

Replacing Projective Data Association with Lucas-Kanade for KinectFusion

Brian Peasley and Stan Birchfield
Electrical and Computer Engineering Dept.
Clemson University, Clemson, SC 29634
{bpeasley, stb}@clemson.edu

Abstract—We propose to overcome a significant limitation of the KinectFusion algorithm, namely, its sole reliance upon geometric information to estimate camera pose. Our approach uses both geometric and color information in a direct manner that uses all the data in order to perform the association of data between two RGBD point clouds. Data association is performed by aligning the two color images associated with the two point clouds by estimating a projective warp using the Lucas-Kanade algorithm. This projective warp is then used to create a correspondence map between the two point clouds, which is then used as the data association for a point-to-plane error minimization. This approach to correspondence allows camera tracking to be maintained through areas of low geometric features. We show that our proposed LKDA data association technique enables accurate scene reconstruction in environments in which low geometric texture causes the existing approach to fail, while at the same time demonstrating that the new technique does not adversely affect results in environments in which the existing technique succeeds.

I. INTRODUCTION

Three-dimensional (3D) reconstruction of an environment is an important problem in robotics that has received much attention in past decades. The advent of inexpensive RGBD sensing (e.g., the Kinect sensor) has facilitated a quantum leap in terms of both the fidelity and speed with which such reconstructions can be made. These capabilities have far-reaching implications for the exploration and navigation of unknown environments, as well as the manipulation of objects in those environments.

Of the many 3D modeling techniques that have been developed, the landmark KinectFusion method [18], [15] has shown perhaps the most impressive results for an accurate, real-time, dense modeling system. Recent work by others [26] has extended this algorithm to operate over large scale environments. However, both the standard and the extended version use geometric information alone to align the camera with the model. As a result, they require the environment to contain sufficient geometric features in order for the iterative closest point (ICP) algorithm implemented in KinectFusion to accurately estimate the camera pose. This

deficiency was noted in [26], where experiments were conducted to explore the advantage of replacing the camera estimation of KinectFusion with the output of a feature-based visual odometry system called FOVIS [13] to provide a more stable estimate of camera pose. The frame-to-frame nature of FOVIS, however, loses the drift-free property of KinectFusion.

Rather than replacing the camera estimation algorithm with an entirely different technique, in this paper we propose a different approach, namely, to modify how data correspondence is obtained by using not only geometric information but also radiometric information. That is, both depth and color are used to align the point clouds obtained from an RGBD sensor.

The KinectFusion variant of ICP uses the projective data association (PDA) point matching algorithm [4] to minimize a point-to-plane error metric [5]. We replace PDA with a data association technique driven by Lucas-Kanade alignment [17], [24], [3]. The resulting Lucas-Kanade Data Association (LKDA) matching algorithm is a *direct* method that uses *all* the data. Another advantage of this direct approach is that it obviates the need for the feature point extraction and feature correspondence steps inherent in feature-based visual odometry systems. By a set of experiments, we show that our LKDA technique enables camera tracking to succeed in areas of low geometry while at the same time maintaining accurate camera tracking in environments in which the standard PDA approach performs well.

II. RELATED WORK

The literature on 3D reconstruction, or simultaneous localization and mapping (SLAM), is immense. In this section we focus on approaches that operate on RGBD sensors or otherwise incorporate color information into the mapping process. One approach to use visual information to improve upon geometric mapping is that of Henry et al. [11], which performs an initial estimate of the 3D camera transformation by applying RANSAC to SIFT feature matches with depth values. These visual feature associations are then combined with dense point associations in an ICP framework to minimize both

geometric point-to-plane error and visual point-to-point error. The final transformations are used to produce a pose graph, which is then optimized in a final post-processing step to yield a globally consistent map as a surfel representation. In a followup paper [12], the authors avoid the computationally expensive ICP step whenever enough visual keypoints can be found.

