A Convex Relaxation Approach to Space Time Multi-view 3D Reconstruction

Martin R. Oswald and Daniel Cremers
Department of Computer Science, TU München*

Abstract

We propose a convex relaxation approach to space-time 3D reconstruction from multiple videos. Generalizing the works [16], [8] to the 4D setting, we cast the problem of reconstruction over time as a binary labeling problem in a 4D space. We propose a variational formulation which combines a photoconsistency based data term with a spatio-temporal total variation regularization. In particular, we propose a novel data term that is both faster to compute and better suited for wide-baseline camera setups when photoconsistency measures are unreliable or missing. The proposed functional can be globally minimized using convex relaxation techniques. Numerous experiments on a variety of publically available data sets demonstrate that we can compute detailed and temporally consistent reconstructions. In particular, the temporal regularization allows to reduce jittering of voxels over time.

1. Introduction

Estimating 3D geometry from a set of images is among the central problems in computer vision. Especially for static scenes significant advances have been made in the last decade that allow for high quality 3D reconstructions. An overview is found in [13]. Unfortunately, the generalization of these techniques to the reconstruction from videos is by no means straightforward. Firstly, there are usually far fewer cameras, the synchronization and simultaneous acquisition from many cameras still being a costly and cumbersome effort. With a wider average baseline, many existing schemes for photoconsistency estimation break down because respective patches are no longer visible in the other images or too distorted for reliable patch comparison. Secondly, accurate reconstructions over time pose huge demands with respect to memory and computation time – in particular if one wishes to exploit the temporal coherence of the reconstruction over consecutive frames. In this work, we tackle the problem of space-time 3D reconstruction by means of a convex optimization approach.

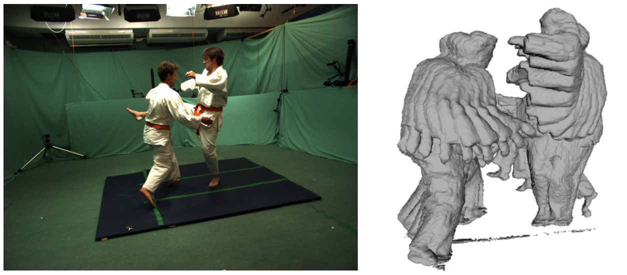


Figure 1. One of the input images and several times frames of a space time surface evolution.

1.1. Contributions

- We generalize the works of Unger et al. [16] and Kolev et al. [8] from the three-dimensional setup to a four dimensional one leading to a mathematically transparent and globally optimal approach for space-time multi-view 3D reconstruction.
- In order to make the 3D reconstruction approach by Kolev et al. [8] work in wide-baseline camera setups we propose a novel data term, which has several desirable properties and improves the one in [8] in several aspects. Firstly, it better preserves surface edges and concavities. Secondly, it has better hole filling abilities when photoconsistency information is weak and sparse. Finally, it does not have a global influence, that is, it does not affect surface parts which are not visible in the respective camera.
- Further, we reduce the computation time per frame from several hours, as reported by [8], to appr. 1-2 minutes for equivalent volume sizes. This aspect is important when processing longer sequences.

1.2. Related Work

Zhang et al. [18] extended the problem of classical binocular stereo matching into the space time domain. Pioneering work on the topic of space-time 3D reconstruction in a multi-view setup has been done by Goldluecke et al. [3], [4]. They described the evolution of a space time surface by means of level set functions which iteratively approach a local minimum of the respective energy.

*This work was supported by the ERC Starting Grant 'Convex Vision'.

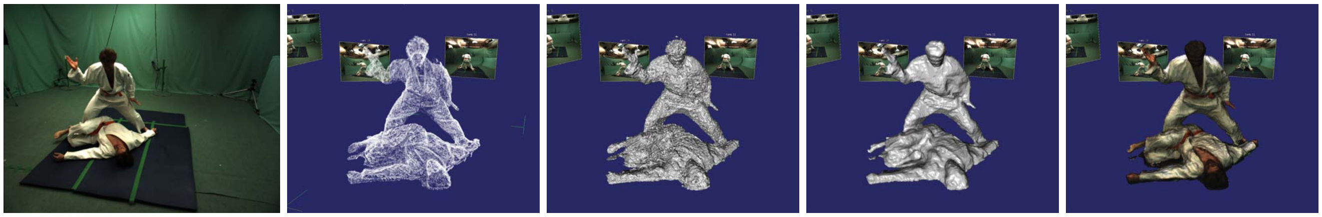


Figure 2. Outline of the proposed space time reconstruction framework. Two men are filmed synchronously by 16 cameras. The figure shows (left to right) one input image, estimated photoconsistencies, a level set of the proposed data term, the final reconstructed mesh shaded and textured.

Generally, these methods rely on a proper initialization to converge to the desired solution. In [1], Aganj et al. proposed to calculate a spatio-temporal coherent mesh from silhouettes using 4D Delaunay meshing. Guillemaut and Hilton [5] jointly solve the problem of multi-layer segmentation and depth estimation within a graph-cut framework. They enforce temporal coherence by means of optical flow measures which are weighted according to their confidence to account for unreliable flow estimates. Richardt et al. [12] recently proposed a method for spatio-temporal filtering and upsampling of RGB-Depth videos. Sharf et al. [14] study the problem of space time reconstruction by means of incompressible flow.

