DEPTH ESTIMATION OF NON-RIGID OBJECTS FOR TIME-OF-FLIGHT IMAGING

James Noraky, Vivienne Sze

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
{jnoraky, sze}@mit.edu

ABSTRACT

Depth sensing is useful for a variety of applications that range from augmented reality to robotics. Time-of-flight (TOF) cameras are appealing because they obtain dense depth measurements with low latency. However, for reasons ranging from power constraints to multi-camera interference, the frequency at which accurate depth measurements can be obtained is reduced. To address this, we propose an algorithm that uses concurrently collected images to estimate the depth of non-rigid objects without using the TOF camera. Our technique models non-rigid objects as locally rigid and uses previous depth measurements along with the optical flow of the images to estimate depth. In particular, we show how we exploit the previous depth measurements to directly estimate pose and how we integrate this with our model to estimate the depth of non-rigid objects by finding the solution to a sparse linear system. We evaluate our technique on a RGB-D dataset of deformable objects, where we estimate depth with a mean relative error of 0.37% and outperform other adapted techniques.

***Index Terms*—** depth estimation, time-of-flight imaging, non-rigid, RGB-D, 3D motion estimation

1. INTRODUCTION

Depth sensing is useful in a variety of applications that range from augmented reality to robotic navigation. One appealing way to measure depth is to use a time-of-flight (TOF) camera, which obtains dense depth measurements by emitting pulses of light and measuring their round trip times [1]. However, for reasons that range from system power constraints to the mitigation of multi-camera interference, the frame rate at which accurate depth measurements can be acquired is often reduced [2, 3, 4]. To address this issue, we present an approach that uses concurrently collected images and previous depth measurements to estimate depth as shown in Figure 1. Images are routinely collected for many applications, and we reuse them to estimate depth maps.

The idea of using image sequences to estimate depth for non-rigid objects has been explored in many related fields [5, 6, 7, 8]. These techniques require only monocular images and many follow a common pipeline that subdivides the image into segments, applies structure-from-motion principles to compute the relative pose, or rotation and translation, of each segment, and then estimate depth up to an unknown scale factor. We can adapt these techniques to address our problem by first using them to estimate depth up to scale and then using previous depth measurements to obtain the scale factor. However, the drawback of adapting these approaches is that they are often complex and inaccurate, which begs the question of whether we can directly incorporate previous depth measurements alongside the image data to minimize computation and increase accuracy. Previously, we presented an algorithm that estimates depth for rigid objects [9]. Here, we extend our work to include objects undergoing non-rigid deformations, which encompasses a larger range of motion.

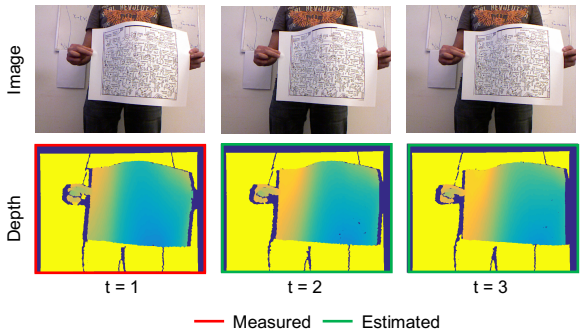

Fig. 1. **Depth Estimation Setup:** Our goal is to use consecutive images and previously measured depth to estimate depth. We denote t as the time (in units of the sampling period) at which frames are acquired.

Our Contribution We introduce an algorithm that estimates the depth of non-rigid objects using consecutive images and previous depth measurements. We model non-rigid objects as locally rigid and use optical flow to estimate the underlying 3D motion and depth. By exploiting the previous depth measurements, we formulate this problem as a sparse linear system, which can be efficiently solved. The resulting algorithm obtains accurate depth compared to other adapted approaches.

