# Spatio-Temporal Upsampling for Free Viewpoint Video Point Clouds

Matthew Moynihan[1], Rafael Pagés[1,2] and Aljosa Smolic[1]
[1]*V-SENSE, School of Computer Science and Statistic, Trinity College Dublin*
[2]*Volograms Ltd, Dublin*
{*mamoynih, smolica*}*@tcd.ie, rafa@volograms.com*

Keywords: Point Clouds, Upsampling, Temporal Coherence, Free Viewpoint Video, Multiview Video

Abstract: This paper presents an approach to upsampling point cloud sequences captured through a wide baseline camera setup in a spatio-temporally consistent manner. The system uses edge-aware scene flow to understand the movement of 3D points across a free-viewpoint video scene to impose temporal consistency. In addition to geometric upsampling, a Hausdorff distance quality metric is used to filter noise and further improve the density of each point cloud. Results show that the system produces temporally consistent point clouds, not only reducing errors and noise but also recovering details that were lost in frame-by-frame dense point cloud reconstruction. The system has been successfully tested in sequences that have been captured via both static or handheld cameras.

## 1 Introduction

Recent years have seen a spike in interest towards virtual/augmented reality (VR/AR), especially at consumer level. The combined maturity and affordability is reducing the barrier to entry for content creators and enthusiasts alike. As a result, it has become largely apparent that there is a need to close the content creation gap, specifically with respect to the capture and reconstruction of live scenes and performances.

Free-viewpoint video (FVV) technology provides the necessary tools for creators to capture and display real-world dynamic scenes. Current state-of-the-art systems usually feature large arrays of high-resolution RGB cameras and IR depth sensors in a professional studio environment (Collet et al., 2015; Liu et al., 2010). These systems normally operate on a frame-by-frame basis, where they compute a dense point cloud using different multi-view stereo (MVS) techniques. With such high-density camera configurations, temporal inconsistencies in the 3D reconstructions are less conspicuous. However, these systems are likely to be inaccessible to low-budget productions and independent content creators. New approaches are emerging that enable FVV capture using wide-baseline camera setups that include only consumer-grade cameras, some of which can even be handheld (Pagés et al., 2018). However, frame-by-frame reconstructions (Mustafa et al., 2015) often include temporal artifacts in the sequence. This is most often due to common fail case scenarios for photogrammetry-based systems such as a lack of texture information or attempting to reconstruct non-lambertian surfaces. In the absence of any spatio-temporal constraints, it can be observed that salient geometric features can become distorted and temporally incoherent across FVV sequences. Figure 1 demonstrates such an example as frame-wise reconstruction suffers a loss of geometric information where the reconstruction system failed to identify enough feature points to guarantee an accurate reconstruction, specifically in extremities such as hands and feet. Temporal inconsistencies can also be observed between frames in the form of structured noise patches and holes in the model.

We propose a system that both, upsamples low density point clouds, and also enforces a temporal constraint which encourages the selective recovery of lost geometric information. They key contributions of our work can be summarised as follows:

- A spatio-temporal consistency system for point cloud sequences that coherently merges consecutive point clouds, based on estimation of the Edge Aware Scene Flow on the original wide-baseline images.
- A self regulating noise filter based on a Hausdorff distance quality metric as the conditioning criterion of the coherent mesh.

As a baseline for improvement we compare this proposed method to temporally-incoherent alterna-

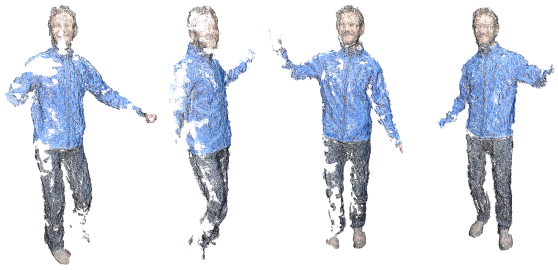

Figure 1: Four frames from a typical FVV sequence. These unprocessed results show inconsistencies and noise due to occlusions, fast-moving elements and sparse feature detection. Some temporally-inconsistent structured noise patches can be observed also. Our system implements spatio-temporal consistency which aims to remove the majority of structured noise patches as well as recovering some lost geometry leading to a more temporally coherent result.

tives whereby point cloud densification is achieved solely by geometric upsampling.

