method utilizes seam-based homography optimization for parallax tolerant stitching [ZL14], which we compare to in our results.

Dynamic panoramas. Several methods have been developed that augment static panoramas with dynamic video elements, e.g., by segmenting dynamic scene parts captured at different times and overlaying them on a static panorama [IAB*96]. Dynamic video textures [AZP*05] are infinitely looping video panoramas that show periodic motions (e.g. fire). Dynamosaics [RPLP05] are made by scanning a dynamic scene with a moving video camera, creating a panorama where all events play simultaneously. Similarly, Pirk et al. [PCD*12] enhance a static gigapixel panorama by locally embedding video clips. All above methods work well for localized, periodic motion, but have not been designed to handle significant motion resulting from a camera array moving through a scene.

Panoramic video. Our aim is to generate fully dynamic panoramic video. One possibility would be to perform 3D reconstruction, e.g., using structure-from-motion and multi-view stereo techniques over the whole input sequence, and then project the reconstructed 3D models to a single virtual camera. However, robustly and efficiently creating photo-realistic, temporally stable geometry reconstructions for entire videos remains a difficult challenge. In particular for non-static scenes with independently moving objects, the typically limited amount of overlap between cameras hinders robust multi-view stereo of the complete panorama. One of the few works that explicitly addresses panoramic video is [ZC12]. Similar to [KSU04] they compute depth only in the overlapping regions and then smoothly extrapolate deformation to non-overlapping regions [JT08]. This method works well for simple motions without strong occlusions, but relies heavily on accurate segmentation of moving objects. Practical issues such as rolling shutter or unsynchronized cameras pose additional challenges for stereo-based methods, which our method can handle to a certain extent due to a general motion field estimation.

Commercial solutions like the Point Grey Ladybug (*www.ptgrey.com*), the FlyCam (*www.fxpal.com*), or the Panacast camera (*www.altiasystems.com*) are based on pre-calibrated, miniaturized camera arrays in order to minimize parallax. The resolution, optical quality, and flexibility of such systems is limited in comparison to larger, high quality cameras and lenses. Similar issues arise for systems like GoPano (*www.gopano.com*), FullView (*www.fullview.com*), or the OmniCam [SFW*13], which rely on catadioptric mirrors [Nay97] to eliminate parallax.

Assessing panorama quality. In order to analyze alignment quality, a standard choice is to measure patch similarity using, e.g., the sum of squared differences (SSD) of pixels in overlapping image regions. This is, however, overly sensitive to intensity differences in less structured image regions, which can be easily fixed after alignment using multi-level blending. More robust measures were proposed for hole filling in videos [WSI07] in order to find similar patches that minimize the SSD of color and gradients. One problem of comparing pixel values in a patch is that it does not consider structural image properties. To alleviate this issue, Kopf et al. [KKDK12] restrict the search space for patches to texture-wise similar patches only, which also improves efficiency. In the context of hole filling, Simakov et al. [SCSI08] achieve semantically meaningful results by enforcing that the filled region is close to known content. Our analysis of parallax error builds on these ideas.

3. Our Algorithm

A common first step for combining views from multiple cameras is to estimate their mutual poses, i.e., to find a basic static image alignment on a user-defined projection surface (such as a hemisphere). In the following we refer to this step as the *reference projection*, and we assume the configuration of cameras to be static over the input sequence.

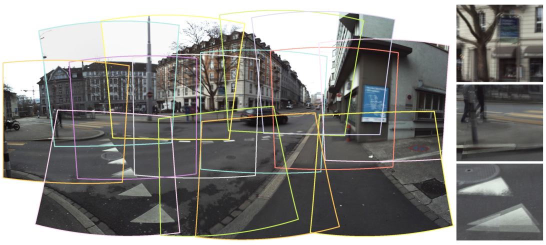

Figure 3: Example of a reference projection generated from 14 unstructured input views. The colored frames represent the fields of view of the individual input cameras. The remaining locally varying parallax manifests as ghosting when averaging the images, as shown in the close-ups. We encourage the reader to zoom in, e.g., on the street and sidewalk, and to see the supplemental material and video (01:05).

Our method only assumes a simple feature-based alignment using pairwise homographies directly on the input footage, which makes it more flexible and straightforward to initialize than techniques requiring special calibration objects like checkerboards. We provide some helpful practical hints on required preprocessing like lens undistortion, exposure compensation, and how to generate such a reference projection in Section 4.2. For the remainder of this section we assume the reference projection to be given. See Fig. 3 for an example.

Our method builds on the concepts for parallax compensation introduced by Shum and Szeliski [SS00]. The basic idea is to estimate a dense correspondence field between overlapping image regions, based on which the regions then can be warped to achieve per-pixel alignment. More formally, let I_i and I_j be two partially overlapping input images that are projected to the subsets Ω_i and Ω_j of the output projection surface, respectively. A pairwise alignment of these two images can be achieved by computing dense motion vector field u_{ij} in the overlap region $\Omega_{ij} = \Omega_i \cap \Omega_j$ using an optical flow method like [BBPW04]. This motion field defines a mapping from pixel x_i in image I_i to its corresponding pixel x_j in I_j via $x_j = x_i + u_{ij}(x_i)$. Thus we can align the image I_j to I_i using a backward warping operation and obtain the aligned image $I_{j \to i}(x_i) = I_j(x_j)$.

When generalizing the pairwise case to N input views, each pixel overlapping with $M < N$ other images has M motion vectors assigned to it. The task is now to combine all the resulting motion fields or warps into a single consistent output image. Previously proposed strategies such as exhaustively computing *all* pairwise motion fields and combining them by averaging [SS00] are, unfortunately, impractical for two reasons. Firstly, motion vectors estimated on real-world data generally contain a significant amount of errors, leading to noticeable artifacts such as ghosting and image deformations as shown in Figure 4. Secondly, for a video with K frames such a procedure has a complexity $\mathcal{O}(N^2K)$. Since motion field estimation is the computational bottleneck, any method based on exhaustive computation quickly becomes infeasible in practice: in our first experiments on input such as the one in Figure 1 we observed computation times in the order of days.

The challenge therefore is to understand which images we should actually warp and combine in order to efficiently create a high quality panorama with minimized parallax errors.