Our approach is related to the space-time 2D tracking framework by Unger et al. [16]. They cast the problem of tracking objects in images over time as a 3D segmentation problem to model temporal smoothness or deal with temporally short occlusions of the tracked object. Although the task and several properties are quite different we use a similar model, but in a 4D rather than a 3D setting.

In [8], Kolev et al. proposed to model the 3D surface as a binary inside-outside labeling in 3D space to convexify the surface reconstruction problem and hence obtain globally optimal solutions for multi-view 3D reconstruction. A similar model to the one in [8] has also recently been used by Ummenhofer and Brox [15] for combined 3D reconstruction and camera pose estimation. We adopt their approach because this model has several desirable properties. It easily deals with topological changes and allows for global optimization. Further, it provides a natural way for surface regularization in 3D which is perfectly suited for a multi-view setup.

Although a variety of useful regularizers for depth maps have been presented in the literature, intuitively they do not provide a good regularization in a multi-view setup because we are usually looking for a connected and locally smooth surface rather than a smooth depth map. 3D reconstruction based on depth maps is a popular approach to this problem and many works exist on this topic e.g. [17],[7]. Inherently these approaches split the overall problem into two separate ones: depth reconstruction followed by surface reconstruction based on these depth maps. As a result, important information such as the consistency of an estimated depth map value is usually not handed over into the following surface reconstruction. In contrast, our goal is to carry as much information as possible into the final global 3D surface optimization.

1.3. Paper Outline

In the following we introduce our space time reconstruction model and subsequently explain how to compute respective terms. In Section 3 we explain the optimization procedure and give some details on the implementation in Section 4. Section 5 presents results on several data sets and Section 6 concludes the paper.

2. Variational Space Time Reconstruction

Let $V \subset \mathbb{R}^3$ describe a volume in space and let $T \subset \mathbb{R}_+$ represent the temporal domain. We are looking for a smooth hypersurface S in the space $V \times T$ which best explains the series of input images with known projections $\{\pi_i\}_{i=1}^N$. For ease of notation we will drop the temporal index whenever the meaning is clear by context. Similar as in [8] we represent surface S by means of a binary labeling function $u : V \times T \mapsto \{0, 1\}$ which indicates surface interior (1) or exterior (0). We follow the path of their work and define an energy function which measures both the surface smoothness and how well the surface fits to the input data.

$$E(u) = \int\limits_{V\times T} \Big(\rho|\nabla_{\boldsymbol{x}} u| + g_t|\nabla_t u|\Big)\, d\boldsymbol{x}dt + \lambda \int\limits_{V\times T} f u \;\; d\boldsymbol{x}dt \quad (1)$$

The second term in Eq. (1), data term f gives local preferences for either an interior or an exterior label and will be defined in Subsection 2.2. It is weighted by parameter $\lambda > 0$ to favor either a smooth surface or a surface that aligns with the potentially noisy data. The task of the first term - the regularization term - is to reject outliers, deal with locations of missing data and to favor a spatially and temporally smooth surface. To account for the inherent difference between spatial and temporal dimensions this term is split into a spatial and a temporal part which then regularizes these dimensions in an anisotropic manner.

The spatial regularization is weighted by function $\rho : V \times T \mapsto \mathbb{R}$ which represents the photoconsistency measure being defined in the following section. Weighting down the penalization of the gradient norm ρ makes the surface boundary snap to probable surface locations which are indicated by a low photoconsistency value ρ.

In Eq. (1) function $g_t : V \times T \mapsto \mathbb{R}$ steers the temporal smoothness. We choose it as a function that depends on the gradient magnitude of the data term:

$$g_t(\boldsymbol{x}, t) = \exp\big(- a|\nabla_t f(\boldsymbol{x}, t)|^b\big) \ . \tag{2}$$

This choice of $g_t(\cdot)$ prevents locations with strong gradients from being over-smoothed which is a favorable property in the presence of fast surface motions. The purpose of the temporal regularization is mainly to suppress temporal noise in the surface reconstruction rather then penalizing surface motion in a dynamic scene. The effects of parameters a, b will be discussed in the experimental section.

2.1. Photoconsistency Estimation

For each camera i we define a cost function[1] $C_i : V \times \mathbb{R} \mapsto \mathbb{R}$ which calculates a matching cost at a location defined by distance d from the camera center towards or through point $\boldsymbol{x}$ based on the normalized cross correlation (NCC)

$$C_i(\boldsymbol{x}, d) = \sum_{j \in \mathcal{C}' \setminus i} w_i^j(\boldsymbol{x}) \cdot \text{NCC}\Big(\pi_i\big(r_i(\boldsymbol{x}, d)\big), \pi_j\big(r_i(\boldsymbol{x}, d)\big)\Big) \ . \tag{3}$$

The function $r_i : V \times \mathbb{R} \mapsto V$ returns points on the ray from camera i through point $\boldsymbol{x}$ according to a given distance d from the camera. To calculate $C_i(\cdot)$ we select a subset of front-facing cameras $\mathcal{C}' \subset \mathcal{C}$ for which the angle between the viewing directions is below γ_{max}=85°. The contribution of each camera is weighted by a normalized Gaussian weight $w_i^j(\boldsymbol{x})$ of the angle between view directions of cameras i and j. Further, we discard unreliable correlation values by means of a threshold $\tau_{\text{ncc}} = 0.3$ and truncate C_i to zero by setting

$$\bar{C}_i(\boldsymbol{x}, d) = \begin{cases} 0, & \text{if } C_i(\boldsymbol{x}, d) < \tau_{\text{ncc}} \\ C_i(\boldsymbol{x}, d), & \text{otherwise} \end{cases} \tag{4}$$

