A TOOLCHAIN FOR CAPTURING AND RENDERING STEREO AND MULTI-VIEW DATASETS

Felix Klose, Christian Lipski, Kai Ruhl, Benjamin Meyer, Marcus Magnor

Institut für Computergraphik, TU Braunschweig

ABSTRACT

We present our toolchain of free-viewpoint video and dynamic scene reconstruction for video and stereoscopic content creation. Our tools take video data from set of sparse unsynchronized cameras and give great freedom during the post production. From input data we can either generate new viewpoints employing the Virtual Video Camera a purely image based system or generate 3D scene models. The approaches are explained and guidelines for weighing their specific advantages and disadvantages in respect to a concrete application are given.

Index Terms— Stereoscopic 3D, Multi-View Stereo, Unsynchronized Cameras.

1. INTRODUCTION

Handling stereoscopic or multi-view material requires a specialized set of algorithms. In order to capture, process and re-render such image sequences, typical challenges arise. One of which is the camera calibration, others are reconstructing geometry or synthesizing novel viewpoints. A special case of novel viewpoints synthesis is the generation of a second eye view for stereoscopic 3D (S3D) rendering.

The relative placement of multiple images or cameras is of special importance for S3D content generation. In a stereo camera rig the spatial distance between the camera viewpoints is the baseline and one of the most important parameters for the viewers 3D experience. Additionally the temporal synchronization of stereo-rig cameras is mandatory to avoid viewing discomfort caused by image content mismatches and vertical disparities.

Therefore camera calibration does not only require recovering the spatial camera parameters, but also the temporal offset. This becomes even more an issue, when using more than two cameras or a recorder setup with heterogeneous frame rates.

When recording with a stereo camera rig, the camera baseline is fixed at the recording time. The possibility to allow modifications to the baseline or the overall 3D impression is very desirable and an active research area. The same holds true for the temporal domain. Resampling a video in time can be used to achieve slow motion effects or upsample footage, that is recorded at slow framrates.

Modifying S3D content on an even finer scale can enable non-linear depth remapping to keep the 3D effects in the viewers comfort space. To achieve this per pixel remapping capabilities, a scene depth reconstruction is necessary. For multi-view setups, more information is available that can be exploited for reconstruction purposes, but the hardware effort required to gen-lock and calibrate cameras greatly increases. With a given 3D scene model depth based rendering enables re-rendering different or adaptive baselines from a single view.

For scene analysis or special effect purposes, it can be important to recover the motion from the recorded material. We show a set of tools and algorithms that tackle those problems, and create a great degree of freedom during the post-production.

2. RELATED WORK

There exists a great deal of research for 3D reconstruction, rendering and production techniques. It would be far beyond the scope of this paper to give a comprehensive overview. However only a few solutions focus on stereo- and multi-view work flows. Some groups have developed similar projects we would like to highlight here:

The University of Surrey in cooperation with the BBC developed iview [6]. Their multi-view free viewpoint system is focused on performance and sport events. It is able to render novel viewpoints such as first person perspective in sports events. The system is based on estimating a 3D reconstruction from multiple broadcast camera views and render the textured geometries.

Recently Disney Research [17] presented their tools for disparity-aware stereo 3D production. They show for the stereoscopic case what can be accomplished in the fields of non-linear depth remapping. Employing image-based warping techniques they show how to regain production freedom to increase the viewer comfort when watching S3D content.

For individual aspects of producing and editing stereoscopic 3D content an increasing number of commercial tools
exist. The Foundry have a stereoscopic editing suite called Ocula based on their compositing framework NUKE, which also allows adaptive image warping to change the 3D impression of video material.

This paper first gives an overview of our tools in Sec. 3. Then the individual aspects are presented in Sec. 4, Sec. 5, and Sec. 6. The applications section Sec. 7 focuses on advantages and disadvantages of different tool choices and Sec. 8 concludes the paper.