Other researchers have followed a similar approach. Engelhard et al. [9] use SURF instead of SIFT features. Endres et al. [10] compare SURF, SIFT, and ORB features on public datasets using a system that also yields globally consistent transformations from post-processing optimization of pose-graph maps. The final representation of their system, however, uses a volumetric OctoMap rather than surfels in order to efficiently encode free spaces and unmapped areas. Our recent work [19] performs similarly, creating volumetric OctoMap representations of large-scale indoor environments by leveraging the Manhattan World assumption to avoid rotational drift without having to perform loop closure. This approach also involves post-processing data using a pose-graph map. Note that the advantage of KinectFusion [18], [15], along with the approach proposed in this paper, is that no such post-processing is necessary.

Another body of work involves improving upon ICP by incorporating color information. Druon et al. [7] segment point clouds based on the hue component of the HSV color space, then perform ICP while requiring matching points to belong to the same color class. Douadi et al. [6] incorporate both geometric and color information into the distance metric used by ICP. Huhle et al. [14] register scans of 3D point data by extending the standard metric with color information using Gaussian mixture models in a color space. Joung et al. [16] extract feature points from images using SIFT to find a set of correspondences between two laser scans, which provide an initial alignment for the standard ICP algorithm.

In work that is perhaps most similar to ours, Tykkälä et al. [25] formulate the 3D registration of point clouds as a direct image-based minimization task. In a manner similar to projective data association (PDA), they store point cloud data as images to avoid expensive nearest-neighbor searches in 3D. Unlike our approach, which adds color to a point-to-plane version of ICP, their approach adds depth to visual odometry. In followup work [1], the same researchers adopt the simpler approach of minimizing the sum-of-squared distances of image intensities under a projective model, without using depth, in order to compute the transformation of the camera in order to reduce the computational load. The resulting system is able to produce 2D reconstructions of environments. Like this work, our approach also involves direct matching of all the image data, thus

avoiding the various problems associated with feature point detection and matching, but leveraging the benefits of the KinectFusion framework.

III. PROJECTIVE DATA ASSOCIATION

Let $P = \{p_i\}_{i=1}^N$ and $Q = \{q_i\}_{i=1}^N$ be two point clouds in 3D Euclidean space, $p_i, q_i \in \mathbb{R}^3$. In the context of KinectFusion, P is obtained from the current scan of the RGBD sensor, while Q is obtained by projecting the world model onto a virtual RGBD sensor located at the camera's estimated location at the previous frame. Note that both point clouds are transformed (after alignment) back into a global coordinate frame in the actual system, although for brevity we omit these details.

ICP (Iterative Closest Point) is a family of algorithms to incrementally align two point clouds. As explained in [21], the ICP approach can be broken into six stages, and different choices within these stages lead to different variations of the algorithm. In this section we summarize the version of the algorithm that is used by KinectFusion [18], [15], namely Projective Data Association (PDA) with point-to-plane error minimization. ICP is an iterative algorithm, with the stages described here being repeated at each iteration. For ease of reference, we indicate by an asterisk (*) the stages that are modified by our approach, as described in the next section.

A. Point Selection

Determine which points from the two clouds to use. KinectFusion uses all points, with no initial filter to remove points to be included for alignment.

B. Point Matching*

The next step is to establish data association, or correspondence between the points in the two clouds. We seek a function $L : \mathbb{Z}_{1:N} \rightarrow \mathbb{Z}_{1:N}$ that maps indices in one point cloud to those of the other, so that $L(i) = j$ indicates a corresponding pair of points $p_i \leftrightarrow q_j$, where $\mathbb{Z}_{1:N} = \{1, \dots, N\}$. This is achieved in KinectFusion by Projective Data Association (PDA) [4], which is a camera-centric approach to data association that is especially suitable to point clouds obtained from a depth sensor. For each point $p_i \in P$, its corresponding point q_i is the closest point in Q to the line of projection of p_i into the camera used to obtain the points of the second cloud, Q .

