Improving Camera Pose Estimation via Temporal EWA Surfel Splatting

Nikolaos Zioulis *† Alexandros Papachristou *‡ Dimitris Zarpalas§ Petros Daras¶

Information Technologies Institute, Centre for Research and Technology - Hellas

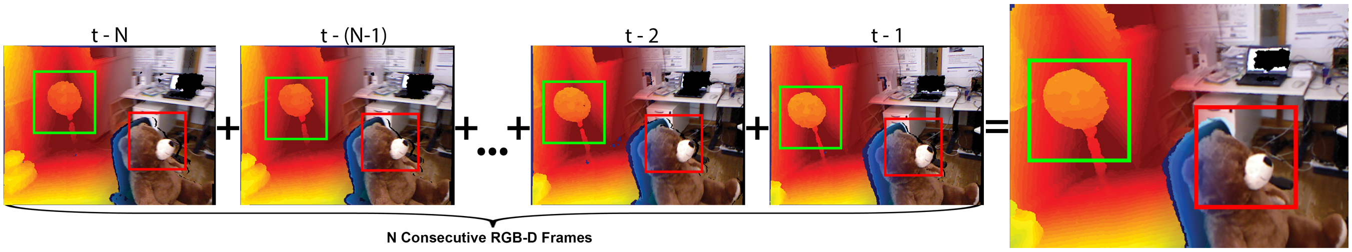

Figure 1: A temporal window RGB-D model is presented in the *fr1/teddy* sequence from the TUM dataset. A number of N consecutive frames $(t-N,\dots,t-1)$ are temporally accumulated and aggregated to the most recent one $(t-1)$ via Elliptical Weighted Average (EWA) splatting using their estimated relative poses. The resultant higher quality color and depth splatted frame (**right**) is then used for estimating the pose of the next incoming frame (t), improving the accuracy of the estimation. Depth fusion and de-noising is highlighted in the green box, while preservation of color details in the red box. RGB-D frames are visualized through a mixed depth and color representation from left to right with a blue-green-red color scale for depth (best viewed in color).

Abstract

Camera pose estimation is a fundamental problem of Augmented Reality and 3D reconstruction systems. Recently, despite the new better performing direct methods being developed, state-of-the-art methods are still estimating erroneous poses due to sensor noise, environmental conditions and challenging trajectories. Adding a back-end mapping process, SLAM systems achieve better performance and are more robust, but require higher computational resources, limiting their applicability. Therefore, lighter solutions to improve the accuracy of pose estimates are required. In this work we demonstrate the effectiveness of lighter data structures, namely surface elements, and exploit the temporality of sensor data streams to accumulate moving camera frames and improve tracking. This representation allows us to ”splat” a photometric and geometric model simultaneously and use it to improve the performance of dense RGB-D camera pose estimation methods. Exploiting Elliptical Weighted Average splatting to produce high quality photometric results also allows us to detect erroneous poses through a novel visual quality analysis process. We show evidence of the EWA temporal model's effectiveness in publicly available datasets and argue that point-based representations are a good candidate for building lighter systems that should be further explored.

Keywords: Camera pose estimation, Surface elements (surfels), Elliptical Weighted Average (EWA), Point-based rendering (PBR), Splatting, Tracking, SLAM, AR, 3D reconstruction, Visual quality analysis (VQA)

Index Terms: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking; I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and Scanning; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities;

1 Introduction

Virtual- and Augmented- Reality (VR & AR) technologies have recently started gaining momentum with forecasts continuously supporting their rise, further fueled by the developments in hardware (headsets, sensors, mobile). These new immersive and interactive experiences enabled by VR/AR are expected to open up new opportunities for various industries like education/training, entertainment, advertising/marketing, product design and gaming. Consequently, the development and refinement of solutions for visual navigation and mapping, as well as 3D capturing of real world scenes and objects, is required.

Camera pose estimation is fundamental computer vision problem that is directly related to AR applications [25] in order to map and track a visual sensor within the environment, and also highly relevant for the provisioning of realistic assets, be it either objects or real world scenes, for VR and/or AR use, through their 3D reconstruction. The associated challenges are varying illumination, the need to support unconstrained camera motion, the high amount - or even severe lack - of complexity within real world scenes and the online operation requirement, limiting the very important accuracy of camera pose estimation. The taxonomy of camera localization solutions includes *sparse*, *hybrid* and *dense* methods, with the former ones utilizing extracted features to establish correspondences, the latter ones leveraging the complete image information assuming pixel wise correspondences, and the hybrid methods lying in the middle, combining elements from both other ones.

While sparse methods were developed mainly for monocular cameras, the increased availability of consumer grade RGB-D sensors (Microsoft Kinect, Asus Xtion and Intel RealSense series), even available on mobile devices (Google Tango, Occipital Structure Sensor) [31, 1], has spurred new higher performing dense methods, mainly utilizing the depth information to produce better pose estimates, pioneered by KinectFusion [27]. Further work focused on additionally using the color information acquired by RGB-D sensors to increase the accuracy of the pose estimates by adding

*Indicates equal contribution
†e-mail: nzioulis@iti.gr
‡e-mail:papachra@iti.gr
§e-mail:zarpalas@iti.gr
¶e-mail:daras@iti.gr

a photo-consistency assumption and directly minimizing both geometric and photometric errors [13, 43, 18, 17], allowing camera localization to be more robust with respect to each data type's limitations (planar regions with low geometric features or texture-less regions). Many extensions and improvements have been proposed [2, 11, 32, 16], increasing the pose estimation accuracy. However, the aforementioned challenges make it difficult to eliminate erroneous estimates and make camera pose estimation suffer from drift due to error accumulation, therefore necessitating the use of corrective techniques, typically by building a pose- graph or chain and optimizing it upon detecting local or global loop closures. Graph optimization methods have evolved and improved in tandem with the higher accuracy camera localization estimates produced by the evolution of dense methods [7].

Simultaneous Localization and Mapping (SLAM) research focuses on fusing the localized depth information into 3D models of the environment while also optimizing it in-the-loop, with recent research making the online reconstruction of challenging scenes in high quality possible [20, 5, 39, 44, 9]. A key element of these approaches, first introduced by [27], is estimating the camera's pose with respect to the incrementally reconstructed model of the captured scene instead of the error-prone and drift suffering frame-to-frame registration. The most common model representation is the Truncated Signed Distance Function (TSDF) [4]. Still, this implicit surface representation has high memory consumption and, while it can also be used to accumulate color information [13, 23] - at the expense of more memory - the quality degrades due to the coarse resolution of the voxel grid, and the blurring induced by the running average weighting scheme customarily used to update the TSDF. As a result, the enhanced camera pose estimation methods for RGB-D sensors need to operate on a frame-to-frame [14], frame-to-multi-frame [5] or frame-to-keyframe [17, 2] basis, or otherwise ignore the color information [20, 39, 43, 32, 15].

With recent research [44, 32, 35, 15] turning to point-based representations to avoid the intermediate implicit surface representation and achieve lighter memory consumption, it was also shown that surface elements (surfels) [30] can also better model photo-consistency in finer resolution and allow for full utilization of both color and depth information [44, 35]. Inspired by [44], where surfels were used to jointly render a dense geometric and photometric representation of the observed scene, this work aims to expand and enhance this point-based representation in order to improve RGB-D camera pose estimation and decouple it from the mapping process to enhance its usability. In summary, the main contributions are:

- The introduction of a temporal surfel model, leveraging on the accurate blending offered by surfel splatting and Elliptical Weighted Average (EWA) Filtering, that is also seamlessly integrated into typical camera pose estimation frameworks.

- A novel visual quality analysis technique aimed at detecting erroneous pose estimates and increasing the robustness of the proposed model by preventing their accumulation into the temporal model.

