A Self-regulating Spatio-Temporal Filter for Volumetric Video Point Clouds

Matthew Moynihan[1](✉), Rafael Pagés[1,2], and Aljosa Smolic[1]

[1] V-SENSE, Trinity College Dublin, Dublin, Ireland
{mamoynih,smolica}@tcd.ie
[2] Volograms, Dublin, Ireland
rafa@volograms.com
https://v-sense.scss.tcd.ie/people/
http://www.volograms.com

Abstract. The following work presents a self-regulating filter that is capable of performing accurate upsampling of dynamic point cloud data sequences captured using wide-baseline multi-view camera setups. This is achieved by using two-way temporal projection of edge-aware upsampled point clouds while imposing coherence and noise filtering via a windowed, self-regulating noise filter. We use a state of the art Spatio-Temporal Edge-Aware scene flow estimation to accurately model the motion of points across a sequence and then, leveraging the spatio-temporal inconsistency of unstructured noise, we perform a weighted Hausdorff distance-based noise filter over a given window. Our results demonstrate that this approach produces temporally coherent, upsampled point clouds while mitigating both additive and unstructured noise. In addition to filtering noise, the algorithm is able to greatly reduce intermittent loss of pertinent geometry. The system performs well in dynamic real world scenarios with both stationary and non-stationary cameras as well as synthetically rendered environments for baseline study.

Keywords: Point clouds · Upsampling · Temporal coherence · Free viewpoint video · Multiview video · Volumetric video

1 Spatio-Temporal Coherence in Volumetric Video

As the popularity of VR and AR consumer devices continues to grow, we can naturally expect an increase in the demand for engaging and aesthetic mixed reality content. The barrier to entry for creative enthusiasts and content creators has begun to decline as more digital frameworks supporting VR/AR become available, however, performance capture and reconstruction of real-world scenes still remains largely out of reach for amateur productions.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant No. 15/RP/2776. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

A. P. Cláudio et al. (Eds.): VISIGRAPP 2019, CCIS 1182, pp. 391–408, 2020.
https://doi.org/10.1007/978-3-030-41590-7_16

Using Free-Viewpoint Video (FVV) or, more specifically Volumetric Video (VV), content creators have the technology to record and reconstruct performances in dynamic real-world scenarios. However, these captures are often restricted by the constraints of highly controlled studio environments, requiring dense arrays of high-resolution RGB cameras and IR depth sensors [5,20]. The reconstruction is usually done in a frame-by-frame manner beginning with the construction of a dense point cloud via Multi-View Stereo (MVS). For such high-budget studios with very dense coverage of the subject, temporal inconsistencies in the resulting point cloud may be visually negligible after the final meshing and tracking process. Yet where low-budget content creation is concerned, such high density coverage may not be achievable and thus any spatio-temporal inconsistencies can become magnified and visually unappealing by the end of the reconstruction pipeline.

In order to address to the demand for low-cost VV and performance capture, new systems have been proposed which enable VV content creation solely on consumer-grade RGB cameras and even hand-held personal devices [30]. However, any such system which features framewise reconstruction [27] will contain spatio-temporal artifacts in the resulting volumetric sequence. This is usually a consequence of some inherent fail cases for photogrammetry-based techniques whereby the subject can contain highly reflective surfaces or a lack of textured material.

Without accounting for spatio-temporal variance, otherwise pertinent geometric features become distorted and inconsistent across VV sequences. This is especially true in the case of small or thin details such as hands or arms. An example of which can be seen in Fig. 1 where the naive frame-by-frame reconstruction fails to distinguish temporally persistent features as portions of limbs lack persistence. Furthermore large sections of geometry in relatively untextured areas may be intermittently present depending on the success of the point cloud reconstruction for that given frame. Structured noise patches can also be intermittently observed.

The proposed system is an expansion to the work presented in [24] that is able to spatio-temporally upsample a point cloud sequence captured via wide-baseline multi-view setups and further support the self-regulating noise filter metric using a new windowed approach to sampling. In summary, the following work proposes:

- A spatio-temporally coherent point cloud sequence upsampling algorithm that selectively merges point cloud projections within a variable window. The projections of which are computed iteratively using a pseudo-scene flow estimate.
- An autonomously regulated noise filter supported by a density-weighted energy term for averaging within a window of frames.

We perform a baseline comparison on the work presented in [24] as well as previous examples.

2 Previous Work

One of the fundamental processes in modern VV pipelines is spatio-temporal consistency. Ensuring this consistency across the sequence of 3D models helps reduce the impact of small geometry differences among frames and surface artifacts, which result in temporal flickering when rendering the VV sequence. Most techniques apply a variation of on the non-rigid ICP algorithm [19,38], such as the coherent drift point method [29], performing a geometric temporal constraint to align the meshes resulting from the 3D reconstruction process on a frame-by-frame basis [14,17]. This works specially well when the 3D models acquired for every frame are detailed and accurate, as registration algorithms are not always robust to big geometry differences or loss of portions of the mesh (something that can often happen for human limbs). A good example of this is the system by Collet et al. [5]: they apply mesh tracking in the final processing stage, both to provide a smoother VV sequence and also to improve data storage efficiency as, between keyframes, only the vertex positions vary while face indices and texture coordinates remain the same. They achieve very appealing results by utilizing a sophisticated, very dense camera setup of over 100 sensors (RGB and IR), ensuring a high degree of accuracy for the reconstructed point clouds on a frame-to-frame basis. This type of temporal consistency is also key in the methods proposed by Dou et al. [7,8], where they are able to perform registration in real-time, using data coming from depth sensors. These methods ensure temporal consistency at the end of their pipeline, but differently to the method proposed, they do not address the loss of geometry in the capture stage, which can only be solved using temporal coherence at the point cloud generation.