3. PRODUCTION WORKFLOW OVERVIEW

Fundamental principle of our work flow is to impose as few restrictions as possible on the recording modalities while preserving the greatest possible artistic freedom during post production. To make multi-view recording feasible, we focus on footage recorded with multiple consumer grade cameras that are not synchronized.

After recording, the processing of input material typically begins with spatially calibrating the cameras by determining their parameters in a common 3D space. Recent structure-from-motion algorithms for unordered image collections solve this problem robustly.

Following the spatial registration a temporal alignment of the camera feeds is performed. To determine the subframe accurate offsets we use an algorithm described in Sec. 4 that is based on the tracking of 2D image features.

With the fully calibrated input material our workflow offers two options. Our Free-Viewpoint rendering system (Sec. 5) is capable of rendering new in between views in space and time. It thereby returns a great deal of freedom to the content editor. The camera viewpoint including the temporal dimension can be varied, S3D content can be obtained either rendering multiple views, or generating depth maps and use depth-based rendering to create S3D. The latter also allows for modification of the depth impression. Being entirely image based, no explicit 3D scene model is necessary to do so.

The second option is to start a multi-view reconstruction process that recovers a dynamic scene model (Sec. 6). The resulting model is patch based and each has an associated position, normal and linear motion vector for each surface patch.

Both option have their own merits and applications, which are discussed in section 7.

4. SUBFRAME TEMPORAL REGISTRATION

Our setup of unsynchronized cameras, while ideally suited to casual capture, requires us to estimate the temporal alignment in post production. Preferably, the alignment is performed with subframe accuracy.

We divide our method into two steps: In the first, we track feature points in two cameras over several frames and match their trajectories using extremal point and binary reduction and string matching. This results in frame-accurate alignment. In the second step, we determine the subframe offset between two cameras over a single frame by optimizing one parameter over the fundamental matrix, similar to [19] and [2].

Both steps assume linear motion between frames. Camera motion is tolerated, but for the first step, it has to be smaller than the tracked object motion.

A more detailed discussion can be found in [13].

4.1. Frame-accurate alignment

To achieve frame-accurate temporal alignment, common feature points followed over multiple frames in all cameras are considered, e.g. a bouncing ball.

Its trajectory is initially followed by a modified version of the KLT tracker. The ideal characteristic movement consists of extremal points of a linear motion, therefore we use the Douglas-Peucker Algorithm with a scale parameter $\epsilon$, which determines the number of remaining points for reduction, see Fig. 1(a). We adjust $\epsilon$ such that a sufficient number of intermediary points remains.

Since we are only interested in the temporal component of the motion, not in the spatial, we can simplify our problem further by reducing the extremal points to binary codes, see Fig. 1(b). To this end, we model extremal points as 1 and intermediary points as 0.

Finally, we use simple string matching to determine the offset in which a motion is most similar in multiple cameras.

The approach requires motion that can be linearly approximated, and camera motion must be smaller than object mo-
from each sequence, using a 9-matrix determined by solving Eq. 4.

ject movement, a unique solution for movable motion exists and camera movement is smaller than observed.

Fig. 2. The cameras ca, cb, cc and cd on the sphere S together with the ground plane given by \( g_{1,2,3} \), define the dimensions elevation and azimuth of \( N \).

Under these conditions, the method described above is both fast and reliable, particularly when coupled with a RANSAC to remove outliers.

4.2. Subframe-accurate alignment

Given the frame accurate alignment, we choose two image pairs \( T \) and \( T' \), where points in time \( p_t \) and \( p_{t+1} \) are related to points \( p'_t \) and \( p'_{t+1} \) such that \( p_t \leq p'_t \leq p_{t+1} \).

