A Coarse and Relevant 3D Representation for Fast and Lightweight RGB-D Mapping

Bruce Canovas, Michele Rombaut, Amaury Negre, Serge Olympieff and Denis Pellerin
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

Keywords: RGB-D, Dense Reconstruction, Superpixel, Surfel, Fusion, Robotics.

Abstract: In this paper we present a novel lightweight and simple 3D representation for real-time dense 3D mapping of static environments with an RGB-D camera. Our approach builds and updates a low resolution 3D model of an observed scene as an unordered set of new primitives called supersurfels, which can be seen as elliptical planar patches, generated from superpixels segmented RGB-D live measurements. While most of the actual solutions focuse on the accuracy of the reconstructed 3D model, the implemented method is well-adapted to run on robots with reduced/limited computing capacity and memory size, which do not need a highly detailed map of their environment but can settle for an approximate one.

1 INTRODUCTION

Live 3D reconstruction from RGB-D data is a major and active research topic in computer vision for robotics. Indeed, to be able to interact in an environment, a robot needs to have access in real-time to its 3D geometry. Several dense visual SLAM and 3D mapping systems able to produce impressive results have been proposed. They enable robots to simultaneously localize themselves, build and update a 3D map of an observed scene.

Depending on the 3D representation they use to model the observed scene, many state of the art 3D reconstruction systems are limited to small areas and/or require heavy and costly hardware to operate properly in real-time. Indeed, most of the softwares operate with really accurate but expensive and memory consuming forms of representation while in many case an approximate one can be sufficient for the needs of a robot which does not have to perform classification tasks, place recognition, or to localize with a centimeter accuracy. A good representation should be adapted to the robotic system running the algorithm and to its mission.

In this paper, we present a method based on a novel representation to build and update iteratively an approximate 3D model of an observed space, as a set of piecewise planar elements called supersurfels, from the segmentation in superpixels of the input live video stream of a moving RGB-D camera. Our contribution is a real-time and memory efficient 3D mapping system, called SupersurfelFusion, that accomplishes rough but light 3D reconstruction.

Our method is not designed to compete with the level of detail of existing dense RGB-D 3D reconstruction approaches. Instead, it aims to enable fast 3D mapping with good scalability on power restricted platforms, or to serve as groundwork for applications that require high efficiency. It has been developed under ROS (Robot Operating System) for flexibility and portability.

2 RELATED WORKS

Most of the actual 3D reconstruction systems with RGB-D camera share a common pipeline, based on a frame-to-model strategy. They build and update a single global3D model of an observed scene, from an input live RGB-D video stream. First the RGB-D measurements are acquired and preprocessed. Then the camera pose is estimated and the data of the current frame are aligned to the predicted 3D model (frame-to-global-model registration strategy). After that, the newly aligned data are integrated/fused into the model. Finally the 3D model is cleaned and can be rendered.

These systems mainly differ in the way they represent the target 3D scene. Volumetric (or voxel-based) representations for 3D reconstruction systems have been popularized with KinectFusion (Izadi et al.,

Canovas, B., Rombaut, M., Negre, A., Olympieff, S. and Pellerin, D.
A Coarse and Relevant 3D Representation for Fast and Lightweight RGB-D Mapping.
DOI: 10.5220/0007381708240831
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 824-831
ISBN: 978-989-758-354-4

2011), one of the first methods to perform real-time dense reconstruction using an RGB-D camera. Systems using volumetric representation build and update a single high-quality 3D model along with live RGB-D data, based on the volumetric fusion method of (Curless and Levoy, 1996). The model is represented as a truncated signed distance function stored in a regular 3D grid volume, known as voxel grid. Volumetric approaches are robust to noises and produce really accurate results but are highly memory consuming and are thus limited to small environments. Furthermore the volume and the resolution of the regular grid have to be predefined. Another drawback of voxel-based representation comes from the need to transit between different data representations: first the RGB-D data are extracted as a point-cloud converted to a continuous signed-distance function, which is then discretised into a regular grid to update the model. Finally the reconstructed model is raycasted to render the resulting reconstruction. Numerous other volumetric approaches have been developed in order to override these drawbacks or to improve the method, such as Kintinuous (Whelan et al., 2012), a spatial extension of KinectFusion, or Patch Volumes (Henry et al., 2013) which presents a more compact representation.