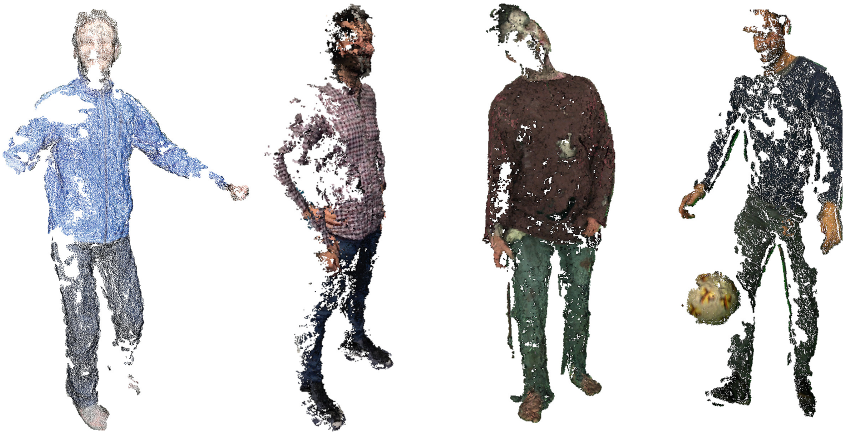

Fig. 1. Input dense point clouds generated using an affordable volumetric video capturing platform [30]. Even after densification via multi-view stereo, the input clouds still exhibit large gaps in structure as well as patches of noise.

Mustafa et al. [25] ensure temporal consistency of their VV sequences by first, using sparse temporal dynamic feature tracking as an initial stage, followed by a shape constraint based on geodesic star convexity for the dense model. These temporal features are used to initialize a constraint which refines the alpha masks used in visual-hull carving and are not directly applied to the input point cloud. The accuracy of their results is not comparable with the methods mentioned above, but they show good performance with a reduced number of viewpoints and wide baseline. Mustafa et al. extended their work to include sequences that are not only temporally but also semantically coherent [26], and even light-field video [28].

An interesting way of pursuing spatio-temporal consistency is by using optical flow. For example, Prada et al. [31] use mesh-based optical flow for adjusting the tracking drift when generating texture atlases for the VV sequence, adding an extra layer of spatio-temporal consistency at the texturing step. It is possible to address temporal coherence by trying to use the scene flow to recover not only motion, but also depth. Examples of this are the works by Basha et al. [2] and Wedel et al. [35]. These techniques require a very dense and accurate motion estimation for every pixel to acquire accurate depth maps, together with a camera setup with a very narrow baseline. Alternatively, our system uses the temporally consistent flow proposed by Lang et al. [18] applied to multi-view sequences, allowing us to track dense point clouds across the sequence even with a wide baseline cameras configurations.

Other ways of improving incomplete 3D reconstructions, such as the ones acquired with wide baseline camera setups, include upsampling or densifying [15,36,37] them in a spatially coherent way. These systems are designed to perform upsampling for a single input point cloud, and not specifically a VV sequence, so they are unable to leverage any of the temporal information within a given sequence of point clouds. As a result, the use of such techniques alone will still suffer from temporally incoherent errors. Our system takes advantage of the geometric accuracy of the state of the art Edge-Aware Point Set Resampling technique proposed by Huang et al. [15] and supports it using the temporal information obtained from the inferred 3D scene flow along with some spatio-temporal noise filtering. The reasoning behind this approach being that increasing the density of coherent points improves the accuracy of surface reconstruction algorithms such as Poisson Surface Reconstruction [16] and thus, propagates visual improvement through th VV pipeline.

3 Proposed System

3.1 Point Cloud Generation and Upsampling

We use a low-cost VV pipeline similar to the system by [30] to generate the input clouds for the proposed algorithm. Such pipelines generally maximise the baseline between cameras in order to reduce the cost of extra hardware while still providing full coverage of the subject. The camera intrinsics are assumed

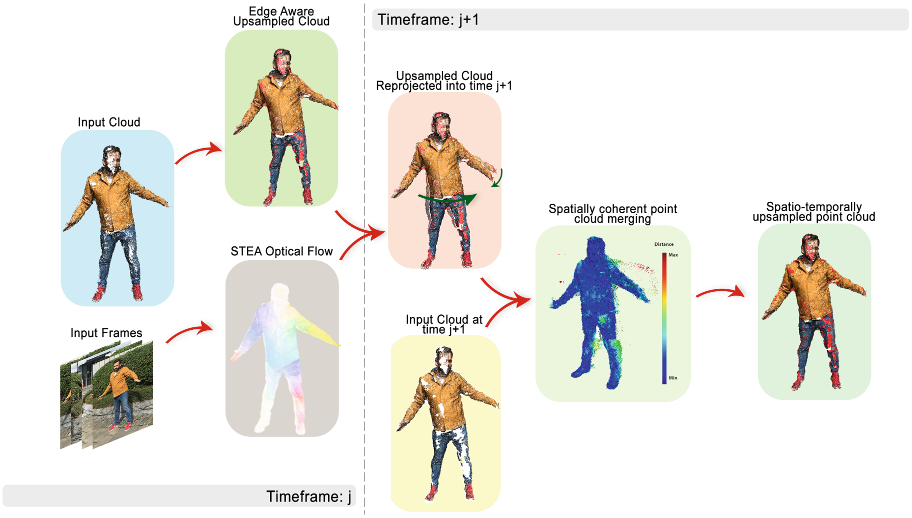
Edge Aware Upsampled Cloud
Timeframe: j+1
Upsampled Cloud Reprojected into time j+1
Input Cloud
Spatially coherent point cloud merging
Distance
Max
Min
Spatio-temporally upsampled point cloud
STEA Optical Flow
Input Frames
Input Cloud at time j+1
Timeframe: j