3.1. Parallax Error Analysis

For assessing parallax error, we design an error function $\Phi(I_{ij})$ that reflects the quality of the image I_{ij} resulting from warping I_j onto I_i using the motion field u_{ij}, and combining both images. A first straightforward idea to define Φ would be the residual warping error of the motion field, i.e. the sum of color differences between corresponding pixels in I_j and I_i. This, however, only considers the *per-pixel* matching between the two images and does not capture *structural* deviations of the warped result I_{ij} from the (unwarped) input I_i and I_j, which are the most noticeable and objectionable artifacts (see Figure 5). Assessing the similarity between I_{ij} and the input I_i and I_j by directly using common measures like sum-of-squared-differences is also not feasible, as this does not consider the parallax between the input images and thus will detect errors even when the final warp quality is good.

Our error measure to detect structural deviations is inspired by patch-based methods [WSI07, KKDK12]. We compare warped output patches to their corresponding patches in the input images, considering the parallax between the views, without directly depending on the (potentially wrong) per-pixel motion field. To better reflect structural deviations of warped patches from corresponding patches in the unwarped reference projection, we define our patches on *gradient* images to ignore low frequency differences (e.g. illumination changes) which can be fixed later during blending (Section 3.4). Here we describe the pairwise case, which, however, can be easily generalized to an arbitrary number of images.

In the following, let G be the gradient image of I computed using the Sobel operator. The computation of our error measure proceeds by visiting each pixel in G_{ij}, selecting a patch $p_{ij} \subset G_{ij}$ around the pixel, and determining a patch $p_i \subset G_i$ or $p_j \subset G_j$ in the gradient images of the reference projections that *best explains* p_{ij}. For assessing how well the patch p_j explains p_{ij}, we need to compensate for the warp between G_j and G_{ij}, without directly relying on the potentially wrong per-pixel motion field. Exhaustive patch search is not an option since the search space of possible transformations would be too large and could be compromised by false positive matches. Instead, we found that by computing a low-order approximation of the local motion field u_{ij} within p_{ij}, we get a reliable estimate of the mapping between p_{ij} and p_j, which is not deteriorated by outliers in the motion field. We implement this idea by fitting a homography H to the motion field u_{ij} [HZ06], which essentially moves

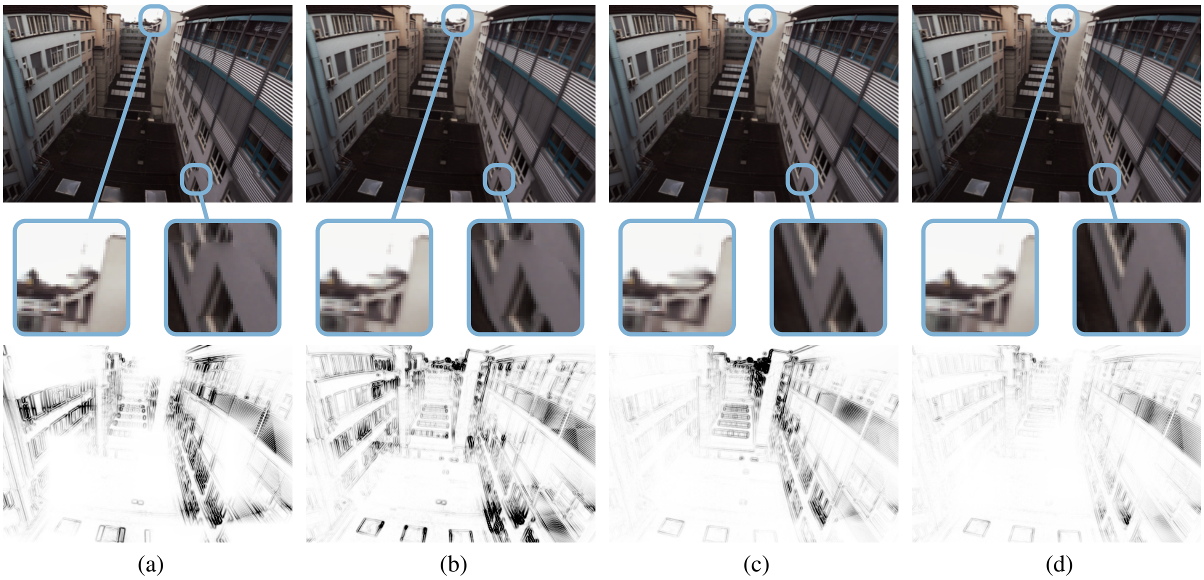

Figure 4: Impact of different warp ordering strategies on data from our array with 14 cameras - further demonstrated in the accompanying video at (00:31). The bottom row shows the output of our error measure. Darker pixels correspond to higher structural error. The respective close-ups focus on two different types of artifacts, deformation (left) and discontinuities (right), caused by wrong correspondence fields. (a) Static alignment with multi-band blending works for well aligned regions, but creates discontinuities in all areas with parallax. (b) Averaging of all correspondence fields (e.g., as in [SS00]) is computationally expensive and is sensitive to incorrect optical flow estimates, which can lead to deformations as well as incomplete alignment. (c) Pairwise warping is computationally efficient, but without the optimal ordering accumulates errors such as deformations, which cannot be corrected. (d) An optimal pairwise ordering computed using our patch-based metric reduces structural errors.

the patch to its corresponding location in G_j. Using this homography we compute the distance d_j between p_{ij} and its corresponding patch $p_j = H \circ p_{ij}$ in G_j as

$$d_j = \left\| G_{ij}[p_{ij}] - G_j[H \circ p_{ij}] \right\|^2 , \qquad (1)$$

where $G[p]$ is the vector obtained by stacking all pixels of G in a patch p. For computing the distance d_i to a patch p_i in the unwarped image G_i, the homography H is the identity. In the following we denote the index of this respective best patch with a '$*$', i.e., the best patch p_* has the minimum patch distance $d_* = \min\{d_i, d_j\}$.

