Paper Review:

Accurate 3-D Reconstruction with RGB-D Cameras using Depth Map Fusion and Pose Refinement

Project Summary:

The goal of our project is to create a software pipeline to reconstruct an accurate 3D model of a baby's head from depth information using mobile devices. The overall goal of the project is to provide pediatricians with a more accurate and quantifiable way of detecting and diagnosing cranial deformities in infants.

Background:

1. Basic Overview

With the rise of consumer-grade RGB-D cameras, RGB images and depth maps are easy to capture, which makes the 3D reconstruction of an object or scene readily available. Depth maps are like RGB images, except the intensity dimension is replaced with depth information. Camera intrinsics can be applied to compute the backprojected depth map, which contains the spatial information of the scene in 3D. These are combined with existing methods of reconstruction to recreate a scene or object. We consider a simple case for our pipeline. First, we record all depth maps as point clouds. We then calculate the transformations between each consecutive cloud and register all point clouds into a single coordinate frame. When everything is registered, we fuse all depth maps together into a single large point cloud. The last step is important, as the fusion of the depth maps has a major impact on the final reconstruction result, which impacts how accurately an object, like a baby's head, can be constructed.

Figure 1: (Left) Depth map of a baby doll (Right) RGB image of a baby doll (Provided by Group 15)

2. Challenges

The simple method outlined above can be prone to many problems. Outliers can lead registration algorithms like iterated closest point (ICP) astray. This produces poor registrations of depth maps which can impact reconstruction negatively, resulting in misaligned point clouds in the final reconstruction.

Fusing point clouds from each depth map also creates many points in the final point cloud. These points are redundant: many of them from different point clouds can overlap or be identical to other points. The redundant points in the point cloud slows down further processing down the line and offer no helpful or additional information about the reconstruction.

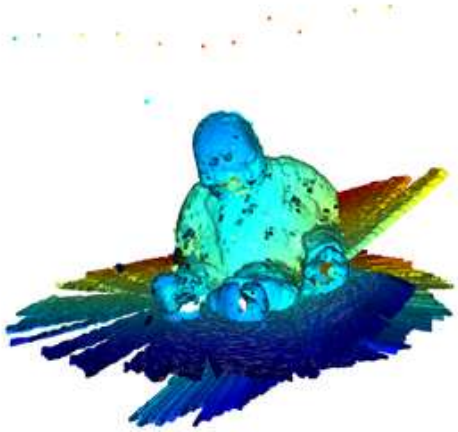

Figure 2: Single misaligned registration leads to poor reconstruction (Provided by Group 15)

Paper Reviewed:

Ylimäki, Markus, Janne Heikkilä, and Juho Kannala. "Accurate 3-d reconstruction with rgb-d cameras using depth map fusion and pose refinement." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.

I picked this paper as it was very relevant to our project, as it discusses different ways of improving 3D reconstruction. Specifically, it provides a pipeline to reduce redundant points in a point cloud, which will be useful to us when we get to the final stages of mesh generation. It gives a good overview of the techniques and the rationale that it uses to improve reconstruction, which can be useful to implement within our own software pipeline as well.

Paper Summary:

The authors of this paper attempt to reconstruct non-redundant point clouds using a mix of fusion and re-registration, expanding on an existing method. They first merge a sequence of depth maps into a point cloud. New points are either added or used to refine the existing points in the point cloud. Their contribution lies

in re-registering the original depth maps back into the fused point cloud to refine camera poses, and repeats the fusion step again, for multiple iterations until satisfactory. They found that this method reduces the number of points in the fused point clouds and leads to better reconstructions.

Paper:

1. Pipeline: the paper follows the pipeline shown in Figure 3.

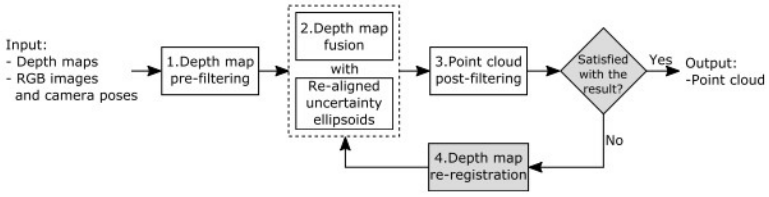

Figure 3: Pipeline outlined by the paper. Author contributions shown in gray. Figure from [1].

Pre-filtering

The pre-filtering step attempts to correct artifacts and outliers generated by multi-path interference or lens distortion. It assumes that the density of points in a backprojected (3D point cloud) depth map near outliers is much smaller than accurate regions. A point is removed if its depth and nearest neighbors with respect to a reference distance is longer than a threshold. This threshold is determined by a constant variable multiplied by the reference distance. The reference distance is determined to be the average distance from a point to its 4^{th} nearest neighbor in a planar point cloud.