## 2 Related Work

Spatio-temporal consistency has been widely addressed by modern FVV systems and dynamic reconstruction algorithms. The addition of this consistency ensures the reconstruction of smooth and realistic sequences with minimal temporal artifacts. However, most techniques apply a registration-based temporal constraint to the final 3D meshes, and not in the early processing stages (Huang et al., 2014; Klaudiny et al., 2012). These techniques normally use some variation of non-rigid ICP (Li et al., 2009; Zollhöfer et al., 2014), such as the coherent drift point algorithm (Myronenko and Song, 2010). An example of this has been demonstrated in the work by Collet et al. (Collet et al., 2015): they apply mesh tracking in the final processing stage, not only to provide a smoother FVV sequence but also to improve data storage efficiency as, between keyframes, only the vertex positions vary while face indices and texture coordinates remain the same. However, they do not apply temporal coherence in any other stage of the process, mainly because their system uses 106 cameras (both RGB and IR) in a studio environment and the resulting dense point clouds are very accurate on a frame-to-frame basis. Another technique has been proposed by Mustafa el al. (Mustafa et al., 2016), where they ensure temporal coherence of the FVV sequence by using sparse temporal dynamic feature tracking, as an initial stage, and also in the dense model, using a shape constraint based on geodesic star convexity. However, 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 these methods are again, highly influenced by the density of viewpoints and baseline width. Furthermore, this constraint is applied at a refinement stage and so the initial point cloud is still temporally unrefined before the poisson mesh has been generated. Other techniques address temporal coherence by trying to find an understanding of the scene flow to recover not only motion, but also depth. Examples of this are the works by Basha et al. (Basha et al., 2013) and Wedel et al. (Wedel et al., 2011). However, these techniques require a very precise and dense motion estimation for almost every pixel in order to acquire accurate depth maps and cameras configured with a very narrow baseline. In our system, we use the temporally consistent flow proposed by Lang et al. (Lang et al., 2012) which we apply to multi-view sequences, allowing us to track dense point clouds across the sequence even when we use cameras with a wide baseline. While not specifically targeting FVV systems, there is a well-established state of the art for improving general 3D reconstruction accuracy via point cloud upsampling or densification (Huang et al., 2013; Wu et al., 2015; Yu et al., 2018). However, given that these systems are designed to perform upsampling for a single input point cloud, 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 noise. 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. (Huang et al., 2013) and supports it using the temporal information obtained from the inferred 3D scene flow along with some spatio-temporal noise filtering. This is performed with the rationale that increasing the density of coherent points improves the accuracy of point cloud meshing processes such as Poisson Surface Reconstruction (Kazhdan and Hoppe, 2013).

## 3 Methodology

### 3.1 Point Cloud Reconstruction & Edge-Aware Upsampling

The input to our system is a temporally-incoherent FVV point cloud sequence captured using an affordable FVV pipeline similar to the system proposed by (Pagés et al., 2018). The target scene is captured across a setup of multi-view videos spanning