This prevents $C_i(\cdot)$ from being negative and the truncation to zero will lead to a neutral behavior for its use in the regularizer as well as in the data term. For the photoconsistency measure ρ we employ the voting scheme of Hernández and Schmitt [2]

$$\rho(\boldsymbol{x}, t) = \exp\Big[-\mu \sum_{i \in \mathcal{C}'} \underbrace{\delta\big(d_i^{\max} = \text{depth}_i(\boldsymbol{x})\big) \cdot \bar{C}_i(\boldsymbol{x}, d_i^{\max})}_{\text{VOTE}_i(\boldsymbol{x})}\Big] \tag{5}$$

which accumulates votes from different cameras only in locations $\boldsymbol{x} \in V$ if the maximum quality along the ray through the center of camera i and $\boldsymbol{x}$ is found at distance $d_i^{\max} = \arg\max_d \bar{C}_i(\boldsymbol{x}, d)$. Thus, every camera ray has exactly one measurement if the corresponding matching score exceeds the threshold. Function $\text{depth}_i : V \mapsto \mathbb{R}$ returns the Euclidean distance of $\boldsymbol{x}$ to the center of camera i. We set scaling parameter to $\mu = 0.15$. Function $\rho(\cdot)$ represents a matching score of how well a small surface patch in $\boldsymbol{x}$ matches both corresponding camera images. It thus indicates probable surface locations with a low value. In the next section we explain how this information can be used for a proper modeling of the data term.

2.2. Data Term for Multi-View Reconstruction

The data term is necessary to avoid trivial solutions when minimizing Eq. (1) and replicates photoconsistency information in form of local labeling preferences. In a multi-view setup, each label of $u(x)$ depends on the labels of all points along all the camera rays passing through x. Considering these dependencies accurately generally leads to an involved non-convex optimization problem. We argue that these dependencies can be approximated by means of unary potentials f. Negative values of f favor an interior label, while positive ones an exterior label of u. The photoconsistency measure defined in the last section gives hints about probable surface locations. However, it is not directly usable to express regional affinity. Our goal is to carry the uncertainties about the surface location indicated by quality functions $C_i(\cdot)$ into the unaries f and thus into the global optimization of energy (1). We assume that the maximum-filtered NCC score at point $\boldsymbol{x}$ has the following relation to the probability that surface S passes through this point:

$$P_i(\boldsymbol{x} \in S) = 1 - \frac{1}{Z} \exp\Big[- \eta \cdot \text{VOTE}_i(\boldsymbol{x})\Big] \tag{6}$$

where Z is a normalization constant. Parameter η steers the exponential relationship between the number of cameras giving a vote, their corresponding voting qualities $\text{VOTE}_i(\boldsymbol{x})$ and the probability that the point $\boldsymbol{x}$ is part of the surface. Each camera ray may give a single vote for a probable surface location. Starting from this location and walking towards the respective camera i we follow the idea that each time we pass another probable surface location, the probability of being in the surface interior further decreases. This idea is expressed in the following equation which defines the probability of point $\boldsymbol{x}$ being in the surface interior for a reference camera i:

$$P_i\big(\boldsymbol{x} \in int(S)\big) = \prod_{j=1}^{N} \prod_{\text{depth}_i(\boldsymbol{x}) < d \leq d_i^{\max}} \Big[1 - P_j\big(r_i(\boldsymbol{x}, d) \in S\big)\Big] \tag{7}$$

[1] The temporal dependency is omitted for better readability.

The inner product integrates the surface probability votes along the ray between $\text{depth}_i(\boldsymbol{x})$ and $d_i^{\max}$ and the outer product accounts for the fact that these probabilities come from other cameras. We assume independence of individual cameras and obtain the overall probability that $\boldsymbol{x}$ is an interior point:

$$P\big(\boldsymbol{x} \in int(S)\big) = \prod_{i=1}^{N} P_i\big(\boldsymbol{x} \in int(S)\big) \tag{8}$$

Finally we define data term f in Eq. (1) as the log-probability ratio:

$$f(\boldsymbol{x}, t) = -\ln\left(\frac{1 - P\big(\boldsymbol{x} \in int(S)\big)}{P\big(\boldsymbol{x} \in int(S)\big)}\right) \quad . \tag{9}$$

Equation (7) is related to the probabilistic visibility model used by Pollard and Mundy [11, Eq.(4)]. They define the visibility $\text{vis}_i(\boldsymbol{x})$ of a point $\boldsymbol{x}$ as the probability that $\boldsymbol{x}$ is not occluded by any other point between $\boldsymbol{x}$ and the camera center:

$$\text{vis}_i(\boldsymbol{x}) = \prod_{0<d<\text{depth}_i(\boldsymbol{x})} \Big[1 - P_i\big(r_i(\boldsymbol{x}, d) \in S\big)\Big] \tag{10}$$