By assuming that the change between the two point clouds to be aligned is small, which is warranted due to the real-time nature of the system, PDA achieves data association in an efficient manner. As shown in Figure 1, correspondences are obtained by projecting the points in one cloud, P , onto the image plane of the camera used to obtain the points of the other cloud, Q . The point in Q associated with the pixel onto which p_i is

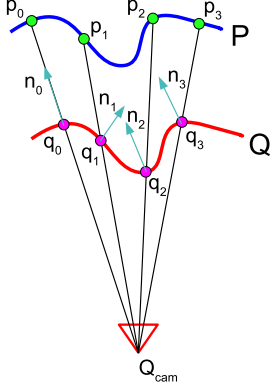


Fig. 1. Projective Data Association (PDA) establishes correspondences between point clouds P and Q by finding the closest point after projecting onto the camera Q_{cam} . The normals (n_i) are used in the point-to-plane error metric.

projected is therefore established as the corresponding point. More formally, if C_Q is the 3×4 projection matrix of the camera used to obtain Q , and $\tilde{p}_i = [p_i^T \ 1]^T$ and $\tilde{q}_i = [q_i^T \ 1]^T$ are the homogeneous coordinates of the points, then PDA establishes correspondence as follows:

$$L(i) = \arg \min_j \|\phi(C_Q \tilde{q}_j) - \phi(C_Q T \tilde{p}_i)\|, \quad (1)$$

where $\phi([x \ y \ w]^T) = [x/w \ y/w]^T$ are the dehomogenized point coordinates, and

$$T = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} \quad (2)$$

is the 4×4 Euclidean transformation between the point clouds, and 0_3 is a 3×1 vector of all zeros.

C. Weighting

As with point selection, KinectFusion does not perform any special step to weight the influence of some points more than others. All point correspondences are weighted equally.

D. Rejection*

The next step is to remove outliers from the correspondence to avoid corrupting the computed transformation. In KinectFusion, the corresponding pair $p_i \leftrightarrow q_i$ is rejected if either of the following conditions is met:

$$\|\hat{p}_i - q_i\| > \tau_{dist} \quad (3)$$

$$\frac{\|n_i \times \hat{p}_i\|}{\|\hat{p}_i\|} > \tau_{ang}, \quad (4)$$

where $\hat{p}_i = [I_3 \ 0_3] T \tilde{p}_i$ is a 3×1 vector, I_3 is the 3×3 identity matrix, and τ_{dist} and τ_{ang} are a predefined distance and angle threshold, respectively.

E. Error Metric

The most obvious error metric is the sum of the Euclidean distances between corresponding points after alignment:

$$E_{\text{point-to-point}}(T; P, Q, L) = \sum_{i=1}^N \|T \tilde{p}_i - \tilde{q}_{L(i)}\|^2. \quad (5)$$

Instead of this point-to-point error metric, KinectFusion uses a point-to-plane error metric [5]:

$$E_{\text{point-to-plane}}(T; P, Q, L) = \sum_{i=1}^N \left((T \tilde{p}_i - \tilde{q}_{L(i)})^T \tilde{n}_{L(i)} \right)^2, \quad (6)$$

where n_i is the surface normal at q_i , obtained by the eigenvector associated with the smallest eigenvalue of the covariance matrix of points in a small neighborhood surrounding q_i , and $\tilde{n}_i = [n_i^T \ 1]^T$. The combination of PDA and the point-to-plane error metric was first implemented by a real-time modelling system [20].

F. Minimize

Since each iteration yields only small changes in the transformation, the rotational component of the transformation matrix can be linearized about the current estimate:

$$T^{(k)} \approx \begin{bmatrix} 1 & -\psi & \theta & t_x \\ \psi & 1 & -\phi & t_y \\ -\theta & \phi & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

where k is the iteration, and $T = T^{(k)} \dots T^{(2)} T^{(1)}$.

The error metric in (6) is minimized by iteratively solving a standard least squares system:

$$Zu = e, \quad (8)$$

where

$$u = [\phi \ \theta \ \psi \ t_x \ t_y \ t_z]^T \quad (9)$$

are the 6 unknowns of the Euclidean transformation (3 Euler angles of the rotation matrix R , and 3 components of the translation vector t), and

$$Z_{\{6 \times 6\}} = \sum_{i=1}^N \begin{bmatrix} (n_i \times \hat{p}_i)(n_i \times \hat{p}_i)^T & (n_i \times \hat{p}_i)n_i^T \\ n_i(n_i \times \hat{p}_i)^T & n_i n_i^T \end{bmatrix} \quad (10)$$

$$e_{\{6 \times 1\}} = \sum_{i=1}^N \begin{bmatrix} (n_i \times \hat{p}_i)((\hat{p}_i - q_i)^T n_i) \\ n_i((\hat{p}_i - q_i)^T n_i) \end{bmatrix}. \quad (11)$$

Each iteration of the least-squares algorithm yields the parameters for the linearized transformation matrix. These parameters are then substituted into (2), and the non-linear transformation matrices from all iterations are multiplied to obtain the overall transformation.