- Comprehensive evaluation of the proposed approach using public datasets, proving its effectiveness in various scenes and different sensors, using state-of-the-art dense RGB-D camera pose estimation methods.

The remainder of this document is organized as follows: Previous related works regarding camera pose estimation, SLAM systems and point-based rendering are presented and discussed in Section 2. In Section 3, preliminaries regarding the state-of-the-art for RGB-D camera pose estimation methods exploiting the entirety of RGB-D sensor data are presented. Then, Section 4 addresses the proposed temporal surfel model building, its implementation and use within a typical RGB-D camera pose estimation framework. Evidence for the effectiveness of the proposed surfel model after evaluating it on various publicly available datasets are presented in detail in Section 5, and finally, in Section 6 we conclude by outlining the results, benefits, limitations and potential future work.

2 Related Work

Camera pose estimation and the reduction of drift have been active research topics that are relevant for both AR and online 3D reconstruction of real-life scenes and objects. Typical approaches include feature tracking and matching (*sparse*), the use of prior knowledge (*model or marker based*), direct image alignment (*dense*) or various combinations of these techniques (*hybrid*). Till the advent of commodity RGB-D sensors, most approaches focused on monocular color cameras where, excluding those using prior knowledge, utilized frame-to-(key)frame based camera localization [24, 21, 26, 37, 19] and generated the environment's map either through bundle adjustment [24, 26, 37, 19] or graph optimization [21, 34, 8]. However, sparse features provide inadequate information about the 3D environment, and thus, direct alignment approaches like MonoSLAM [6], DTAM [28] and LSD-SLAM [10] appeared and, in conjunction with incremental model building and frame-to-model pose estimates, offered a promising alternative that allows for higher quality scene reconstruction, interactions and rendering. As this work focuses on RGB-D camera localization, the reader is referred to an extended survey [25] offering more details about monocular methods.

2.1 Dense RGB-D Camera Pose Estimation

One of the first methods to use dense depth information for camera pose estimation was KinectFusion [27], where depth frames are registered using ICP with projective data association to an incrementally fused volumetric representation (TSDF). After each pose estimate, the current frame was integrated into the volume, further refining and expanding it. This allowed for more accurate tracking as each new pose was estimated against a continuously de-noised model instead of the sensor's noisy output. Another approach is to minimize the photometric error between consecutive RGB-D frames and was shown to have better performance in typical handheld camera tracking with small displacements [33]. Naturally, the combination of simultaneous photometric and geometric error minimization provides a more robust and accurate method as the disadvantages and limitations of each error formulation are complemented by the other term. Color based estimation is sensitive to lighting variations, non diffuse surfaces and minimal texture, while relying on depth only is prone to errors when dealing with planar regions and is also an inherently noisier representation. The color [33] and depth [27] terms' influence on the minimized function are either weighted heuristically [38] or empirically [44, 43, 13].

A probabilistic formulation of camera pose estimation minimizing a color consistency term was contributed by [18] which allowed for the incorporation of a sensor noise model, and was later extended in DVO-SLAM [17] to also include geometric consistency in the form of a point-to-point error term, contrary to KinectFusion's point-to-plane error. However, by modeling the outliers with a t-distribution the more prone to noise point-to-point error was shown to be alleviated. In addition, the effect of the color and depth terms was automatically weighted and adapted based on the t-distribution model. Further extensions accommodated rolling shutter cameras [16] and the addition of depth sensor specific noise models for either infrared pattern projection (Kinect 1, Intel RealSense) [2] or ToF (Kinect 2) [41] sensors. In [11], it was also extended to an inverse depth formulation, also shown to better model the non-linear error distribution of the depth term.

As demonstrated by KinectFusion, camera pose estimation accuracy greatly increases when estimating the sensor's pose with respect to a fused model instead of the previous frame, typically implicitly represented by a Truncated Signed Distance Function (TSDF) [4]. While TSDF models are limited in the space they can cover, volumetric hashing [29] and shifting [43] approaches have extended their effective spatial resolution. Despite their advantages (incremental data integration, implicit de-noising through a running average scheme, surface extraction), they still occupy large amounts of memory and cannot sufficiently model color in high quality as color-per-voxel representations result in blurry renders with little details. Therefore, state-of-the-art SLAM systems resort either to frame-to-keyframe tracking [17], or frame-to-multiframe tracking [5] to exploit the advantages of complete RGB-D tracking, or otherwise omit color information during pose error minimization [39].

2.2 Point-based rendering

Compared to volumetric representations, point-based methods offer the advantage of a unified representation for both rendering and modeling that comes with a lower memory footprint, removing the extra burder of transitioning between volumetric fields and triangulated meshes. Surface elements (surfels) were first introduced in [30] as an alternative geometric type for computer graphics that requires no explicit connectivity and can model continuous and piecewise smooth surfaces. In the context of RGB-D pose estimation, surfels were initially utilized in MRS-SLAM [35] where a space partitioning data structure was used to facilitate the data associations between surfels for pose estimation, and in [15] where the position of each surfel contributed to pose estimation by rasterising hole-free depth maps through the use of a higher resolution index map. Similarly, the SLAM system of [32] utilizes surfels for planar data association but still uses only the depth information during pose tracking by minimizing a point-to-plane error metric. Unlike, previous approaches, ElasticFusion [44] was the first SLAM system to incorporate surfel color into its map model and render temporally proximate surfels in order to use the resultant model color prediction for complete RGB-D frame-to-model pose estimation using both photometric and geometric errors. The surface rendering method used though did not perform high quality surfel blending, but only rasterised elliptical splats to produce a continuous surface, by also exploiting its underlying mapping backend to refine the surfel set. Moreover, highly performing SLAM systems [44, 5] require big amounts of processing power as their accurate tracking is tightly coupled to the map building process. While information can be streamed to powerful workstations, it is not always feasible or within the envisaged scope of an application (e.g. mobile AR). Inspired by ElasticFusion, in this work the surfel based model registration is extended with temporal high quality blending to improve camera pose estimation using a lower amount of extra resources.

3 RGB-D Camera pose estimation

RGB-D cameras produce a stream of color C and depth D data in the image domain Ω:

$$F_t = \langle C_t(\mathbf{p}) \in \mathbb{R}^3, D_t(\mathbf{p}) \in \mathbb{R} \rangle$$

where $\mathbf{p} = (x,y)^{\mathrm{T}} \in \Omega \subset \mathbb{N}^2$ and t being the discrete time point that each frame was generated. Camera pose estimation for a moving camera seeks to find the relative pose $T \in \mathbb{SE}_3$ between two temporally neighboring frames F_t and F_{t-1}, transforming F_{t-1} to F_t, in a direct manner by assuming small camera motion and using projective data association for pixels $\mathbf{p}$ between consecutive frames. A pinhole camera projection model is used for both color and depth cameras with π denoting the projection function of a vertex $\mathbf{v} \in \mathbb{R}^3$, $\mathbf{p} = \pi(\mathbf{v})$, and π^{-1} the inverse projection function $\mathbf{v}(\mathbf{p}) = \pi^{-1}(\mathbf{p}, D(\mathbf{p}))$, producing the vertex map in the image domain Ω from depth map D.

