Coloured signed distance fields for full 3D object reconstruction

Wadim Kehl[1]
kehl@in.tum.de
Nassir Navab[1]
navab@in.tum.de
Slobodan Ilic[2]
slobodan.ilic@siemens.com

[1] CAMP Chair
Computer Science Department
TU Munich, Germany

[2] Siemens AG
Research & Technology Center
Munich, Germany

Abstract

We propose a novel 3D object reconstruction framework that is able to fully capture the accurate coloured geometry of an object using an RGB-D sensor. Building on visual odometry for trajectory estimation, we perform pose graph optimisation on collected keyframes and reconstruct the scan variationally via coloured signed distance fields. To capture the full geometry we conduct multiple scans while changing the object's pose. After collecting all coloured fields we perform an automated dense registration over all collected scans to create one coherent model. We show on eight reconstructed real-life objects that the proposed pipeline outperforms the state-of-the-art in visual quality as well as geometrical fidelity.

1 Introduction

3D reconstruction has been the focus of computer vision and robotics for decades. The advent of low-cost RGB-D sensors further widened the focus since it enabled many different user groups to create coloured, metrically accurate reconstructions. They become more and more important for many different fields including manufacturing verification, human entertainment like gaming or augmented reality as well as tasks in robotics such as object recognition and grasping.

To properly reconstruct an object the global camera pose has to be known at every point of time. Usually, an incremental update of the camera pose between two subsequent frames is estimated by robust means. Tracking of an RGB-D sensor can either be done sparsely by, e.g., matching image keypoints [5] in 3D or densely by ICP variants [20]. KinectFusion [15] allows to create reconstructions of high fidelity but mainly due to the inability of ICP to cope with symmetric or non-distinctive geometry, the method often fails when reconstructing various objects. Furthermore, it still requires the user to specify volume boundaries and to sometimes deal with post-processing steps after meshing when singling out objects. As an alternative to ICP, visual odometry approaches use both colour and depth information and estimate the warp that encodes the camera motion [24]. They can overcome situations where ICP fails and thus, provide for a better estimation of camera movements. However, current state-of-the-art reconstruction approaches often assume a static scene which prohibits the

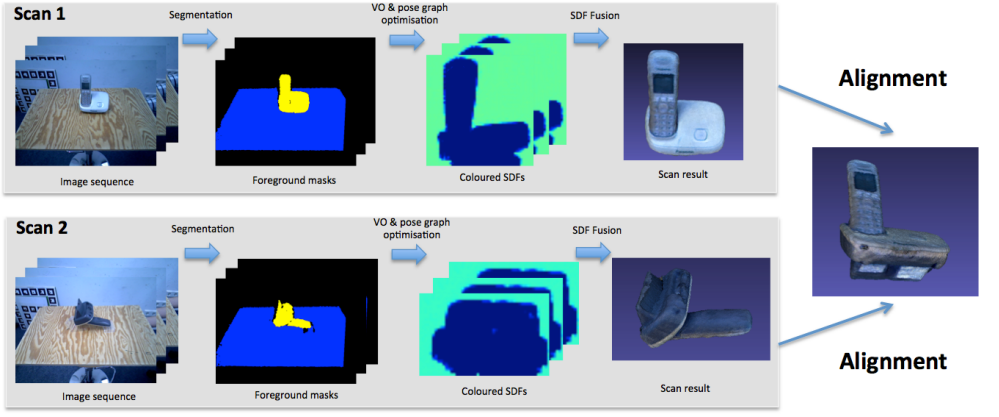

Figure 1: The pipeline visualised. Each scan sequence is masked, pose-optimised and fused to create a model. Then all scans get aligned to one coherent model.

displacement of objects during scanning. As a result, incomplete object geometry is usually obtained with the bottom or some self-occluded parts missing from the reconstruction.

The overall trend and need in robotics and computer vision goes towards unsupervised, automatic and/or autonomous methods which try to reduce human input to create precise, fully coloured 3D meshed models. We therefore propose a framework, combining the strengths of related approaches, to fully automate 3D reconstructions and produce high-quality textured 3D models from low-cost RGB-D sensors.