One could argue that $1 - \text{vis}_i(\boldsymbol{x})$ is also a good indicator for being in the surface interior. However, as long as none of the $P_i\big(\boldsymbol{x} \in S\big)$ equals exactly one, $\text{vis}_i(\boldsymbol{x})$ never reaches zero and will influence the probability of $\boldsymbol{x}$ being inside the surface far behind the camera vote. This model propagates the uncertainty that a ray from the camera center has passed a surface forward infinitely into the scene. In contrast, we propose a more conservative approach: we propagate the uncertainty of a ray-surface intersection from the local camera vote only towards the respective camera centers. This way the uncertainty is only distributed in between the camera and the location of its vote. Figure 3 illustrates the shape of these probability distributions schematically. Visually speaking, every camera vote carves its way towards the camera with its corresponding probability measure and the multiplication of all such camera bundles gives the probability of being in the surface interior. As a desirable result, this approach does not influence areas where photoconsistency information is missing. This way the data term favors the photo hull wherever photoconsistency information is missing or unreliable. Note that we do not need to assume any minimal surface thickness as it is usually done in approaches dealing with truncated signed distance functions (e.g. [17]). In contrast to the data term proposed in [8] our approach does not influence the estimates of other surfaces behind the camera vote.

3. Global Optimization

To minimize energy (1) we relax the image of function u to $[0, 1]$ and employ the preconditioned primal-dual algorithm by Pock and Chambolle [10]. Eq.(1) can be rewritten by introducing a dual variable $p : V \times T \mapsto \mathbb{R}^4$ that helps to deal with the non-differentiability of the total variation norm. The derivations follow the ones of Unger et al. [16]:

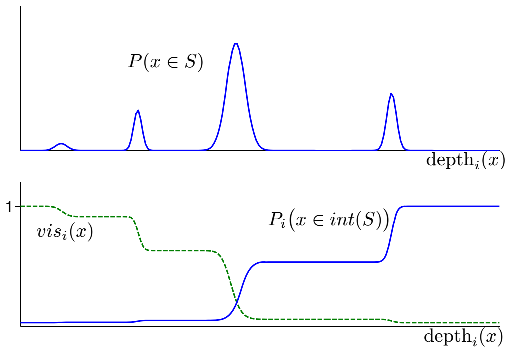


Figure 3. Schematic plots of probabilities along a camera ray. The center of camera i is in the coordinate origin. $P_i\big(\boldsymbol{x} \in int(S)\big)$ and $vis_i(\boldsymbol{x})$ multiplicatively integrate the probabilities $P(\boldsymbol{x} \in S)$ along the ray before and behind location $\boldsymbol{x}$ respectively (when looking from the camera).

$$E(u) = \max_{\|p\| \leq 1} \int_{V \times T} \langle u, -\text{div}(p) \rangle \;\, d\boldsymbol{x}dt + \lambda \int_{V \times T} fu \;\, d\boldsymbol{x}dt \tag{11}$$

This saddle point problem is optimized by means of an iterative update scheme performing a gradient ascent in the dual and a gradient descent in the primal variable:

$$\begin{aligned} p^{n+1} &= \Pi_C \left[p^n + \sigma \nabla \bar{u}^n\right] \\ u^{n+1} &= \Pi_{[0,1]} \left[u^n + \tau(\text{div}(p^{n+1}) - \lambda f)\right] \\ \bar{u}^{n+1} &= 2u^{n+1} - u^n \end{aligned} \tag{12}$$

The projection Π of u onto the unit interval $[0, 1]$ is done by thresholding. Projection onto the set $C = \{q = (q_{\boldsymbol{x}}, q_t)^T : V \times T \mapsto \mathbb{R}^4 \big| \, \|q_{\boldsymbol{x}}\| \leq 1, |q_t| \leq 1\}$ is a projection on a 4D hyperball and can be done as follows:

$$\Pi_C(q) = \left(\frac{q_{\boldsymbol{x}}}{\max(1, \frac{\|q_{\boldsymbol{x}}\|}{\rho})}, \max\big(-g_t, \min(g_t, q_t)\big)\right)^T \tag{13}$$

The step sizes σ and τ are chosen adaptively by keeping track of the corresponding operator norms as suggested in [10]. For the primal variable u we assume von Neumann boundary conditions for both spatial and temporal derivatives and corresponding Dirichlet boundary conditions for p, that is $\nabla u\Big|_{\partial(V \times T)} = 0$ and $p\Big|_{\partial(V \times T)} = 0$. The update scheme (12) provably converges to a global minimum of relaxed energy (1). The corresponding optimal binary labeling can be found by simple thresholding of the relaxed solution [10].

4. Implementation

Both the photoconsistency estimation as well as the energy optimization have been implemented on the GPU using the NVidia CUDA framework. The optimization scheme in Eq. (12) lends itself to a parallel implementation. In the result section we also briefly detail the implementation of the photoconsistency estimation.

A limiting factor of our method is memory requirement. Overall, the method needs $8|V||T| \cdot 4$ bytes, one volume for the data term and photoconsistency each, two for the primal and four volumes for the dual variable. The second primal variable is needed because of the over-relaxation step in Eq. (12). In practice memory resources are limited and smoothing over too many frames is usually not meaningful in dynamic scenes. Therefore, we limit $|T|$ to a fixed number of frames and process longer sequences with a sliding window approach for which we take the center frame of the window as the smooth solution.