Furthermore, we assume that a point in time \( p_{t+a} \) can be approximated by a parameter \( a \) with \( 0 \leq a \leq 1 \) using linear interpolation:

\[
    p_{t+a} = (1-a)p_t + a p_{t+1} \quad (1)
\]

Our basic strategy is to find an \( a \) solving for the fundamental matrix \( F_t \) such that the basic fundamental matrix equation

\[
    (p'_t)^T F_t p_t = 0 \quad (2)
\]

is approximated by

\[
    (p'_t)^T F_t ((1-a)p_t + a p_{t+1}) = 0 \quad (3)
\]

To this end, we need 9 independent motion trajectories from each sequence, using a 9x9 matrix \( M(a) \) and an unknown 9-vector \( f_t \) such that Eq. 4 becomes

\[
    M(a) f_t = 0 \quad (4)
\]

If the constraint \( det(M(a)) = 0 \) is satisfied, i.e. a suitable motion exists and camera movement is smaller than object movement, a unique solution for \( a \) exists and is found by solving Eq. 4.

Given the above constraints, the offset \( \Delta t \) is uniquely determined by \( a \) in just one single step. Accuracy can be further improved by calculating the statistical mean of several such offsets for subsequent frames. This effectively removes outliers.

5. IMAGE-BASED FREE VIEWPOINT VIDEO

With our Virtual Video Camera rendering system it is possible to viewpoint-navigate through space and time of complex real-world, dynamic scenes. As stated earlier our approach accepts unsynchronized multi-video footage as input. Inexpensive, consumer-grade camcorders suffice to acquire arbitrary scenes, e.g., in the outdoors, without elaborate recording setup procedures, allowing also for hand-held recordings. Instead of scene depth estimation, layer segmentation, or 3D reconstruction, our approach is based on dense image correspondences, treating view interpolation uniformly in space and time: spatial viewpoint navigation, slow motion, or freeze-and-rotate effects can all be created in this unified framework.

The view generation system naturally extends to generating stereoscopic content. The two different approaches to synthesize a second view (see Sec. 5.2.

While we give a overview of the system in this paper, discussion in great detail can found in [10].

5.1. Preprocessing

Our goal is to explore the captured scene in an intuitive way and render a (virtual) view \( I_v \) of it, corresponding to a combination of viewing direction and time. To this end, we choose to define a 3-dimensional navigation space \( N \) that represents spatial camera coordinates as well as the temporal dimension. All new virtual viewpoints are described in respect to this navigation space.

To create this space cameras are placed on the surface of a virtual sphere, their orientations are defined by azimuth and elevation (Fig. 5). Together with the temporal dimension \( t \), azimuth and elevation span our 3-dimensional navigation space \( N \). If cameras are arranged in an arc or curve around the scene, elevation is fixed, reducing \( N \) to two dimensions. A sparse set of world points provided by the structure from motion system, is used to define a common ground plane for the navigation space.

After mapping the input images from the calibrated euclidean space into \( N \), we compute a Delaunay triangulation. This enables us to partition \( N \) into a set of tetrahedra. Each tetrahedra is defined by four vertices corresponding to four input images and six edges. For each edge between two images \( I_i \) and \( I_j \), we compute two dense correspondence maps \( W_{ij} \) and \( W_{ji} \).

5.2. View Synthesis

Having subdivided navigation space \( N \) into tetrahedra, each point \( v \) is defined by the vertices of the enclosing tetrahedron \( \lambda = \{ v_i \}, i = 1, \ldots, 4 \). Its position can be uniquely expressed as \( v = \sum_{i=1}^{4} \mu_i v_i \), where \( \mu_i \) are the barycentric coordinates of \( v \) Fig 5. Each of the 4 vertices \( v_i \) of the tetrahedron corresponds to a recorded image \( I_i \). Each of the 12 edges \( e_{ij} \)
correspond to a correspondence map \( W_{ij} \), that defines a translation of a pixel location \( x \) on the image plane. We are now able to synthesize a novel image \( I_v \) for every point \( v \) inside the recording hull of the navigation space \( \mathcal{N} \) by multi-image interpolation:

\[
I_v = \sum_{i=1}^{4} \mu_i \tilde{I}_i,
\]

where

\[
\tilde{I}_i \left( \Pi_i(x) + \sum_{j=1,\ldots,4,j \neq i} \mu_j \Pi_j(W_{ij}(x)) \right) = I_i(x)
\]

are the forward-warped images \( \{I_i\} \). \( \{\Pi_i\} \) defines a set of re-projection matrices that map each image \( I_i \) onto the image plane of \( I_v \), as proposed by Seitz and Dyer [15]. Those matrices can be easily derived from camera calibration. Since the virtual image \( I_v \) is always oriented towards the center of the scene, this re-projection corrects the skew of optical axes potentially introduced by our loose camera setup and also accounts for jittering introduced by dynamic cameras. Image re-projection is done on the GPU without image data resampling.