We found that we can drastically increase robustness by considering *all patches* containing a pixel for computing its respective error instead of using the patch distances directly. In this way we measure the error that a particular pixel introduces into the panorama and ignore the error that other pixels in its vicinity create. Hence we accumulate the error in each pixel as the sum of errors from all patches containing this pixel. Let the per-pixel difference in patch p_* be

$$\delta_{p_*}(x) = |G_{ij}(x) - G_*(Hx)|^2 , \qquad (2)$$

with H as defined above. The combined error with contributions from all patches containing that pixel is computed as

$$\phi(x) = \sum_{p_* \ni x} \delta_{p_*}(x) , \qquad (3)$$

where $p_* \ni x$ denotes the set of all patches p_* that contain x. The final error measure Φ of the combined image I_{ij} is given as the average error over all pixels in the gradient image

$$\Phi(I_{ij}) = \sum_{x \in G_{ij}} \phi(x) . \qquad (4)$$

Note that in practice we compute and store the per-pixel error from Eq. (2) for all pixels inside a patch rather than just for center pixel x, so that Eq. (3) can be efficiently evaluated. We generally computed the error measure using a patch size of 25^2 pixels. See Figure 5 for error visualizations.

3.2. Warp Order Estimation

We can now use the error measure Φ to compute which input images to combine in what order to achieve a high quality output panorama. As illustrated in Figure 4, a sub-optimally chosen combination of images can lead to unrecoverable errors. In order to keep the computational complexity low and to quickly converge towards a complete, low error panorama, we propose a strategy based on finding an optimal sequence of pairwise warps, illustrated in Figure 6. An optimal pairwise ordering is computed once on one or more representative frames, and the same ordering is then used for stitching all frames of the input video. Note that only the ordering is

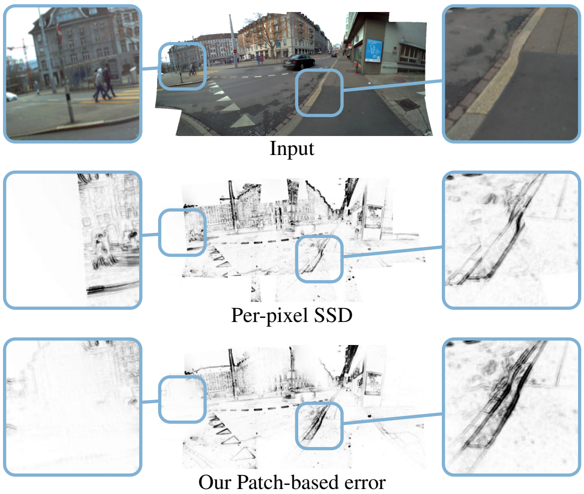

Figure 5: Comparison of a per-pixel SSD error to our patch-based approach. Note how the per-pixel SSD identifies large errors in regions with no visible warping (left), while under-detecting areas with significant distortion (right).

re-used. Motion fields are computed for each output frame individually to ensure optimal parallax removal.

The basic idea of our algorithm is to iteratively find *sets* of pairs of input views, which can be independently warped with minimum parallax error. Each pair results in a spatially separate panoramic fragment. We iterate this process until only a single fragment remains. See Figure 6 for illustration.

More formally, the problem can be formulated as an iterative *maximum weighted graph matching* [Gal83]. To this end we represent the images of our panorama as nodes V of a graph $G=(V,E)$, where undirected edges $e_{ij}\in E$ connect overlapping images I_i and I_j. Each edge has an associated positive weight w_{ij} representing the quality of merging images I_i and I_j. Beside the the quality of the minimum error alignment, we also take the mutual overlap (given by the size of $|\Omega_{ij}|$) into account:

$$w_{ij}=1-\gamma\cdot\frac{\min\left\{\Phi(I_{ij}),\Phi(I_{ji})\right\}}{|\Omega_{ij}|^{\alpha}} \tag{5}$$

where I_{ij} is the fragment resulting from combining I_i and $I_{j\to i}$, and γ is a normalization factor such that edge weights are scaled to $[0,1]$. We set $\alpha=0.5$ in order to achieve a sub-linear influence of the overlap area, since for large overlaps a good matching quality should not be overruled by lower quality matches that simply share a larger overlap. In order to avoid matching images with insufficient overlap, we remove edges if the corresponding images have less than 10% overlap. On the resulting graph, our algorithm computes the maximum weighted matching [Gal83], which, in the first iteration, results in pairs of input views that can be warped together independently with low error (Iteration 1 in Figure 6). Given a computed matching, i.e., one or more pairs of nodes $(v_i,v_j)\in V^2$, we remove for each pair the respective original nodes in G and create a new node v_{ij} corresponding to the fragment I_{ij} that has been created from the warp between images I_i and I_j which had the lower error; see Eq. (5). The weights of the updated graph are then recomputed based on the remaining, unprocessed images and the newly generated fragments, and the procedure is iterated until only one node remains. The ordering of pairwise warps resulting from this procedure ensures that only images with a minimal mutual parallax error are warped and combined.

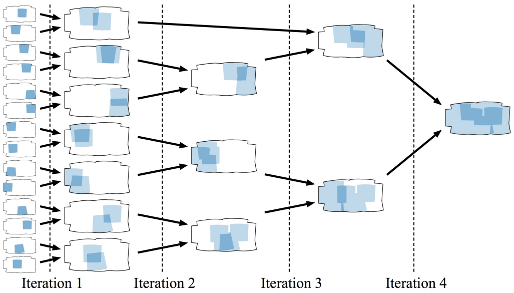

Figure 6: Illustration of the selection strategy for computing an optimal warping sequence on the example from Figure 3. Starting with the input views (first column), an iteration begins with computing all pairwise motion fields and evaluating overlap and matching quality according to Eq. (5). The algorithm then picks a maximal set of pairs of images which produce the smallest alignment error, and warps them into panoramic fragments (second column). This process of estimating motion fields, evaluating matching quality, and selecting maximum sets of pairs is iterated until only one fragment remains. The found ordering is then used for computing all frames of the panoramic output video.

Once an ordering is found, the pairwise pyramidal warping strategy illustrated in Figure 6 reduces the complexity in terms of motion field estimations per panorama output frame from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, which renders the computation tractable even for a large number of input views as in Figure 3. A comparison of a warped result with and without our optimized ordering is shown in Fig. 4.

3.3. Globally Coherent Warping

The warp ordering computed in the previous section tells us in what order to warp the images of every input frame in order to robustly handle parallax. However, for spatiotemporally stable results, two important issues have to be considered.