Alternatively, point-based representations may also be used. They represent the model as a set of unordered 3D points or surfels. A surfel is a circular surface element which principally encodes a position, a color, a normal, a radius (the point size), and a confidence value that quantifies the reliability of the surfel (states stable or unstable). A surfel is considered as stable, reliable, if it has been repeatedly observed. In surfel-based approaches, such as PointBasedFusion (Keller et al., 2013) and ElasticFusion (Whelan et al., 2015), the acquired RGB-D data are directly stored and accumulated in a model composed of surfels. Surfel-based representations are directly obtained from an RGB-D frame and can be used for rendering without converting to an other form of representation. Unlike volumetric approaches, point-based systems do not provide a continuous surface reconstruction. However, the resolution of the model does not have to be predefined because each surfel size is adapted to the accuracy limit of the sensor. Furthermore, free spaces do not need to be represented which makes these methods more memory efficient. Even if surfel-based reconstructions are usually lighter than volumetrics, they are still expensive in memory and computation (a 3D model can count about millions of surfels).

Other approaches were developed too, such as (Thomas and Sugimoto, 2013) (Salas-Moreno et al., 2014), proposing to use sets of planes in their 3D scene representation to make it more compact and still achieve accurate reconstruction. The method presented by (Bódis-Szomorú et al., 2014) shares many similarities with ours. Their multi-view stereo algorithm combines sparse structure-from-motion with superpixels to generate a lightweight, piecewise-planar surface reconstruction. However it is not capable of real-time performances.

In the subsequent section, we introduce a new 3D scene representation as a set of supersurfels, which are simply oriented and colored elliptical planar patchs, generated from the superpixel segmentation of RGB-D frames. The use of superpixels allows us to considerably reduce the quantity of data to process while preserving the meaningful information. Supersurfels share similarities with surfels: they can be displayed directly through a traditional graphic pipeline such as OpenGL and are easily updated. They allow us to store a less accurate, but still relevant, and more compact 3D model in memory, so as to achieve fast mapping on multitasks or less performing embedded platforms.

3 PIPELINE AND SUPERSURFEL REPRESENTATION

The system developed, SupersurfelFusion, achieves low resolution 3D reconstruction of static environments using an RGB-D camera. It builds, completes and updates a single global model in world space coordinate $\mathfrak{R}_W$, denoted as $\mathcal{G}$, as a set of unordered 3D primitives named supersurfels (Figure 1). The model $\mathcal{G}$ is updated at each acquisition from the set of supersurfels $\mathcal{F}$ associated to the current frame.

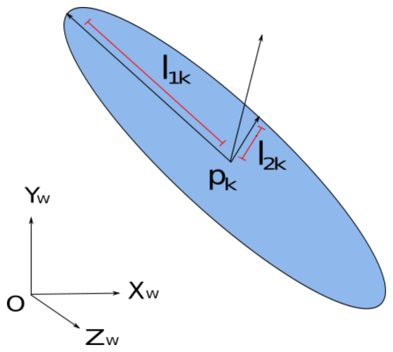

Figure 1: Oriented supersurfel defined in world space coordinate system $\mathfrak{R}_W(O;X_w,Y_w,Z_w)$, with position $\boldsymbol{p_k}$, length $\boldsymbol{l_{1k}}$ and width $\boldsymbol{l_{2k}}$.

A supersurfel can be seen as an approximation of the 3D reprojection (from 2D to 3D space) of an associated superpixel. A Superpixel (Ren and Malik, 2003) defines a group of connected pixels sharing homogeneous informations (color and surface for RGB-D superpixels). A supersurfel $\mathcal{G}^k \in \mathcal{G}$ is simply an elliptical planar patch in 3D space, that encodes its centroid position $\boldsymbol{p_k} \in \mathbb{R}^3$, an orientation $\boldsymbol{R_k} \in \mathbb{SO}^3$, a color $\boldsymbol{c_k} \in \mathbb{R}^3$, longitudinal and lateral elongations $\boldsymbol{l_{1k}} \in \mathbb{R}$ and $\boldsymbol{l_{2k}} \in \mathbb{R}$, a timestamp $\boldsymbol{t_k} \in \mathbb{N}$ to store the last time it has been observed, a confidence weight $\boldsymbol{w_k} \in \mathbb{R}_+$ to quantify its state (stable or unstable) and a 3D covariance matrix $\boldsymbol{\Sigma_k} \in \mathcal{M}_3(\mathbb{R})$ to describe its shape.