Up to this point the view synthesis is monocular. In order to create stereoscopic content a second view has to be synthesized. Since the navigation space has an azimuth parameter creating the image for the second eye can be accomplished by synthesizing a new viewpoint from a slightly offset azimuth position. By varying the offset the baseline can easily be adjusted for a pleasant S3D viewing experience. Problematic are model violations of the linear movement assumption underlying the 2D image correspondences. Those as well as misalignment of the cameras can lead to slight vertical disparities, which are known to cause viewer discomfort. Another issue is, that our system is limited to view interpolation, which makes creating new views impossible if the recording setup has no neighbouring camera on the horizontal axis.

For the most recent production application [20] we therefore opted for another approach [9]. The dense image correspondences in combination with the camera calibration can be used to triangulate dense depth maps. Those depth maps can in turn be used to employ depth image based rendering to create two stereoscopic views.

6. RECONSTRUCTING SHAPE AND MOTION

Instead of directly synthesizing the new viewpoints with the Virtual Video Camera system, the calibration and timing information together with the recorded images can be used to recover geometric information. The challenge for a 3D reconstruction arises from the unconstrained nature of input data. Unsynchronized cameras in combination with dynamic scenes make traditional static stereo, or multi-view reconstructions inapplicable, since objects may vary their position between any two input images.

One of the assumptions we make is that the input video streams show multiple views of the same scene. Since we aim to reconstruct a geometric model, we expect the scene to consist of opaque objects with mostly diffuse reflective properties. An additional assumption is made about the scene dynamics. To be able to recover the motion, we assume a linear motion for all points in the scene.

Our scene model represents the scene geometry as a set of small tangent plane patches. The goal is to reconstruct a tangent patch for the entire visible surface. Each patch is described by its position \( \vec{c} \) at \( t = 0 \), the normal \( \vec{n} \) and a velocity vector \( \vec{v} \). The position of a patch at a given time \( t \) follows as

\[
x(t) = \vec{c} + t \cdot \vec{v}
\]

One of the advantages of a patch based scene model is, that properties such as position, motion and visibility are inherently defined per patch. This gives the scene model the necessary degrees of freedom to describe complex objects with non rigid motions. For static scenes Furukawa et al. [4] demonstrated the potential of this approaches with their Patch-based Multi-view Stereo.

6.1. The Reconstruction Process

Our algorithm starts by creating a sparse set of seed patches in an initialization phase, and grows the seeds to cover the visible surface by iterating expansion, optimization and filter steps.

The algorithm processes a group of images at a time. The images are chosen by their respective temporal and spatial parameters. All patches extracted from an image group collectively form the dynamic model of the scene. This model is valid for the time ranging from the acquisition of the first image of the group to the time the last selected image was recorded.

We start by calculating points of interest selected by evaluating the eigenvalues from the covariation matrix of the gradient image. A SURF descriptor [1] is then calculated for those points.
The object path has to intersect the rays \( \vec{r} \). The red dots on the image planes show the three matched features from which the path of the scene object will be recovered. The object path has to intersect the rays \( \vec{r}_{0,1,2} \).

In a matching step features are matched across all images yielding multiple sets of points. The points in each set possibly belonging to the same position in the scene. The set is evaluated and if possible, one patch per set is recovered.