Firstly, warps computed only on the overlap region Ω_{ij} between two images lead to visible discontinuities [JT08]. Secondly, computing and applying warps on a per-frame basis is inherently temporally incoherent. With changing scene content, the motion fields change and so does the applied image deformation. These deformations accumulate over the

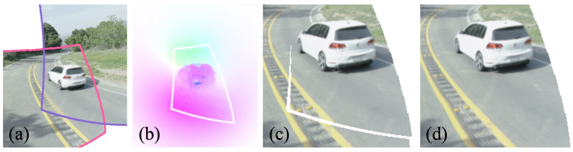

Figure 7: Importance of parallax extrapolation. (a) Two views aligned in the reference projection. (b) The interior of the white border visualizes the flow field (color encodes angle, saturation amplitude) computed on the overlap region for warping the bottom left image in (a) onto the top right image. Outside the white border the flow field has been smoothly extrapolated. (c) The warped result using only the interior of the overlap region. Note the tearing artifacts. (d) The warped result using the extrapolated flow in (b).

pairwise warping steps and differ between each video frame, and hence change the global shape of the output panorama over time with respect to the reference projection.

We propose a solution to these issues based on two components. A *weighted warp extrapolation* smoothly extrapolates the motion field u_{ij} into the non-overlap region of I_j during each pairwise warping step (see Figure 7) while remaining close to the reference projection. At the end of the warping procedure, a global relaxation step, which is constrained by the reference projection, further ensures temporal stability of the panorama shape.

Weighted warp extrapolation. We formulate the extrapolation as an energy minimization problem of the form

$$E(\widetilde{u}_{ij}) = \int_{\overline{\Omega}_j} |\nabla \widetilde{u}_{ij}|^2 \, \mathrm{d}x \, , \qquad (6)$$

where $\overline{\Omega}_j = \Omega_j \backslash \Omega_{ij}$ denotes the non-overlapping image region of I_j for which the extrapolated motion field $\widetilde{u}_{ij}$ is to be computed. In order to minimize Eq. (6), we solve the corresponding Euler-Lagrange equation $\Delta \widetilde{u}_{ij} = 0$, which is known as the Poisson equation. For the solution we assume Dirichlet boundary conditions $\widetilde{u}_{ij} = u_{ij}$ along the boundary $\partial\Omega_{ij}$ of the overlap region Ω_{ij}.

To smoothly attenuate the effects of the extrapolation we augment Eq. (6) with another set of Dirichlet boundary conditions $\widetilde{u}_{ij} = 0$ along the level set $L_c(f) = c$. The function $f(x)$ measures the minimum distance of a pixel x from any point in Ω_{ij} [FH12]. In our experiments we set $c \approx 10\%$ of the resolution of the output panorama. After discretization of the equation, the resulting linear system can be solved by any standard solver, e.g., conjugate gradient. The attenuation ensures that pixels sufficiently far away from the overlap region remain close to their position in the reference projection.

This weighted extrapolation is performed whenever we compute a warp from one image to another, including the computation of the warp ordering in Section 3.2.

Final global relaxation. This step accounts for remaining warp deformation and brings each computed output frame as close as possible back to the reference projection. We achieve this by a weighted global relaxation, in which non-overlap regions act as stronger positional constraints towards the reference projection (as their pixels are influenced less during the pairwise warping), while overlap regions are allowed to move more freely.

From the warping phase we know for each pixel of our current output panorama from which pixel(s) in the input images it originated. We can therefore define a mapping function v that maps a pixel back to its position in the reference projection. In the case where multiple input pixels contributed to a final output pixel, i.e., for pixels in overlapping image regions that are affected by parallax, we let v map the output pixel to the average position of its contributing pixels. We then compute the relaxed map v_s as the minimizer of the energy

$$E(v_s) = \int_{\Omega} w(x)\, |v_s - v|^2 \; + \; |\nabla v_s|^2 \, \mathrm{d}x \, , \qquad (7)$$

where Ω denotes the panorama domain. The first term tries to move each pixel back to its origin by penalizing deviations from the reference projection. The adaptive weights $w(x) : \Omega \to [0,1]$ are computed as a simple smooth function as in Eq. (6), which assigns a value of 1 to outer panorama boundaries and a value of 0 to overlap regions.

For minimizing the energy in Eq. (7) we solve the corresponding Euler-Lagrange equation $wv_s - \lambda \Delta v_s = wv$. We augment the equation with Neumann boundary conditions and solve it in the same way as the parallax extrapolation.

The contribution of our weighted extrapolation and global relaxation towards temporal stability of the output panoramic video are best perceived in the accompanying video (00:59 - 02:08).

3.4. Final Panorama Generation

In order to avoid multiple resampling of images, we concatenate the motion fields computed throughout our algorithm wherever possible, such that the input images have to be warped only once in order to be parallax corrected. Also we compute all above warping steps on each image separately, without performing any actual compositing of individual images yet. Only after computing the final warped projections of each input view, we blend and composite the images using the method described in [BA83] to create a seamless result.

Note that for very high resolution input the motion field computation can become a bottleneck. However, we found that computing the optical flow on 2k resolution and upsampling to the full image resolution produces satisfying results while being orders of magnitude faster. We provide a more detailed discussion of the computational performance of our method in the next section.

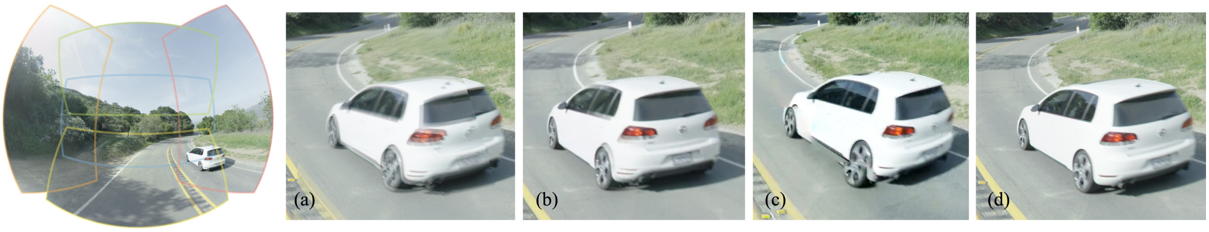

Figure 8: A frame (03:16) from a 60 megapixel panorama with a modified configuration of the RED camera array (field of views of the individual cameras are highlighted). From left to right, the close-ups show comparisons to the following methods: (a) image averaging to illustrate parallax, (b) seam optimization via graph cuts [RWSG13], (c) Autopano Video (commercial software for panoramic video generation, *www.kolor.com*), and (d) our result. Seam-based approaches generally try to "hide" parallax artifacts by cutting through well aligned image regions, rather than actively removing parallax between views. While such strategies work well on still images and specific types of video, for cluttered and highly dynamic scenes such unnoticeable seams often do not exist, so that warping becomes essential. A further problem of per-frame seam computation is temporal stability in video. In our result virtually all parallax has been removed.