Fig. 2. Proposed pipeline: The input to the algorithm requires a sequence of temporally independent point clouds along with the corresponding RGB images and calibration data. At timeframe j, the input cloud is upsampled and projected into the subsequent frame $t + 1$. This is done via an edge-aware scene flow generated from the input RGB images. Expanding on [24], this is performed iteratively across a window of frames centered about the input frame i.e. we recursively project frames within the given window toward the center frame. The output consists of a spatio-temporally coherent merge and averaging system which upsamples the input point clouds and filters against temporal noise.

to be known from prior calibration while extrinsics can be calculated automatically using sparse feature matching and incremental structure-from-motion [23]. In some cases the cameras may be handheld, whereby more advanced techniques like CoSLAM [39] can be applied to better produce dynamic poses. The input sparse clouds are further densified using multi-view stereo. The examples presented within the context of our system use the sparse point cloud estimation system by Berjón et al. [3] and are then further densified by using the unstructured MVS system of Schönberger et al. [34]. Formally, we define $S = \{s_{i=1}, ..., s_m\}$ as the set of all m video sequences, where $s_i(j)$, $j \in \{1, ..., J\}$ denotes the jth frame of a video sequence $s_i \in S$, with J frames. Then for every frame j, there will be an estimated point cloud $\mathcal{P}_j$. In a single iteration, $\mathcal{P}_j$ is taken as the input cloud which is upsampled using Edge-Aware Resampling (EAR) [15]. This initializes the geometry recovery process with a densified point cloud prior which will be temporally projected into the next time frame $j + 1$ and geometrically filtered to ensure both temporal and spatial coherence. With the windowed filtering approach this iteration is performed recursively in such a way that each frame within the window is iteratively projected toward the center frame via it's respective intermediate frames (Fig. 2).

3.2 Spatio-Temporal Edge-Aware Scene Flow

Accurately projecting geometry from between different timeframes is directly dependent on the accuracy of the scene flow used to achieve it. In the context of this paper the scene flow used is actually a dense, pseudo-scene flow which is generated from multi-view videos as opposed to directly extracting it from the clouds themselves. This scene flow is calculated as an extension to dense 2D flow, thus, for every sequence s_i we compute its corresponding scene flow f_i. This view-independent approach ensures that the system is robust to wide baseline input.

To retain edge-aware accuracy and reduce additive noise we have chosen a dense optical flow pipeline that guarantees spatio-temporal accuracy:

- Initial dense optical flow is calculated from the RGB input frames using the Coarse to fine Patch Match (CPM) approach described in [13].
- The dense optical flow is then refined using a spatio-temporal edge aware filter based on the Domain Transform [18].

The CPM optical flow is used to initialize a spatio-temporal edge aware (STEA) filter which regularizes the flow across a video sequence, further improving edge-preservation and noise reduction.

While the STEA can be initialized with most dense optical flow techniques such as the popular Gunnar-Farnebäck algorithm [9], the proposed system uses the coarse-to-fine patch match algorithm by [13] as recommended in [33]. Table 1 provides a breakdown of the amount of pertinent geometry recovered via different optical flow techniques.

Table 1. Investigation by [24] on the effect of STEA filter initialization on geometry recovered expressed as % increase in points. Tested on a synthetic ground-truth sequence. Flow algorithms tested: Coarse-to-Fine Patch Match (CPM) [13], Fast Edge-Preserving Patch Match (FEPPM) [1], Pyramidial Lukas-Kanade (PyLK) [4] and Gunnar-Farnebäck (FB) [9].

STEA initialization	Area increase (%)
CPM	**37.73**
FEPPM	34.9
PyLK	34.77
FB	29.7

The STEA filter consists of the following implementation as in [18]. This implementation further builds upon the Domain Transform [11] extending into the spatial and temporal domains given the optical flow as the target application:

1. The filter is initialized as in [33], using coarse-to-fine patch match [13]. The CPM algorithm estimates optical flow as a quasi-dense nearest neighbour field (NNF) using a subsampled grid.

2. The edges of the RGB input are then calculated using the Structure Edge Detection Toolbox [6].
3. Using the calculated edges, the dense optical flow is then interpolated using Edge-Preserving Interpolation of Correspondences [32].

This dense optical flow field is then regulated by the STEA filter via multiple spatio-temporal domain iterations to reduce temporal noise. Figure 3 visualizes the intermediate stages of the flow processing pipeline.

3.3 Scene Flow Point Projection

Given known per-camera intrinsics ($C_{j_1}, ..., C_{j_m}$, at timeframe j), the set of scene flows ($f_{j_1}, ..., f_{j_m}$), and the set of point clouds ($\mathcal{P}_j, ..., \mathcal{P}_J$), the motion of any given point across a sequence can be estimated. To achieve this, each point is back-projected $\mathbf{P}_k \in \mathcal{P}_j$ to each 2D flow f_i at that specific frame j. We check the sign of the dot product between the camera pointing vector and the normal of the point $\mathbf{P}_k$ to prune any point projections which may otherwise have been occluded for the given view. Using the flow, we can predict the position of the back-projected 2D points $\mathbf{p}_{ik}$ in sequential frames, $\mathbf{p}'_{ik}$.

The set of projected 3D points $\mathcal{P}'_j$, at frame $j+1$, is then acquired by triangulating the flow-projected 2D points $\mathbf{p}'_{ik}$, using the camera parameters of frame $j+1$. This is done by solving a set of overdetermined homogeneous systems in the form of $H\mathbf{P}'_k = \mathbf{0}$, where $\mathbf{P}'_k$ is the estimated 3D point and matrix H is