We present a full 3D reconstruction pipeline combining visual odometry and KinectFusion ideas. Initially, we rely on visual odometry [4, 24] to compute the camera trajectory on a foreground-segmented sequence. As output we obtain a number of keyframes with precise camera poses and associated 3D data clouds. We then cast them into signed distance fields and integrate them into one common field, following [23, 27], while also solving for the colour component. We then displace the object to expose its previously hidden geometry and repeat proposed procedure to obtain multiple scans of the object in form of coloured SDFs (CSDF). Thereafter, we propose a novel automatic registration framework to robustly fuse multiple CSDFs into one coherent model. See Figure 1 for a visualisation. We evaluated on multiple diverse real-life objects to show the capabilities and precision of our approach. We compared with KinectFusion both qualitatively and quantitatively. For smaller and rather symmetric objects our method succeeded while KinectFusion failed. We were always able to recover the full textured geometries. Comparison with ground truth CAD data also revealed our superior metrical precision even with geometries that are distinctive enough for proper KinectFusion tracking.

1.1 Related work

3D object reconstruction with range data is widely covered in the literature together with a good overview given in [1]. One can usually roughly divide the literature up into a stationary set-up where the object sits on top of a (rotating) support surface and a dynamic set-up where the object of interest is scanned in-hand.

Although the in-hand approach is naturally more appealing, proper background segmentation as well as transformation estimation is hard to accomplish. The works [21, 25] remove the background by using coloured gloves which are detected and filtered out. In-hand set-ups work well if the object has a distinctive appearance and the movement between frames is small. Methodologically, it fails for objects with poor geometrical and textural discriminance (or even symmetries) since the registration between frames becomes unreliable. The work [10] follows the same idea but uses a robot and its arm pose to recover the transformation between frames without visual estimation.

The static set-up has the advantage of being able to use markers on the support surface [8] or even act without markers by relying on a sufficiently textured turntable [4]. In case of using markers no constraints on the object's general appearance are imposed. Furthermore, the surface assumption allows for straight-forwardly extracting segmentation masks [4, 22]. In this set-up however, the bottom of the object is never reconstructed. In order to obtain a full scan, the object would need to be put into another pose to be scanned again resulting in two partial scans of the same object related by an unknown transformation.

The inherent problem of automatically registering two independent scans remains to be another challenge to date. An early work on automatic registration of multiple 3D scans can be found in [9] where range scans of an object are transformed into partial meshes, matched pairwise and put into a global graph optimisation problem to find the most consistent connected subgraph. In [12] the authors solve for the alignment by defining correlation functions and computing Fourier transforms with a subsequent verification stage. These works register data which is represented sparsely of either points or surface approximations. Another way to represent depth is using signed distance fields [3] which have been subsequently used in many recent works including scene reconstruction and camera tracking [2, 6, 15, 26]. The advantage of such a representation is the dense space on which it is defined. This allows to introduce operators working on functions while still being able to extract a (possibly sparse) surface at a level-set. Furthermore, it has also been shown in [19] that a richer data representation can, in fact, help in registering when moving from simple point-based metrics to ones using implicit shape representations.

2 The pipeline

We propose a full reconstruction framework with a RGB-D sensor, requiring no marker boards and allowing for objects to be displaced during scanning. It consists of multiple stages which are outlined in the next paragraphs. Even though parts of our approach come from related work the proposed framework is unique and provides a novel fusion and registration procedure for CSDFs resulting in complete 3D models with high fidelity. It is suitable for a large variety of objects and outperforms a state-of-the-art KinectFusion implementation.

The first step in our pipeline is the camera trajectory estimation via RGB-D visual odometry. We move the support surface while collecting keyframes along the way and eventually refine the trajectory globally with a pose-graph optimisation after loop closure detection. A detailed explanation follows in subsection 2.1.

After one full scan and pose refinement we refer to our final result as a hemisphere $\mathcal{H}$ consisting of a number of RGB-D keyframes with associated poses. We create a 3D model ϕ by fusing the data in a variational fashion using coloured signed distance fields and an approximate L^1 minimisation. More details can be found in subsection 2.2.

Usually, one such scan does not expose the full geometry of the object. To this end, we

propose to create multiple scans of the same object but placed differently in order to reveal hitherto unseen parts, thus acquiring multiple hemispheres $\mathcal{H}_j$. Then the transformations Ξ_j that map the models from all hemispheres to the first one $\mathcal{H}_0$ need to be determined. In order to retrieve those Ξ_j we use the reconstructed models ϕ_j and align them automatically using a dense approximate-L^1 registration framework. See 2.3 for details.