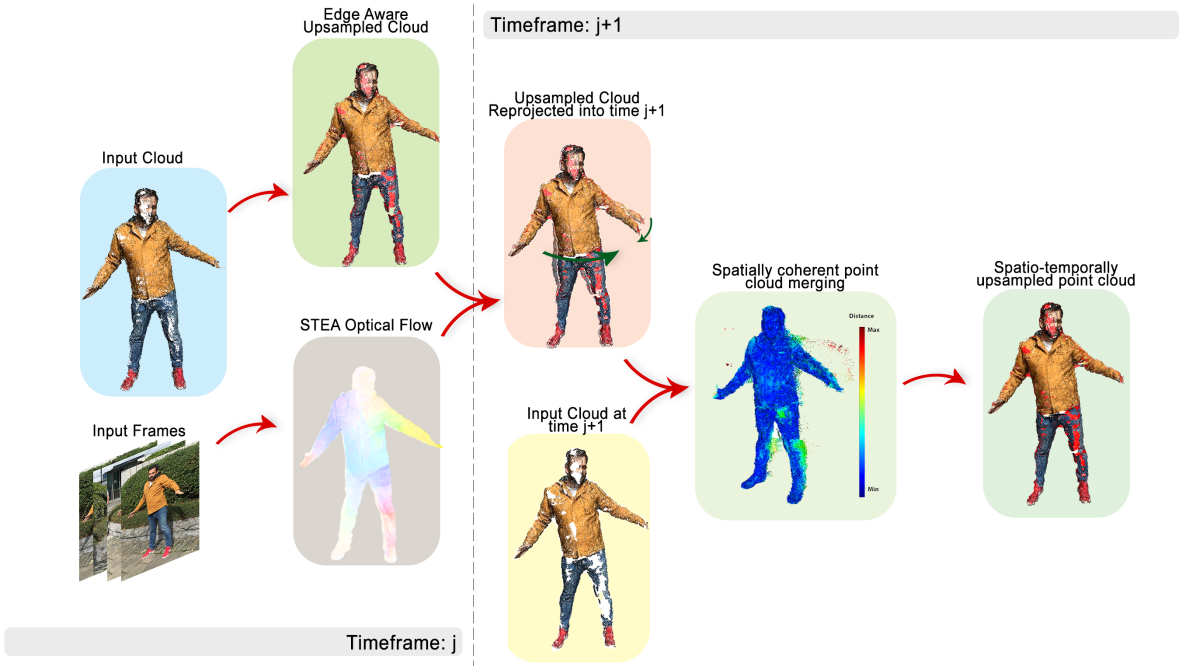


Figure 2: Temporally-Coherent upsampling and filtering: system overview. The system input is the framewise-independent point cloud sequence as well as the RGB images and calibration parameters used to generate it. The system upsamples the input point cloud for a given timeframe $j$, then calculates the edge-aware scene flow to project the upsampled cloud into timeframe $j+1$. The final output is the result of a temporally-coherent merging and filtering process which retains upsampled geometric information from the previous frame as well as pertinent data from following frame.

wide baselines with known camera intrinsics. Extrinsics are automatically calibrated using sparse feature matching and incremental Structure from Motion (Moulon et al., 2012). When the cameras are handheld, other more advanced techniques such as CoSLAM (Zou and Tan, 2013), can be used to estimate their position and rotation. At every frame, a point cloud is initially calculated using structure from motion and densified using multi-view stereo. For instance, the examples shown in this paper use a denser sparse point cloud estimation proposed by Berjón et al. (Berjón et al., 2016), which is later densified even further using the unstructured MVS technique proposed by Schönberger et al. (Schönberger et al., 2016). 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 $j$th 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) (Huang et al., 2013). 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. Figure 2 presents an overview of the proposed pipeline following the acquisition process in which we present our temporally-coherent filtering and upsampling algorithm.

## 3.2 Spatio-Temporal Edge-Aware Scene Flow

We use a pseudo scene flow in order to project as much pertinent geometry from the previous frame as possible. In the context of the proposed system, scene flow is defined as an extension of 2D optical flow to include depth information and provide a framework for tracking point clouds in 3D. Dense scene flow information is generated by computing the 2D optical flow for each input video, thus, for every sequence $s_i$ we compute its corresponding scene flow $f_i$. This approach of accumulating multiple 2D flows ensures a robustness to wide-baseline input in as each input is calculated independently.

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 (Hu et al., 2016).
- The dense optical flow is then refined using a

Table 1: 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) (Hu et al., 2016), Fast Edge-Preserving Patch Match (FEPPM) (Bao et al., 2014), Pyramidial Lukas-Kanade (PyLK) (Bouguet, 2001) and Gunnar-Farnebäck (FB) (Farnebäck, 2003).

| STEA Initialization | Area Increase (%) |
|---|---|
| **CPM** | **37.73** |
| FEPPM | 34.9 |
| PyLK | 34.77 |
| FB | 29.7 |

spatio-temporal edge aware filter based on the Domain Transfer (Lang et al., 2012).