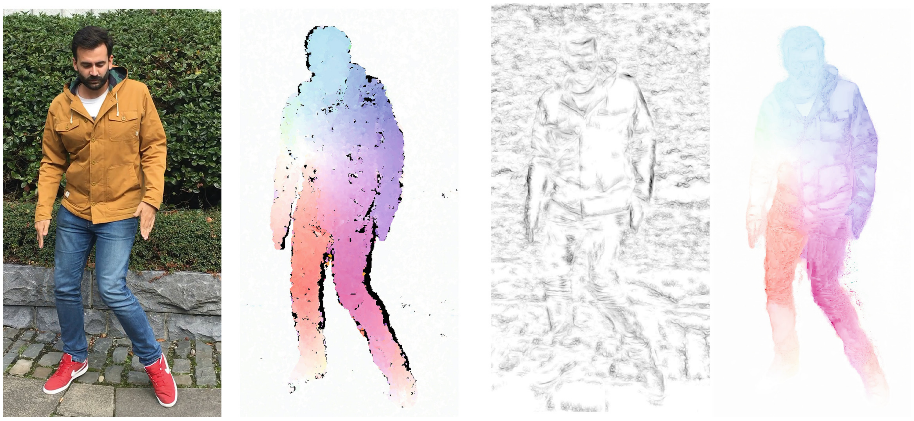

Fig. 3. From left to right, dense optical flow calculation: For a particular viewpoint, the input RGB image, (1) nearest neighbour field estimate from CPM, (2) SED detected edges, (3) interpolated dense STEA output. Conventional colour coding has been used to illustrate the orientation and intensity of the optical flow vectors. Orientation is indicated by means of hue while vector magnitude is proportional to the saturation i.e. negligible motion is represented by white, high-speed motion is shown in highly saturated color [24].

defined by the Direct Linear Transformation algorithm [12]. The reprojection error is minimized using a Gauss-Markov weighted non-linear optimisation [22].

3.4 Windowed Hausdorff Filter

The aforementioned point cloud projection framework can now be used to support the coherent merging and noise filtering process. For a given window of width w for frames $\left\{j_{(c-w/2)}...j_c...j_{(c+w/2)}\right\} \subset J$ where c is the center frame, we project the point cloud at each frame towards the center frame using the above method in a recursive manner. In this way structural information is retained and propagated. However, this also has the effect of accumulating any inherent noise within this window. For this reason we extend the two-way Hausdorf filter in [24] with the addition of an energy density term E_{dens}. This density term takes into account the average voxel density of the merged window of frames which is essentially the sum of the propagated clouds. Using density as a conditioning term leverages the temporal inconsistency in that statistically, occupancy due to noise is far less common than occupancy due to pertinent geometry.

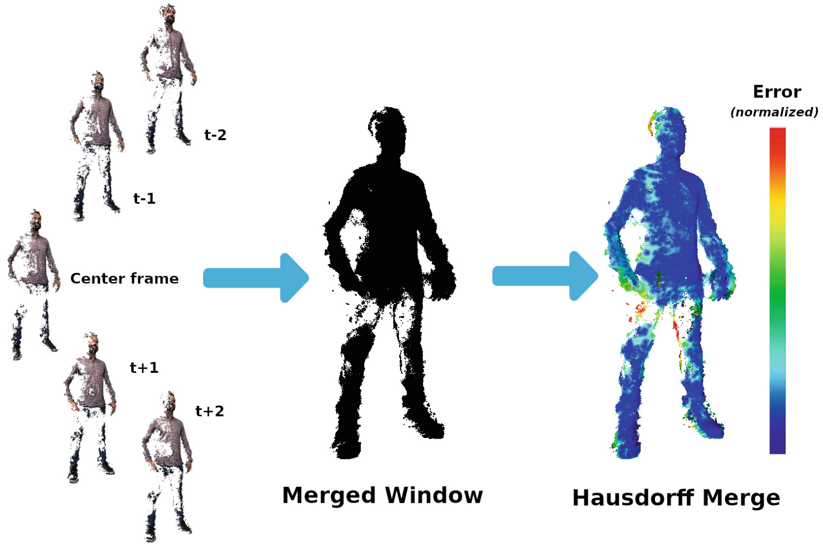

Fig. 4. The windowed merge process. Left: a 5-frame window of input clouds, Middle: the cumulative merge of the upsampled and projected input clouds. Right: the filtered merge process visualized with normalized error given by distance of each point to it's corresponding match in the input cloud. This error term is then augmented with the energy terms $E_{dynamic}$ and E_{dens}.

The coherent merged cloud $\mathcal{P}^*_{j+1}$ is given by the logical definition in Eq. 1 where $D_{\mathcal{P}'_j}$ is the summed result of projecting all point clouds within window w recursively toward the center frame j.

Given an ordered array of values $D_{\mathcal{P}'_j}$ such that $D_{\mathcal{P}'_j(k)}$ is the distance from point $\mathcal{P}_j(k)'$ to its indexed match in $\mathcal{P}_{j+1}$. We also define $D_{\mathcal{P}_{j+1}}$ as an array of distances in the direction of $\mathcal{P}_{j+1}$ to $\mathcal{P}'_j$. We then define the merged cloud to be the union of two subsets $M \subset \mathcal{P}'_j$ and $T \subset \mathcal{P}_{j+1}$ such that,

$$\begin{aligned} M \subset \mathcal{P}'_j \; \forall \; \mathcal{P}'_j(k) \; : \; D_{\mathcal{P}'_j(k)} < \; d_j, \; k \in \{1...j\}\,, \\ T \subset \mathcal{P}_{j+1} \; \forall \; \mathcal{P}_{j+1}(k) \; : \; D_{\mathcal{P}_{j+1}}(k) < \; d_j, \; k \in \{1...j\}\,, \\ \mathcal{P}^*_{j+1} = M \cup T \end{aligned} \tag{1}$$

By this definition, $\mathcal{P}^*_{j+1}$ contains only the points in $\mathcal{P}_{j+1}$ and $\mathcal{P}'_j$ whose distance to their nearest neighbour in the other point cloud is less than the computed threshold d_j. The intention of this design is effectively to remove any large outliers and incoherent points while encouraging consistent and improved point density. Figure 4 shows an example of how the coherent merge works.