2.1 3D object scanning

We follow the concept of [4] by creating a sequence with a fixed sensor and a rotating support surface. In order to reliably assess the camera movement a separation of foreground from background has to be performed. We define a RGB-D sensor pair $[I:\Omega_2 \to [0,1]^3, D:\Omega_2 \to \mathbb{R}^+]$ and the camera projection $\pi:\mathbb{R}^3 \to \Omega_2$. By fitting a plane into the cloud data $C := \pi_D^{-1}$ and computing the prism spanned by the support plane along its normal direction the 3D points lying inside the prism are determined and projected into the image plane to create foreground segmentation masks as done in [22]. If the masking fails (happens very rarely, e.g. another plane larger than the support surface present) the pair is discarded.

Then we estimate the camera transformation via visual odometry using RGB-D data [24]. The goal is to compute the rigid-body movement $\Xi \in SE(3)$ of the camera between two consecutive masked sensor pairs $[I_0,D_0]$,$[I_1,D_1]$ by maximising the photo-consistency

$$E(\Xi) = \int_{\Omega_2} [I_1(w_\Xi(x)) - I_0(x)]^2 dx \tag{1}$$

with a warp function $w_\Xi:\Omega_2 \to \Omega_2$ defined via the depth maps as $w_\Xi(x) = \pi_{D_1}(\Xi \cdot \pi_{D_0}^{-1}(x))$ that transforms and projects the coloured point cloud from one frame into the other. To solve this least-squares problem, the authors employ a Gauss-Newton approach with a coarse-to-fine scheme. We refer to [24] for details.

The advantage of using RGB-D visual odometry as opposed to plain ICP (as for example done in [15]) is that we can handle scenes and/or objects which suffer from poor geometrical discriminance. Although the odometry approach is naturally also prone to drifting, it showed to be far more reliable for the problem at hand as long as the support surface and the object exhibit a fair amount of texture.

Due to computational constraints in the optimisation stages we only want to collect a certain number of keyframes from a sequence. While estimating the frame-wise transformation update, we therefore only store keyframe pairs $[I_i,D_i]$ plus poses P_i after having seen a sufficient amount of transformational change between pose P_i and pose P_{i-1} from the last stored keyframe (in our implementation, 10 degrees and 10cm). To detect loop closure, we make sure that we have already observed a substantial amount of transformation in comparison to the initial frame (330 degrees or 2m) and then start comparing incoming pairs with the initial pair via colour-ICP and the inlier ratio. Eventually, we run a pose-graph optimisation using the g2o framework [12] in order to refine the camera poses while we base the error measure on the visual odometry energy.

2.2 Variational sensor data fusion

Given one hemisphere $\mathcal{H} = \{(I_i,D_i,P_i)_i\}$ consisting of masked sensor pairs and poses, we fuse the data into a coherent model. Analogously to [3, 7, 11, 15, 18, 23, 26, 27], we cast our data into volumetric fields $f_i:\Omega_3 \subset \mathbb{R}^3 \to \mathbb{R}$ in order to smoothly integrate them into

one fused model. The idea for the geometrical field is to create a truncated SDF by shooting rays and computing the signed distance from the surface for each point $\mathbf{x} \in \Omega_3$ along the line of sight. Points in front of the object receive positive values while points behind the surface become negative. We scale with a divisor δ and truncate to $[-1,+1]$ written as

$$f_i(\mathbf{x}) = \psi(D_i(\pi(\mathbf{x})) - ||\mathbf{x} - P_i^{xyz}||) \quad , \quad \psi(d) = \begin{cases} sgn(d) & \text{if } |d| > \delta \\ \frac{d}{\delta} & \text{else} \end{cases} \tag{2}$$

with P_i^{xyz} being the camera center for pose P_i. The parameter δ can be regarded as a tolerance towards measurement noise in the depth data and should be set in respect to the depth sensor's fidelity (we choose $\delta = 2mm$). Every f_i has a binary weighting function $w_i : \Omega_3 \to \{0,1\}$ that signifies which parts of the SDF should be taken into account during the fusion process:

$$w_i(\mathbf{x}) = \begin{cases} 1 & \text{if } D_i(\pi(\mathbf{x})) - ||\mathbf{x} - P_i^{xyz}|| < -\eta \\ 0 & \text{else} \end{cases} . \tag{3}$$