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 (Farnebäck, 2003), the given initialization is less sensitive to temporal noise and emphasizes edge-aware constraints at input, thus producing more coherent results. We analysed other approaches from the state-of-the-art and concluded that they lack global regularization, edge-preservation or are sensitive to large displacement motion. Table 1 demonstrates how initializing the filter with different flow algorithms affects the geometry recovered by the proposed algorithm.

The STEA filter is implemented as in (Lang et al., 2012) which features an extension to the Domain Transform (Gastal and Oliveira, 2011) in the spatial and temporal domains using optical flow as a primary application:

1. The filter is initialized as suggested in (Schaffner et al., 2018), using coarse-to-fine patch match (Hu et al., 2016). 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 (Dollár and Zitnick, 2013).
3. Using the calculated edges, the dense optical flow is then interpolated using Edge-Preserving Interpolation of Correspondences (Revaud et al., 2015).

The interpolated dense optical flow is then fed into the STEA filter as an optical flow video sequence where it is filtered in multiple passes through the spatial and temporal domains to reduce temporal inconsistencies and improve edge fidelity. An example of the STEA processing pipeline can be seen illustrated in Figure 3.

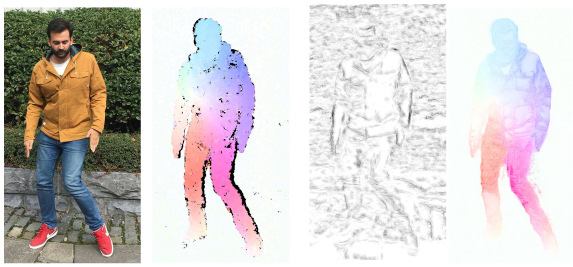

Figure 3: Pictured left to right, the STEA flow processing pipeline: input RGB image from a given viewpoint, (1) CPM nearest neighbour field estimate, (2) SED detected edges, (3) interpolated dense STEA output. Conventional colour coding has been used to illustrate the orientation and intensity of the 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.

## 3.3 Point Cloud Motion Estimation

Knowing the camera parameters $(C_{j_1},...,C_{j_m}$, at the $j$th frame), the set of scene flows $(f_{j_1},...,f_{j_m})$, and the set of point clouds $(\mathcal{P}_j,...,\mathcal{P}_J)$, we can predict how a certain point cloud moves across the sequence. For this, we back-project every point $\mathbf{P}_k \in \mathcal{P}_j$ to each flow $f_i$ at that specific frame $j$. To avoid the back-projection of occluded points, we check the sign of the dot product between the camera pointing vector and the normal of the point $\mathbf{P}_k$. Using the flow, we can predict the position of the back-projected 2D points $\mathbf{p}_{ik}$ in sequential frames, $\mathbf{p}'_{ik}$.

Therefore, the predicted point cloud $\mathcal{P}'_j$, at frame $j+1$, is the result of triangulating the set of predicted 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 defined by the Direct Linear Transformation algorithm (Hartley and Zisserman, 2004). The resulting point undergoes a Gauss-Markov weighted non-linear optimisation which minimises the reprojection error (Luhmann et al., 2007).

## 3.4 Geometry-Based Filtering & Reconstruction

The last step of the proposed system involves performing a coherent merging of the predicted point cloud and the target frame. This coherent merge uses a Hausdorff distance-based quality metric to allow

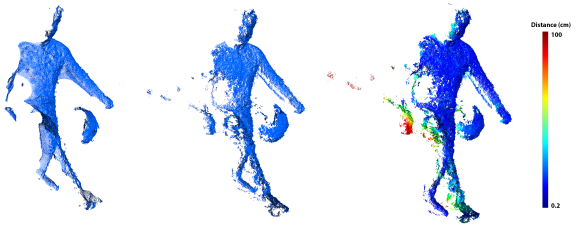


Figure 4: A visual representation of the coherent merge process. Pictured is the result of merging the predicted point cloud (left) with the target cloud (middle) . All points are color-coded with respect to the distance between their nearest-neighbour match in the other cloud. Points labelled hihger than the threshold for the given frame will be removed from the merged result.