5. Results

We applied our algorithm to several data sets provided by the INRIA 4D repository [6] and the free viewpoint video data sets from Tsinghua University provided by Liu et al. [9]. Both data sets also provide silhouette information which is quite useful in a sparse camera setup. We used the silhouette information provided with the data sets to speed up photoconsistency matching and optimization by restricting all computations to the interior of the visual hull. In some frames the silhouettes are incorrect and lead to missing scene parts in some experiments. All experiments have been computed on a Intel Xeon E5520 PC with 12GB RAM, equipped with an NVidia Tesla C2070 card and running a recent Linux distribution.

Given the relaxed solution of energy (1) we extracted an isosurface at $u = 0.5$ with the Marching Cubes algorithm. To better see the jittering reduction all experiments show pure results of our algorithm after Marching Cubes without any mesh smoothing, filtering or remeshing. The following section details the photoconsistency and data term computation to explain differences and compare to previous work.

5.1. Photoconsistency and Data Term Evaluation

As explained in Section 2.2 the data term is built based on the photoconsistency measure ρ. The quality of this measure directly influences the quality of the data term. Kolev et al. [8] iteratively improved the quality of the photoconsistency by calculating the NCC scores based on a surface normal estimate which they first take from the visual hull and later update with the solution of the surface reconstruction in an iterative manner. In the photoconsistency voting scheme as described in [8] each point $\boldsymbol{x}$ defines a ray to each camera. Point $\boldsymbol{x}$ only gets a vote if the normal corresponding to $\boldsymbol{x}$ maximizes the NCC along the whole ray in point $\boldsymbol{x}$. This means that for every point $\boldsymbol{x}$ the photoconsistency has to be calculated for all points on the corresponding camera rays with respect to the same normal. This makes the photoconsistency estimation inherently slow and explains the long (up to 10 hours for one scene) computation times reported in [8]. In our 4D setup we dropped this dependency by maximizing the photoconsistencies along rays independent of the normal direction. This way the photoconsistency calculations can be done independently and thus easily be parallelized to speed up computations. We simply use the viewing direction of the reference camera towards $\boldsymbol{x}$ as the surface normal estimate. We compared the results with our reimplementation of the normal dependent maximization and experienced fairly similar results. Figure 4 shows exemplarily results for these different photoconsistency estimation schemes. As result, we experienced speedups of one or several orders of magnitude (depending on the volume resolution) for getting comparable results.

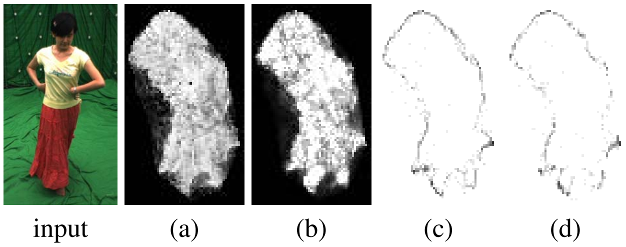
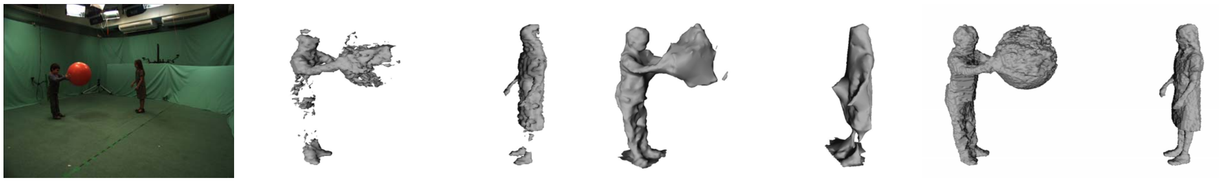


Figure 4. Comparison of data term from [8] (a) and the proposed one (b) for a lower cross section of the skirt. Shown are the voxels' probability of being inside (white) and outside (black) the surface. Corresponding photoconsistencies are respectively displayed in (c) and (d). Dark pixels represent higher matching scores. Although the photoconsistency is slightly worse, the proposed data term yields sharper contours and better carves out concavities because only front facing cameras determine their shape, rather than all cameras. The volume resolution was 128x256x192.

On the left part of Fig. 4 we compare the proposed data term with the one in [8]. We briefly repeat its definition to clarify the differences. They also define a quality measure for each camera ray defined by point $\boldsymbol{x}$ and camera j:

$$\begin{aligned}\rho^j_{int}(\boldsymbol{x}) =& H(d_i^{\max} - d) \cdot (1 - f(\bar{C}_i(\boldsymbol{x}, d))) \\ &+ (1 - H(d_i^{\max} - d)) \cdot f(\bar{C}_i(\boldsymbol{x}, d))\end{aligned} \quad (14)$$

H is the Heavyside step function switching between two different costs depending on whether d is larger or smaller than $d_i^{\max}$, i.e. if the point $\boldsymbol{x}$ is either before or behind the voting location. The data term is then defined as an average of $\rho^j_{int}(\cdot)$ over all cameras. The key difference to our proposed approach is the fact that this model influences the data term before *and* behind the camera vote while the proposed approach only influences the data term in between the camera and the camera vote. This global influence degrades the quality of back faces and other object parts which



1/16 input images Kolev et al. [8] Jancosek and Pajdla [7] proposed

Figure 5. Comparison of the reconstruction results using the data term by Kolev et al. [8] and the proposed one. Further we show the result of the method by Jancosek and Pajdla [7]. The ball has low texture information and further exhibits strong reflections which makes it difficult to reconstruct.