The parameter η defines how much has been seen behind the observed surface and assumed to be solid (we fixed $\eta = 1cm$). Since we are interested in recovering the colour of the object as well, we furthermore define a colour volume $c_i : \Omega_3 \to [0,1]^3$ with

$$c_i(\mathbf{x}) = I_i(\pi(\mathbf{x})). \tag{4}$$

We will refer to the joint representation $[f_i, c_i]$ as a CSDF. The goal now is to recover functions u, v which hold the object's reconstructed geometry and colouring, respectively. Following [23], we cast the problem into a variational energy optimisation formulation where we seek the minimisers of the functional

$$\mathcal{E}(u,v) = \int_{\Omega_3} \left[\mathcal{D}(\mathbf{f},\mathbf{w},\mathbf{c},u,v) + \alpha\mathcal{S}(\nabla u) + \beta\mathcal{S}(\nabla v)\right] \, d\mathbf{x} \tag{5}$$

with a data term $\mathcal{D}$ that strives to uphold the solution's fidelity to all the observations $\mathbf{f} = \{f_1,...,f_n\}, \mathbf{c} = \{c_1,...,c_n\}$ and two regularisers $\mathcal{S}(\nabla u)$ and $\mathcal{S}(\nabla v)$, weighted with α and β respectively, that force the minimisers to be smooth. Note that in contrast to the original work [23] which only fuses the geometrical fields, we also include colour information into the formulation and solve simultaneously for both.

A suitable data term for many problems in reconstruction and segmentation usually involves an outlier-robust L^1-norm whereas for regularisation purposes the total variation (TV) of the function is often employed:

$$\mathcal{D}(\mathbf{f},\mathbf{w},\mathbf{c},u,v) = \frac{1}{\varepsilon + \sum_i w_i} \sum_i w_i \cdot (|u - f_i| + |v - c_i|) \quad , \quad \mathcal{S}(\nabla u) = |\nabla u| \tag{6}$$

Due to the problematic aspect of solving such energies, specific minimisation schemes are employed (e.g. a ROF-variant [27] or (iterated) primal-dual solutions [6, 16]). An alternative has been proposed in [23] where the problematic terms have been replaced with a smooth $epsL^1$ [13] approximation $\Gamma(x) := \sqrt{x^2 + \varepsilon}$. We define it similarly

$$\mathcal{D}(\mathbf{f},\mathbf{w},\mathbf{c},u,v) = \Gamma(\sum_i w_i)^{-1} \sum_i w_i \cdot (\Gamma(u - f_i) + \Gamma(v - c_i)) \quad , \quad \mathcal{S}(\nabla u) = \Gamma(|\nabla u|) \tag{7}$$

where we regard the weighted approximate absolute differences together with an additional normalisation factor and an approximate TV-regulariser. This regulariser penalises the perimeter of the level sets and therefore leads to the removal of isolated small-scale features and the shaping of a low-genus isosurface of u.

With this strictly convex and differentiable formulation, the global minimisers can be found at the steady state of the gradient descent equations

$$\partial u = \alpha \cdot \mathrm{div}(\mathcal{S}'(\nabla u)) - \frac{\partial \mathcal{D}}{\partial u}(\mathbf{f},\mathbf{w},\mathbf{c},u,v) \;\;,\;\; \partial v = \beta \cdot \mathrm{div}(\mathcal{S}'(\nabla v)) - \frac{\partial \mathcal{D}}{\partial v}(\mathbf{f},\mathbf{w},\mathbf{c},u,v) \tag{8}$$

which we determine and denote as u^*, v^*. We now also define our reconstructed CSDF $\phi : \Omega_3 \to [-1,+1] \times [0,1]^3$ with $\phi = [u^*, v^*]$. This presented fusion method is applied to each of the collected hemispheres $\mathcal{H}_j$ to produce a corresponding model, i.e. CSDF ϕ_j.