IV. LUCAS-KANADE DATA ASSOCIATION

The KinectFusion ICP algorithm described in the previous section relies solely upon geometric information, i.e., the spatial coordinates of the points in the two clouds. As a result, environments that do not provide high geometric texture (e.g., a plane such as a wall) can cause camera tracking to fail. To improve the alignment of point clouds in such environments, we propose to use the RGB color information. While existing systems such as [13] and [12] use visual cues alongside the depth information for camera tracking, these approaches rely upon extracted sparse feature points to estimate the relative transformation of a camera. In contrast, we describe in this section our approach for incorporating *all* the RGB information in a natural way to replace PDA as the data association technique for KinectFusion.

A. Lucas-Kanade

The Lucas-Kanade algorithm is a differential method for computing the optical flow of an image. The goal of Lucas-Kanade is to find the parameters ζ that minimize the SSD error

$$E_{LK} = \sum_x \sum_y (I(W^{-1}(x, y; \zeta)) - J(x, y))^2, \quad (12)$$

where I and J are the two images, the double summation is over all the pixels, and $W(x, y; \zeta)$ is a parametric warp function that brings the two images into alignment. This equation is minimized by linearizing about the current estimate and repeatedly solving a linear system:

$$\Delta\zeta = H^{-1} \sum_x \sum_y \left(\nabla J \frac{\partial W}{\partial P} \right)^T \delta_{IJ}, \quad (13)$$

where

$$H = \sum_x \sum_y \left(\nabla J \frac{\partial W}{\partial P} \right)^T \nabla J \frac{\partial W}{\partial T} \quad (14)$$

$$\delta_{IJ} = I(W^{-1}(x, y; \zeta)) - J(x, y), \quad (15)$$

and where ∇J is the 1×2 vector containing the x and y gradients of image J . $\Delta\zeta$ is computed incrementally until the algorithm converges, or a maximum number of iterations has been reached. For efficiency, we use the inverse compositional algorithm [2].

The above formulation is valid for any warp function. Since we are interested in a global, featureless mapping technique that warps an entire image into another, we use a projective warp to more accurately model the relationship between the two images:

$$\tilde{W}(x, y; \zeta) = \underbrace{\begin{bmatrix} R_{xx} + 1 & R_{xy} & a_x \\ R_{yx} & R_{yy} + 1 & a_y \\ s_x & s_y & 1 \end{bmatrix}}_{\text{projective warp}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (16)$$

which in inhomogeneous coordinates, is

$$W(x, y; \zeta) = \begin{bmatrix} \frac{x(R_{xx}+1)+yR_{xy}+a_x}{s_x x + s_y y + 1} \\ \frac{xR_{yx}+y(R_{yy}+1)+a_y}{s_x x + s_y y + 1} \end{bmatrix}. \quad (17)$$

Differentiating leads to

$$\frac{\partial W}{\partial \zeta} = \frac{1}{\lambda} \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & \frac{\partial W_x}{\partial s_x} & \frac{\partial W_x}{\partial s_y} \\ 0 & 0 & 0 & x & y & 1 & \frac{\partial W_y}{\partial s_x} & \frac{\partial W_y}{\partial s_y} \end{bmatrix}, \quad (18)$$

where

$$\lambda = xs_x + ys_y + 1, \quad (19)$$

and the derivatives of the two elements of W with respect to the parameters s_x and s_y are given by

$$\frac{\partial W_x}{\partial s_x} = -\frac{x(a_x + yR_{xy} + x(R_{xx} + 1))}{\lambda} \quad (20)$$

$$\frac{\partial W_x}{\partial s_y} = -\frac{y(a_x + xR_{yx} + x(R_{xx} + 1))}{\lambda} \quad (21)$$

$$\frac{\partial W_y}{\partial s_x} = -\frac{x(a_y + xR_{yx} + y(R_{yy} + 1))}{\lambda} \quad (22)$$

$$\frac{\partial W_y}{\partial s_y} = -\frac{y(a_y + xR_{yx} + y(R_{yy} + 1))}{\lambda}. \quad (23)$$