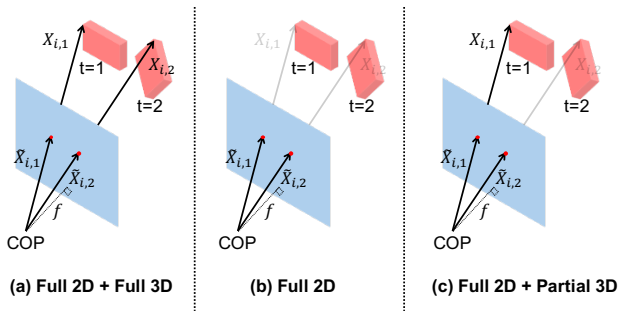

Fig. 2. **Pose Estimation:** We depict the different scenarios where pose can be estimated as the object moves between the first ($t = 1$) and second frame ($t = 2$). The dark arrows indicate the known vectors.

2. DIRECT POSE ESTIMATION

In this section, we describe how we estimate the pose, or rotation and translation, of a rigid segment, which is a prerequisite step in many techniques that estimate the depth of non-rigid objects. In contrast to standard approaches, we show how we directly estimate rotation and *absolute* translation with linear least squares by exploiting previous depth measurements.

To illustrate how pose is estimated, we denote the 3D coordinate of the ith pixel in the first frame as $X_{i,1}$ and its 3D correspondence in the second frame as $X_{i,2}$, which is depicted in Figure 2(a). The pose is the rotation, R, and translation, T, such that:

$$RX_{i,1} + T = X_{i,2} \tag{1}$$

When $X_{i,1}$ and $X_{i,2}$ are not known, which is the case with monocular depth estimation techniques as shown in Figure 2(b), the pose can still be determined in a two-step process, where the essential matrix [10] is first estimated and then factored to obtain R and T (which is known only up to scale). Techniques to estimate the essential matrix range from performing singular value decomposition [10] to finding the roots of a tenth order polynomial [11].

In contrast, because we have previous depth measurements, we have partial 3D information as shown in Figure 2(c). We then exploit the fact that when $X_{i,1}$ is rotated and translated, it is collinear with $\tilde{X}_{i,2}$, which is the vector that connects the center of projection (COP) to its corresponding pixel in the second frame. This results in the following constraint:

$$\tilde{X}_{i,2} \times (RX_{i,1} + T) = 0 \tag{2}$$

where $\times$ denotes the cross product. We further simplify Eq. (2) by assuming that the rotation between frames is small to approximate rotation as:

$$RX_{i,1} \approx X_{i,1} + \omega \times X_{i,1} \tag{3}$$

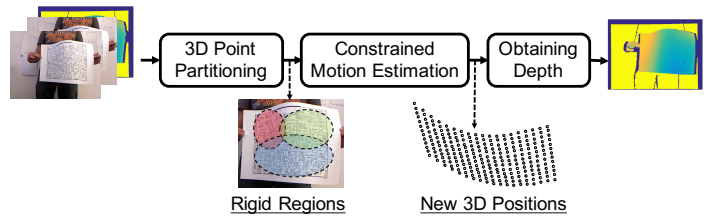

Fig. 3. **Pipeline:** Our algorithm takes as input consecutive images and previous depth measurements. Nearby pixels are partitioned into regions that have the same rigid motion. Because regions can overlap, we solve a constrained optimization problem to estimate the new 3D position of each point and then obtain depth.

for some $\omega \in \mathbb{R}^3$. Using Eq. (3), we rewrite the constraint in Eq. (2) as follows:

$$\begin{pmatrix}[\tilde{X}_{i,2}]_\times & \tilde{X}_{i,2}^T X_{i,1} I - X_{i,1}\tilde{X}_{i,2}^T\end{pmatrix}\begin{pmatrix}T\\ \omega\end{pmatrix} = X_{i,1} \times \tilde{X}_{i,2} \tag{4}$$

where $[\tilde{X}_{i,2}]_\times$ is the skew-symmetric matrix such that $[\tilde{X}_{i,2}]_\times T = \tilde{X}_{i,2} \times T$. We refer to Eq. (4) as the *rigidity assumption* and T and ω as the rigid motion that moves the point $X_{i,1}$ so that it lines up with $\tilde{X}_{i,2}$. This also holds for every pair of corresponding points in a rigid segment and we solve for the pose directly using least squares as opposed to the two steps required with standard techniques [10, 11]. Our approach also only requires three correspondences compared to the minimum of five required for the essential matrix based approaches [11]. This is beneficial when used with RANSAC [12], which obtains robust pose estimates by using this technique to efficiently generate pose hypotheses.