2.3 Automatic alignment of CSDFs

Let us resume to the problem of aligning all the models ϕ_j from hemispheres $\mathcal{H}_j$, which we reduce to solving pairwise problems of aligning models ϕ_j to ϕ_0. Thus, a 3D rigid-body transformation Ξ_j needs to be determined. We are faced with six degrees of freedom, therefore choosing a minimal representation of our transformation parametrised via a twist vector $\xi = [\omega_x, \omega_y, \omega_z, t_x, t_y, t_z]^T \in \mathbb{R}^6$, its corresponding Lie algebra twist $\hat{\xi} \in se(3)$ and exponential mapping from the Lie group $\Xi \in SE(3)$:

$$\hat{\xi} = \begin{pmatrix} 0 & -\omega_z & \omega_y & t_x \\ \omega_z & 0 & -\omega_x & t_y \\ -\omega_y & \omega_x & 0 & t_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \;\;,\;\; \Xi = \exp(\hat{\xi}) = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \tag{9}$$

with the goal that for every point $\mathbf{x}$ we achieve $\phi_0(\mathbf{x}) = \phi_j(\Xi(\mathbf{x}))$, where $\Xi(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^3$ denotes, for the sake of notation, the homogeneous transformation parametrised by ξ and applied to point $\mathbf{x}$. Since exact solutions usually do not exist in practice, we try to minimise their distance instead by an energy formulation with an appropriate measure M:

$$\mathcal{E}(\Xi) = \int_{\Omega_3} M(\phi_0(\mathbf{x}), \phi_j(\Xi(\mathbf{x}))) \; d\mathbf{x}. \tag{10}$$

The striking difference to other KinectFusion-based approaches [15, 26] is that we tackle the registration problem in a fully continuous global manner while solving an algebraic error, inspired by [17, 19]. It is noteworthy that in [2, 11] the authors walk half the way by formulating a point-SDF distance measure. Our approach can be regarded as an algebraic (i.e. non-geometric) generalisation of ICP to dense volumetric representations. One differentiable measure that one can use here is the L^2-norm which leads to

$$\mathcal{E}(\Xi)_{L^2} = \int_{\Omega_3} \frac{1}{2}(\phi_0(\mathbf{x}) - \phi_j(\Xi(\mathbf{x})))^2 \; d\mathbf{x}. \tag{11}$$

In order to optimise the given non-linear energy, the mentioned related work (using an SSD measure over points) usually rely on a Gauss-Newton scheme, where they iteratively linearise the problem and solve linear equation systems. Since tracking speed in real-time is an important issue for them, their method is the most appealing due to its convergence speed

and small incremental camera changes. We, on the other hand, are interested in precise alignment with larger transformations and therefore propose a more robust energy that employs the approximative L^1-counterpart:

$$\mathcal{E}(\Xi)_{L^1} = \int_{\Omega_3} \Gamma(\phi_0(\mathbf{x}) - \phi_j(\Xi(\mathbf{x}))) \; d\mathbf{x}. \tag{12}$$

In theory, one could use a variety of differentiable distance measures like correlation ratios or mutual information. We apply a gradient descent scheme to update the transformation parameters. Starting with an initial transformation Ξ^0, we iteratively solve (depending on the measure):

$$\nabla\xi = -\tfrac{1}{|\Omega_3|}(-\tfrac{\partial\phi_j}{\partial\Xi^i}\cdot\tfrac{\partial\Xi^i}{\partial\xi})\cdot(\phi_0-\phi_j(\Xi^i)) \quad (L^2) \tag{13}$$

$$\nabla\xi = -\tfrac{1}{|\Omega_3|}(-\tfrac{\partial\phi_j}{\partial\Xi^i}\cdot\tfrac{\partial\Xi^i}{\partial\xi})\cdot\Gamma'(\phi_0-\phi_j(\Xi^i)) \quad (L^1) \tag{14}$$

$$\Xi^{i+1} = \exp(\tau\cdot\hat{\nabla\xi})\cdot\Xi^i \tag{15}$$

having a twist update $\nabla\xi$ for either energy with the Jacobian $\frac{\partial\phi_j}{\partial\Xi^i}\cdot\frac{\partial\Xi^i}{\partial\xi}$, a gradient step size τ and an additional normalisation $\frac{1}{|\Omega_3|}$ to ensure a proper numerical update.

The proposed energy admits local optima and therefore depends on the initialisation. While we generally observe a very robust convergence, we still have the problem of large rotational differences in between hemispheres. To this end, we search in a spherical grid by sampling spherical coordinates in the ranges $\theta \in (0,\pi), \kappa \in (0,2\pi)$ in discrete steps and run the alignment until convergence. To speed up the registration process, we employ a coarse-to-fine pyramid scheme over three levels where spatially down-sampled fields are roughly aligned and the resulting alignment is taken as the initialisation into the next pyramid stage.