3.1 Iterative Closest Point

One approach to solve this problem is the iterative closest point algorithm (ICP) used to minimize a geometric point-to-plane error metric:

$$E_{geom}(T) = \sum_{\mathbf{p}\in\Omega_t} \left\| \left(T^{-1}\mathbf{v}_t(\mathbf{p}) - \mathbf{v}_{t-1}(\mathbf{p})\right) \mathbf{n}_{t-1}(\mathbf{p}) \right\|_2^2, \qquad (1)$$

where $\mathbf{n}(\mathbf{p}) \in \mathbb{R}^3$ is a map of normal vectors associated with each pixel $\mathbf{p}$ in the image domain Ω. The lifting of $\mathbf{v}$ to a homogeneous vector $\in \mathbb{R}^4$ to enable transformation by pose T^{-1}, is implicitly assumed. After extracting the scalar valued intensity image $I(\mathbf{p}, C) \in \mathbb{R}$, a photometric error can also be defined:

$$E_{photo}(T) = \sum_{\mathbf{p}\in\Omega_t} \left(I_t(\mathbf{p}) - I_{t-1}(w^-(T^{-1},\mathbf{p}))\right)^2, \qquad (2)$$

with $w^-(T^{-1},\mathbf{p}) = \pi(T^{-1}\mathbf{v}_t(\mathbf{p}))$ being a warping function from I_t to I_{t-1}. By combining both error metrics in a weighted manner, the robustness of pose estimation can be increased, with the final minimized energy being:

$$E_{icp} = E_{photo} + \lambda_{geom} E_{geom}\,. \qquad (3)$$

During solving, a minimal *twist* $\xi \in \mathfrak{se}_3$ representation of pose $T^{-1} = exp(\xi)$ is applied. While the energy of equation 3 is nonlinear, the small motion assumption allows for solving with acceptable accuracy using Gauss-Newton iterations i of linear approximations ξ_i, combined to form the final pose estimate ξ. To increase leniency to larger motions, a coarse-to-fine pyramidal scheme in the image domain Ω is also utilized, as the linear approximation assumption can only hold for small twists ξ_i near the identity.

3.2 Probabilistic Formulation

A probabilistic formulation of the above problem can also be derived where the color e_{photo} and depth e_{depth} error metrics:

$$e_{photo}(T) = I_t(w^+(T,\mathbf{p})) - I_{t-1}(\mathbf{p}), \mathbf{p} \in \Omega_{t-1} \qquad (4)$$

$$e_{geom}(T) = D_t(w^+(T,\mathbf{p})) - \lfloor T\mathbf{v}_{t-1}(\mathbf{p}) \rfloor_z, \mathbf{p} \in \Omega_{t-1} \qquad (5)$$

with $w^+(T,\mathbf{p})$ being a warping function from F_{t-1} to F_t and $\lfloor\cdot\rfloor_z$ extracting the z coordinate of a given point. Again, using a minimal *twist* ξ parameterisation (with $T = exp(\xi)$ in this case), these are combined into a bi-variate random variable $\mathbf{r}(\xi) = (\mathbf{e}_{photo}(\xi), \mathbf{e}_{geom}(\xi))^{\mathrm{T}}$ instead of linearly like in (3). This joint modeling lifts the underlying error independence assumption and the extra linear weight λ_{geom} parameter setting. Then, the objective is to estimate ξ by maximizing the posterior probability

$$\boldsymbol{\xi}_{MAP} = \arg\max_{\boldsymbol{\xi}} \mathrm{p}(\boldsymbol{\xi} \,|\, \mathbf{r})\,. \qquad (6)$$

Besides the advantage of joint modeling of the two errors, this formulation also offers the potential to directly incorporate suitable error models into the probability $\mathrm{p}(\boldsymbol{\xi} \,|\, \mathbf{r})$. Previous work [18, 17] showed that the error functions are better modeled by a zero mean bi-variate t-distribution $\mathrm{p_t}(\mathbf{0}, \Sigma, \nu)$, with $\nu = 5$ degrees of freedom and Σ its covariance matrix. After defining the probability of equation 6 as a t-distribution it then becomes an iteratively re-weighted least squares problem:

$$\boldsymbol{\xi} = \arg\min_{\boldsymbol{\xi}} \sum_{\mathbf{p}\in\Omega_{t-1}} w_{\mathbf{p}} \mathbf{r}_{\mathbf{p}}^{\mathrm{T}} \Sigma^{-1} \mathbf{r}_{\mathbf{p}}\,, \qquad (7)$$

with more details offered in [18, 17]. Other work has also followed extending the formulation with a sensor specific noise model [2], further improving the overall pose estimation performance. Similar to the ICP approach, minimizing equation (7) involves Gauss-Newton iterations in a coarse-to-fine scheme.

4 Surfel Model Camera Tracking

Surfels (surface elements) [30], a point-based representation with no explicit connectivity information, are defined as oriented disks that locally approximate a small neighborhood of a surface. Each surfel's $\boldsymbol{s}$ attributes form a tuple: $\boldsymbol{s} = \langle \boldsymbol{v}, \boldsymbol{n}, \boldsymbol{c}, r \rangle$, with a position $\boldsymbol{v} \in \mathbb{R}^3$, an orientation $\boldsymbol{n} \in \mathbb{R}^3$, a color $\boldsymbol{c} \in \mathbb{R}^3$ and a radius $r \in \mathbb{R}$. Unordered sets of surfels constitute a lightweight, in terms of memory and data structure complexity, continuous surface representation. Such point-based representations are suitable for sampled geometry, which fits naturally with RGB-D sensors. Each color and depth frame F_t can be directly mapped to a surfel set

$$S_t(\mathbf{p}) = \langle\, \mathbf{v}_t(\mathbf{p})\,, \mathbf{n}_t(\mathbf{p})\,, C_t(\mathbf{p})\,, \rho_t(\mathbf{v}_t(\mathbf{p}))\,\rangle,\ \mathbf{p} \in \Omega_t$$

with ρ estimating the radius r of each surfel $\boldsymbol{s}$ as a function of its position $\mathbf{v}$ and the camera's characteristics.

4.1 High Quality Surface Splatting

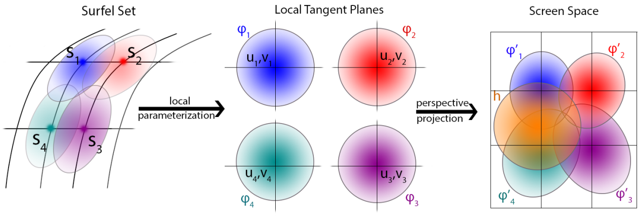

Figure 2: Each surfel s is represented by an oriented disk residing on a plane tangent to the surface. Its reconstruction filter ϕ is locally parameterized on this 2D tangent plane by (u,v). The projection of this filter ϕ' is an ellipsis on screen space, which is band limited by the low-pass filter h.

Surfels can represent and reconstruct continuous surfaces through splatting [45], a computer graphics point-based rendering technique that renders high quality surfaces from unconnected sampled geometry. Previous approaches using surfel splatting for camera pose estimation [44, 32, 15] only used a single pass approach that is prone to z-fighting and does not accurately blend the contributions of neighboring surfels, resulting in loss of detail, lower quality color output and susceptibility to noise.

Elliptical Weighted Average (EWA) [46, 47] splatting offers higher quality results by minimizing aliasing and blurring. EWA is based on the concept of Gaussian *resampling filters* that are formed by the combination of a Gaussian *reconstruction filter* and a screen space *low-pass filter*. EWA offers high quality renders due to its anti-aliasing ability, as a result of the band-limiting screen space filter (Figure 2). Each surfel partakes in surface reconstruction through its local tangent plane, defined by vectors $(\boldsymbol{u}, \boldsymbol{v})$ perpendicular to the surfel's normal, forming a local coordinate system. A 2D Gaussian *reconstruction filter* $\phi_k(\mathrm{u},\mathrm{v})$, defined on the tangent plane and centered on the surfel's position $\mathbf{v}$, reconstructs the surface in screen space in a radius r area by projecting ϕ_k to the rendered image domain, yielding the screen space warped filter $\phi'_k(\mathrm{u},\mathrm{v})$, typically called a "splat". Each tangent plane point's (u,v) influence is weighted by its distance from the surfel position. The accumulation of all reconstruction filters in screen space renders the final continuous surface's attributes a:

$$g(\mathbf{x}) = \sum_k \phi'_k(\mathbf{x}) a_k, \text{for } a_k \begin{cases} \mathbf{v}_k, \text{rendering the surface's depth image} \\ \mathbf{n}_k, \text{rendering the surface's normals} \\ \mathbf{c}_k, \text{rendering the surface's color image} \end{cases}$$

where $\mathbf{x}$ are screen space coordinates. To reduce aliasing each ellipse (the projection of a disk) is band-limited by convolving with a 2D screen space Gaussian h resulting in the final *resampling filter* $g'(\mathbf{x}) = g(\mathbf{x}) \otimes h(\mathbf{x})$, with the overall effect resembling anisotropic filtering, and being able to produce high quality output even at extreme perspectives.