B. Integrating LKDA into KinectFusion

As we saw in (1), projective data association (PDA) establishes correspondence by finding the closest point from the other cloud as projected onto the image plane. In a similar manner, Lucas-Kanade data association (LKDA) also finds the closest point on the image plane, but only after first transforming the projected coordinates according to the warp function found by Lucas-Kanade:

$$L(i) = \arg \min_j \|\phi(C_Q \tilde{q}_j) - \phi(\tilde{W}(C_P \tilde{p}_i; \zeta))\|. \quad (24)$$

In PDA, point correspondences are computed by projecting one point cloud onto the other with respect to the other's camera. In our LKDA approach, the warping function from Lucas-Kanade yields the correspondence. The modified ICP steps are as follows.

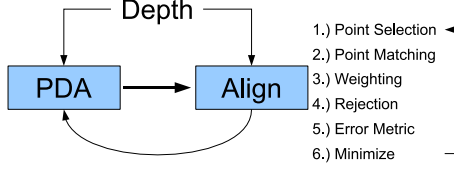
***Point Matching:** Correspondences are found by computing the parameters, ζ , that warp image I to image J . Although these correspondences could be recomputed in each iteration of ICP, as in the traditional approach, we have found this to be unnecessary. As a result, once the warping parameters have been found, they are used to generate a correspondence map C_{map} , so that the problem of finding a corresponding point simply requires a lookup in this map. Pseudocode for computing the correspondence map is shown in Algorithm 1, where $d_P(x, y)$ is the depth value at pixel (x, y) , $Proj^{-1}$ computes the 3D point using the camera's internal calibration parameters, $V_P(x, y) \in \mathbb{R}^3$.

Algorithm 1 LKDA Correspondence Map

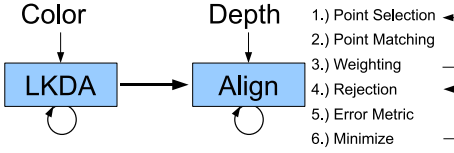
```

 $\zeta \leftarrow$  Perform Lucas-Kanade
for each pixel  $(x, y)$  in image domain do
     $C_{map}(x, y) \leftarrow W(x, y; \zeta)$ 
end for

```



Pipeline of the ICP implementation in KinectFusion.



Pipeline of our LKDA approach outlined in this paper.

Fig. 2. Unlike the ICP process used by KinectFusion, our approach utilizes both color and depth information. Another difference is that correspondences are established once by LKDA allowing us to achieve more robust results with less computation.

$$Proj^{-1} = \begin{bmatrix} \frac{(x-c_x)}{f_x} z & \frac{(y-c_y)}{f_y} z & z \end{bmatrix}^T \quad (25)$$

***Rejection:** Rejection of point correspondences is done as described earlier in section III-D with the added constraint of requiring the corresponding points to have a similar color:

$$\|I(x, y) - J(W(x, y; \zeta))\| > \tau_{color}, \quad (26)$$

where γ is a predefined color threshold. Algorithm 2 presents our alignment process without finding new correspondences in each iteration.

An illustration of our approach compared with the PDA approach used by KinectFusion is shown in Figure 2. As mentioned above, our approach uses the same correspondences during each iteration of the cloud alignment process rather than having to re-establish correspondences each time. Like KinectFusion, we also use a point-to-plane error metric, although any error metric may be used.