The system takes as input live registered RGB-D pairs of images $(\boldsymbol{C_t}, \boldsymbol{Z_t})$, with $\boldsymbol{C_t} : \boldsymbol{\Omega} \rightarrow \mathbb{R}^3$ the color image at time $\boldsymbol{t}$, $\boldsymbol{Z_t} : \boldsymbol{\Omega} \rightarrow \mathbb{R}$ the associated depth map and $\boldsymbol{\Omega}$ the image plane. The 3D reconstruction pipeline can then be divided in 4 steps:

1. First the incoming RGB-D frame is segmented in superpixels.
2. The resulting superpixels are then used to generate a set of supersurfels $\mathcal{F}$ in camera space $\mathfrak{R}_C$.
3. After that the 6DOF pose of the camera in world space $\mathfrak{R}_W$ is estimated from a visual odometry solution such as ORB-SLAM2 (Mur-Artal and Tardós, 2017), a feature-based SLAM system, or the open source OpenCV RGB-D odometry based on the direct frame-to-frame registration method of (Steinbrücker et al., 2011).
4. Finally, the pose of the camera is used to transform current frame supersurfels $\mathcal{F}$ from camera $\mathfrak{R}_C$ to world space $\mathfrak{R}_W$ and enable the update of the model $\mathcal{G}$.

The segmentation in superpixels, the generation of supersurfels and the fusion of data have been implemented on GPU with CUDA library, to benefit from the high computational power of this device. In order for our algorithm to be flexible and easily integrated to a complete robotic system, it has been designed to work under ROS.

4 GENERATION OF SUPERSURFELS

This section describes the process that compute the set $\mathcal{F}$ of current frame supersurfels from color and depth data.

4.1 Segmentation in Superpixels

First, the newly acquired RGB-D frame is segmented into N superpixels $C = \{\boldsymbol{C^h},\ h = 1,...,N\}$. A superpixel is a group of homogeneous pixels $\boldsymbol{u_j}$, for $j = 1,2,...,M$ with M the size of the superpixel, sharing similar color and for which their 3D reprojections can be fit to an identical plane in 3D space. A CUDA implementation of the approach described in (Yamaguchi et al., 2014) is applied to partition the current frame into suitable superpixels, that is to say which preserve as much as possible boundaries and depth discontinuities. The algorithm starts from the breakdown of an image into a regular grid (for instance the image is divided into groups of 20x20 pixels) and iteratively shifts the boundaries of the superpixels by minimizing a cost function which preserves the topology.

4.2 Extraction of Supersurfels

Then a supersurfel $\mathcal{F}^i \in \mathcal{F}$, where $\mathcal{F}$ defines the set of supersurfels associated to the current frame, is generated for each suitable superpixel $\boldsymbol{C^h}$, as shown in Figure 2. The position $\boldsymbol{p_i}$ of the supersurfel $\mathcal{F}^i$, is the mean value of the 3D reprojections $\pi(\boldsymbol{u_j}, \boldsymbol{Z_t}(\boldsymbol{u_j}))$ of the $\boldsymbol{u_j}$ pixels contained in $\boldsymbol{C^h}$. The color $\boldsymbol{c_i}$ is set as the average colors, in CIELAB color space, of the pixels contained in the superpixel. Principal Component Analysis (PCA) technique is adopted to estimate the orientation $\boldsymbol{R_i}$ and the longitudinal and lateral elongations $\boldsymbol{l_{1i}}$, $\boldsymbol{l_{2i}}$. The covariance matrix $\boldsymbol{\Sigma_i}$ is computed according to the following formula:

$$\Sigma_i = \frac{1}{M-1} \sum_{j=1}^{M} (\pi(u_j, Z_t(u_j)) - p_i)(\pi(u_j, Z_t(u_j)) - p_i)^T. \tag{1}$$

The matrix is diagonalized, which gives three eigenvectors $\boldsymbol{e_{1i}}, \boldsymbol{e_{2i}}, \boldsymbol{e_{3i}}$ and their associated eigenvalues $\boldsymbol{\lambda_{1i}}, \boldsymbol{\lambda_{2i}}, \boldsymbol{\lambda_{3i}}$ ($\boldsymbol{\lambda_{1i}} > \boldsymbol{\lambda_{2i}} > \boldsymbol{\lambda_{3i}}$). The orientation $\boldsymbol{R_i}$ corresponds to the eigenvectors, with $\boldsymbol{e_{3i}}$ the eigenvector belonging to the smallest eigenvalue being an estimate for the normal of the supersurfel. The longitudinal and lateral elongations $\boldsymbol{l_{1i}}, \boldsymbol{l_{2i}}$ are respectively defined by the square root of the largest $\boldsymbol{\lambda_{1i}}$ and the medium eigenvalues $\boldsymbol{\lambda_{2i}}$, scaled by a constant factor $\boldsymbol{s}$. For instance, $\boldsymbol{s} = 2.4477$ gives us the 95% confidence ellipse associated to the set of points defined by the $\pi(\boldsymbol{u_j}, \boldsymbol{Z_t}(\boldsymbol{u_j}))$ 3D reprojections. The timestamp $\boldsymbol{t_i}$ simply received the current time $\boldsymbol{t}$. The confidence weight $\boldsymbol{w_i}$ is set to a low value at initialization, corresponding to the ratio between the number of pixels with valid depth (that is to say for which the depth is provided by the depth map $\boldsymbol{Z_t}$) in superpixel $\boldsymbol{C^h}$ and the total number of pixels it contains:

$$w_i \leftarrow pixels_{valid} / pixels_{total}. \tag{2}$$

Disproportionate supersurfels, for which $\boldsymbol{l_{1i}} / \boldsymbol{l_{2i}}$ surpasses a fixed threshold, are subdivided lengthwise in two parts, so as to ensure supersurfels of regular sizes. Like (Keller et al., 2013) with surfels, we discard