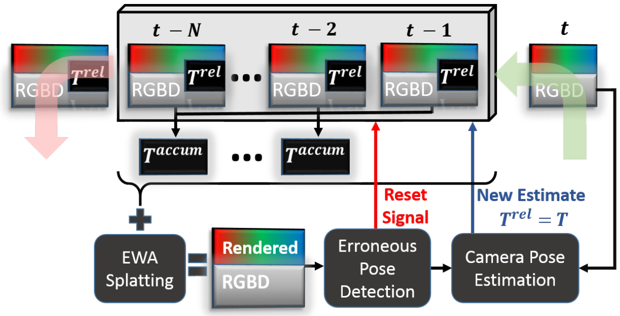

Figure 3: EWA temporal model pipeline: Each new frame t is aligned with the rendered frame consisting of N consequent frames splatted to frame $t-1$ via their relative poses T_i^{rel}, forming a pose-chain to the most recent frame $t-1$. Before estimating the camera's pose T, the splatted frame's visual quality score against the previous color frame $t-1$ is calculated, and used to assess the accuracy of the latest pose estimate T_{t-2}^{rel}. If considered erroneous the temporal model is reset, keeping only the previous frame $t-1$ into the sliding window. After alignment with the splatted frame, new frame t gets inserted into the sliding window along with the relative pose estimate $T_{t-1}^{rel} = T$, pushing the oldest frame out.

High quality splatting is implemented on modern GPUs with a multi pass rendering algorithm [3]. Initially, a ***visibility*** pass sets up a z-buffer that will be used for depth testing to determine the visible surfels. To allow for splat blending, the original depth values are offset by a small amount ε along the viewing rays which determines the maximum allowed depth distances of overlapping surfels. Then, in the ***attribute*** pass, the weighted contributions of each splatted surfel are additively blended for all attributes a. Finally, since the weighted contributions of all splats are irregular and require normalizing, a ***normalization*** pass divides the accumulated weighted contributions by the weight sum to produce the final rendered attribute output.

4.2 Temporal Surfel Model

Given a stream of RGB-D frames we can estimate the pose of each new frame F_t with a splatted frame instead of the previous one, by exploiting the stream's temporality and accumulating a window of N previous frames into a higher quality one (Figure 1). The challenge lies in the uncertainty involved with noisy sensor measurements and imperfect pose estimates. While implicit representations like TSDFs can deal with these issues, surfels are the more appropriate explicit geometry representation and lightweight data structure to increase robustness to uncertainty, mainly due to their simplicity, the removal of topology bookkeeping and their better accommodation of noisy and duplicate measurements through the high quality blending and robust averaging of EWA splatting.

A temporal surfel model is defined within a window of N neighboring frames:

$$F_{t-N}, F_{t-(N-1)}, \cdots, F_{t-1}$$

and $N-1$ relative poses between consecutive frames:

$$T_{t-N}^{rel}, T_{t-(N-1)}^{rel}, \cdots, T_{t-2}^{rel},$$

with $T_i^{rel} : F_i \rightarrow F_{i+1} \in \mathbb{SE}_3$ transforming frame i to $i+1$ (Figure 3).

Each frame F_i within this temporal window can be aligned with the most recent frame $t-1$, through the accumulated pose-chain:

$$T_i^{accum} = \prod_i^{t-2} T_i^{rel}. \tag{8}$$

Therefore, the N frames of the temporal window form a set of surfels on the same coordinate system:

$$\boldsymbol{S} := \{\hat{S}_{t-N}(\mathbf{p}), \ldots, \hat{S}_{t-1}(\mathbf{p})\}, \hat{S}_i(\mathbf{p}) = \mathbb{S}(F_i, T_i^{accum}),$$

where operator $\mathbb{S}$ creates a surfel set from a RGB-D frame and transforms its elements' positions and normals with pose T_i^{accum}. Estimation of the normals from the associated depth map is done through the cross product of central differences.

High quality GPU accelerated EWA splatting of $\boldsymbol{S}$ produces the temporally accumulated rendered model of frame $t-1$. Our implementation is based on [3] which uses an approximation of the complete EWA splatting algorithm [47] to speed up rendering time and increase throughput. After experimenting with [47] we found no discernible quality loss and that the advantages of perspective accurate splatting at grazing and extreme angles are almost non existent when rendering from the sensor's fixed perspective and using a small finite window size that limits the extend of its motion. Thus, the quicker approximation was preferred that enforces a minimum screen space size for each surfel to ensure anti-aliasing. Further, each surfel's radius is computed as $\rho(\mathbf{v}(\mathbf{p})) = \sqrt{2}\lfloor \mathbf{v}(\mathbf{p}) \rfloor_z / f$, with f being the sensor's focal length, inline with previous work [32, 44].

The outcomes are images $\mathbf{D}^r$, $\mathbf{C}^r$, $\mathbf{N}^r$ aligned with the previously streamed frame $t-1$, resulting from the splatting of the position $\mathbf{v}$, color $\mathbf{c}$ and normal $\mathbf{n}$ attributes respectively. Therefore, each new frame's pose is estimated against the temporal model's splatted outcome $F^{\mathbf{r}} = \langle \mathbf{C^r}, \mathbf{D^r} \rangle$ instead of the previous frame, improving the estimation and then sliding the window to include the new frame along with its more accurately estimated pose. Essentially, $\mathbf{D^r}$ replaces D_{t-1} and I_{t-1} is extracted from $\mathbf{C^r}$ instead of C_{t-1} in equations (1), (2), (4), (5). This sliding process allows us to exploit temporal coherence to reduce the rendering processing time by modifying the visibility pass. Instead of splatting all surfels, we can transform the previously splatted depth image with the new pose estimate and re-splat it again, to render the z-buffer that is used to discard occluded surfels during the blending pass, which in turn will produce the new splatted model images that the incoming frame will be registered to.

The depth $\mathbf{D}^r$ and normal $\mathbf{N}^r$ images are hole free and de-noised, while the color image $\mathbf{C}^r$ does not suffer from color bleeding, excessive blurriness or loss of detail plaguing other representations like colored TSDFs (Figure 4). In addition, we observe crisp image edges and textures that can better guide the minimization process. It should be noted that the color image could be neglected and only the de-noised depth and normal images be used for pose estimation, still offering a lighter alternative to a complete TSDF model fusion and tracking.

4.3 Erroneous Pose Estimate Detection

Despite the progress being made with dense camera pose estimation and the evidently higher accuracy results, in practice they are not always of sufficient accuracy. Be it either because of fast camera motion violating the linearization assumption as described in Section 3, large rotations that prevent solving from converging or due to external reasons like varying frame rates, automatic exposure, featureless regions and problematic materials (specular, reflective, absorbing), erroneous pose estimates are an issue that needs to be circumvented. SLAM systems address this by identifying loop closures and refining the pose- graph or chain using bundle adjustment or graph error propagation techniques. Recent work has also harnessed the power of latest generation GPUs to globally optimize keyframes as they are being inserted to the map [5]. These techniques exploit the back-end mapping process running in parallel with camera tracking. At the same time, purely from a camera pose estimation perspective and de-coupled from the mapping process, estimates are judged and accepted based on re-projection error thresholds [5], total residual cost, percentage of valid correspondences or the solved system's covariance [44, 27].