3.5 Dynamic Motion Energy Term

Due to the distance-based nature of the Hausdorff-based filter, it is often observed that fast-moving objects are pruned after being projected into the next frame. This approach to filtering greatly reduces the amount of temporally inconsistent noise, but simultaneously, it over-filters dynamic objects due to the lack of spatial overlap between frames. This is especially true for sequences captured at 30 fps or less, which is often the case for affordable VV setups where bandwidth and storage are concerned. To address this issue, we supplement the distance-based threshold term with a dynamic motion energy which is designed to add bias towards fast-moving objects. This energy term is proportional to the average motion observed across the scene-flow estimates for a given timeframe. For faster-moving objects, higher confidence is assigned to clusters of fast-moving points. Given that $\mathcal{P}'_j$ is a prediction for frame $j+1$, we validate each predicted point by back-projecting $\mathcal{P}'_j$ into the respective scene flow frames for time $j+1$. The flow values for the pixels in each view is then averages to calculate the motion for a given pixel at that time. As in Sect. 3.3 we again filter out occluded points using the dot product of the camera pointing vector and the point normal.

3.6 Spatio-Temporal Density Term

The proposed system offers an expansion to the two-way Hausdorff-based filter presented in [24] by sampling a window of frames about the current timestamp. While the two-way filter is robust to temporal noise it isn't capable of recovering large sections of missing geometry over a spanning timeframe. As illustrated in Fig. 6, the two-way approach fails to recover much from the sequence where large patches are missing over a longer time period. To address this, the proposed system introduces a windowed approach which combines the projected information from multiple frames while retaining comparable noise filtering. In order to reduce the added noise we propose an additional energy term for the filtering threshold based on patio-temporal density within the given window. The new threshold score criteria is then given by:

$$E_{th} = d - (E_{dens} + E_{dynamic}) \tag{2}$$

The E_{dens} term is calculated as follows:

- For a window of width w we iteratively project each frame into the current timestamp such that a single point cloud object is created consisting of the points projected from the frame range $\left\{t_{(c-w/2)}...t_c...t_{(c+w/2)}\right\}$
- An octree-based occupancy grid is then constructed on this object where each leaf is assigned a normalized density score. This score is the E_{dens} term for any point given its index within the occupancy grid.

Figure 4 illustrates this process for any given window. The size of this window is variable but is limited by practical limitations of computation time and the trade-off of adding multiple sources of noise. For our purposes we concluded that a window size of 5 was within practical time constraints while still providing good results. As with any filtering or averaging algorithm, there is an inherent risk of over-smoothing data and thus, such decisions may differ for various sequences depending on the degree of dynamic motion.

4 Results

In Fig. 5 we demonstrate a side-by-side comparison of the process results vs unprocessed input for two challenging yet conventional scenarios. We evaluate the system on a number of sequences captured outdoors with as little as 6 to 12 handheld devices (i.e. smartphones, tablets etc.) as well as a controlled green screen environment comprised of 12 high-end, rigidly mounted cameras (6 4K resolution, 6 Full HD). A ground-truth comparison is also presented by comparing reconstruction results against a known synthetic model within a virtual environment with rendered cameras.

4.1 Outdoor Handheld Camera Sequences

Shooting outdoors with heterogenous handheld devices can present a number of challenging factors including: non-uniform dynamic backgrounds, increased margin of error for intrinsics and extrinsics calculations, instability of automatic foreground segmentation methods and more. The cumulative effect of these factors results in temporal inconsistencies with the reconstructed point cloud sequence as well as the addition of structured noise and omission of pertinent geometry. Figure 5 (left model) shows the difference between using framewise reconstruction (a) and the proposed system (b). A significant portion of structured noise has been removed whilst also managing to fill-in gaps in the subject.

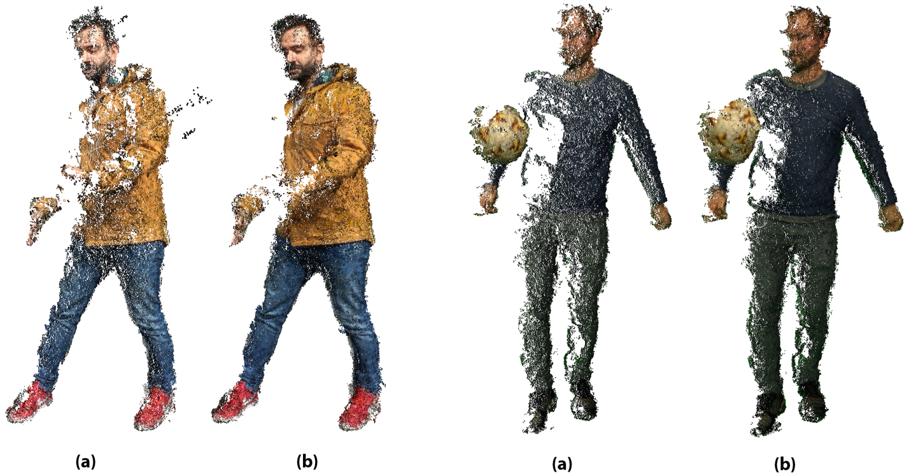

Fig. 5. An example of the proposed upsampling and filtering system. Pictured left: a sequence captured outdoors with handheld devices. Pictured right: a sequence captured in a low-cost controlled studio environment with fast-moving objects. For both sequences, (a) corresponds to the input cloud prior to filtering while (b) represents the upsampled and filtered result [24].

To further demonstrate the impact of our system targeting volumetric reconstruction, we present the effect of applying screened poisson surface reconstruction (PSR) [16] to the input point cloud. In general, the direct application of PSR creates a fully closed surface which usually creates bulging or "inflated-looking" surface meshes. Instead we use the input cloud to prune outlying faces from the PSR mesh such that the output surface mesh more accurately represents the captured data. Thus, in Fig. 6 the gaps in the input data can be visualized clearly. This figure also shows the appreciable increase in pertinent surface area after spatio-temporal upsampling.