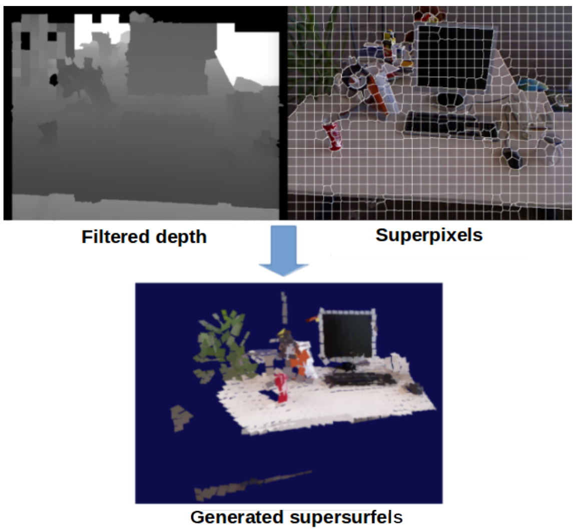

Figure 2: Generation of supersurfels from a superpixels segmented RGB frame and its associated filtered depth (even if considered as elliptical planar patches, supersurfels are displayed as rectangular patches to ease and accelerate the rendering).

any supersurfel having its normal seen under an incidence angle θ larger than θ_{max} and we only generate supersurfels when the depth of the associated superpixels is less than z_{max}, with z_{max} set according to the accuracy of the sensor.

5 MODEL UPDATE

The system maintains and extends a single global model $\mathcal{G}$ as a set of supersurfels $\mathcal{G}^k$ stored in a flat array indexed by $k \in \mathbb{N}$. Newly generated supersurfels $\mathcal{F}$, from the current frame, are either added in the model or merged with similar supersurfels from the model. Merging a supersurfel $\mathcal{F}^i \in \mathcal{F}$ with $\mathcal{G}^k$ will increase the confidence weight w_k of $\mathcal{G}^k$. That way, a supersurfel from the model will change its status from unstable to stable if it is repeatedly observed. As in (Keller et al., 2013), supersurfels with $w_k > w_{stable}$ are considered as stable.

5.1 Data Association

In this step, we look for each newly generated supersurfel $\mathcal{F}^i$, whether a similar supersurfel $\mathcal{G}^k$ already exists in the predicted 3D model $\mathcal{G}$. The purpose is to fuse alike supersurfels, in order to avoid duplications, redundancies, and to refine the model.

First, knowing the current pose of the sensor (provided by ground-truth measurements or estimated by an external visual-odometry system), supersurfels $\mathcal{F}^i$ in $\mathcal{F}$ are aligned with the model. Then, we carry out a nearest neighbours search for each $\mathcal{F}^i$, through a Bounding Volume Hierarchy (BVH) (Kay and Kajiya, 1986). The BVH is build based on the supersurfels from the predicted model which are contained in the field of view of the camera at current time. The search is performed in camera space and the BVH is generated following the implementation on GPU by (Karras, 2012). It is an acceleration structure, which allows to organize the 3D reconstructed scene in a binary search tree of bounding volumes. It thus reduces the time complexity of the research. Leaf nodes are supersurfels from the model wrapped in minimum axis-aligned bounding box. The minimum axis-aligned bounding box of an object is the smallest box containing all of the points of the object and having its edges parallel to the reference coordinate system axes. Leaf nodes are grouped and enclosed by bigger bounding boxes representing the nodes of the tree. Each node is an axis aligned bounding box of its children. All the leaf supersurfels where the associated bounding volumes contain the center p_i of the supersurfel $\mathcal{F}^i$ from the current frame, are added to the set of nearest neighbours.