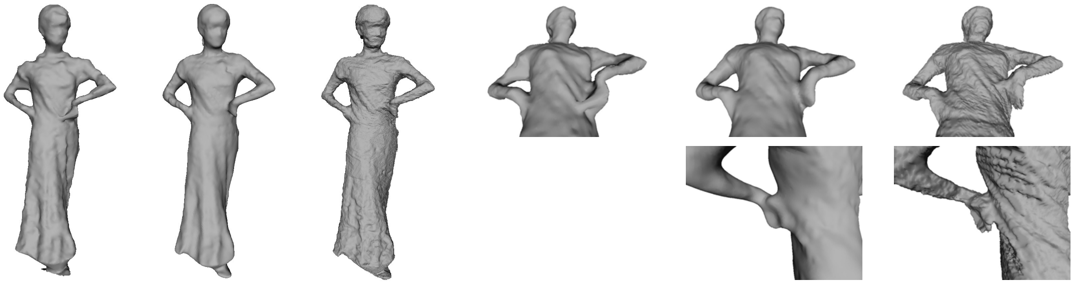


Figure 6. Comparison of the proposed method for $|T| = 1$ with other 3D reconstruction methods. Respectively from left to right (twice): Jancosek and Pajdla [7], Liu et al. [9] and the proposed method. The approach by Jancosek and Pajdla wrongly connects points at the hand and the armpits. Our approach method better preserves several details like the hand.

are unrelated to the camera vote. This is visible in Fig. 4 showing the differences in the data term, as well as in Fig. 5 which depicts a resulting surface reconstruction. For comparison we also show the reconstruction result of Jancosek and Pajdla [7]. The scenes with the gymnastic ball are especially challenging because the ball surface has low texture information and a shiny surface. In Fig. 6 we compared the output of our method with the methods by Jancosek and Pajdla [7] and to the ones of Liu et al. [9] who provided the data. Both methods yield much smoother surface reconstructions, but also blur fine scale details like the hand. Table 1 lists average computation times for the experiments depicted in Fig. 9.

data set	volume size	pc+d	opt
kick one	384^3	89	28/93/-
cartwheel	$384 \times 384 \times 256$	21	18/59/-
playing	$384 \times 384 \times 256$	18	18/60/-
adult child	384^3	43	31/91/-
red skirt	256^3	90	10/31/88

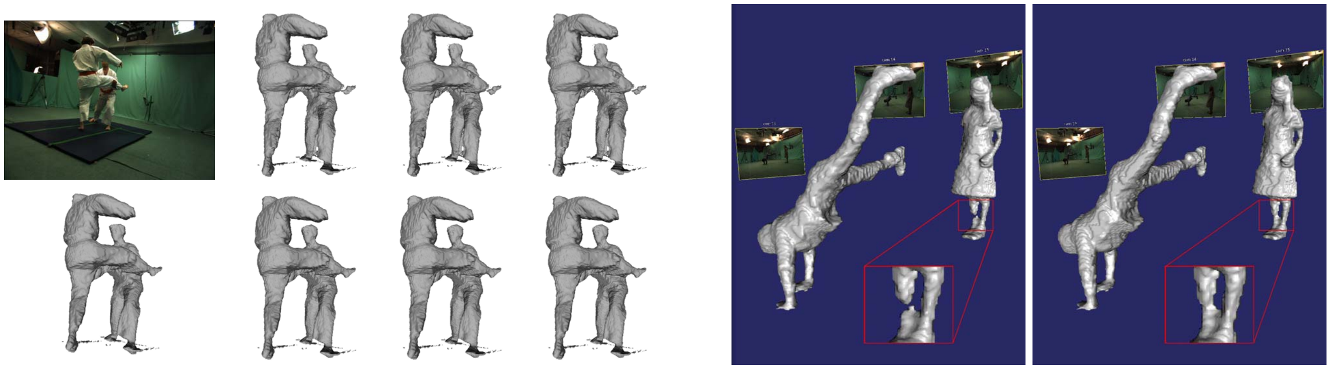
Table 1. Average runtimes per frame for our method on different data sets for the photoconsistency and data term estimation (pc+d) and the surface optimization (opt) for different sizes of $|T| \in \{1, 3, 5\}$. Timings are in seconds. In comparison the method by Jancosek and Pajdla [7] computed 10-20 min/frame.