3. DEPTH ESTIMATION OF NON-RIGID OBJECTS

Our algorithm assumes that non-rigid objects are composed of locally rigid segments. As such, our technique first partitions the pixels into rigid regions. We then use the optical flow to estimate the 3D motion and depth of each region. Figure 3 summarizes the pipeline of our approach.

3.1. 3D Point Partitioning

Our approach assumes that nearby points undergo the same rigid motion. As such, we first group these points together using their 3D coordinates. By perspective projection, the ith pixel in the first frame, $x_{i,1} = (u_i, v_i)$, corresponds to the 3D point:

$$X_{i,1} = \frac{z_{i,1}}{f}(u_i - u_c, v_i - v_c, f)^T \tag{5}$$

where (u_c, v_c) is the principal point, f is the focal length, and $z_{i,1}$ is its depth. We then assign each point to regions that are

centered on predefined points. The region centered on the ith point, for example, is defined as the following set of points:

$$C_i = \{X_{j,1} : ||X_{i,1} - X_{j,1}||_2 < \epsilon \quad j = 1, 2, \ldots, N\} \quad (6)$$

where ϵ is the radius and N is the number of points. Because neighboring pixels have similar motion and depth, we only partition the pixels on a subsampled grid and space regions uniformly across it. For our experiments, we subsampled the pixels to a 10×15 grid and centered regions (with $\epsilon = 70$) on every tenth pixel.

Because the points in each region are rigid, they must satisfy Eq. (4). We leverage the rigidity assumption and use RANSAC to retain the points that have residual norms within a carefully selected threshold, ρ. In our experiments, we used $\rho = 0.5$. To obtain $\tilde{X}_{i,2}$ required for Eq. (4), we use optical flow estimated using [13] to find the corresponding pixels. This additional step improves the accuracy of our depth estimates, which we describe in Section 4.2.2.

3.2. Constrained Motion Estimation

Once the points are partitioned, we want to estimate the rigid motion for each region to obtain the new 3D positions. However, some points belong to multiple regions (as shown in Figure 3), and we need to ensure that the new 3D position of each point is consistent across the regions it belongs to. For these points, we have the following *consistency constraint*:

$$\omega_k \times X_{i,1} + T_k = \omega_l \times X_{i,1} + T_l \quad (7)$$

where $X_{i,1} \in C_k \cap C_l$. We can rewrite Eq. (7) in matrix form, which is convenient for our final formulation.

$$\begin{pmatrix} I & -[X_{i,1}]_\times & -I & [X_{i,1}]_\times \end{pmatrix} \begin{pmatrix} T_k \\ \omega_k \\ T_l \\ \omega_l \end{pmatrix} = 0 \quad (8)$$

Putting It All Together Combining the rigidity assumption in Eq. (4) and the consistency constraint in Eq. (8), we can formulate an optimization problem to estimate the motion within each region as:

$$\min \quad \frac{1}{2}||Ap - b||_2^2 \quad \text{s.t.} \quad Dp = 0 \quad (9)$$

where $p = (T_1, \omega_1, \ldots, T_N, \omega_N)^T$ is the concatenation of the motion of all the regions, A and b are the concatenation of the left- and right-hand side of the rigidity assumption in Eq. (4), respectively, and D is the concatenation of the consistency constraints in Eq. (8) for the points that belong to multiple regions. It should be noted that the constraint, $Dp = 0$, forces regions with at least 3 overlapping points to have the same rigid motion. While this may seem like a limitation, it makes intuitive sense that regions with significant overlap should have the same rigid motion. Furthermore, with careful placement of the region centers and selection of the ρ parameter for RANSAC, we can avoid overlapping regions unless the regions have the same rigid motion.