4.2 Indoor Studio Sequences

In general, sequences shot in controlled studio environments exhibit far less temporal noise and structural inconsistencies in comparison to "in-the-wild" dynamic outdoor shots. To further test our system we introduce an extra degree of challenge in the form of multiple, fast-moving objects while still using no more than 12 cameras for full, 360-degree coverage. This introduces further difficulty due to occlusions caused when the ball passes in front of performer as well as testing the limits of the flow-based projection system. In spite of these challenges, the proposed system is still able to filter a lot of the noise generated and can recover a modest amount of missing geometry, Fig. 5, (right model).

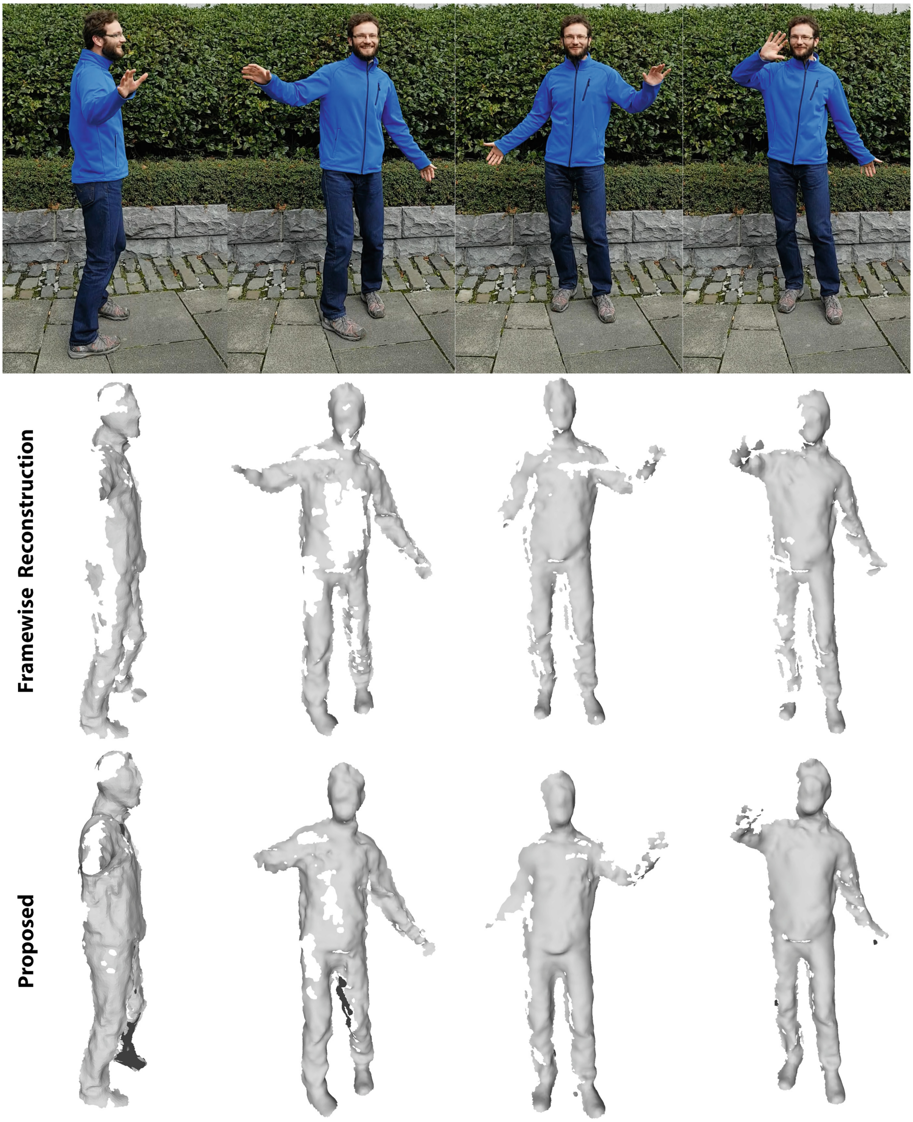

Fig. 6. A non-sequential set of frames from an outdoor VV shoot using handheld cameras. (Top): The RGB input to the system. (Middle): The result of applying poisson reconstruction to the unprocessed, temporally incoherent point clouds. (Bottom): The same poisson reconstruction method applied to the upsampled and filtered output of the proposed system [24].

4.3 Synthetic Data Sequences

As a baseline for ground-truth quantitative benchmarking, we evaluate our system using a synthetic virtual scenario. This synthetic data consists of a short

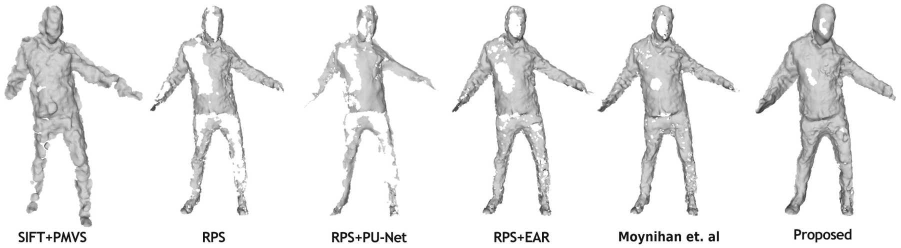

Fig. 7. A qualitative comparison of surface areas recovered from PSR meshing of point clouds from comparable systems. All meshes were created using the same octree depth for PSR and same distance threshold for outlier removal. From left to right: SIFT+PMVS [10,21], RPS [30], RPS+PU-Net [37], RPS+EAR [15], Proposed system applied in two-frame, forward direction only [24], the proposed system with windowed temporal filter centered on a window of 5 frames.

sequence featuring a human model performing a simple animated dance within a realistic environment. 12 virtual cameras were evenly spaced around a 180° arc centered about the animated character model. The images rendered from these virtual cameras provided the input to the VV systems for testing. Using this data we compare our results with those of temporally incoherent VV systems by applying PSR to the output point clouds and using the Hausdroff distance as an error metric. This is shown in Fig. 9.