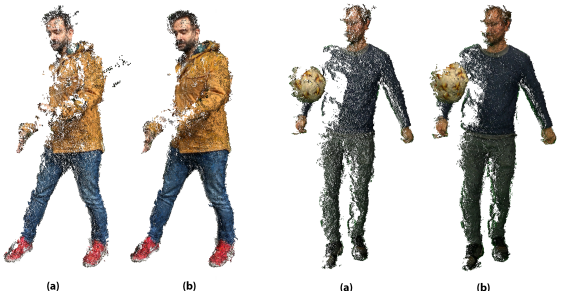


Figure 5: Filtered point clouds from two sequences, one extracted from hand held cameras in outdoor setting (left) and the other captured in a green screen studio (right). Both show a comparison between the point cloud extracted from framewise reconstruction (a), and the filtered results (b).

neighbouring geometry to persist and deform naturally whilst also removing noise in an adaptive manner. The Hausdorff distance threshold is computed as the average resolution of the predicted and target point clouds reduced by one order of magnitude. This constrains the threshold to be set at some small distance relative to the point cloud resolution which ensures that only pertinent points remain. Formally, $d_j$ is the Hausdorff distance threshold between the flow-predicted point cloud $\mathcal{P}'_j$ and the target sequential point cloud $\mathcal{P}_{j+1}$.

The coherent merged cloud $\mathcal{P}^*_{j+1}$ is given by the logical definition in equation 1.

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,

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

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 Object Point Validation

The result of the coherent merge, described in Section 3.4, are the points upon which the input cloud and the projected cloud agree. While this co-dependence is well-suited to filtering noise, it fails to recover pertinent geometry that doesn't happen to reside within the distance threshold. In particular, faster-moving objects tend to be trimmed as the overlap between frames can be small. To further improve the recovery of geometry we added a validation process which considers a confidence value for projected points in $\mathcal{P}'_j$. 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 average magnitude of the optical flow vectors for each view of the given point is then used as a confidence value for that point. In this way, points for which a high flow magnitude exists in the sequential frame can be considered dynamically tracked. A confidence value proportional to the average scene flow magnitude is applied as a weight to adaptively adjust the distance threshold $d_j$ for dynamically tracked points. This allows for the retention of pertinent, fast-moving geometry without hindering the performance of the noise filter.

## 4 Results

Figure 5 shows a direct comparison between two frames from two typical yet challenging FVV sequences. The performance of the system was evaluated qualitatively on sequences captured outdoors using handheld devices (i.e. phone and tablet device cameras) and on sequences captured in a modest green screen studio using 12 mounted (6 full HD and 6 4K) cameras. A synthetic sequence was used to evaluate the results on a quantitative basis. This sequence consists of a digitally created character placed in a virtual environment with simulated cameras.

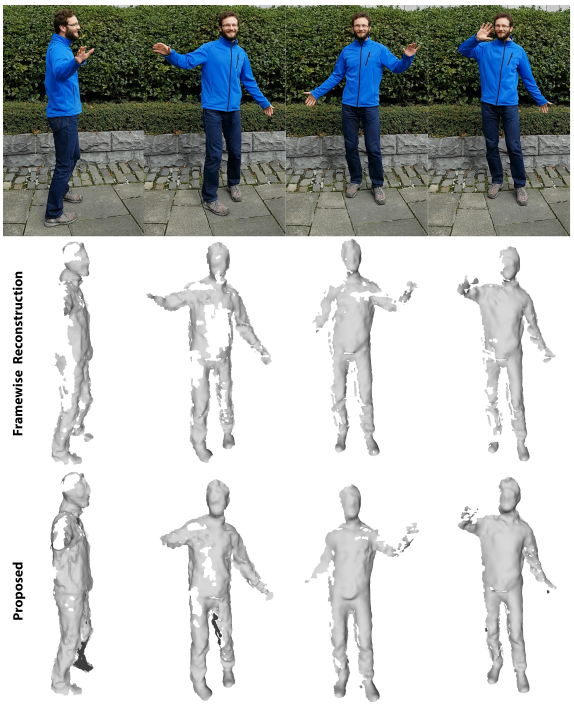


Figure 6: A selection of frames from a handheld outdoor sequence. The RGB input from a single camera (top), the result of poisson reconstruction on raw input (middle), the result of poisson reconstruction on proposed method (bottom).