We use the symmetric Kullback-Leibler divergence to find out the $\mathcal{G}^k$ most similar to $\mathcal{F}^i$, among the supersurfels from the model $\mathcal{G}$ selected as nearest neighbours. The symmetric Kullback-Leibler divergence can be defined as:

$$KLD(\mathcal{F}^i||\mathcal{G}^k) = \frac{1}{2}\{tr(\Sigma_i^{-1}\Sigma_k + \Sigma_k^{-1}\Sigma_i) + (p_i - p_k)^T(\Sigma_i^{-1} + \Sigma_k^{-1})(p_i - p_k)\} - 3. \quad (3)$$

The use of the Kullback-Leiber divergence as a distance has been proposed by (Davis and Dhillon, 2006), in order to perform gaussian clustering. It allows to value the dissimilarity between two supersufels in terms of position, orientation and shape.

To ensure that supersurfels too different are not associated we also proceed to a series of checkups:

1. We check that the divergence angle between the normals is small: $arcos(<n_i,n_k>) \times 180/\pi < \varepsilon_1$, with ε_1 usually set to 10 degrees.
2. We compare their areas: $|a < l_{1i}l_{2i}/l_{1k}l_{2k}| < b$, with $a,b \in \Re$ (for instance set to $a = 0.5$ and $b = 2$ if we want to excludes supersurfels which are at least twice bigger or twice smaller than $\mathcal{F}^i$).
3. We look up if their colors are close: $\Delta E^*_{c_ic_k} < \varepsilon_2$ where $\Delta E^*_{c_ic_k}$ is the distance between two colors in CIELAB color space, ignoring the lightness components so as to consider only chromatic information and be robust against light intensity variations. The threshold ε_2 can be fixed to 10.

If no correspondent $\mathcal{G}^k$ is found in the model $\mathcal{G}$, $\mathcal{F}^i$ is simply added into the model. Else, the supersurfel $\mathcal{G}^k$ from the model $\mathcal{G}$, which minimizes the symmetric Kullback-Leibler divergence (Equation 3) is selected as a match and data fusion is applied.

5.2 Fusion

After data association, for each pair of correspondences $(\mathcal{G}^k, \mathcal{F}^i)$ found, we update the attributes of $\mathcal{G}^k$ from $\mathcal{F}^i$ ones so as to refine the reconstruction. The new position $\boldsymbol{p}'_k$ and covariance $\boldsymbol{\Sigma}'_k$ are computed by applying covariance intersection strategy:

$$\Sigma_k'^{-1} = \alpha\Sigma_k^{-1} + (1-\alpha)\Sigma_i^{-1}, \tag{4}$$

$$p'_k = \Sigma'_k(\alpha\Sigma_k^{-1}p_k + (1-\alpha)\Sigma_i^{-1}p_i), \tag{5}$$

$$\alpha = \frac{w_k}{w_k + w_i}. \tag{6}$$

The updated color $\boldsymbol{c}'_k$ is obtained by a weighted average in CIELAB color space:

$$c'_k = \frac{w_k c_k + w_i c_i}{w_k + w_i}. \tag{7}$$

To compute updated values for the orientation $\boldsymbol{R}'_k$ and the longitudinal and lateral elongations $\boldsymbol{l}'_{1k}$ and $\boldsymbol{l}'_{2k}$, the same PCA procedure as during the supersurfels generation step is applied. The confidence weight $\boldsymbol{w}'_k$ is incremented:

$$w'_k = min(w_k + w_i, w_{max}). \tag{8}$$

The truncation of the confidence weight over a maximum value $\boldsymbol{w_{max}}$ is needed if we want new supersurfels with low confidence to still have a minimum influence on old stable supersurfels. A supersurfel is considered stable when its confidence exceed a fixed threshold $\boldsymbol{w_{stable}}$. The timestamp is also updated with current time value:

$$t'_k = t. \tag{9}$$

5.3 Removal of Outliers

Lastly, different strategies are applied to remove outliers, due to noisy data, and filter the 3D model. Supersurfels that stay in an unstable state for too long are removed after $\boldsymbol{\Delta t}$ steps and the confidence value of supersurfels from the model in the field of view of the camera, but not updated, is decreased.

A simple detection of free-space violations is also performed to remove all the supersurfels from the predicted 3D global model $\mathcal{G}$ that lie in front of newly updated supersurfels, with regard to the camera. To find these supersurfels to be removed, we compare the value of the $\boldsymbol{z}$ coordinate of the center $\boldsymbol{p_k}$ of each supersurfel $\mathcal{G}^k$ from $\mathcal{G}$, expressed in camera space at time $\boldsymbol{t-1}$, with the value of the current frame depth map $\boldsymbol{Z_t}$ at pixel $\boldsymbol{u_k}$. Pixel $\boldsymbol{u_k}$ corresponds to the location of the perspective projection of the center $\boldsymbol{p_k}$ in the image plane $\boldsymbol{\Omega}$. A supersurfel $\mathcal{G}^k$ is removed if the following relation is verified:

$$z < Z_t(u_k). \tag{10}$$

6 EXPERIMENTS AND RESULTS

We used an Asus Xtion Live Pro camera (640x480 image resolution, acquisition frequency of 30 fps) for our experiments and a laptop equipped with an Nvidia GTX 950M GPU and an Intel Core i5-6300HQ CPU. We performed quantitative and qualitative evaluations of our solution using our own video sequences, sequences from the TUM RGB-D benchmark dataset (Sturm et al., 2012) and from the ICL-NUIM dataset (Handa et al., 2014). Supersurfels are rendered as rectangular planar patches to accelerate the viewing.