To determine position \( \vec{c} \) and velocity \( \vec{v} \), a linear equation system is formulated. (Fig. 4) The line of movement \( \vec{q} \) must intersect the viewing rays \( \vec{q}_i \) that originate from the camera center \( \Phi_i \) and are cast through the image plane at the pixel position where the patch was observed in image. The time when an input image \( I_i \) was recorded is denoted by \( t_i \):

\[
\begin{pmatrix}
I_{d^2 \times 3} & I_{d^2 \times 3} \cdot t_{0} & -\vec{q}_0^T & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
I_{d^2 \times 3} & I_{d^2 \times 3} \cdot t_{i} & 0 & 0 & -\vec{q}_i^T
\end{pmatrix}
\begin{pmatrix}
\vec{c}^T \\
\vec{v}^T \\
\Phi_i^T
\end{pmatrix} = \begin{pmatrix}
\Phi_0^T \\
\Phi_0^T \\
\Phi_i^T
\end{pmatrix} \tag{8}
\]

The variables \( \alpha_0 \) to \( \alpha_i \) give the scene depth in respect to the camera center \( \Phi_0 \). The overdetermined linear system is solved with a SVD solver.

Using \( \vec{c}, \vec{v} \) and the camera calibration the reconstructed patch is reprojected into the input images. The reprojection error is used to evaluate the validity of the reconstruction. If the point set from which \( \vec{c} \) and \( \vec{v} \) were calculated has more points than needed for the reconstruction, RANSAC is used to remove outliers.

If a reconstruction is found to be valid, the normal for the new patch needs to be determined. Therefore it is initialized as pointing directly at one of the cameras \( \Phi_i \), from which the patch was reconstructed. To refine the normal the normalized cross correlation (NCC) across all images is maximized using a conjugate gradient optimization framework. In the same optimization the position and velocity are refined. Details about the cost function and the patch refinement can be found in the original paper [7].

The number of patches after this initial reconstruction is very low. To incrementally cover the entire visible surface, the existing patches are expanded along the object surfaces.

For a given patch, the surrounding pixels in the images are tested and if no other patch exists at this position a new patch is created at that position. A viewing ray is cast through the center of the empty pixel and intersected with the plane defined by the patch position and its normal. The intersection point is the center position for the newly created patch. The velocity and normal of the new patch are initialized with the values from the source patch. The new patch is then subjected to the NCC based refinement to improve the initial values for position \( \vec{c} \), velocity \( \vec{v} \) and normal \( \vec{n} \).

After evaluating all patches for possible expansions, a filtering step is performed. The filtering eliminates inconsistencies in the scene model introduced by the last expansion step. Simple depth and occlusion tests are used to filter patches that lie within or before the actual surface. A third filter test enforces a light regularization. Therefore a similarity measure is introduced.

To determine whether two patches are similar in a given image, their position \( \vec{x}_0, \vec{x}_1 \) and normals \( \vec{n}_0, \vec{n}_1 \) are used to evaluate the inequality

\[
(\vec{x}_1 - \vec{x}_0) \cdot \vec{n}_0 + (\vec{x}_0 - \vec{x}_1) \cdot \vec{n}_1 < \kappa. \tag{9}
\]

The comparison value \( \kappa \) is calculated from the size of the local region in which patches are compared. If the inequality holds, the two patches are similar.

All surrounding \( c \) patches in the local region are evaluated with [9]. The quotient of the number \( c' \) of patches similar to the original patch in relation to the total number of surrounding patches \( c \) is the regularization criterion: \( \frac{c'}{c} < \zeta \). Our results show that a quotient of \( z = 0.25 \) leads to satisfying results.

After a predefined amount of the input images is covered, the iteration process of growing and filtering the patches is stopped.

### 7. APPLICATIONS

To demonstrate the capabilities of the Virtual Video Camera and the dynamic scene reconstruction we show footage from a recent music video project [20] in which our tools where applied in a production environment (Fig. 5).

When preparing recorded datasets for post processing the most basic decision to be made is what type of model to use for analysing the data. The two most popular choices are a geometry based scene model or image-based approaches. Traditionally a lot of video and image post processing like segmenting, compositing, color grading and even basic relighting is done purely in the image domain. A geometry model is hand created, or reconstructed for special effects purposes, or more recently to create S3D content.