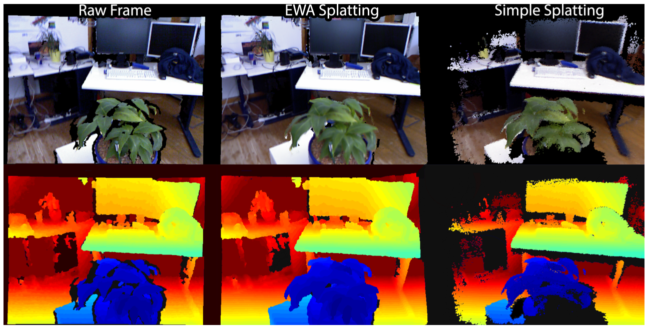

Figure 4: Visual comparison of the EWA temporal model (**middle column**) splatted color (**top**) and depth (**bottom**) against the raw input (**left column**) and the result of the surfel model of [44] (**right column**). Best viewed in color.

In our temporal model building approach, highly erroneous pose estimates will accumulate and corrupt the model as their influence will not be limited to a single time point, but instead be mixed into all subsequent frames within the window. While there is a certain amount of tolerance, as a result of blending multiple contributions, in practice it was observed that the aforementioned checks could not sufficiently reject erroneous estimates. However, one advantage of the proposed temporal splatting scheme, is that unlike approaches using only the previous frame's color information C_{t-1} along with a TSDF representation for geometry, we can assess the quality of the pose estimate directly through visual quality analysis. Given that EWA splatting produces a high quality - within the limits of sensor noise - photometric model when receiving accurate poses, errors in poses would manifest as structural distortions in the rendered color image. Therefore, by analyzing the visual quality of the splatted color image $\mathbf{C}^r$ against the color of the previous frame C_{t-1} the detection of errors in pose estimation is possible.

We use the established Structural Similarity Index Metric (SSIM) [40] for visual quality analysis. The SSIM for two image patches $\mathbf{x}$ and $\mathbf{y}$ of size $M \times M$ involves the joint comparison of 3 terms, the luminance $\mathbf{l}(\mathbf{x},\mathbf{y})$, contrast $\mathbf{c}(\mathbf{x},\mathbf{y})$ and structural $\mathbf{s}(\mathbf{x},\mathbf{y})$ scores, weighted by $\alpha, \beta, \gamma > 0$ respectively, and defined as: $SSIM(\mathbf{x},\mathbf{y}) = \mathbf{l}(\mathbf{x},\mathbf{y})^{\alpha}\, \mathbf{c}(\mathbf{x},\mathbf{y})^{\beta}\, \mathbf{s}(\mathbf{x},\mathbf{y})^{\gamma}$, with

$$\mathbf{l}(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \mathbf{c}(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$\mathbf{s}(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3},$$

where μ_x, μ_y and σ_x, σ_y are the mean and standard deviations of patches $\mathbf{x}$ and $\mathbf{y}$, and σ_{xy} their cross-covariance. The similarity between two images $\mathbf{X}$ and $\mathbf{Y}$ is the mean SSIM score over all their corresponding image patches.

Consequently, we can determine the accuracy of the latest pose estimate through the similarity $ssim_t = SSIM(\mathbf{C}^r, C_{t-1})$. One approach would be to ascertain a threshold for the SSIM but after examining multiple RGB-D sequences from various datasets and extracting their similarity score time series, no optimal parameter could be determined that would detect errors accurately. Thus, a sliding window statistics approach was opted for. We search for erroneous estimates within the similarity measurement time series $ssim_t$ by utilizing the median absolute deviation (MAD) [22] which for a time series $A(t)$ is defined as $MAD(A) = median(|A - median(A)|)$. Using the median is a more robust variability measure than the mean and standard deviation, which is largely influenced by outliers [22]. We can approximate the standard deviation for a normal distribution as $1.4826 \times MAD$ and therefore consider a measurement t as an outlier when its absolute deviation from the median exceeds 3 or 4 times the MAD. Instead of using the complete stream we resort to performing this check within a window W of the latest similarity measurements. We also tried the PSNR metric, but did not find it to perform adequately as the resultant time series was very unstable and noisy. Upon identifying an outlier, we can prevent its accumulation into the temporal model by reseting it, i.e. emptying the window and starting to sliding it again from the current time point, effectively reducing its detrimental effect to the following frames' estimates.

5 Results

We evaluate the temporal surfel model using two state-of-the-art RGB-D camera pose estimation methods, representative of the currently two most established approaches. The DVO probabilistic method of [17] and the linearly weighted method of [44] are used, with the first one using a pure depth error for the geometric term and the second one a point-to-plane error metric (the splatted normal map is also utilized in this case), and both using the same photoconsistency term. They will be referred to as DVO and ICP respectively for the remainder of this section. The parameters of each method are fixed across all experiments with DVO implemented efficiently on the CPU taking advantage of SIMD instructions and ICP implemented on the GPU using CUDA. DVO is calculated using a 4-level pyramid and a precision threshold of $5e^{-7}$ and ICP using a 3-level pyramid with a maximum of $4, 5$ and 10 iterations per level in a coarse-to-fine order. The linear weight between the geometric and photometric terms is set to $\lambda_{geom} = 10$. While residuals are modeled with a t-distribution in DVO, the ICP method discards outliers based on thresholds $\varepsilon_d = 10$ cm for the depth distance and $\varepsilon_a = 20^{\text{o}}$ for the normal angle between correspondences. In addition, a color only rotation estimation step is applied to the ICP method for a maximum of 10 iterations, used to bootstrap the initial pose estimate's rotation component. This step minimizes:

$$E_{rot}(\omega) = \sum_{\mathbf{p}\in\Omega_t} \left(I_t(\mathbf{p}) - I_{t-1}(w_{so_3}(\omega,\mathbf{p}))\right)^2, \omega \in \mathfrak{so}_3 \qquad (9)$$

with $w_{so_3}(\omega,\mathbf{p}) = \pi(exp(\omega)\pi^{-1}(\mathbf{p}))$. Finally, each incoming frame's depth image is bilaterally filtered to reduce noise and produce higher quality normals.

We use the RMSE of the translational component of the relative pose error (RPE) metric, as defined in [36], measuring the drift of the estimated trajectory over a fixed interval, set to one second in line with previous works. In the following subsections we present results on three publicly available datasets, consisting of synthetic and realistic data using various sensors, and focusing on handheld trajectories in area exploring or complete object coverage scenarios, as is usually the case for AR and 3D reconstruction applications.

5.1 ICL-NUIM

The ICL-NUIM dataset [12] comprises synthetic ray-traced sequences of virtual indoor environments with both noise-free and noisy offered, with the noisy ones containing artificial noise. We only used the noiseless ones to assess performance gains when using high quality RGB-D input data. Table 2 presents the results for four ICL-NUIM sequences for both ICP and DVO tracking, comparing the performance of frame-to-frame (F2F) tracking and with the use of our temporal surfel model (EWA) without the reset functionality (Section 4.3), using a window size of $N = 7$ frames. Due to the high quality of the synthetic data, which are not plagued by missing and noisy depth measurements, even the frame-to-frame tracking is very robust with the overall errors lying in the millimeter range. We observe that in noise-free data the temporal surfel model, does not improve the performance of camera pose estimation, and even slightly deteriorates the results, although the differences usually lie in the sub-millimeter range and can thus can be considered as negligible. Of course, real world depth data are of worse quality, and as we will show in the next subsections, the EWA temporal surfel model generally improves camera pose estimates.