4. Experiments and Results

In this section we first describe our capture setups and provide some practical tips related to preprocessing panoramic video. We then discuss a number of results created with our method. Panoramic video results and high resolution frames are provided in the accompanying video and supplemental material. Input data and results are publicly available on our project webpage for facilitating future research in this area.

4.1. Capture Setups

We used three different camera arrays, each with varying configurations. See Figure 2 for two exemplary implementations.

The most portable rig was built from 5 GoPro Hero3 cameras, each operating at 1920×1080 resolution with a horizontal view of about 94 degrees. This rig is compact and can be easily operated by a single person. Despite its small size, this rig still enabled us to produce panoramic video at resolutions of up to 3000×1500. We also built a rig consisting of 14 Lumenera LT425 machine vision cameras with Kowa 12.5mm C mount lenses, each recording at a resolution of 2048×2048 and a field of view of approximately 180×140 degrees. We show sample panoramas from both of these setups in Figure 10. Finally, the highest quality footage has been captured with a large custom rig consisting of five movie-grade cameras (RED Epic-X) with Nikkor 14 mm lenses, each capturing at a resolution of 5120×2700 pixels. The corresponding videos have been captured with the rig mounted on a helicopter and to the back of a car. We show results with resolutions up to 15339×10665, with an approximate field of view of 160×140 degrees (Figures 1,8 and 13).

4.2. Preprocessing

Due to details of the various camera hardware and APIs, it can be a difficult task to perfectly synchronize all cameras using hardware triggers alone. We therefore precompute temporal offsets for each video using the method of Wang et al. [WSZ*14]. Before feeding the captured data into our parallax correction pipeline, we apply the following common preprocessing steps, which are readily available in existing toolkits for static panorama generation such as Hugin (*hugin.sourceforge.net*).

Compensating lens distortion. For lenses with strong radial distortion it is advisable to undistort the images first in order to achieve a rectilinear projection. Nominal lens parameters for many models can be found in the LensFun database (*lensfun.berlios.de*).

Generation of the reference projection. In principle, the basic static alignment for generating the reference projection can be achieved in a variety of ways, ranging from manual alignment to feature-based methods and full 3D calibration, e.g., using calibration targets. Since our method does not require pre-calibration, we use a simple feature-based approach that estimates pairwise homographies between the images as initial alignment [HZ06]. The aligned images are then individually projected onto a common projection surface, corresponding to the desired projection model for the panorama, e.g., rectilinear, cylindrical, fisheye, etc. For our particular examples we generally used a fisheye projection as commonly required for spherical dome projections.

Exposure compensation. For handling strong exposure differences between the cameras, we perform a radiometric alignment between the input cameras [d'A07]. This helps to improve the parallax compensation as the motion estimation using optical flow can be sensitive to strong brightness or color differences between images. Remaining differences are handled by the final blending step (Section 3.4).

4.3. Results and Discussion

Figures 1, 8, and 13 show panoramic video results captured with two different configurations of the RED camera

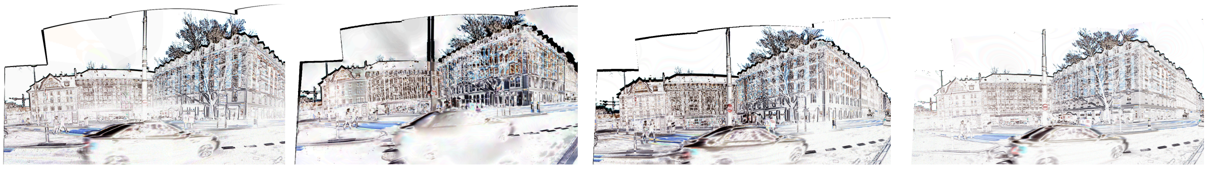
[Mic14] [ZL14] Per-frame warp order Our strategy

Figure 9: Frame-differences between sequential frames visualize the temporal warping artifacts of different approaches. Dark regions of static content correspond to temporal instability, especially visible around the border. Images have been enhanced for visibility. Please see the accompanying video (02:10) for more comparisons.

rig. In Figure 8, we show comparisons of our results to a standard static alignment, a seam-based method using graph cuts [RWSG13], and a commercial video stitching software. Figure 9 visualizes temporal stability between existing approaches and our proposed method. In Figure 12 we compare various different warp ordering strategies to our proposed strategy on input from the 14 camera array. The figure also empirically demonstrates that the choice of the frame to compute the ordering is not critical and that any choice of frame that is representative for the sequence provides a good solution. The reason for this is twofold: (1) the amount of overlap influencing motion estimation is fixed throughout the sequence and (2) the scene content viewed from each camera remains similar, e.g., downward facing cameras always point the ground, while upward facing typically see the sky, building tops and trees. In Figure 13 the close-ups show scene detail in overlap regions where our algorithm removed parallax between the views.

Figure 10 shows results captured with our 14 camera array, and additional results are provided in the video. Compared to the street example, e.g., shown in Figure 1, we reduced the mutual overlap between the field of views of the individual cameras in order to increase the overall field of view of the array. These datasets posed several considerable challenges. In particular dynamic scene elements such as objects passing directly in front of the array combined with camera ego-motion creates significant parallax differences between views. Despite these challenges our method was able to produce acceptable results for the majority of the video. We refer to the video for further validation.

A major challenge of the GoPro array (see Figure 10) was that rolling shutter effects create significant artifacts in each video. The situation is further complicated by the random orientation of the cameras (Figure 2), such that the individual scanning directions and, hence, rolling shutter artifacts, created inconsistent distortions, causing some degree of "bouncing" in the resulting video. The GoPro footage therefore represented a significant challenge, but the resulting panoramic output video still looks acceptable. In the video we compare our result to a state-of-the-art commercial software package, which clearly struggles with the above mentioned difficulties. The pairwise optimal selection and global relaxation can also handle closed cyclic view configurations without modification. We show a corresponding result in Figure 11. We encourage the reader to refer to the accompanying video (02:10) for additional comparisons with existing state-of-the-art methods and commercial softwares such as [Mic14] and Autopano Video.