## 4.1 Outdoor Handheld Camera Sequences

Unique challenges arise from filtering point clouds extracted from unstable cameras with a non-uniform and dynamic background. Errors in the camera extrinsics, differences in colour balance, and irregular lighting conditions result in reconstruction errors: inconsistency in the frame-by-frame reconstruction and a significant amount of noise. An example of this is shown in Figure 5 (left model): the figure shows the difference between using framewise reconstruction (a) and our method (b). As can be seen, large holes in the subject have been filled and most of the undesirable noise has been filtered. However, while the Hausdorff quality metric is able to remove most of the noise, the system is still sensitive to structured noise patches, typical of MVS reconstruction inaccuracies.

Figure 6 shows four non-consecutive frames for another outdoor sequence shot on the same location. In the top row, the input images from one of the handheld cameras. The second and third rows demonstrate the result of applying Screened Poisson Reconstruction (PSR) (Kazhdan and Hoppe, 2013) to the resulting point clouds. The meshes shown are the result of sampling the initial PSR-generated mesh with the input point cloud to remove outlier vertices. As a result, holes in the input point cloud become apparent in the resulting mesh. This figure demonstrates the effect of coherent point cloud upsampling on reducing the perforations in the mesh.

## 4.2 Indoor Studio Sequences

The use of stabilized, high resolution cameras in a green screen studio brings many advantages to filtering the reconstruction such as more accurate flow information and far less temporal noise. In order to add an extra degree of challenge to this sequence an additional and fast-moving dynamic object has been added to the scene by having the subject volley a soccer ball. While this setup enables the estimation of compelling dense point clouds, the relatively sparse camera array still suffers from occlusions, as demonstrated when the ball crosses in front of the subject (Figure 5, right model). Despite this, it can be seen that similar to outdoor sequences, large portions of the subject have been recovered while retaining the fast-moving football.

## 4.3 Synthetic Data Sequences

In order to conduct a ground-truth analysis we have performed an evaluation of our system using a synthetic dataset. The dataset consists of a short 25 frame sequence in which a digitally created human performs some dynamic motion against an otherwise static background. In this dataset, 12 camera views arranged in a 180° arc, with known parameters, have been synthesized to provide the input multi-view video sequences. We compare the result of our system with the results of framewise reconstructions by meshing the output point clouds using PSR and calculating their Hausdorff distance with respect to the original model. Figure 9 illustrates the error heatmap of the reconstructed mesh in the absence of point cloud processing and following the proposed temporally-coherent system. It can be seen that the proposed coherent upsampling approach manages to recover accurate geometry that would be otherwise missing for the same frame.

As a baseline for comparison we have measured the performance of our system against two framewise reconstructions, SIFT+PMVS (Furukawa and Ponce, 2010) and RPS (Pagés et al., 2018) as well as some state of the art upsampling algorithms using the RPS method as input; PU-Net (Yu et al., 2018) and the Edge-Aware Resampling (Huang et al., 2013)

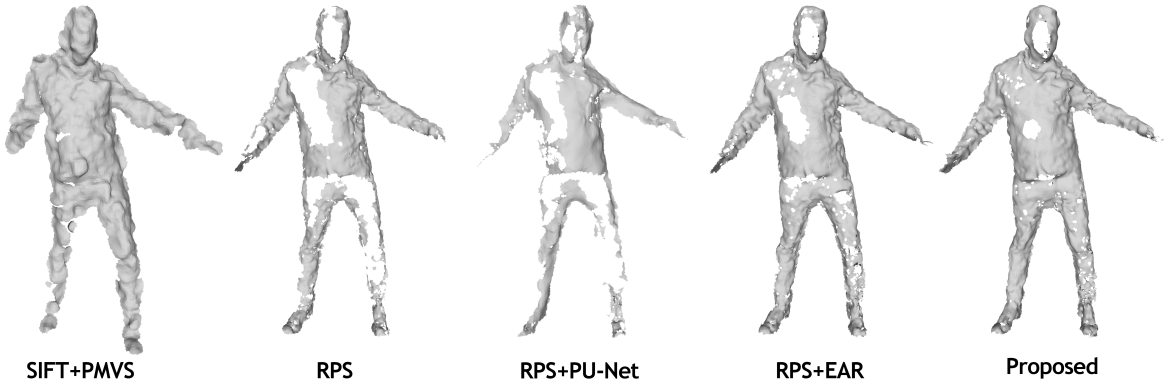


Figure 7: Results of applying PSR to resulting pointclouds. PSR is first applied and then the input cloud is used to clean the resulting mesh by removing faces which exceed a given distance to any input vertices. All inputs were processed using the same octree depth and distance threshold for cleaning.