ICL-NUIM Sequence	ICP (cm / s)		DVO (cm / s)	
	F2F	EWA	F2F	EWA
living room 1	**0.189**	0.207	**0.172**	0.204
living room 2	0.466	**0.465**	**0.452**	0.467
office room 1	**0.444**	0.450	**0.447**	0.464
office room 2	0.503	**0.499**	0.506	**0.500**

Table 2: The RMSE of the RPE metric (cm / s) of frame-to-frame and frame-to-temporal-model tracking on the synthetic ICL-NUIM sequences for both ICP and DVO methods.

5.2 TUM

The TUM RGB-D dataset [36] contains a variety of hand-held trajectories fitting both AR scenarios (desk and room navigations) or 3D reconstruction (object navigations). Two types of sensors are used, the Microsoft Kinect 1.0 and the Asus Xtion with both sensing depth through a projected infrared pattern. Again we present results for both ICP and DVO methods, comparing the frame-to-frame performance with the EWA surfel temporal model, with the reset functionality disabled, using the same window size of $N = 7$. Furthermore, we also compare our EWA approach to a simpler splatting approach [44, 15, 32] , where each surfel is rendered as a splat, with no explicit blending, aimed mainly at hole filling and accumulating temporal information into the rendered images, after also de-coupling it from the mapping process.

TUM Sequence	DVO (cm / s)		%	ICP (cm / s)		%
	EWA	EWA+R		EWA	EWA+R	
fr1/desk	4.735	**2.820**	+40%	2.897	**2.851**	+2%
fr1/desk2	4.830	**4.747**	+2%	4.599	4.599	±0%
fr1/room	6.270	**6.194**	+1%	**5.518**	5.533	±0%
fr1/teddy	**6.671**	6.758	-1%	5.009	**4.804**	+4%
fr2/desk	1.612	**1.556**	+3%	1.295	**1.277**	+1%
fr3/office	2.067	**1.884**	+9%	1.379	**1.313**	+5%
fr1/plant	5.425	**5.029**	+7%	2.647	**2.554**	+4%
fr3/cabinet	**4.016**	4.091	-2%	2.914	**2.851**	+2%
	total		**+7%**	**total**		**+2%**

Table 3: Reseting the model upon identifying erroneous pose estimates improves the accuracy of both methods. A comparison between the temporal surfel model with and without reseting capabilities is shown for both DVO and ICP methods. The RMSE of the RPE metric (cm / s) is used.

As presented in Table 1, we observe an overall reduction of drift when using temporal EWA splatting (12% for DVO, 28% for ICP). However, in the case of DVO for sequences *fr1/desk* and *fr1/plant* we find that frame-to-frame pose estimation produces less drift. These, along with*fr1/desk2* and*fr1/room* where smaller gains are observed, are challenging sequences with large motion (either translational or rotational) that introduce erroneous pose estimates.

TUM Sequence	DVO (cm / s)			DVO EWA %		ICP (cm / s)			ICP EWA %	
	F2F	Simple	EWA	vs F2F	vs Simple	F2F	Simple	EWA	vs F2F	vs Simple
fr1/desk	**4.381**	5.051	4.735	-8%	+6%	3.948	**2.874**	2.897	+27%	-1%
fr1/desk2	5.698	5.029	**4.830**	+15%	+4%	6.385	4.948	**4.599**	+28%	+7%
fr1/room	6.738	6.874	**6.270**	+7%	+9%	5.838	**5.468**	5.518	+5%	-1%
fr1/teddy	7.226	**6.444**	6.671	+8%	-4%	**4.902**	6.406	5.009	-2%	+22%
fr2/desk	2.511	1.760	**1.612**	+36%	+8%	2.134	9.533	**1.295**	+39%	+86%
fr3/office	3.776	4.590	**2.067**	+45%	+55%	3.058	1.686	**1.379**	+55%	+18%
fr1/plant	**3.611**	4.903	5.425	-50%	-11%	3.303	3.025	**2.647**	+20%	+12%
fr3/cabinet	7.004	5.666	**4.016**	+43%	+29%	6.555	3.747	**2.914**	+56%	+22%
		total		**+12%**	**+12%**		**total**		**+28%**	**+21%**

Table 1: The RMSE of the RPE metric (cm / s) on the TUM sequences using the DVO and ICP methods. A comparison of frame-to-frame and frame-to-temporal-model tracking is presented, showcasing the gains when using the EWA temporal model. In addition, results against a simple splatting model are also presented along with the gains of the EWA temporal splatting against the simpler splatting scheme.

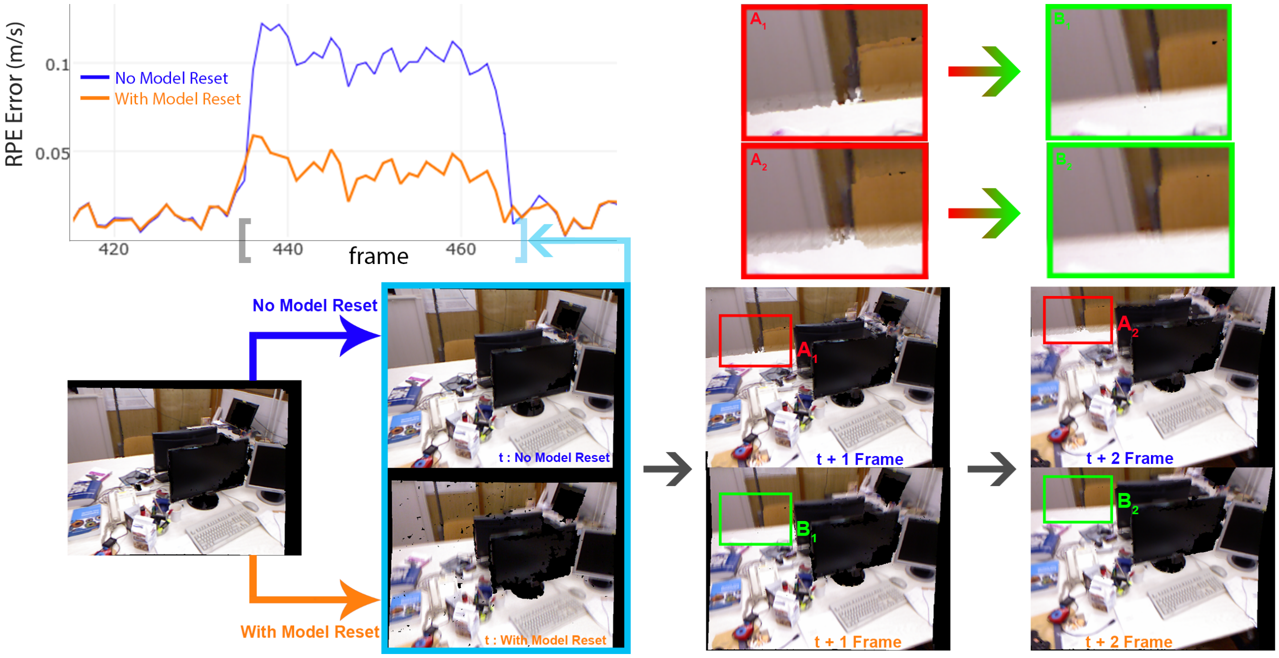

Figure 5: The erroneous pose detection results are presented. The top plot shows the RPE metric for each frame of the *fr1/desk* sequence. On the left is the initial splatted frame at time point $t-1$. The blue arrow sequence (and RPE plot) shows the stream advancing without the reset capability toggled, while the orange arrow shows the same stream sequence with the reset capability turned on. At time point t, denoted by the larger acqua box, the orange sequence detects an erroneous estimate due to an outlier in the visual quality time series and resets. At time points $t+1$ and $t+2$ it is evident that the erroneous estimate is distorting the results, as pinpointed and illustrated in zoom in the green and red boxes. Meanwhile, the orange sequence shows crisper results and allows for more accurate subsequent registrations. The RPE plot on top shows the resulting drift with the erroneous pose accumulating into the model (blue series), and the reduction when it is reset (orange series). It should be noted that due to how the RPE is calculated, the error is peaking at an earlier time point than the erroneous pose estimate.