Fusion

The depth map fusion step is based on an existing method in [2] and calculates a location uncertainty of a point given by the covariance matrix:

$$\mathbf{C} = \begin{bmatrix} \lambda_1(\frac{\beta_x z}{\sqrt{12}})^2 & 0 & 0 \\ 0 & \lambda_1(\frac{\beta_y z}{\sqrt{12}})^2 & 0 \\ 0 & 0 & \lambda_2(\alpha_2 z^2 + \alpha_1 z + \alpha_0)^2 \end{bmatrix}$$

Figure 4: Covariance matrix for the uncertainty of a point. Figure from [1].

λ_1 and λ_2 are parameters to scale the variances.

α_0, α_1 and α_2 are parameters of a quadratic depth variance function described in [2].

β_x, β_y define the width and height of a back projected pixel one meter away from the sensor.

z is the depth of the point.

The uncertainties extracted from this covariance matrix determines how the depth map is fused together. For visualization purposes, we can think of these uncertainties as ellipses around each point. If a new point does not land within the ellipse of any existing measurement, then it is added to the fused point cloud. If the new point lands within the ellipse of an existing measurement, the new measurement

is used to refine the existing measurement, giving more weight to the measurement with a lower uncertainty. This effect is shown in Figure 5. The points are given by the vector p. The new point is given by the index n while the existing point is given by the index e. The covariances or uncertainties are given by the vector c. Since the new point and the existing point have overlapping ellipses, the existing point can be refined to p'. It gives more weight to the point with the smaller uncertainty, which is why it is closer to the existing point. The formula for the weights can be calculated from [2].

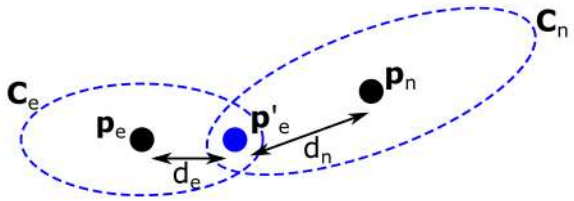

Figure 5: Establishing the new point from the new point and existing point. Figure from [1].

Post-filtering

The post-filtering step is based on visibility violation. If a point has reached a threshold of visibility violations, then it can be considered as an outlier. In other words, if nearby points in the backprojected depth maps project to the same pixel in the 2-D depth map, then they are violating the visibilities of each other. The visibility violation is shown in Figure 6. The vectors v are normalized vectors from two points towards the camera. The vectors n are the normals of the two points. The measurements s are distances from the camera and the two points. The indices are the same from the fusion step. If these criteria are met, meaning that both points point in the same direction and the difference between the two are small, then they violate visibility.

$$\arccos(\mathbf{n}_e \bullet \mathbf{v}_e) < \frac{\pi}{2} \ , \ \arccos(\mathbf{n}_n \bullet \mathbf{v}_n) < \frac{\pi}{2} \text{ and}$$

$$|s_e - s_n| < 0.1 s_n$$

Figure 6: Conditions for violating visibility. Figure from [1].

Pose Refinement

This portion is the main contributions from the authors of this paper. It uses the ICP algorithm to align the original depth maps with the last fused point cloud to minimize the distance between the two. After this ICP, the camera poses are refined, and the fusion portion is repeated. This process can be repeated until the result look good, or if the results do not improve. Though simple, this small step does have positive effects on the reconstruction shown in the experiments performed by the authors.

2. Experiments and Results

The experiments were performed on three datasets: CCorner, Office1, and Office2, captured on the Kinect V2, and were compared to baselines in [2] and [3]. The reconstructions were from 59, 98, and 114 views respectively for each dataset.

Qualitative

The authors first performed a visual experiment shown in Figure 7:

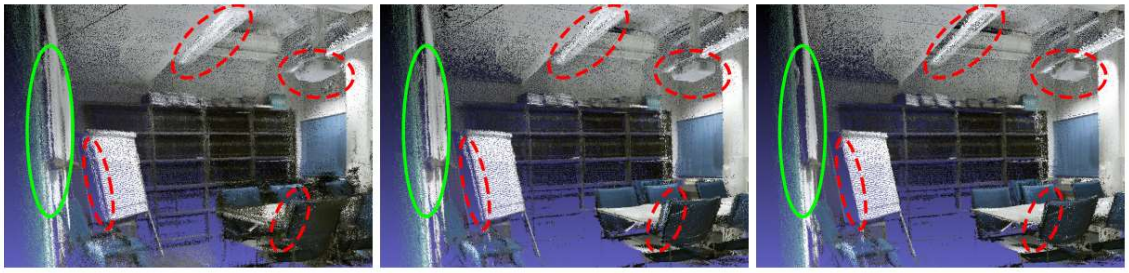

Figure 7: (Left) Method in [2]. (Middle) Method in [3]. (Right) Method described by the authors. Figure from [1].

The red highlighted areas where object boundaries were sharpened compared to baseline methods and the green highlighted surfaces that were reconstructed better. This was done visually.