The solution, p, to Eq. (9) can be found by solving the following linear system:

$$\begin{pmatrix} A^T A & D^T \\ D & 0 \end{pmatrix} \begin{pmatrix} p \\ \lambda \end{pmatrix} = \begin{pmatrix} A^T b \\ 0 \end{pmatrix} \quad (10)$$

where λ is a vector of the Lagrange multipliers that enforce the equality constraints. The matrix in Eq. (10) is *sparse*, where $A^T A$ is block-wise diagonal and every row of D contains at most six elements, and this system can be solved efficiently. For Eq. (10) to have a unique solution, the columns of D^T and $(A, D)^T$ must all be linearly independent, and we can select them using QR factorization. Once the motion within each region is estimated, we can obtain the new 3D position for the point $X_{i,1}$ in the kth region as follows:

$$X_{i,2} = X_{i,1} + \omega_k \times X_{i,1} + T_k \quad (11)$$

3.3. Obtaining Depth

To obtain a depth map, we project the depth of each point using the camera intrinsics. Because we estimate motion on a subsampled grid, the resulting depth map is sparse, but we obtain a dense depth map using linear interpolation.

3.4. Limitations of Our Approach

Our algorithm was designed with the assumption that the frame rate at which images are acquired is high, allowing us to approximate rotation, and that optical flow can be accurately estimated. Our technique will disappoint if the frame rate is low, or if the images are textureless.

4. ALGORITHM EVALUATION

4.1. Results

Implementation We implement our algorithm on a laptop with an i5-5257U CPU and an embedded platform [14] with an Exynos 5422 processor. Our laptop implementation can estimate a dense (640×480) and sparse depth map in approximately 0.06 and 0.02 seconds, respectively, and the bottleneck is linear interpolation. When using only the low power Cortex-A7 cores on the embedded platform, which consumes 352 mW (Idle Power: 178 mW), our algorithm obtains dense and sparse depth estimates in 0.3 and 0.09 seconds, respectively. In contrast, approaches like [5] require minutes to obtain depth on a computer with an i7 processor.

Dataset We evaluate our algorithm on both synthetic and real data that have substantial changes in depth from frame to frame. We synthesize 640×480 planar images bending smoothly (*syn_bend*) and being sharply folded in the middle

	Frame Number			
Sequence	2	3	4	Mean
kinect_paper	0.19	0.43	0.23	0.28
kinect_tshirt	0.35	0.52	1.16	0.68
syn_bend	0.27	0.25	0.24	0.26
syn_crease	0.27	0.27	0.27	0.27
Mean - Real	0.27	0.48	0.69	0.48
Mean	0.27	0.37	0.47	0.37

Table 1. **Algorithm Evaluation:** We present the percent MRE for each sequence and frame number for both real (*kinect_paper* and *kinect_tshirt*) and synthetic sequences (*syn_bend* and *syn_crease*).

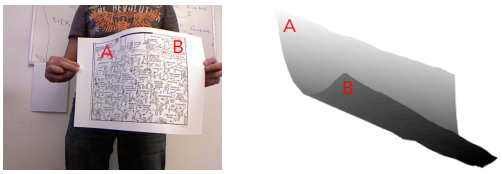

Fig. 4. **Reconstruction of *kinect_paper*:** We present the 3D reconstruction of the second frame ($t = 2$) from Figure 1, which is rotated to show the contours of the paper between points A and B.

(*syn_crease*). We also test our algorithm on the RGB-D sequences *kinect_paper* and *kinect_tshirt* from [15]. We crop out the regions undergoing the non-rigid deformation.

Methodology We assume that we only have depth measurements in the first frame and estimate it until the fourth frame. We quantify the accuracy of our N estimates using the percent mean relative error (MRE): $\frac{100}{N}\sum_{i=1}^{N}\frac{|\hat{z}_i - z_i|}{z_i}$ where $\hat{z}_i$ and z_i are the estimated and ground truth depth for the ith pixel, respectively. We summarize the percent MRE for each sequence and frame in Table 1. The average MRE across all sequences and frames is 0.37%. We also show an example of a 3D reconstruction of the *kinect_paper* sequence in Figure 4.

4.2. Discussion

4.2.1. Algorithm Outperforms Adapted Approach

We compare our algorithm to an adapted approach that first uses techniques like [5] to obtain depth to scale, and then use the previous depth measurements to estimate the unknown scale factor. For approaches that perform monocular depth estimation, this same procedure is followed to evaluate the accuracy of their reconstruction. The authors of [5] also use this procedure to benchmark the performance of their algorithm and other competing techniques on sequences from [15].