5.2. Temporal Regularization

For evaluation we studied the influence of the temporal window size $|T|$ and weighting $g_t = \exp(-a|\nabla_t f|^b)$ in Eq. (2). Fig. 7 gives an overview for $|T| \in \{3, 5, 7\}$ (horizontal) and different $a \in \{0.001, 1\}$ (left, vertical). The effect of g_t on the solution is mainly governed by parameter a. When a approaches zero the temporal regularization gets maximal and the reconstructed surface tends towards the intersection with neighboring time slices (see the disappearance of the lower leg part in Fig. 7, top row). We could not experience significant visible differences for varying values of b and set $b = 1$ in all experiments. The differences are largest between window sizes $|T| = 1$ and $|T| = 3$. Choosing larger window sizes only led to subtle differences which do not pay off the increase in computation time and memory resources. Since no other 4D reconstruction implementations are publicly available and it is difficult to obtain ground truth geometry, we visually compare our method with (a) time-independent reconstruction by Jancosek and Pajdla [7], (b) time-independent reconstruction as proposed with $|T| = 1$, (c) temporal Gaussian smoothing of (b) as post processing for temporal smoothness, and (d) the proposed method with $|T| = 3$. In particular, we compute a smoothed occupancy labeling $\bar{u}$ from the time-independent result $\hat{u}$ as follows:

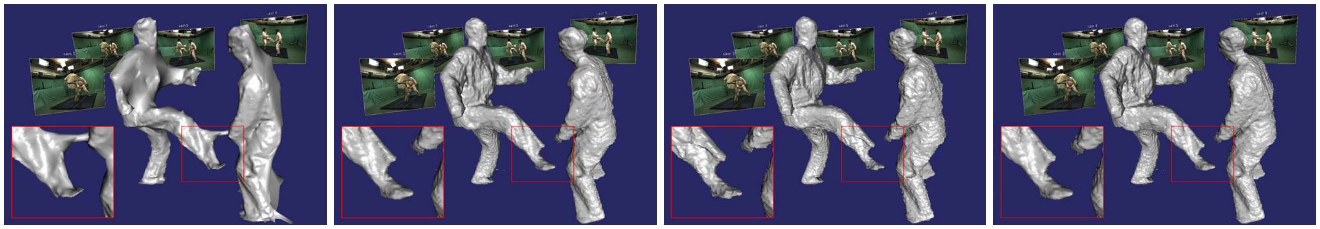
$$\bar{u}(x,t) = \frac{1}{Z} \sum_{i=0}^{|T|-1} \exp\left[-\frac{(i-|T|/2)^2}{2\sigma^2}\right] \hat{u}(x, t+i-|T|/2) \tag{15}$$

Fig. 8 shows a representative frame for each method. Generally, the Gaussian filtering cannot reach the same level of smoothness as (d) while preserving fast moving object parts. For preserving fast movements σ needs to be chosen very small such that voxel jittering is barely reduced. The proposed method balances these issues much better.



$|T| = 1$ $|T| = 3$ $|T| = 5$ $|T| = 7$ $|T| = 1$ $|T| = 3, a = 0.5$

Figure 7. Effect of the temporal regularization. The approach allows to impose temporal regularity over multiple time steps $|T|$. For a small weight of temporal smoothness ($a = 1$, left bottom row) the regularity reduces the jittering of voxels over time (see supplementary video), where as for strong temporal smoothness ($a = 0.001$, left top row) the regularization starts to deteriorate fast moving structures like the right foot. Temporal coherence also improves reconstructions with weak photoconsistencies in single time frames (right).



(a) Jancosek and Pajdla [7] (b) time-indep. ($|T| = 1$) (c) temp. filtering ($|T| = 3$) (d) proposed ($|T| = 3$)

Figure 8. Comparison of different reconstruction techniques. (a) produces strong surface jittering, wrongly connects the leg and hand and misses parts of the head. (b) Voxel jittering is visible. (c) Voxel jittering can be reduced, but fast moving object parts start disappearing, e.g. the foot. The edge on the lower leg is an artifact of the averaging of consecutive time frames. (d) Due to the weighting and the TV-regularization the problems of (c) can be balanced much better (see also supplementary video).

6. Conclusion

We presented a novel approach to space time multi-view 3D reconstruction that generalizes several previous works into a 4D setting. In order to get competitive reconstructions on wide-baseline camera setups we further proposed a novel data term that better preserves concavities and fine details. 3D reconstruction results compare favorably to other works. Our approach directly accounts for temporal surface coherence within the reconstruction process. In comparison to single frame-by-frame reconstruction our approach clearly reduces the amount of noise on the estimated surface. In several experiments we showed the viability of the proposed framework. To our knowledge, this is the first time that space-time 3D reconstruction was formulated as a convex variational problem. The solutions are provably optimal, independent of initialization and recover fine details.

References

[1] E. Aganj, J.-P. Pons, F. Sgonne, and R. Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *ICCV*, pages 1–8. IEEE, 2007. 2

[2] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *CVIU*, 96(3):367–392, Dec. 2004. 3

[3] B. Goldluecke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE TPAMI*, 29(7):1194–1208, July 2007. 1

[4] B. Goldluecke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *CVPR*, volume I, pages 350–355, July 2004. 1

[5] J.-Y. Guillemaut and A. Hilton. Space-time joint multi-layer segmentation and depth estimation. In *3DIMPVT*, pages 440–447, 2012. 2

[6] Institut national de recherche en informatique et en automatique (INRIA) Rhône Alpes. 4d repository. http://4drepository.inrialpes.fr/. 5

[7] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR*, pages 3121–3128, 2011. 2, 6, 7

[8] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *IJCV*, 84(1):80–96, August 2009. 1, 2, 4, 5, 6

[9] Y. Liu, Q. Dai, and W. Xu. A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):407–418, May 2010. 5, 6

[10] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *ICCV*, pages 1762–1769, Washington, DC, USA, 2011. 4