The sliding window approach will inherently accumulate these errors into subsequent frames, corrupting the EWA splatting model. Comparing with the simple splatting approach, we observe an overall drift reduction of 12% and 21%, for DVO and ICP respectively. Additionally, we observe that in some of the challenging sequences, like *fr1/desk*, *fr1/teddy* and *fr1/room*, the simple splatting approach performs slightly better. By omitting the visibility pass, the final rendering results of simple splatting only allow for a single contribution per fragment, as dictated by the depth testing process. As a result, in the case of erroneous poses the outcome greatly depends on how the last frame will be posed and, depending on the type of motion involved, might occlude or get occluded by previous frames. While the mapping process can greatly alleviate such issues as demonstrated by [44], when the temporal simple splatting is de-coupled from it, we find that EWA splatting performs better.

Consequently, we evaluate our approach with the reset mechanism outlined in Section 4.3, referred to as EWA+R, in the same sequences for both RGB-D camera pose estimation methods. We consider as visual quality outliers, those measurements exceeding 4 standard deviations as defined by the MAD within a temporal window of 21 frames, which was empirically selected after testing several indicative window sizes. The results are illustrated in Table 3, with the overall percentage gains in drift reduction also presented. As it can be seen, for the *fr1/desk* sequence, where the frame-to-frame tracking produced better results than the temporal EWA splatting due to erroneous pose accumulation and corruption of the aggregated splats, the error is greatly reduced after identifying those erroneous poses. Figure 5 showcases one such outlier detection and its detrimental effects on the temporal model. At the same time, the gains after resetting the temporal window are clearly visible. It is also apparent from the RPE plot that an erroneous pose estimate greatly influenced the sequence's drift and that its effects were reduced after the erroneous pose identification. Overall, we observe that in most sequences the visual quality analysis outlier detection improves the temporal model's performance or at least keeps it at the same levels. More similar examples can be found on the supplementary video, as well as visual comparison between the tracking of complete sequences with frame-to-frame, frame-to-temporal-model and frame-to-simple-splatting.

Finally, in Table 4 we compare our results against previous SLAM systems, utilizing a mapping and optimization back-end on a common intersection of sequences. The first is RGB-D SLAM [9], whose trajectories are publicly available, and the second is a recent work focusing on improving the estimation accuracy of DVO by better modeling depth errors [2], using the results reported in their work. The third is ElasticFusion [44] where we used the author's publicly available implementation. While already offering better drift reduction, our proposed model can also benefit from sensor noise modeling approaches like in [2]. It should be noted though that we do not perform pose-chain optimization of any kind, and therefore complete SLAM systems offer a better performance with respect to the global trajectory as measured by the absolute trajectory error (ATE) [36].

TUM Sequence	DVO EWA+R	RGB-D SLAM	σ-DVO	ICP EWA+R	Elastic Fusion
fr1/desk	**2.820**	3.325	3.900	2.851	**2.844**
fr1/desk2	**4.747**	5.441	6.500	**4.599**	4.636
fr1/room	**6.194**	9.536	6.300	**5.533**	5.846
fr2/desk	**1.556**	1.670	1.600	**1.277**	1.464
fr3/office	1.884	-	**1.400**	**1.313**	1.323

Table 4: Comparison with other SLAM systems using the RMSE of the RPE (cm / s). For RGB-D SLAM the publicly available trajectories are used, for sigma-DVO we use the numbers reported in their published work, while for ElasticFusion [44] we used the author's implementation.

5.3 CoRBS

In order to demonstrate sensor invariance, our approach is also evaluated with the CoRBS dataset [42], captured by a Microsoft Kinect 2.0, a time-of-flight depth sensing device. Since the Kinect 2.0 offers a higher resolution color image, we downsampled it to the depth image's resolution. Table 5 shows similar gains to those presented in Section 5.2 using the ICP method. In general, the EWA temporal model offers a reduction in pose estimation drift, while the erroneous pose detection makes it a bit more robust.

5.4 Performance

The experiments were run on a system with an Intel i7-4790K @ 4GHz CPU with 16GB of memory and a NVIDIA GTX 960 GPU with 2GB of memory. In Table 6 an analysis regarding performance gains versus resources used is presented. Increasing the window size reduces drift, but at the cost of memory. However, considering a standard resolution of 640 × 480 for both color and depth and a window of size $N = 7$ plus the overhead of the allocated framebuffers needed for splatting, the amount of extra memory required (around 22 MB) is 17% of a TSDF voxel grid of resolution 256^3 (around 128 MB for the distance and weight values).

CoRBS Sequence	ICP (cm / s)			EWA	EWA+R
	F2F	EWA	EWA+R	vs F2F %	vs EWA %
D1	3.553	2.153	**2.034**	+39%	+6%
D2	3.203	2.004	**1.960**	+37%	+2%
D3	10.437	5.997	**5.718**	+43%	+5%
D4	4.382	2.563	**2.444**	+42%	+5%
D5	4.392	**2.988**	3.033	+32%	-2%
E1	3.818	**3.452**	3.484	+10%	-1%
E2	3.332	2.398	**2.311**	+28%	+4%
E4	2.730	**2.453**	2.445	+10%	±0%
E5	2.454	1.820	**1.780**	+26%	+2%
H1	2.192	1.650	**1.645**	+25%	±0%
H2	2.426	1.304	**1.269**	+46%	+3%
H3	3.252	2.733	**2.674**	+16%	+2%
		total		**+29%**	**+2%**

Table 5: The RMSE of the RPE metric (cm / s) for the CoRBS dataset sequences. Results are shown for both the ICP method for the frame-to-frame tracking and the frame-to-temporal-model tracking with and without the reseting capability.

However, while increasing the model's sliding window size offers performance gains, it also adds more surfels to the model, and therefore, more fragments during rasterization, decreasing the computational performance. Table 7 presents the time required for each different pass of the EWA splatting algorithm without reset enabled for a window of size $N = 7$. Surfel slatting rendering techniques are bottlenecked by the huge amount of fragments produced during rendering. The temporal visibility check described in Section 4.2 decreases the computational load, however increasing the number of frames, increases the processing time of the blending pass. Most mobile solutions resort to using half the resolution of the input frames [14], with its processing time also reported.

TUM Sequence	Window Size N				
	N = 3	N = 5	N =7	N = 9	N = 11
fr1/desk2	5.467	4.991	4.599	4.358	**4.215**
fr2/desk	1.478	1.299	**1.295**	1.321	1.335
fr3/cabinet	4.222	3.183	2.914	2.440	**2.171**
fr3/office	2.006	1.528	1.379	1.190	**1.125**

Table 6: The RMSE of the RPE metric (cm / s) is shown for the ICP method using different window sizes. Bigger window sizes typically perform better than smaller ones. The EWA method without resetting is used.

Shader		Resolution			
		Half (ms)		Full (ms)	
Visibility Pass	Temporal Visibility Pass	5.36	1.47	11.88	1.47
Blending Pass		6.77		16.07	
Normalization Pass		0.04		0.04	

Table 7: The processing time (in milliseconds) for each shading pass is reported. We obtain higher performance by utilizing the temporal visibility pass. The timings for using half of the original resolution are also reported.