As summarized in Table 2, the authors report a MRE of 3.22% for *kinect_paper* and a MRE of 4.20% for *kinect_tshirt*

Sequence	This Work	[5]	Best in [5]
kinect_paper	0.28	4.76	3.22
kinect_tshirt	0.68	4.80	4.20

Table 2. **Method Comparison:** We compare the MRE of our approach to techniques benchmarked in [5].

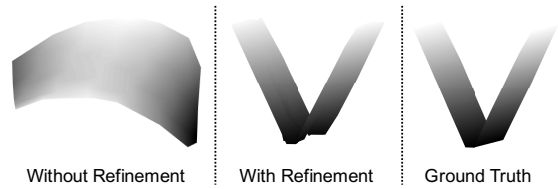

Fig. 5. **Comparing Reconstructed Shape:** Using RANSAC to refine our point partition preserves the underlying shape.

for the best techniques. In contrast, our algorithm achieves a lower MRE of 0.28% and 0.68% for these respective sequences. This suggests that integrating previous depth measurements directly into the depth estimation process not only simplifies our algorithm but also improves its accuracy.

4.2.2. Impact of RANSAC in 3D Point Partitioning on MRE

In our experiments, we find that using RANSAC to refine our regions lowers the MRE by up to 25%. Because the MRE is an average statistic computed over all the pixels, this metric alone does not reflect how our refinement step preserves the underlying structure. To show that the additional refinement step preserves the underlying structure of the depth map, we test our algorithm on *syn_crease* and compare the 3D reconstructions in Figure 5. We see that without this refinement step, the presence of noisy optical flow estimates and our selection of the point partitioning radius results in a curved plane instead of a sharply folded sheet. This failure mode makes sense because nearby points with different motions are partitioned together, and RANSAC mitigates this.

5. CONCLUSION

In this paper, we present a technique to estimate the depth of non-rigid objects using consecutive images and previous depth measurements. Instead of adapting existing techniques to address this problem, we exploit previous depth measurements and our assumption of locally rigid objects to estimate depth using linear least squares. Our proposed solution outperforms adapted techniques and achieves a MRE of 0.37% when evaluated on a RGB-D dataset of deformable objects.

6. ACKNOWLEDGEMENTS

We thank Analog Devices for funding this work and the research scientists within the company for helpful discussions.

7. REFERENCES

[1] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud, *Time-of-Flight Cameras*, SpringerBriefs in Computer Science. Springer London, London, 2013.

[2] Bernhard Büttgen, M'Hamed Ali El Mechat, Felix Lustenberger, and Peter Seitz, "Pseudonoise optical modulation for real-time 3-D imaging with minimum interference," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 10, pp. 2109–2119, 2007.

[3] Bernhard Büttgen and Peter Seitz, "Robust optical time-of-flight range imaging based on smart pixel structures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1512–1525, 2008.

[4] Lianhua Li, Sen Xiang, You Yang, and Li Yu, "Multi-camera interference cancellation of time-of-flight (TOF) cameras," in *International Conference on Image Processing*, 2015, pp. 556–560.

[5] Suryansh Kumar, Yuchao Dai, and Hongdong Li, "Monocular Dense 3D Reconstruction of a Complex Dynamic Scene from Two Perspective Frames," *International Conference on Computer Vision*, pp. 4649–4657, 2017.

[6] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun, "Dense Monocular Depth Estimation in Complex Dynamic Scenes," *Conference on Computer Vision and Pattern Recognition*, pp. 4058–4066, 2016.

[7] Chris Russell, Rui Yu, and Lourdes Agapito, "Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8695 LNCS, pp. 583–598. 2014.

[8] Jonathan Taylor, "Non-Rigid Structure from Locally Rigid Motion," *Science*, pp. 275–287, 2014.

[9] James Noraky and Vivienne Sze, "Low Power Depth Estimation For Time-of-Flight Imaging," *International Conference on Image Processing*, 2017.

[10] Hugh Christopher Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, sep 1981.

[11] David Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, jun 2004.

[12] Martin A Fischler and Robert C Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[13] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, San Francisco, CA, USA, 1981, IJCAI'81, pp. 674–679, Morgan Kaufmann Publishers Inc.

[14] ODROID, "ODROID-XU3," `www.hardkernel.com/main/products/prdt_info.php?g_code=g140448267127`.

[15] Aydin Varol, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun, "A constrained latent variable model," in *Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2248–2255.