In the context of the Virtual Video Camera from our workflow, we use an image-based technique to generate novel view-
Fig. 5. (a,b) Two input views of the static set background painted with graffiti, (c) a view rendered from a position in between (a) and (b), (d) textured patches as reconstructed from the multi-view reconstruction, (e,f) input views of the foreground live action recording, (g) synthesized in between viewpoint masked with a color key, (h) dynamic scene reconstruction results.

points. The main limitation with the current rendering approach is that only interpolation between camera views is possible. Views that leave the camera manifold cannot be synthesized.

Since the underlying concept is dense image correspondences, that are used to warp the input material, no explicit 3D model needs to be reconstructed. This allows the rendering amorphous objects and phenomena such as fire, water or smoke, by finding visually plausible correspondences without the constraint of geometric validity.

The output material rendered from the Virtual Video Camera is dense by construction. When having a reconstruction approach in mind it is noteworthy, that the results are not affected by low texture regions.

Although the image correspondences are not geometrically constrained, they can be used to estimate dense depth maps. However we found them to be accurate enough to create high quality S3D content by depth image based rendering. By modifying the depth images before the rendering step it is even possible to achieve a depth remapping to further enhance the 3D experience.

Additionally only one resampling step from the recorded images to the final view is required. This has the advantage of faithfully preserving the original scene appearance.

The major drawback of staying in the 2D image domain is, that occlusions and disocclusions are not apparent from the correspondence fields. It is possible to enforce symmetric correspondences and in case of non-symmetry assume a (dis-)occlusion, in some cases this heuristic is still prone to error. Depending on the scene (no smoke, fire, water), it is possible to use the dense depth maps as an approximation of the real scene geometry to resolve these issues. This approach yields high quality results.

With the above in mind, there are cases, when a geometric reconstruction seems a better choice.

For a 3D reconstruction to give satisfying results the common scene restrictions apply: only opaque objects. Glass, fog or fire, mirrors and in fact any material that has strongly view dependent appearance will decrease the reconstruction result as it gets harder to match across multiple images.

Besides those restrictions a 3D model as described in Sec. 6 is a description of the scene holding much less redundancy. Which is an advantage when dealing with large datasets. Reconstructing data in a 3D domain instead of purely 2D image correspondences gives additional constraints that allow explicit handling of occlusion and disocclusion.

When the textured 3D data is recovered, it integrates well with existing production tools. Tasks as relighting or compositing other 3D content into the scene is straightforward. However to gain this data in the highest possible quality additional steps as surface reconstruction, view-dependent texture mapping and error concealment during rendering become necessary.

As mentioned earlier depending on the scene structure a geometry reconstruction is often only possible in the salient regions of the input images. This poses a problem especially for the surface reconstruction steps.

Under the given restrictions a textured scene model is still of value, since the camera viewpoint can be chosen freely in the scene. In contrast to the Virtual Video Camera, extrapolating camera positions is possible. However when moving the synthesized viewpoint away from the original input camera viewpoints hole filling strategies have to be applied to fill areas where no data was observed.

8. CONCLUSION

We presented our toolchain for free-viewpoint 3D stereo video, the virtual video camera including a 3D reconstruction approach. Its main strength in the context of 3D stereo video...
is the flexible choice of the right tool for the desired application. Both tools the Virtual Video Camera and the dynamic scene reconstruction allow for easy post production variation of parameters such as baseline that are usually fixed with contemporary stereo recording rigs. The low requirements for the camera are make it cost-effective and easy to set up.

The natural limit of our system is the sparsity of cameras and frame rates. Any high speed motion of small objects or ultra-wide camera spacing must be accounted for with appropriate camera equipment.

Ultimately, we believe that post production flexibility in stereo parameters including the camera position is highly desirable and as we have shown very well feasible.

9. ACKNOWLEDGEMENTS

This work has been funded by the European Research Council ERC under contract No.256941 Reality CG and by the German Science Foundation, DFG MA 2555/1-3.

10. REFERENCES