TUM Sequence	DVO (cm / s)			DVO EWA+R %		ICP (cm / s)			ICP EWA+R %	
	F2F	EWA	EWA+R	vs F2F	vs EWA	F2F	EWA	EWA+R	vs F2F	vs EWA
fr3/sitting_halfsphere	**3.767**	7.430	6.694	-78%	+10%	**4.981**	5.030	5.029	-1%	±0%
fr3/sitting_static	1.330	1.124	**1.041**	+22%	+7%	1.874	1.436	**1.332**	+29%	+7%
fr3/walking_halfsphere	39.391	32.530	**31.769**	+19%	+2%	36.376	30.906	**30.256**	+17%	+2%
fr3/walking_static	29.562	26.630	**25.866**	+13%	+3%	**24.892**	34.022	29.872	-20%	+12%
fr3/walking_xyz	**46.548**	50.040	48.527	-4%	+3%	47.842	43.627	**42.472**	+11%	+3%
		total		**-6%**	**+5%**		**total**		**+7%**	**+5%**

Table 8: The RMSE of the RPE metric (cm / s) on the TUM sequences containing moving objects using DVO and ICP methods. A comparison of frame-to-frame and frame-to-temporal-model (with and without reseting) tracking is presented, also presenting the gains of each method.

6 Conclusion

We have presented a temporal surfel EWA splatting model to improve the accuracy of camera pose estimation methods. Compared to the simple splatting scheme used by previous works, we have demonstrated the efficacy of complete EWA splatting with high quality blending in various datasets. It was shown that although gains cannot be discerned for high quality synthetic data, in realistic conditions we observe a reduction of drift in the range of around 25%. The advantage of lightweight point-based representations also lies in their simplicity, alleviating the burden of transitioning between different representations and being mindful of their consistency. Unlike implicit representations like TSDFs, exploiting the RGB-D stream's temporality offers a low memory alternative for achieving registration with a de-noised and accumulated representation and is also de-coupled from the mapping process. Nonetheless, this scheme can be potentially combined with existing SLAM systems, as aggregated frame splatting can be utilized to produce de-noised keyframes. Further, a novel visual quality erroneous pose estimate detection technique was presented to complement the proposed temporal model, showcasing an interesting potential and supporting the temporal EWA model to prevent it from getting corrupt by bad poses estimates.

While the parameters used within this work were experimentally judged as better performing, there were sequences that produced better trajectories through different parameterizations, be it either the window size or the outlier detection parameters. At the same time, certain visible color degradations were not identified by the MAD outlier detector. Finally, realistic AR scenarios involve more challenging scenarios with the most problematic for temporal aggregation methods being those where the scene loses its rigidity. To that end, we also offer results in dynamic scenes from the TUM dataset that include moving objects in Table 8, with some sequences suffering from very high drifts, which constitutes a tracking failure. In these situations, while the temporal aggregation can overally increase the tracking performance and reduce drift in those segments where the scene is rigid, it will deteriorate the pose estimation results when scene objects start to move as they will be blended multiple times in the splatted images. At the same time, the rendered color image's similarity timeseries will be very unstable and fail to reset the window sufficiently to improve the performance as required. Concluding, being able to accurately detect bad pose estimates is still an open research task that still needs to be addressed and also dealing with dynamic scenes is the next natural step towards increasing AR's applicability.

Acknowledgements

This work was supported and received funding from the European Union Horizon 2020 Framework Programme for Research and Innovation Action under Grant Agreement no. 723277 (Factory2Fit).

References

[1] T. Araújo, R. Roberto, J. M. Teixeira, F. Simões, V. Teichrieb, J. P. Lima, and E. Arruda. Life cycle of a slam system: Implementation, evaluation and port to the project tango device. In *Virtual and Augmented Reality (SVR), 2016 XVIII Symposium on*, pages 10–19. IEEE, 2016.

[2] B. W. Babu, S. Kim, Z. Yan, and L. Ren. σ-dvo: Sensor noise model meets dense visual odometry. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pages 18–26. IEEE, 2016.

[3] M. Botsch, A. Hornung, M. Zwicker, and L. Kobbelt. High-quality surface splatting on today's gpus. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 17–141. IEEE, 2005.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[5] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *arXiv preprint arXiv:1604.01093*, 2016.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6), 2007.

[7] G. Dubbelman and B. Browning. Closed-form online pose-chain slam. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5190–5197. IEEE, 2013.

[8] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[9] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[10] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[11] D. Gutiérrez-Gómez, W. Mayol-Cuevas, and J. J. Guerrero. Inverse depth for accurate photometric and geometric error minimisation in rgb-d dense visual odometry. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 83–89. IEEE, 2015.

[12] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and automation (ICRA), 2014 IEEE international conference on*, pages 1524–1531. IEEE, 2014.

[13] P. Henry, D. Fox, A. Bhowmik, and R. Mongia. Patch volumes: Segmentation-based consistent mapping with rgb-d cameras. In *3DTV-Conference, 2013 International Conference on*, pages 398–405. IEEE, 2013.

[14] I. Ihm, Y. Kim, J. Lee, J. Jeong, and I. Park. Low-cost depth camera pose tracking for mobile platforms. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 123–126, Sept 2016.

[15] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb.

Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3DTV-Conference, 2013 International Conference on*, pages 1–8. IEEE, 2013.

[16] C. Kerl, J. Stuckler, and D. Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2264–2272, 2015.

[17] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2100–2106. IEEE, 2013.

[18] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3748–3754. IEEE, 2013.

[19] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[20] D. Lefloch, M. Kluge, H. Sarbolandi, T. Weyrich, and A. Kolb. Comprehensive use of curvature for robust and accurate online surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[21] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[22] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, 2013.

[23] F. Li, Y. Du, and R. Liu. Color-introduced frame-to-model registration for 3d reconstruction. In *International Conference on Multimedia Modeling*, pages 112–123. Springer, 2017.

[24] H. Liu, G. Zhang, and H. Bao. Robust keyframe-based monocular slam for augmented reality. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pages 1–10. IEEE, 2016.

[25] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2016.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[28] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[29] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[30] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.

[31] R. Roberto, J. P. Lima, T. Araújo, and V. Teichrieb. Evaluation of motion tracking and depth sensing accuracy of the tango tablet. In *Mixed and Augmented Reality (ISMAR-Adjunct), 2016 IEEE International Symposium on*, pages 231–234. IEEE, 2016.

[32] R. F. Salas-Moreno, B. Glocken, P. H. Kelly, and A. J. Davison. Dense planar slam. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 157–164. IEEE, 2014.

[33] F. Steinbrücker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE, 2011.

[34] H. Strasdat, J. Montiel, and A. J. Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2010.

[35] J. Stückler and S. Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147, 2014.

[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[37] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao. Robust monocular slam in dynamic environments. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 209–218. IEEE, 2013.

[38] T. Tykkälä, C. Audras, and A. I. Comport. Direct iterative closest point for real-time visual odometry. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2050–2056. IEEE, 2011.

[39] H. Wang, J. Wang, and W. Liang. Online reconstruction of indoor scenes from rgb-d streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3271–3279, 2016.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[41] O. Wasenmüller, M. D. Ansari, and D. Stricker. Dna-slam: Dense noise aware slam for tof rgb-d cameras. In *Asian Conference on Computer Vision Workshop (ACCV workshop), Springer*, 2016.

[42] O. Wasenmüller, M. Meyer, and D. Stricker. Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–7. IEEE, 2016.

[43] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015.

[44] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, volume 11, 2015.

[45] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378. ACM, 2001.

[46] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.

[47] M. Zwicker, J. Räsänen, M. Botsch, C. Dachsbacher, and M. Pauly. Perspective accurate splatting. In *Proceedings of Graphics interface 2004*, pages 247–254. Canadian Human-Computer Communications Society, 2004.