We compare our results against similar framewise point cloud reconstruction systems, SIFT+PMVS [10] and RPS [30] as well as some state of the art upsampling algorithms for which we provide the method of RPS as input; PU-Net [37] and the Edge-Aware Resampling [15] method. Benchmarking against RPS+EAR also provides a form of ablation study for the effect of the proposed method as this is the approach used to initialize the system.

The proposed system demonstrates an overall improvement in quality in Table 2 yet the synthetic dataset lacks the noise which would be inherent to data captured in a real-world scenario. We would expect further improvements in such a scenario where the input error for the framewise reconstruction systems would be higher. Figure 7 qualitatively shows the effect of applying the proposed system to much noisier input data.

4.4 Flow Initialization

While practically any dense optical flow approach can be used to initialize the STEA filter in Sect. 3.2, improvements can be achieved by application-appropriate initialization. We show the results of initializing the STEA filter with CPM against other dense-flow alternatives in Table 1. The advanced edge-preservation of CPM results in it out-performing the alternatives but comparable results can achieved using GPU-based alternatives which may somewhat trade off accuracy for speed [1] (Fig. 8).

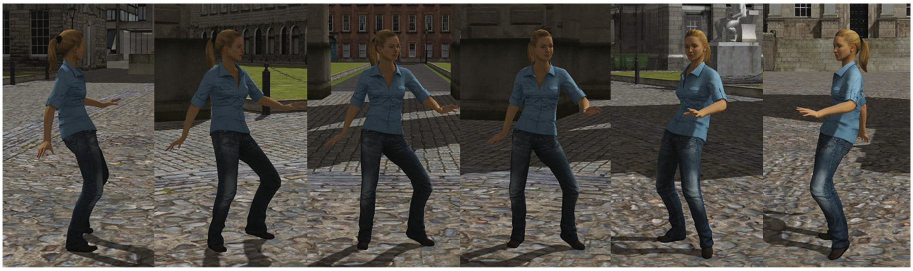

Fig. 8. An animated character model within a realistic virtual environment to generate synthetic test data [24].

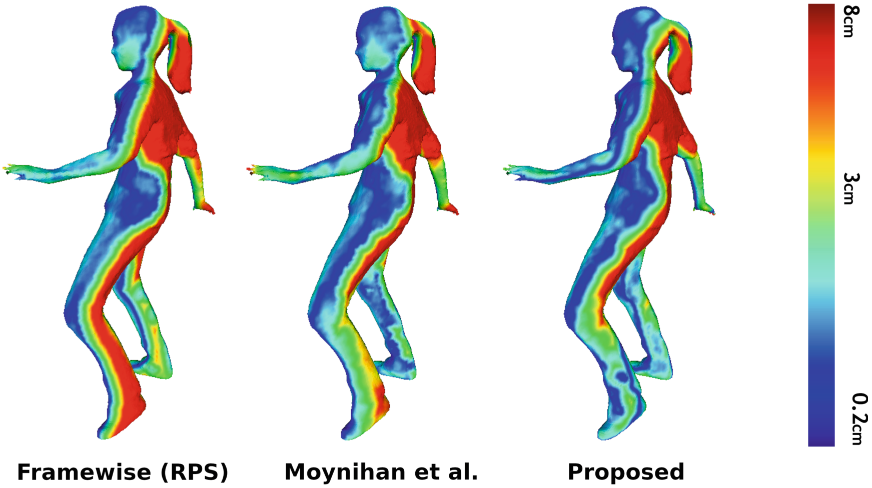

Fig. 9. Ground-truth evaluation of the proposed system against the virtual reference model using Hausdorff distance as the error metric. The left model shows a frame generated using framewise reconstruction [30], the middle model is the forward-projection, two frame filter [24], while the right shows the proposed systems [24] for a filtering window of 5 frames.

5 Limitations and Future Work

Due to the temporal nature of the algorithm, it is not possible to directly parallelize the proposed system as the most accurate scene flow is generated by providing the full length of the video sequence. Yet, if parallelism is a necessity, a compromise can be achieved in the form of a keyframe-based system whereby the input timeline is divided in reasonably-sized portions. Future work may employ some automatic keyframe detection which could maximise inter-keyframe similarity.

Table 2. Synthetic baseline comparison between the proposed method and similar state of the art approaches. Figures represent the Hausdorff distance metric with respect to the bounding box diagonal of the ground truth (%) [24].

Method	Mean error (%)	RMS error (%)
SIFT+PMVS	6.18	8.09
RPS	2.17	3.27
RPS + PU-Net	2.44	3.50
RPS + EAR	2.40	3.64
Moynihan et al.	1.78	2.72
Proposed	**1.56**	**2.30**

6 Conclusions

As the barrier to entry for VV content creation lowers, we still see a large disparity between content from affordable systems and that from high-budget studios. Sparse and dynamic, in-the-wild studio setups will always have to overcome the characteristic spatio-temporal errors of systems which continue to lower the cost to entry while maintaining creative freedom. These limitations are difficult to overcome but we have demonstrated that improvements are achievable by extending upsampling and filtering techniques into the spatio-temporal domain.

Our approach can efficiently filter temporally incoherent noise without overcorrecting for otherwise pertinent geometry. We also demonstrate the ability to perform a framewise upsampling which not only creates new coherent geometry but also propagates existing, spatially-coherent geometry across a variable frame window. This expansion to [24] shows improved results over the framewise, two-way projection and filter.

The most appreciable results emerge for the most challenging sequences. Handheld, outdoor VV captures tend to be the most error prone and thus stand to benefit the most from the proposed upsampling and filtering method as can be seen in the qualitative results presented. However, despite being less susceptible to error our qualitative analysis via synthetic ground truth data shows a marked improvement over the framewise approach. We also demonstrate an improvement over the work presented in [24] with the addition of a variable temporal window that further explores the persistence of coherent geometry against noise.