The quantitative results are given for two different levels of resolution of SupersurfelFusion reconstruction: 3D supersurfels reconstruction obtained from the segmentation of input RGB-D pairs in superpixels of about 100 pixels and the one obtained from the segmentation in superpixels of about 400 pixels. Smaller superpixels generate more and finer supersurfels and tend to produce a more accurate map, as we can see on Figure 5, even if that is not the purpose of our method. In man made environment with large planar surfaces, wide superpixels allow a sufficiently suitable 3D reconstruction. However when the observed area is composed of thin and detailed objects (and according to the level of detail required by the user) smaller superpixels better fit the boundaries and enable to build a more complete model.

6.1 Qualitative Results

Figure 3 shows some reconstructed scenes. The obtained 3D model is dense and even some fine or curved elements such as the legs of the table or the back of the chair are well reproduced. This outlines that our representation is able to preserve meaningfull informations and is not a simple point cloud extracted from a downsized or decimated RGB-D frame. The approach is well adapted to local mapping tasks because it generates large supersurfels for distant elements which are then refined by the fusion procedure when the sensor comes closer. Close elements are displayed with higher accuracy, represented by smaller supersurfels, contrary to distant ones.

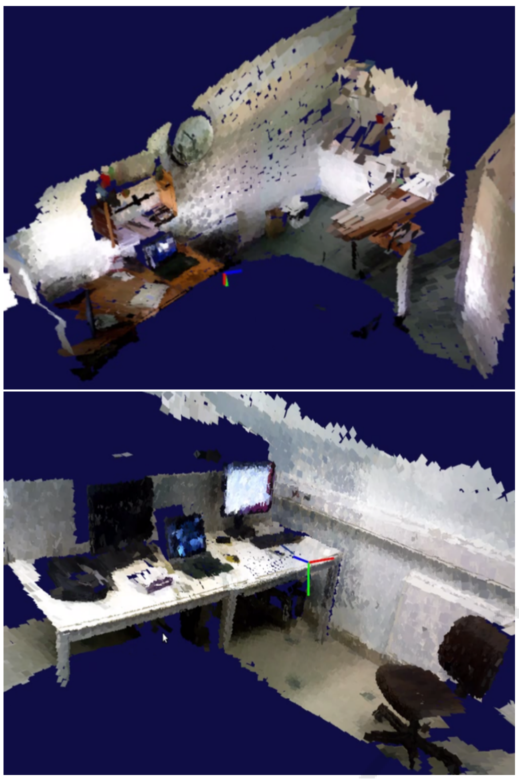
Figure 3: Supersurfels reconstructions of our office environments (superpixel size $\simeq$ 100 pixels).

6.2 Accuracy of the Surface Estimation

Although accuracy is not the purpose of our method, it is important that the 3D map built by SupersurfelFusion is relevant and represents well the real environment. We evaluate the quality of the surface reconstructed by our approach on the living room scene of the ICL-NUIM dataset of (Handa et al., 2014). This dataset provides four synthetic noisy RGB-D video sequences, with associated ground-truth trajectories and a ground-truth 3D model of the environment. An example of our system running on this dataset can be viewed on Figure 4. To evaluate the surface estimation produced by our approach we convert our supersurfels model to a dense point cloud by oversampling the surface of each supersurfel. We then compute the mean distances from each point of the obtained point cloud to the nearest surface of the ground-truth 3D model.

Results presented Table 1 show that even if SupersurfelFusion generates a rough 3D model, a certain accuracy can still be maintained in simple indoor environments. We also add results obtained with the state of the art accurate solution ElasticFusion (Whelan et al., 2015), running with its default configuration with provided ground-truth.

Table 1: Surface reconstruction accuracy results on the ICL-NUIM (Handa et al., 2014) synthetic dataset, for SupersurfelFusion with superpixels of about 100 pixels (SFusion 100), with superpixels of about 400 pixels (SFusion 400) and ElasticFusion (EFusion). Mean distances (m) from each point of the reconstruction to the nearest surface in the ground-truth 3D model are shown.