Algorithm 2 LKDA Point Cloud Alignment

```

// Compute vertex and normal maps
for each pixel  $(x, y)$  in the depth map(s) do
     $V_P(x, y) \leftarrow Proj^{-1}(x, y, d_P(x, y))$ 
     $V_Q(x, y) \leftarrow Proj^{-1}(x, y, d_Q(x, y))$ 
     $N(x, y) \leftarrow$  normal vector of  $V_Q(x, y)$ 
end for
// Compute alignment
while not aligned do
    for each pixel  $(x, y)$  in depth map  $d_P$  do
         $p \leftarrow TV_P(x, y)$ 
         $q \leftarrow V_Q(C_{map}(x, y))$ 
         $n \leftarrow N(C_{map}(x, y))$ 
        if  $\|p - q\| > \tau_{dist}$  or  $\|n \times p\| / \|p\| > \tau_{ang}$  or
             $\|I(x, y) - J(W(x, y; \zeta))\| > \tau_{color}$ 
        then
            Reject point correspondence
        end if
    end for
     $T^{(k)} \leftarrow$  solve linear system
    Update  $T \leftarrow T^{(k)} \dots T^{(2)} T^{(1)}$ 
end while

```

V. EXPERIMENTS

We integrated LKDA into the open source implementation of KinectFusion, namely KinFu, which is part of the Point Cloud Library (PCL) [22]. The PDA technique used to find correspondences was replaced by the correspondence map generated from Lucas-Kanade, and the rejection step was given an additional color constraint, both of which are described in section IV. The remainder of the system was left unaltered.

Since the main contribution of this paper is to improve the robustness of camera tracking of the KinectFusion algorithm by replacing PDA with LKDA, we compare the results of KinectFusion with PDA versus with LKDA. Our approach was tested in environments in which there are few geometric features, as well as environments in which the relatively large distance of objects to the RGB-D sensor causes the depth data to be noisy as a result of the limited range of the sensor. Results from the two approaches for an environment with few geometric features can be seen in Figure 3. With PDA, the KinectFusion algorithm is unable to maintain camera tracking, therefore producing an incorrect reconstruction of the scene. Notice, for example, that the poster is too close to the table, and the electrical outlet has disappeared entirely. On the other hand, KinectFusion with LKDA is able to maintain camera tracking throughout the sequence, resulting in an accurate reconstruction. This is also clearly seen in the plot in Figure 3d, where PDA fails to continue to track the upward (negative y) motion of the camera after frame 100.



Fig. 5. 3D reconstruction of a person using LKDA for data association. The resulting point cloud was converted to a triangular mesh for rendering.

The second environment contained plenty of geometric features, but these features were farther away than the effective range of our RGB-D (Kinect) sensor, which is approximately 3 meters. This caused the depth measurements to be noisy which, in turn, caused the PDA algorithm to fail. In contrast, the robustness of our LKDA approach allowed KinectFusion to maintain correct camera tracking despite the presence of noisy depth data. Results of a reconstruction in this environment can be seen in Figure 4. Notice that the person is painted onto the wall by PDA rather than being detected as standing in front of the wall; in other words, the depth information has been lost. (This behavior is more clearly seen in the supplemental video which can be found at <http://www.ces.clemson.edu/~stb/research/LKDA>) This particular sequence was taken by a camera that rotated about its center, so that its x , y , and z coordinates did not change. The plot in Figure 4d shows that LKDA, unlike PDA, maintained consistent coordinates over time.

Since LKDA finds correspondence from the color information and not the geometric information, it is necessary to show that LKDA still performs in areas with rich geometric features. These tests show that, despite its ignorance of geometry for the purpose of establishing correspondence, LKDA still produces high quality results in areas with many geometric features, therefore maintaining accurate camera tracking over extended frame sequences. Figure 5 shows the reconstruction of a person, and Figure 6 shows the reconstruction of an office scene. Thus we see that LKDA not only improves upon PDA in environments in which the latter fails, but it also does not hinder the performance of KinectFusion in environments in which it succeeds.

VI. CONCLUSION

This paper has addressed the problem of aligning point clouds obtained from an RGBD sensor. Replacing the traditional projective data association (PDA) that



Fig. 6. Reconstruction of a scene using LKDA over an extended period of time (1100 frames of data).