Figure 8: A selection of the images used to generate synthetic data for a ground-truth analysis of the fvv reconstruction system.

method. The comparison with RPS+EAR also functions as an ablation study as this is used as the initializer for the proposed system.

The results of Table 2 show improvement on the compared methods but may also be hindered by the synthetic nature of the test data. This is, in part, due to the lack of natural noise that one would expect for the equivalent real-world application. In such a scenario where more temporally-incoherent structured noise is more prominent it would be expected that a further margin of improvement could be achieved. We have provided Figure 7 as a qualitative demonstration of the margin of improvement achievable by the proposed system when applied to noisy scenario. It should also be noted that while the SIFT+PMVS method demonstrates a more complete mesh, it is largely contaminated with noisey data as evidenced by the results of the quantitative study in Table 2.

## 4.4 Flow Initialization

The STEA filter described in section 3.2 is robust in that it can be initialized using practically any dense optical flow algorithm, but in order to retain spatial accuracy with regards to point projection it requires an appropriate selection. Table 1 shows the effect of initialization using the chosen CPM method in comparison to popular alternatives. CPM demonstrably

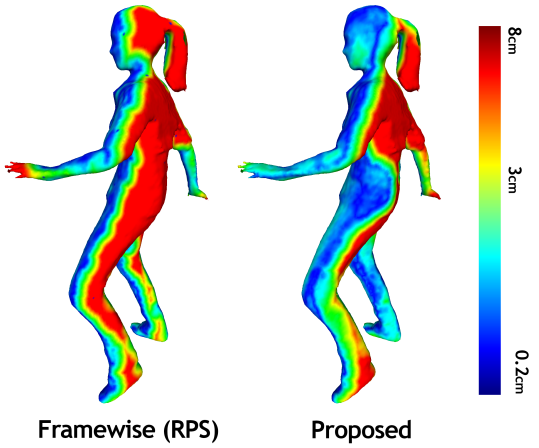


Figure 9: Hausdorff distance with respect to the synthetic model. On the left, using the result of a framewise reconstruction. On the right, using our system. As the model is synthetic, the units were scaled with respect to the bounding box diagonal such that it's length becomes 150cm.

out-performs the chosen alternatives due to its edge-preserving application. While FEPPM (Bao et al., 2014) uses an edge-preserving patch match and NNF approach, cpm improves upon typical NNF field type matching by adding global regularization.

# 5 Conclusions

It remains a challenge for amateur and low-budget productions to produce FVV content on a comparable scale with that of more affluent studios. Wide-baseline FVV systems are likely to always be more susceptible to inherent noise in the form of occlusions and photogrammetry errors. While this noise presents a difficult obstacle we have shown that it is often tem-

Table 2: Hausdorff error (mean and root mean square (RMS)) comparison between reconstruction results and ground truth synthetic dataset. Figures presented are expressed as % with respect to bounding box diagonal of the ground truth.

| 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 |
| **Proposed** | **1.78** | **2.72** |

porally incoherent and so it can be corrected by enforcing spatio-temporal constraints.

By leveraging the permanence of temporally coherent geometry, our system is able to effectively filter noise while retaining pertinent geometric data which has been lost on a frame to frame basis. By enforcing this spatio-temporal consistency we demonstrate the improvements that our system will have for modern and future FVV systems alike.