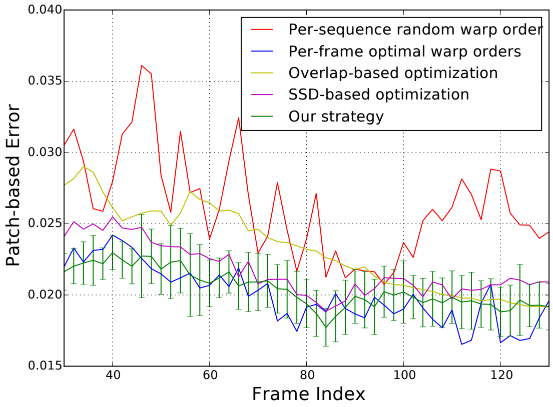

Figure 12: Error produced by different pairwise warp ordering strategies for the street sequence with 14 input views. Red: a random warp ordering fixed over the entire sequence produces high parallax error. Yellow: a warp ordering which only maximizes pairwise overlap between images does not account for parallax errors. Blue: optimal per-frame warp ordering, computed at each frame independently, generates a low error but causes temporal instability, see Figure 9. Green: our strategy is to compute an optimal warp ordering on a single reference frame, and then apply the same order to the whole frame sequence. The green plot shows the average error and standard deviation of our strategy over all possible reference frames. The results demonstrate that the choice of the reference frame to compute the ordering is not critical and parallax removal is consistent over the entire sequence. Magenta: in comparison, warp ordering based on SSD error (see Figure 5) produces inferior results.

Timing. In Table 1 we report detailed running times for each intermediate step of our algorithm at different (horizontal) pixel resolutions. All timings have been measured on a single core Intel Xeon 2.20 GHz processor. As the smooth relaxation and extrapolation can be computed at a

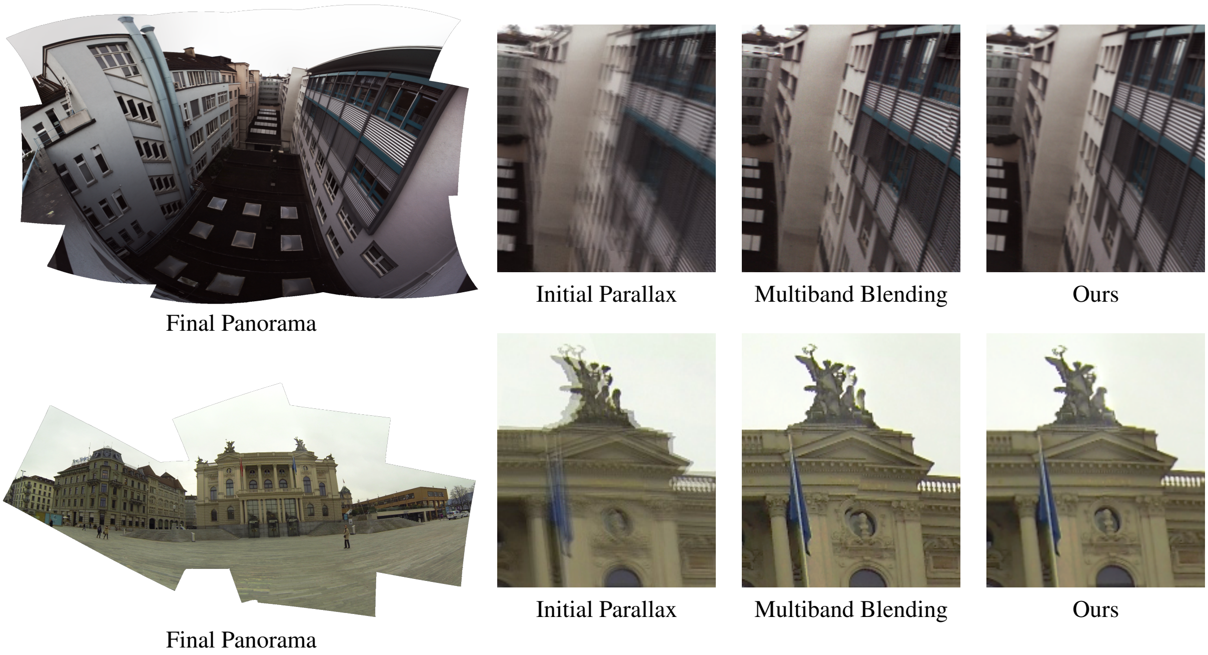

Figure 10: Results generated with our 14 camera array - top, (00:31) - and using 5 (randomly assembled) GoPro cameras - bottom, (01:30) - shown in Figure 2. Inset are a close up of the reference panorama showing the initial parallax, the result after standard multiband blending, and the result after applying our method. Note the ghosting, duplication, and discontinuities in high frequency image content which are removed in our result. Please see the video for full results.

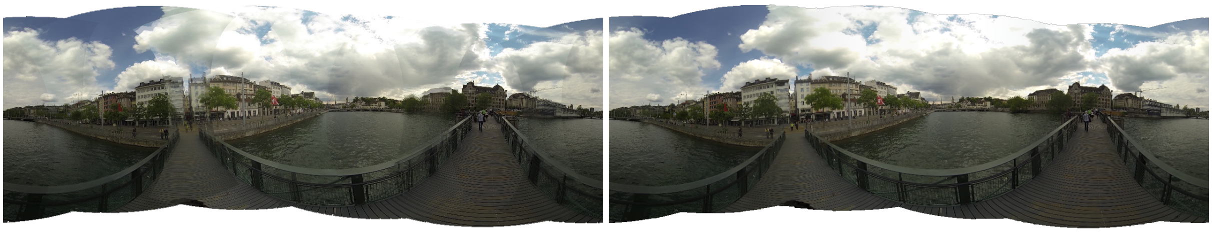

Figure 11: Our method also handles closed 360 degree configurations, here we used 5 GoPros. Left: the reference projection with averaged images to demonstrate ghosting, in particular around the rail, buildings, and the clouds. Right: our stitched result. Please zoom in for details and refer to the accompanying video (03:06).