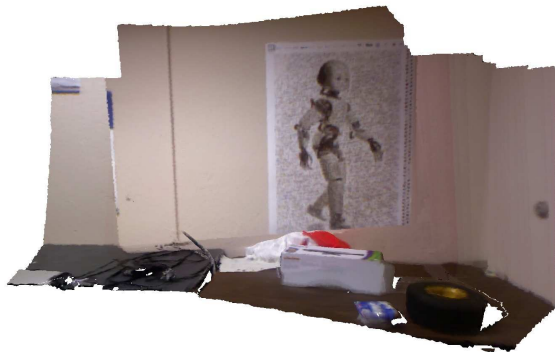
only uses geometric information, our approach of Lucas-Kanade projective data association (LKDA) uses color information to establish correspondence between the two point clouds. Once established, this correspondence can be used in subsequent iterations of ICP in order to avoid having to recompute correspondence each time. Our approach was integrated into the KinectFusion algorithm, showing its ability to maintain camera tracking through areas with few geometric features, as well as in environments in which the objects are beyond the range of accurate distance measurements. Moreover, even when geometric information is sufficient, using LKDA instead of the traditional PDA does not result in a loss of accuracy. The use of LKDA will not hinder the real-time performance in the KinectFusion algorithm due to its ability to be implemented on a GPU to run in real time [23], [8]. Future work should focus upon using depth information in the Lucas-Kanade step, applying LKDA to larger environments, and further experiments to validate its applicability in other scenarios.

VII. ACKNOWLEDGMENT

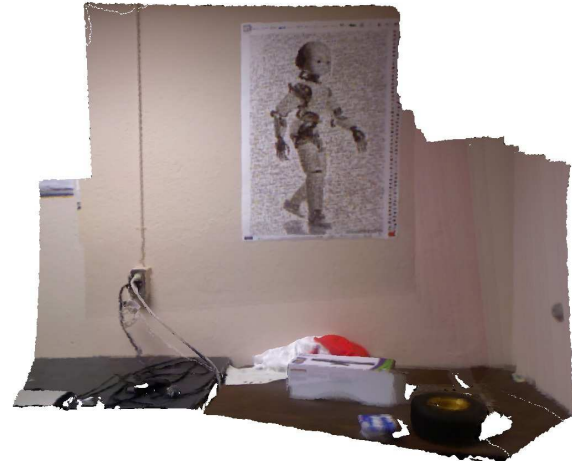
This research was supported by the U.S. National Science Foundation under grant IIS-1017007.

REFERENCES

- [1] C. Audras, A. I. Comport, M. Meilland, and P. Rives. Real-time dense RGB-D localisation and mapping. In *Australian Conference on Robotics and Automation*, Dec. 2011.
- [2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2001.
- [3] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [4] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):820–824, Aug. 1995.



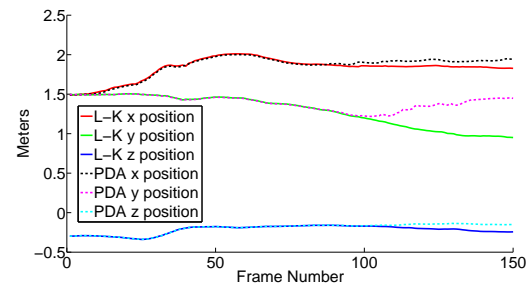
(a) Projective Data Association



(b) Lucas-Kanade Projective Data Association



(c) Image of environment



(d) Estimated position of camera

Fig. 3. Top: 3D reconstructions from the traditional PDA approach and our LKDA approach. When compared with an image of the scene (bottom-left), note that PDA misplaces the poster and averages out the wall pixels which results in a complete loss of the electrical outlet. These errors are due to the planar nature of the scene, which does not provide enough geometric information for PDA. This limitation is overcome by LKDA by using color information. The bottom-right shows a plot of the position of the camera as estimated by the two algorithms. After frame 100, LKDA fails to track the upward movement of the camera. (Note that upward movement is seen as decreasing y values.)