After deciding for the best alignment Ξ_j we fuse both hemispheres by simply merging their elements into the first $\mathcal{H}_0 := \{(I_i^0, D_i^0, P_i^0)_i, (I_k^j, D_k^j, \Xi_j \cdot P_k^j)_k\}$ while transforming the poses. Eventually, we run Marching Cubes to extract a mesh at the 0-level set of ϕ_0.

3 Evaluation

The proposed algorithm was implemented on the CPU in C++ and has been used to reconstruct eight real-life objects, namely *book*, a 3D print of Stanford's *bunny*, *drill*, *mango*, *milk*, *phone*, *tape* and *turbine*. They were placed on a table and two sequences of about 800 images have been taken for each object. Computing the visual odometry as well as masking and loop closure detection is accomplished in real-time. For 32 keyframes, pose graph optimisation and data fusion take around 2-3 minutes each whereas the alignment of two CSDFs is depending on the object's geometry and size and can take up to 10 minutes.

3.1 KinectFusion versus our method

We compared our method to the commercial state-of-the-art KinectFusion implementation RecFusion. The voxel size was fixed to $1mm$ and we ran the methods on exactly the same sequences. The reconstruction results are presented in Figure 2. Even though KinectFusion performed well it failed for the objects *mango*, *milk* and *tape* due to poor geometry leading to tracking failure. For the two models *bunny* and *turbine* ground-truth data was available

Figure 2: Each pair depicts the results from KinectFusion (left) and our approach (right). We clearly recover richer texture as well as geometry. We are even able to fully reconstruct symmetric objects like *tape* or geometrically poor objects like *mango* and *milk*.

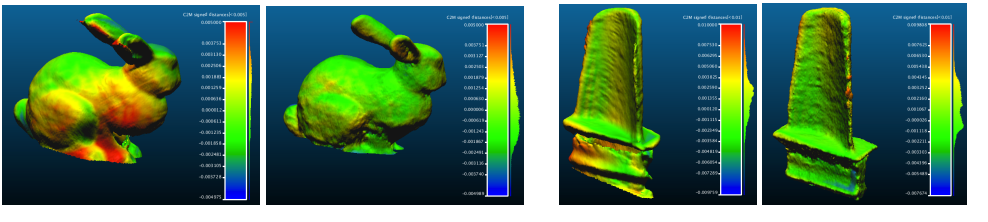

Figure 3: Colour-coded reconstruction accuracy in meters in respect to ground truth data of *bunny* and *turbine*. Left-right pair: KinectFusion - our approach. Our method is able to recover finer details due to optimising both the pose graph and the sensor data integration.

Object	book	bunny	drill	mango	milk	phone	tape	turbine
# Pos. L^1 runs	4	6	7	2	3	6	11	3
# Pos. L^2 runs	4	5	6	2	3	5	10	3
Avg. L^1 iters	289	204	363	102	282	261	120	425
Avg. L^2 iters	344	284	501	134	352	462	148	411

Figure 4: Registration results using both measures for the given objects. A total of 16 runs with different rotational initialisations have been performed. The L^1 registration clearly outperforms the L^2 formulation both in terms of its wider convergence basin and speed.

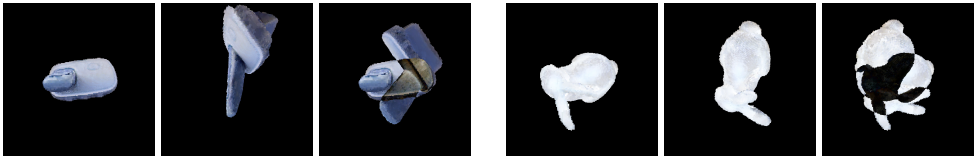

Figure 5: L^2-registration performance for *phone* and *bunny* visualised as a difference image of ray-casted volumes. From left to right: Target, initialisation, result. The L^2-energy got stuck locally whereas the L^1-energy converged globally in these cases.

and was used to measure the metrical error of the reconstructions (see Figure 3). To be fair for the latter, we compared the KinectFusion results with ours only by reconstructing from one hemisphere. Thus, both methods worked on the same data since we wanted to demonstrate the accuracy obtained with our optimisation pipeline. We clearly boost the geometrical fidelity due to the pose graph optimisation and the L^1 sensor fusion.