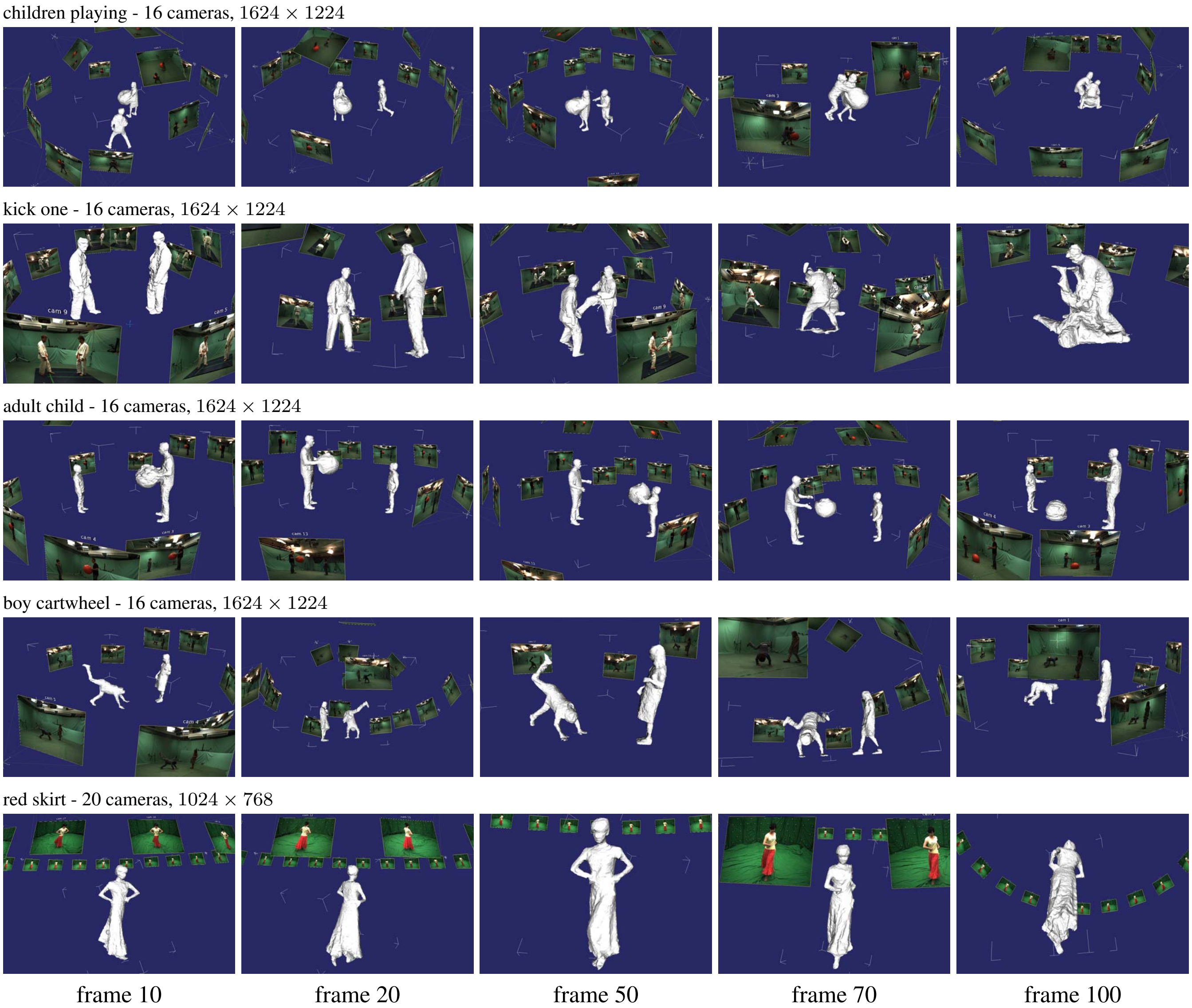

children playing - 16 cameras, 1624 × 1224
kick one - 16 cameras, 1624 × 1224
adult child - 16 cameras, 1624 × 1224
boy cartwheel - 16 cameras, 1624 × 1224
red skirt - 20 cameras, 1024 × 768
frame 10
frame 20
frame 50
frame 70
frame 100

Figure 9. Results of our framework on several data sets for $|T| = 3$. For the cartwheel sequence we selected different frame numbers (120,130,222,347,442) and for red skirt (41,45,50,55,58). Please refer to the supplementary material for video sequences.

[11] T. Pollard and J. L. Mundy. Change detection in a 3-d world. In *CVPR*, pages 1 – 6, Minneapolis, USA, 2007. IEEE. 4

[12] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2), May 2012. 2

[13] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society. 1

[14] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. In *ACM SIGGRAPH Asia 2008 papers*, pages 110:1–110:10, New York, NY, USA, 2008. ACM. 2

[15] B. Ummenhofer and T. Brox. Dense 3d reconstruction with a hand-held camera. In *DAGM/OAGM Symposium'12*, pages 103–112, 2012. 2

[16] M. Unger, T. Mauthner, T. Pock, and H. Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted tv. In *EMMCVPR*, pages 193–206, Berlin, Heidelberg, 2009. Springer-Verlag. 1, 2, 4

[17] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *ICCV*, pages 1–8, 2007. 2, 4

[18] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *CVPR*, pages 367–374, June 2003. 1