Overall we present an efficient method for improving the quality of greatly constrained VV capture setups in order to meet the growing demand for affordable virtual and augmented reality content creation.

References

1. Bao, L., Yang, Q., Jin, H.: Fast edge-preserving PatchMatch for large displacement optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3534–3541 (2014)
2. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: a view centered variational approach. Int. J. Comput. Vision **101**(1), 6–21 (2013)
3. Berjón, D., Pagés, R., Morán, F.: Fast feature matching for detailed point cloud generation. In: 2016 6th International Conference on Image Processing Theory Tools and Applications (IPTA), pp. 1–6. IEEE (2016)
4. Bouguet, J.Y.: Pyramidal implementation of the affine Lucas-Kanade feature tracker. Intel Corporation (2001)
5. Collet, A., et al.: High-quality streamable free-viewpoint video. ACM Trans. Graph. (ToG) **34**(4), 69 (2015)
6. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 1841–1848. IEEE (2013)
7. Dou, M., et al.: Motion2fusion: real-time volumetric performance capture. ACM Trans. Graph. (TOG) **36**(6), 246 (2017)
8. Dou, M., et al.: Fusion4d: real-time performance capture of challenging scenes. ACM Trans. Graph. (TOG) **35**(4), 114 (2016)
9. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45103-X_50
10. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
11. Gastal, E.S., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Graph. (ToG) **30**, 69 (2011)
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York (2004)
13. Hu, Y., Song, R., Li, Y.: Efficient coarse-to-fine PatchMatch for large displacement optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5704–5712 (2016)
14. Huang, C.H., Boyer, E., Navab, N., Ilic, S.: Human shape and pose tracking using keyframes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3446–3453 (2014)
15. Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.: Edge-aware point set resampling. ACM Trans. Graph. **32**, 9:1–9:12 (2013)
16. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. (ToG) **32**(3), 29 (2013)
17. Klaudiny, M., Budd, C., Hilton, A.: Towards optimal non-rigid surface tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 743–756. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_53
18. Lang, M., Wang, O., Aydin, T.O., Smolic, A., Gross, M.H.: Practical temporal consistency for image-based graphics applications. ACM Trans. Graph. **31**(4), 1–8 (2012)
19. Li, H., Adams, B., Guibas, L.J., Pauly, M.: Robust single-view geometry and motion reconstruction. ACM Trans. Graph. (ToG) **28**, 175 (2009)
20. Liu, Y., Dai, Q., Xu, W.: A point-cloud-based multiview stereo algorithm for free-viewpoint video. IEEE Trans. Visual Comput. Graph. **16**(3), 407–418 (2010)

21. Lowe, D.G.: Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image, uS Patent 6,711,293, 23 March 2004
22. Luhmann, T., Robson, S., Kyle, S., Harley, I.: Close Range Photogrammetry. Wiley, New York (2007)
23. Moulon, P., Monasse, P., Marlet, R.: Adaptive structure from motion with *a Contrario* model estimation. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7727, pp. 257–270. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37447-0_20
24. Moynihan, M., Pagéés, R., Smolic, A.: Spatio-temporal upsampling for free viewpoint video point clouds. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, pp. 684–692. INSTICC, SciTePress (2019). https://doi.org/10.5220/0007361606840692
25. Mustafa, A., Kim, H., Guillemaut, J.Y., Hilton, A.: Temporally coherent 4D reconstruction of complex dynamic scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4660–4669, June 2016. https://doi.org/10.1109/CVPR.2016.504
26. Mustafa, A., Hilton, A.: Semantically coherent co-segmentation and reconstruction of dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 422–431 (2017)
27. Mustafa, A., Kim, H., Guillemaut, J.Y., Hilton, A.: General dynamic scene reconstruction from multiple view video. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 900–908 (2015)
28. Mustafa, A., Volino, M., Guillemaut, J.Y., Hilton, A.: 4D temporally coherent light-field video. In: 2017 International Conference on 3D Vision (3DV), pp. 29–37. IEEE (2017)
29. Myronenko, A., Song, X.: Point set registration: coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2262–2275 (2010)
30. Pagés, R., Amplianitis, K., Monaghan, D., Ondřej, J., Smolic, A.: Affordable content creation for free-viewpoint video and VR/AR applications. J. Vis. Commun. Image Representat. **53**, 192–201 (2018). https://doi.org/10.1016/j.jvcir.2018.03.012. http://www.sciencedirect.com/science/article/pii/S1047320318300683
31. Prada, F., Kazhdan, M., Chuang, M., Collet, A., Hoppe, H.: Spatiotemporal atlas parameterization for evolving meshes. ACM Trans. Graph. (TOG) **36**(4), 58 (2017)
32. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1164–1172 (2015)
33. Schaffner, M., Scheidegger, F., Cavigelli, L., Kaeslin, H., Benini, L., Smolic, A.: Towards edge-aware spatio-temporal filtering in real-time. IEEE Trans. Image Process. **27**(1), 265–280 (2018)
34. Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 501–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31
35. Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., Cremers, D.: Stereoscopic scene flow computation for 3D motion understanding. Int. J. Comput. Vision **95**(1), 29–51 (2011)
36. Wu, S., Huang, H., Gong, M., Zwicker, M., Cohen-Or, D.: Deep points consolidation. ACM Trans. Graph. (ToG) **34**(6), 176 (2015)

37. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: PU-NET: point cloud upsampling network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2799 (2018)
38. Zollhöfer, M., et al.: Real-time non-rigid reconstruction using an RGB-D camera. ACM Trans. Graph. (ToG) **33**(4), 156 (2014)
39. Zou, D., Tan, P.: CoSLAM: collaborative visual SLAM in dynamic environments. IEEE Trans. Pattern Anal. Mach. Intell. **35**(2), 354–366 (2013)