Panorama Resolution	**512**	**1024**	**2048**	**4096**
Reference Projection	0.18	0.30	0.96	1.95
Pairwise Parallax	0.83	3.14	7.64	54.03
Extrapolation	0.40	2.20	8.54	85.62
Relaxation	1.09	4.70	9.23	78.28
Blending	0.59	1.95	7.10	29.59

Table 1: Running time in seconds for the steps required for a panoramic frame at different horizontal resolutions.

low resolution, the overall bottleneck is the motion estimation for which we use the well established method of Brox et al. [BBPW04]. As mentioned in Section 3.4, we compute motion fields on maximally 2k resolution where the flow estimation per pair of input images takes about 7 seconds, stressing the importance of reducing the necessary amount of pairwise motion field computations (Section 3.2). The overall time to compute one panoramic video frame for the 14 camera input was 57.2 seconds and 512.4 seconds for the very high resolution results from the RED camera array. Warp ordering estimation, which is only computed once per panorama, took 793.1 seconds for the 14 camera array and 128.3 seconds for the RED array, using 32 cores in parallel. Most of the computation time was spent evaluating the error measure which takes around 24.3 seconds on an overlapping region of size 256^2 pixels. The performance could be easily improved by computing the patch-error in a pyramidal fashion using smaller patch sizes instead of our single resolution implementation with patches of size 25^2.

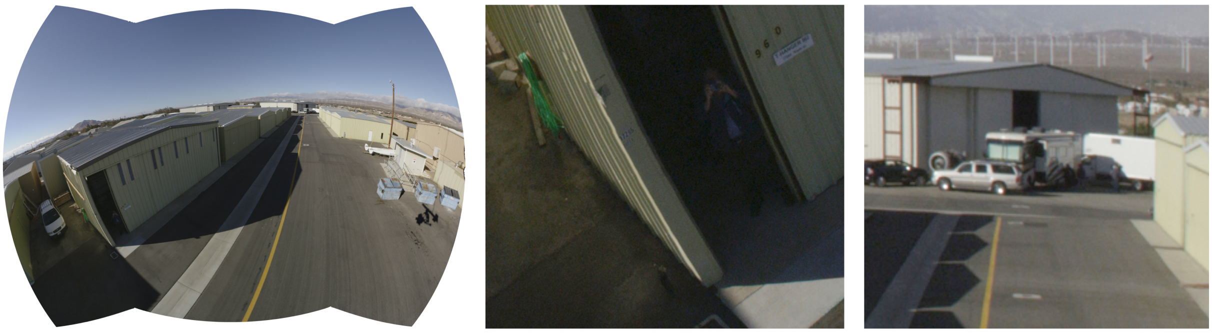

Figure 13: Additional result computed at 40 megapixel resolution, showing the full field of view of all five cameras as well as close-ups into different overlap regions, where our algorithm removed parallax between the images (03:45).

It is important to note that our method is trivial to parallelize, as each frame of the output panorama is computed independently from all other frames. We believe that this is a key property in order to be able to create high resolution output, as processing an entire video volume rather than individual frames becomes intractable with such large amounts of image-data. In the future we would like to investigate faster motion field estimators and more efficient implementations of our patch-based method. For this work we focused on the quality of the output videos rather than optimizing for speed.

Limitations and Future Work. Our method is currently limited by the properties of the basic motion estimation and warping. We build on and extend a well established concept, namely parallax removal by image warping [SS00], which we found to provide higher quality and temporally more stable results on video than existing alternatives such as seam-optimization. However, optic flow methods such as [BBPW04] have not been optimized for motion estimation in situations with large displacements between images, lack of texture, rolling shutter effects, and motion blur. In some frames of our accompanying video, artifacts caused by these issues become visible, e.g., one of the buildings in the center of the street panorama (01:24) breaks apart due to large misalignment and small overlap between input views. Furthermore our algorithm is susceptible to the quality of the reference projections. In this work we address only the correction of parallax effects and not the creation of temporally stable reference projections. Therefore effects such as jittering and wobbling observed in the input videos are preserved by our algorithm. An example can be observed in the top region of the GoPro street sequence in Figure 10, bottom - (01:30). Hence, we believe that research on the basic view synthesis aspects of our approach and the removal of temporal artifacts present in the input images is a very promising direction for future research. Our input data and results will be made publicly available to facilitate future research in these directions.

5. Conclusions

We presented an algorithm and processing pipeline for creating panoramic video from camera arrays, which focuses on the particular challenges arising for high resolution video input: spatiotemporally coherent output and globally minimized distortion despite considerable scene motion and parallax. We demonstrated that, using our algorithm, it is possible to create panoramic videos even from large arrays with challenging unstructured camera configurations and practical issues such as lack of perfect temporal synchronization or rolling shutter. To the best of our knowledge, these issues have not been explored in previous work.

We believe that algorithms for jointly processing video from multiple cameras for applications such as panorama stitching will become more important in the coming years, as it is much easier in practice to combine cameras into arrays than to increase the resolution and quality of a single camera. Assembling a set of portable GoPro cameras into an array is nowadays easily possible even for non-professional users, and camera arrays are currently even being integrated into mobile phones (*www.pelicanimaging.com*). Our aim with this work was to provide a step towards the direction of ubiquitous panoramic video capture, similar to how panoramic image capture has become part of every consumer camera.

Our input data and results are publicly available on our project webpage.

6. Acknowledgements

We are grateful to Yael Pritch for her substantial contributions during the initial phase of this work. We thank Andreas Baumann and Carlo Coppola for their invaluable support during the construction of the machine vision camera array, and Feng Liu and Fan Zhang for kindly providing results of their method [ZL14]. Finally, special thanks to Marianne McLean, Max Penner, and the entire production team at Walt Disney Imagineering and ParadiseFX. This work was supported by an SNF award 200021_143598.