3.2 L^1 versus L^2 registration

In order to show the superiority of the approximate L^1 registration in comparison to L^2, we ran the alignment (given an initial transformation) on all reconstructed objects. The rotations were sampled from spherical coordinates in discrete steps (resulting in 16 runs) and the SDFs were mean-centered. We measured both the number of successful runs (i.e. global convergence) and the number of average gradient descent iterations over all successful runs. Figure 4 summarises the registration results. We can observe that the L^1 formulation consistently leads to an energy that allows for easier global convergence while reducing the number of gradient descent steps. Figure 5 shows a typical case where the L^1-energy was able to globally converge whereas the L^2-energy got stuck in a local minimum.

4 Conclusion

We presented a novel pipeline for full coloured 3D reconstruction. We showed that by using optimisation over camera poses, data fusion and an automatic registration we outperform the state-of-the-art in terms of textural and geometrical accuracy. Future work will include bundle adjustment over the coloured SDF representations as well as a computationally more efficient fusion. The authors would also like to thank Toyota Motor Corporation for supporting and funding the work.

References

[1] Fausto Bernardini and Holly Rushmeier. The 3D Model Aquisition Pipeline. *Computer Graphics Forum*, 21, 2002.

[2] Erik Bylow, Jürgen Sturm, Christian Kerl, Frederik Kahl, and Daniel Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *RSS*, 2013.

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[4] Maria Dimashova, Ilya Lysenkov, Vincent Rabaud, and Victor Eruhimov. Tabletop Object Scanning with an RGB-D Sensor. *ICRA Workshop*, 2013.

[5] Felix Endres, Juergen Hess, Nikolas Engelhard, Juergen Sturm, and Wolfram Burgard. An Evaluation of the RGB-D SLAM System. In *ICRA*, 2012.

[6] Gottfried Graber, Thomas Pock, and Horst Bischof. Online 3D reconstruction using convex optimization. In *ICCV Workshop*, 2011.

[7] Peter Henry, Dieter Fox, Achintya Bhowmik, and Rajiv Mongia. Patch Volumes: Segmentation-Based Consistent Mapping with RGB-D Cameras. In *3DV*, 2013.

[8] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradsky, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *ACCV*, 2012.

[9] Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3D data sets. *IVC*, 21, 2003.

[10] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3D object modeling. *IJRR*, 2011.

[11] Daniel B. Kubacki, Huy Q. Bui, S Babacan, and Minh Do. Registration and integration of multiple depth images using signed distance function. In *SPIE 8296, Computational Imaging X*, February 2012.

[12] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization, 2011.

[13] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Ng. Efficient L1 Regularized Logistic Regression. In *AAAI*, 2006.

[14] Ameesh Makadia, Alexander Patterson, and Kostas Daniilidis. Fully Automatic Registration of 3D Point Clouds. In *CVPR*, 2006.

[15] Richard A. Newcombe, Andrew J. Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011.

[16] Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. An Iterated L1 Algorithm for Non-smooth Non-convex Optimization in Computer Vision. In *CVPR*, June 2013.

[17] Nikos Paragios, Mikeal Rousson, and Visvanathan Ramesh. Matching distance functions: A shape-to-area variational approach for global-to-local registration. In *ECCV*, 2002.

[18] Carl Yuheng Ren and Ian Reid. A unified energy minimization framework for model fitting in depth. In *ECCV*, 2012.

[19] Mohammad Rouhani and Angel Sappa. The Richer Representation the Better Registration. In *IEEE TIP*, 2013.

[20] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *3DIM*, 2001.

[21] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-Time 3D Model Acquisition. In *SIGGRAPH*, 2002.

[22] Radu Bogdan Rusu, Andreas Holzbach, Nico Blodow, and Michael Beetz. Fast Geometric Point Labeling using Conditional Random Fields. In *IROS*, 2009.

[23] Christopher Schroers, Henning Zimmer, Levi Valgaerts, Oliver Demetz, and Joachim Weickert. Anisotropic Range Image Integration. In *Pattern Recognition, LNCS*, 2012.

[24] Frank Steinbrucker, Jurgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *ICCV Workshop*, 2011.

[25] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. Online loop closure for real-time interactive 3D scanning. *CVIU*, 2011.

[26] Thomas Whelan, Hordur Johannsson, Michael Kaess, John J. Leonard, and John McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *ICRA*, 2013.

[27] Christopher Zach, Thomas Pock, and Horst Bischof. A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration. In *ICCV*, 2007.