Quantitative

For quantitative experiments, they compared the number of points in the fused point clouds both with and without their method. They show that the ratio of reduction was higher for their method compared to the other baselines, which meant that they successfully condensed the point cloud.

$$Ratio\ of\ Reduction = 1 - \frac{\text{Count}_{\text{final}}}{\text{Count}_{\text{original}}}$$

Dataset	View count	Original point count	Method	Final point count	Ratio of reduction
CCorner	59	9 307 296	[11]	1 299 555	86.0%
			[12]	939 730	89.9%
			Ours	881 994	90.5%
Office1	98	16 690 662	[11]	5 930 663	64.5%
			[12]	4 352 962	73.9%
			Ours	4 252 937	74.5%
Office2	114	20 400 588	[11]	6 777 222	66.7%
			[12]	5 221 117	74.4%
			Ours	4 956 266	75.7%

Figure : (Left) Ratio of Reduction Formula. (Right) Table comparing different ratios of reduction between datasets and methodologies and the associated number of views used. [11] is [2] and [12] is [3] in our references. Figure from [1].

Secondly, the authors also compared the error on the CCorner dataset with its corresponding ground truth. They calculated the errors by calculating the distance between the reconstructed point and the closest point on the ground truth. By plotting a cumulative occupancy of the distribution of errors, they show that their method creates more points with low error than the baseline methods.

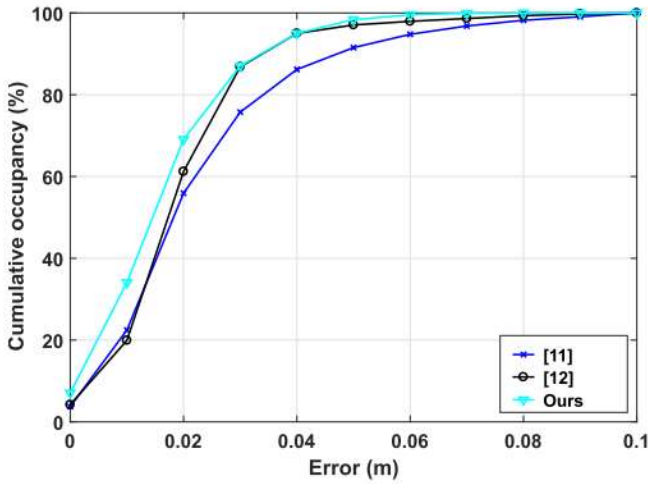

Figure 8: Cumulative occupancy graph detailing the increase in points with lower error. [11] is [2] and [12] is [3] in our references. Figure from [1].

Finally, the authors computed a voxel-based evaluation metric. It uses the Jaccard index, which is calculated by comparing the voxel representation of a reconstruction with that of the ground truth. It measures the number of voxels occupied by both reconstructions over the total number of voxels within a certain threshold. This measures the completeness of the reconstruction. The compactness of the reconstruction is measured by the ratio between the number of points in the ground truth and the reconstruction itself.

		Method		
		[11]	[12]	Ours
Compression ratio		0.443	0.612	0.652
Jaccard index with voxel size	**5mm**	0.027	0.026	0.045
	20mm	0.162	0.174	0.190
	45mm	0.309	0.350	0.355
	85mm	0.388	0.440	0.456

Figure 9: Authors showing that their method achieves a higher Jaccard index and higher compression ratios than other methods. [11] is [2] and [12] is [3] in our references. Figure from [1].

Thoughts:

1. Strengths:
    - This paper was well-formed and provides a strong augmentation to an existing method. The problems and ideas fixing them are described well. Our project has certainly run into some of these problems and can use the ideas provided in this paper to fix some of them.
    - The experiments done were simple and easy to understand and show a solid grasp of what the authors were trying to accomplish and how they measured their success.
2. Weaknesses:
    - The pre-filtering step is described poorly. A visualization of the nearest neighbors, average distances, and the planar point map is better than writing it out in words. It does not provide a reference to any other section that implements this step.
    - Re-registering depth maps for multiple iterations will obviously consume more time and is useful for offline analysis of depth maps, but not real-time analysis. Given a set of hundreds of depth maps from for example, a video source, this re-registration step takes a lot of time.
    - No convergence criteria are discussed for the re-registration step. Do we stop when the ICP stopping criteria is small enough? The authors said that they used six iterations, but do not give any meaningful quantification of why they chose six, or if there were any other criteria to consider. For example, specifics of their ICP algorithm could also be shared or discussed.
    - Pseudocode or a GitHub repository outlining their work would be nice to see.
    - More datasets ran with this method would also be nice to see, as this was only tested on 3 different datasets with a varying number of maps.

References:

[1] Ylimäki, Markus, Janne Heikkilä, and Juho Kannala. "Accurate 3-D Reconstruction with RGB-D Cameras using Depth Map Fusion and Pose Refinement." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.

[2] Kyöstilä, Tomi, et al. "Merging Overlapping Depth Maps into a Nonredundant Point Cloud." Scandinavian Conference on Image Analysis. Springer, Berlin, Heidelberg, 2013.

[3] Ylimäki, Markus, Juho Kannala, and Janne Heikkilä. "Robust and practical depth map fusion for time-of-flight cameras." Scandinavian Conference on Image Analysis. Springer, Cham, 2017.