System	kt0	kt1	kt2	kt3
SFusion 100	0.009	0.011	0.017	0.010
SFusion 400	0.013	0.013	0.020	0.021
EFusion	0.003	0.005	0.002	0.004

We recall that our goal is not to challenge the accuracy of others state of the art systems. Instead we want to show that our strategy for approximate 3D reconstruction, based on the supersurfels representation, enable a performance improvement in speed and memory usage.

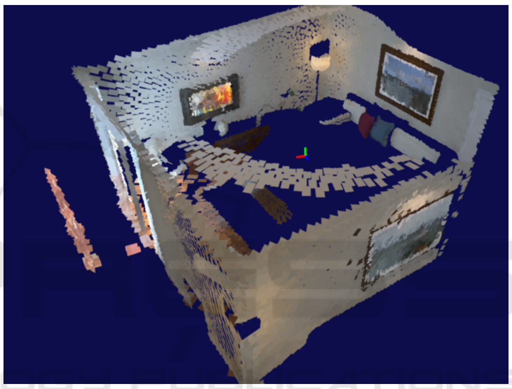
Figure 4: Supersurfels reconstruction of the living room for the sequence kt1 of the ICL-NUIM dataset (Handa et al., 2014).

6.3 Computational Performance

To evaluate the performance of our solution we took videos from (Sturm et al., 2012). This dataset provides real RGB-D video sequences with associated ground-truth measurements of the pose of the camera that we used as replacement of the visual odometry block. As a means of comparison, here again we present results acquired with ElasticFusion (Whelan et al., 2015), running with its default configuration with provided ground-truth, on our test platform.

Table 2 presents the execution time of SupersurfelFusion and ElasticFusion systems for 4 different videos. Real-time execution is achieved for both levels of resolution with SupersurfelFusion on non professional GPU, even if as expected using smaller superpixels slow down the process. There is no interest or advantages of using our method with too small superpixels because it would run with similar or worst performance than other standard approaches

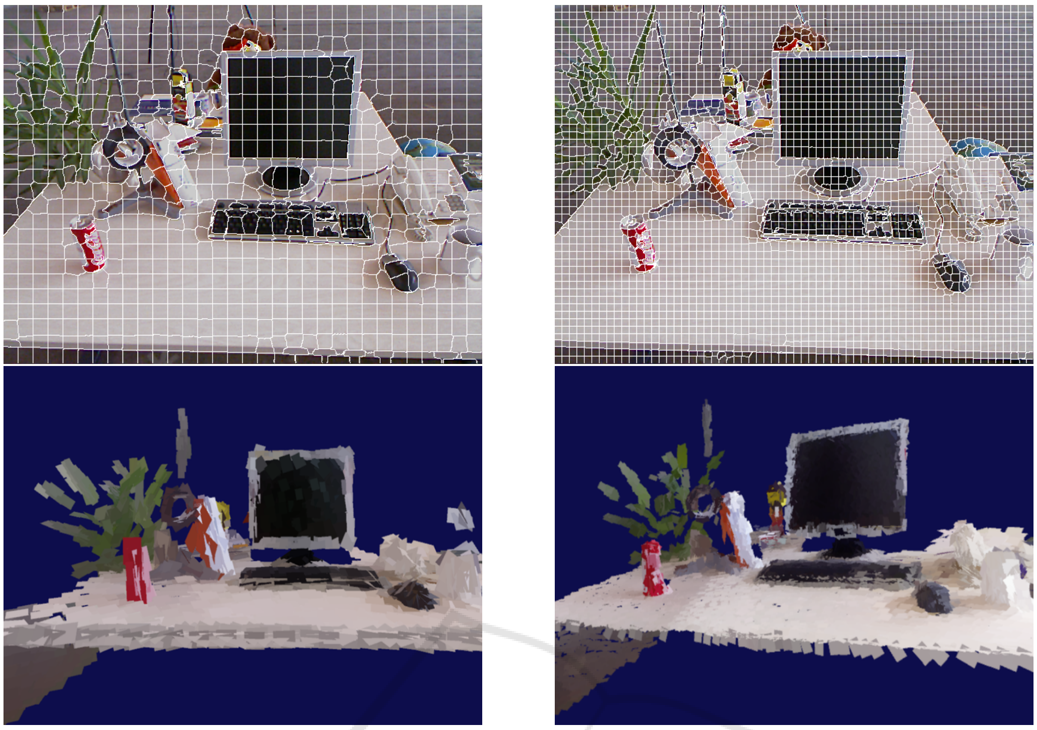

Figure 5: Comparison between 3D reconstruction results on the *fr2/rpy* sequence, from the TUM RGB-D benchmark dataset (Sturm et al., 2012), using SupersurfelFusion configured to generate superpixels of about 400 pixels (left column) and superpixels containing about 100 pixels (right column).

using every pixels, due to the time consuming superpixels segmentation task which is mainly dependant of the input image size.