References

[AAC*06] AGARWALA A., AGRAWALA M., COHEN M. F., SALESIN D., SZELISKI R.: Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph. 25*, 3 (2006), 853–861. 3

[AZP*05] AGARWALA A., ZHENG K. C., PAL C., AGRAWALA M., COHEN M. F., CURLESS B., SALESIN D., SZELISKI R.: Panoramic video textures. *ACM Trans. Graph. 24*, 3 (2005), 821–827. 3

[BA83] BURT P. J., ADELSON E. H.: A multiresolution spline with application to image mosaics. *ACM Trans. Graph. 2*, 4 (1983), 217–236. 2, 7

[BBPW04] BROX T., BRUHN A., PAPENBERG N., WEICKERT J.: High accuracy optical flow estimation based on a theory for warping. In *ECCV* (2004). 4, 10, 11

[d'A07] D'ANGELO P.: Radiometric alignment and vignetting calibration. *Camera Calibration Methods for Computer Vision Systems* (2007). 8

[EdDM*08] EISEMANN M., DE DECKER B., MAGNOR M. A., BEKAERT P., DE AGUIAR E., AHMED N., THEOBALT C., SELLENT A.: Floating textures. *Comput. Graph. Forum 27*, 2 (2008), 409–418. 3

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *SIGGRAPH* (2001), pp. 341–346. 2

[FH12] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Distance transforms of sampled functions. *Theory of Computing 8*, 1 (2012), 415–428. 7

[Gal83] GALIL Z.: Efficient algorithms for finding maximal matching in graphs. In *CAAP* (1983), pp. 90–113. 6

[HCS13] HE K., CHANG H., SUN J.: Rectangling panoramic images via warping. *ACM Trans. Graph. 32*, 4 (2013), 79. 2

[HZ06] HARTLEY A., ZISSERMAN A.: *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006. 4, 8

[IAB*96] IRANI M., ANANDAN P., BERGEN J., KUMAR R., HSU S. C.: Efficient representations of video sequences and their applications. *Sig. Proc.: Image Comm. 8*, 4 (1996), 327–351. 3

[JT08] JIA J., TANG C.-K.: Image stitching using structure deformation. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 4 (2008), 617–631. 3, 6

[KCSC10] KOPF J., CHEN B., SZELISKI R., COHEN M. F.: Street slide: browsing street level imagery. *ACM Trans. Graph. 29*, 4 (2010). 3

[KKDK12] KOPF J., KIENZLE W., DRUCKER S. M., KANG S. B.: Quality prediction for image completion. *ACM Trans. Graph. 31*, 6 (2012), 131:1–131:8. 3, 4

[KLD*09] KOPF J., LISCHINSKI D., DEUSSEN O., COHEN-OR D., COHEN M. F.: Locally adapted projections to reduce panorama distortions. *Comput. Graph. Forum 28*, 4 (2009), 1083–1089. 2

[KSU04] KANG S. B., SZELISKI R., UYTTENDAELE M.: *Seamless Stitching using Multi-Perspective Plane Sweep*. Tech. Rep. MSR-TR-2004-48, Microsoft Research, 2004. 3

[KUDC07] KOPF J., UYTTENDAELE M., DEUSSEN O., COHEN M. F.: Capturing and viewing gigapixel images. *ACM Trans. Graph. 26*, 3 (2007). 2

[LLM*11] LIN W.-Y., LIU S., MATSUSHITA Y., NG T.-T., CHEONG L. F.: Smoothly varying affine stitching. In *CVPR* (2011). 3

[LYTS13] LIU S., YUAN L., TAN P., SUN J.: Bundled camera paths for video stabilization. *ACM Trans. Graph. 32*, 4 (2013), 78. 3

[LZPW04] LEVIN A., ZOMET A., PELEG S., WEISS Y.: Seamless image stitching in the gradient domain. In *ECCV* (2004). 2

[Mic14] MICROSOFT: Microsoft image composite editor, 2014. 9

[Nay97] NAYAR S. K.: Catadioptric omnidirectional camera. *CVPR* (1997). 3

[NZN07] NOMURA Y., ZHANG L., NAYAR S. K.: Scene collages and flexible camera arrays. In *Rendering Techniques* (2007), pp. 127–138. 3

[PCD*12] PIRK S., COHEN M. F., DEUSSEN O., UYTTENDAELE M., KOPF J.: Video enhanced gigapixel panoramas. *SIGGRAPH Asia Technical Briefs* (2012), 7:1–7:4. 3

[PRRAZ00] PELEG S., ROUSSO B., RAV-ACHA A., ZOMET A.: Mosaicing on adaptive manifolds. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 10 (2000), 1144–1154. 3

[RPLP05] RAV-ACHA A., PRITCH Y., LISCHINSKI D., PELEG S.: Dynamosaics: Video mosaics with non-chronological time. In *CVPR* (2005). 3

[RWSG13] RÜEGG J., WANG O., SMOLIC A., GROSS M. H.: Ducttake: Spatiotemporal video compositing. *Comput. Graph. Forum 32*, 2 (2013), 51–61. 8, 9

[SCSI08] SIMAKOV D., CASPI Y., SHECHTMAN E., IRANI M.: Summarizing visual data using bidirectional similarity. In *CVPR* (2008). 3

[SFW*13] SCHREER O., FELDMANN I., WEISSIG C., KAUFF P., SCHÄFER R.: Ultrahigh-resolution panoramic imaging for format-agnostic video production. *Proceedings of the IEEE 101*, 1 (2013), 99–114. 3

[SS00] SHUM H., SZELISKI R.: Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *IJCV 36*, 2 (2000), 101–130. 2, 3, 4, 5, 11

[STP12] SUMMA B., TIERNY J., PASCUCCI V.: Panorama weaving: fast and flexible seam processing. *ACM Trans. Graph. 31*, 4 (2012), 83. 2

[Sze06] SZELISKI R.: Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision 2*, 1 (2006). 2

[WSI07] WEXLER Y., SHECHTMAN E., IRANI M.: Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 3 (2007), 463–476. 3, 4

[WSZ*14] WANG O., SCHROERS C., ZIMMER H., GROSS M., SORKINE-HORNUNG A.: Videosnapping: Interactive synchronization for multiple videos. *ACM Trans. Graph. 33*, 4 (2014). 8

[ZC12] ZHI Q., COOPERSTOCK J. R.: Toward dynamic image mosaic generation with robustness to parallax. *IEEE Transactions on Image Processing 21*, 1 (2012), 366–378. 3

[ZCT*14] ZARAGOZA J., CHIN T., TRAN Q., BROWN M. S., SUTER D.: As-projective-as-possible image stitching with moving DLT. *IEEE Trans. Pattern Anal. Mach. Intell. 36*, 7 (2014), 1285–1298. 3

[ZL14] ZHANG F., LIU F.: Parallax-tolerant image stitching. In *CVPR* (2014). 3, 9, 11

[ZMP07] ZELNIK-MANOR L., PERONA P.: Automating joiners. In *NPAR* (2007), pp. 121–131. 3