We have shown that our system is suited to filtering point clouds from both studio setups and handheld "dynamic camera" outdoor scenes. Although the effects are most appreciable for dynamic outdoor scenes in which there tends to be much more noise, the advantage of more accurate flow information demonstrates visible improvements for indoor, studio-based sequences also. Some inherent limitations exist in the amount of noise which can be filtered whilst retaining important geometry, as is typical of many signal-to-noise filtering systems. This is particularly evident in the case of fast moving objects but our system alleviates this problem by using a dense optical flow method with demonstrably good sensitivity to large displacement as well as our proposed dynamic object tracking constraint.

In comparison to temporally-naive geometric upsampling approaches we can see that supplying spatio-temporal information leads to more accurate results and provides tighter framework for seeding geometric upsampling processes. This is confirmed by the results obtained from the synthetic dataset test whereby the most accurate approach was achieved by spatio-temporal filtering of an edge-aware upsampled point cloud.

## Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant No. 15/RP/2776.

## REFERENCES

Bao, L., Yang, Q., and Jin, H. (2014). Fast edge-preserving patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3534–3541.

Basha, T., Moses, Y., and Kiryati, N. (2013). Multi-view scene flow estimation: A view centered variational approach. *International journal of computer vision*, 101(1):6–21.

Berjón, D., Pagés, R., and Morán, F. (2016). Fast feature matching for detailed point cloud generation. In *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*, pages 1–6. IEEE.

Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas-kanade feature tracker.

Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., and Sullivan, S. (2015). High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):69.

Dollár, P. and Zitnick, C. L. (2013). Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer.

Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376.

Gastal, E. S. and Oliveira, M. M. (2011). Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, volume 30, page 69. ACM.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hu, Y., Song, R., and Li, Y. (2016). Efficient coarse-to-fine patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5712.

Huang, C.-H., Boyer, E., Navab, N., and Ilic, S. (2014). Human shape and pose tracking using keyframes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3446–3453.

Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., and Zhang, H. (2013). Edge-aware point set resampling. *ACM Transactions on Graphics*, 32:9:1–9:12.

Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29.

Klaudiny, M., Budd, C., and Hilton, A. (2012). Towards optimal non-rigid surface tracking. In *European Conference on Computer Vision*, pages 743–756.

Lang, M., Wang, O., Aydin, T. O., Smolic, A., and Gross, M. H. (2012). Practical temporal consistency

for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34–1.

Li, H., Adams, B., Guibas, L. J., and Pauly, M. (2009). Robust single-view geometry and motion reconstruction. In *ACM Transactions on Graphics (ToG)*, volume 28, page 175. ACM.

Liu, Y., Dai, Q., and Xu, W. (2010). A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE transactions on visualization and computer graphics*, 16(3):407–418.

Luhmann, T., Robson, S., Kyle, S., and Harley, I. (2007). *Close range phoToGrammetry*. Wiley.

Moulon, P., Monasse, P., and Marlet, R. (2012). Adaptive structure from motion with a contrario model estimation. In *Asian Conference on Computer Vision*, pages 257–270. Springer.

Mustafa, A., Kim, H., Guillemaut, J.-Y., and Hilton, A. (2015). General dynamic scene reconstruction from multiple view video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–908.

Mustafa, A., Kim, H., Guillemaut, J. Y., and Hilton, A. (2016). Temporally coherent 4d reconstruction of complex dynamic scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4660–4669.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275.

Pagés, R., Amplianitis, K., Monaghan, D., Ondej, J., and Smolic, A. (2018). Affordable content creation for free-viewpoint video and vr/ar applications. *Journal of Visual Communication and Image Representation*, 53:192 – 201.

Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2015). Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172.

Schaffner, M., Scheidegger, F., Cavigelli, L., Kaeslin, H., Benini, L., and Smolic, A. (2018). Towards edge-aware spatio-temporal filtering in real-time. *IEEE Transactions on Image Processing*, 27(1):265–280.

Schönberger, J. L., Zheng, E., Frahm, J.-M., and Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer.

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., and Cremers, D. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 95(1):29–51.

Wu, S., Huang, H., Gong, M., Zwicker, M., and Cohen-Or, D. (2015). Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6):176.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. (2018). Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799.

Zollhöfer, M., Nießner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., et al. (2014). Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics (ToG)*, 33(4):156.

Zou, D. and Tan, P. (2013). CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366.