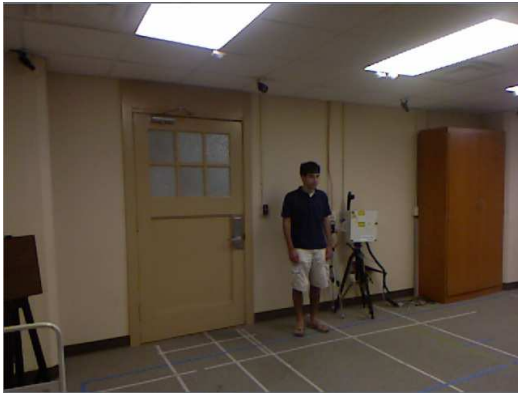
- [5] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, volume 3, pages 2724–2729, Apr 1991.
- [6] L. Douadi, M.-J. Aldon, and A. Crosnier. Pair-wise registration of 3D/color data sets with ICP. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 663 – 668, Oct 2006.
- [7] S. Druon, M. Aldon, and A. Crosnier. Color constrained ICP for registration of large unstructured 3D color data sets. In *IEEE International Conference on Information Acquisition*, pages 249 – 255, Aug 2006.
- [8] B. Duvénage, J. P. Delport, and J. de Villiers. Implementation of the lucas-kanade image registration algorithm on a GPU for 3D computational platform stabilisation. In *International Conference on Virtual Reality, Computer Graphics, Visualization and Interaction in Africa (AFRIGRAPH)*, pages 83–90, 2010.
- [9] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3D visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Apr 2011.
- [10] J. H. F. Endres, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Proceedings of the International Conference on Robotics and Automation*, May 2012.
- [11] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Dec. 2010.
- [12] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663, Apr 2012.
- [13] A. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research (ISRR)*, Aug. 2011.
- [14] B. Huhle, M. Magnusson, W. Strasser, and A. J. Lilienthal. Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform. In *Proceedings of the International Conference on Robotics and Automation*, pages 4025 – 4030, May 2008.
- [15] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of*



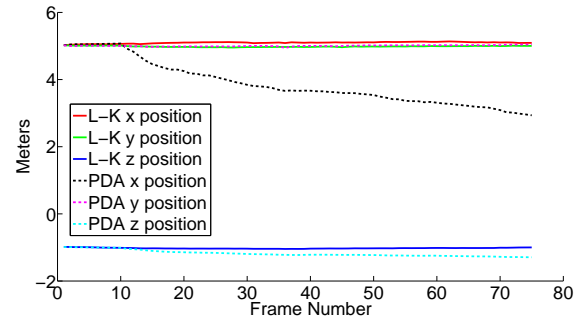
(a) Projective Data Association



(b) Lucas-Kanade Projective Data Association



(c) Image of environment



(d) Estimated position of camera

Fig. 4. 3D reconstructions of a different environment. KinectFusion with PDA is unable to maintain tracking, resulting in a flat 3D reconstruction that does not capture the geometry of the scene. For this sequence, the camera was rotated about its center, so that its position remained stationary. Note that the x position when using PDA is inaccurate, with an error of nearly 2 meters after only 75 frames.

- the 24th ACM Symposium on User Interface Software and Technology, pages 559–568, 2011.
- [16] J. H. Joung, K. H. An, J. W. Kang, M. J. Chung, and W. Yu. 3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3082 – 3088, Oct. 2009.
 - [17] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
 - [18] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.
 - [19] B. Peasley, S. Birchfield, A. Cunningham, and F. Dellaert. Accurate on-line 3D occupancy grids using Manhattan world constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012.
 - [20] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (SIGGRAPH)*, 2002.
 - [21] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.
 - [22] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the International Conference on Robotics and Automation*, 2011.
 - [23] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *EDGE Workshop on Edge Computing Using New Commodity Architectures*, May 2006.
 - [24] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.
 - [25] T. Tykkälä, C. Audras, and A. I. Comport. Direct iterative closest point for real-time visual odometry. In *Second International Workshop on Computer Vision in Vehicle Technology*, Nov. 2011.
 - [26] T. Whelan, J. B. McDonald, M. Kaess, M. F. Fallon, H. Johannsson, and J. J. Leonard. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, July 2012.