Table 2: Run time (Mean ± Std ms) of SupersurfelFusion with superpixels of about 100 pixels (SFusion 100), 400 pixels (SFusion 400) and ElasticFusion (EFusion). Evaluation performed on sequences from the TUM RGB-D benchmark dataset (Sturm et al., 2012).

System	**fr1/room**	**fr1/plant**	**fr2/rpy**	**fr2/desk**
SFusion 100	35.3 ± 13.7	33.2 ± 6.83	25.9 ± 4.41	28.3 ± 5.79
SFusion 400	23.7 ± 2.27	23.8 ± 3.72	21.2 ± 9.01	22.2 ± 2.66
EFusion	58.1 ± 9.26	53.0 ± 7.81	50.4 ± 1.06	73.1 ± 2.02

Table 3 shows the memory footprint of the model reconstructed by SupersurfelFusion and ElasticFusion. The method requires a small amount of memory to store the model, thanks to the fact that unlike traditional point-based reconstruction approaches, which generate a surfel for each pixel, we only considered a restricted set of 3D superpixel-based primitives for each image. However each supersurfel uses bigger storage than a basic surfel structure. The short memory usage offers the possibility to reconstruct large size 3D models. We can see that ElasticFusion uses way more memory than our coarse 3D reconstruction method (about 10 times with regard to SupersurfelFusion using superpixels of about 400 pixels).

Table 3: Maximal memory usage (MB) of the reconstructed model for SupersurfelFusion with superpixels of about 100 pixels (SFusion 100), 400 pixels (SFusion 400) and ElasticFusion (EFusion). Evaluation performed on sequences from the TUM RGB-D benchmark dataset (Sturm et al., 2012).

System	**fr1/room**	**fr1/plant**	**fr2/rpy**	**fr2/desk**
SFusion 100	20.9	18.3	5.83	7.88
SFusion 400	5.23	4.41	1.45	2.37
EFusion	52.3	33.0	19.1	58.6

As expected, the lower the resolution of SupersurfelFusion is, the better are the results in terms of speed and memory usage. The user should fix the adapted resolution according to the task and the working environment of the robot. The system has been tested on an NVIDIA Jetson TX1 embedded board too, showing similar performance and results.

7 CONCLUSION

In this paper, a novel representation for 3D reconstruction of static environments with RGB-D cameras is presented. The reconstructed 3D model is defined as a set of oriented and colored elliptical planar patches, extracted from the segmentation in superpixels of a live RGB-D video stream. The use of superpixels to generate the coarse 3D primitives (supersurfels) composing the predicted model guarantee the preservation of most of the relevant and meaningful information from the observed scene.

The reconstruction system proposed, Supersurfel-Fusion, based on this representation, rebuild a low-resolution but pertinent 3D model with high speed performance and low memory usage. The use of this system is of interest for robots which do not need a very accurate but rather an efficient, fast and lightweight 3D map generation, enabling them to perform other tasks at the same time without consuming too much of the resources available.

As future works, we would like to integrate our own camera tracking solution to this system, as frame to model registration strategy, and make it robust to dynamic environments by detecting moving objects at a superpixel level and tracking them. Further optimization can be achieved to speed up the computation of supersurfels attributes by minimizing branch divergence.

REFERENCES

Bódis-Szomorú, A., Riemenschneider, H., and Gool, L. V. (2014). Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.

Davis, J. V. and Dhillon, I. (2006). Differential entropic clustering of multivariate gaussians. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06. MIT Press.

Handa, A., Whelan, T., McDonald, J., and Davison, A. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*.

Henry, P., Fox, D., Bhowmik, A., and Mongia, R. (2013). Patch volumes: Segmentation-based consistent mapping with rgb-d cameras. In *3DV*. IEEE Computer Society.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM.

Karras, T. (2012). Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, EGGH-HPG'12. Eurographics Association.

Kay, T. L. and Kajiya, J. T. (1986). Ray tracing complex scenes. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86. ACM.

Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., and Kolb, A. (2013). Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision - 3DV 2013*.

Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*.

Ren and Malik (2003). Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*.

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

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

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*.

Thomas, D. and Sugimoto, A. (2013). A flexible scene representation for 3d reconstruction using an rgb-d camera. In *2013 IEEE International Conference on Computer Vision*, pages 2800–2807.

Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia.

Whelan, T., Leutenegger, S., Moreno, R. S., Glocker, B., and Davison, A. (2015). Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems*.

Yamaguchi, K., McAllester, D., and Urtasun, R. (2014). Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*.