

Efficient Tracking with the Bounded Hough Transform

Michael Greenspan^{1,2,4} Limin Shang¹ Piotr Jasiobedzki³

¹Dept. of Electrical & Computer Engineering, ²School of Computing, Queen's University, Canada

³MDRobotics, 9445 Airport Rd., Brampton, Ontario, Canada

⁴corresponding author: michael.greenspan@ece.queensu.ca

Abstract

The Bounded Hough Transform is introduced to track objects in a sequence of sparse range images. The method is based upon a variation of the General Hough Transform that exploits the coherence across image frames that results from the relationship between known bounds on the object's velocity and the sensor frame rate. It is extremely efficient, running in $O(N)$ for N range data points, and effectively trades off localization precision for runtime efficiency.

The method has been implemented and tested on a variety of objects, including freeform surfaces, using both simulated and real data from Lidar and stereovision sensors. The motion bounds allow the inter-frame transformation space to be reduced to a reasonable, and indeed small size, containing only 729 possible states. In a variation, the rotational subspace is projected onto the translational subspace, which further reduces the transformation space to only 54 states. Experimental results confirm that the technique works well with very sparse data, possibly comprising only tens of points per frame, and that it is also robust to measurement error and outliers.

Keywords: tracking, pose determination, hough transform, range image

1 Introduction

Tracking objects in a time sequence of images is a problem of general interest in the computer vision literature. Given an initial estimate of an object's pose, the goal is to efficiently localize the moving object in each subsequent image frame. Tracking is related to pose determination, although it is made simpler due to the high degree of coherence between successive image frames. So long as the frame rate is fast enough with respect to the object's velocity, then the pose estimate can be propagated to each subsequent frame. This effectively reduces the size of the search space, and most tracking methods exploit this coherence to improve efficiency.

In intensity images, one class of tracking technique establishes correspondences across frames between extracted image features. For 3D data it is more common to track us-

ing variants of the Iterative Closest Point Algorithm (ICP) [1]. This is primarily because range data is more expensive to collect, and so the images tend to be sparse, which makes it difficult to extract meaningful features. Examples of ICP-based tracking are [2, 3] and recently [4], which simultaneously reconstructs while tracking.

The Hough Transform is a well known and effective method of feature extraction and pose determination that has been explored thoroughly in the literature [5]. Many variations of the Hough Transform have been proposed [6], some of which are specifically tailored to tracking. The Velocity Hough Transform (VHT) [7] included a specific velocity term in the parametric expression of a circle. This increased the dimension of the parameter space, and was recently extended [8] to allow for arbitrary motions. Another Hough variation used motion bounds to establish correspondences between Hough space peaks across successive frames to track line features in range image sequences for purposes of robotic navigation [9].

In this paper we introduce the Bounded Hough Transform (BHT). The BHT is a variation of the General Hough Transform that exploits coherence across image frames and effectively trades off localization precision for runtime efficiency.

2 Problem Definition

Let \mathbf{M} be an object defined within an egocentric coordinate system \mathcal{M} . Our goal is to track the pose of \mathbf{M} as it is transformed rigidly through a time sequence of frames. At each frame t , a set of range data $\mathbf{P}_t = \{\vec{p}_i\}_1^{N_t}$ is acquired by sampling the surfaces of \mathbf{M} within the sensor coordinate system \mathcal{S} . We assume that the data is acquired with a conventional range sensor such as a time-of-flight, triangulation, or stereovision sensor. Each datum $\vec{p}_i \in \mathbf{P}_t$ is therefore a noisy measurement of some 3D coordinate on a surface of \mathbf{M} that is non-occluded with respect to the sensor vantage. We implicitly assume that all data points are acquired at the same time instance. This is strictly correct for range measurements obtained from stereovision where the full images are captured simultaneously, but it is only an approxima-

tion of the acquisition process using scanning rangefinders where the data is acquired sequentially. This approximation is only accurate when the object motion during one scan can be neglected. Our technique attempts to reduce the number of necessary measurements, which helps to uphold this approximation. We allow that the cardinality of each \mathbf{P}_t may (but need not) be small, comprising only a small number (possibly only tens) of points.

We assume that an initial transformation A_0 is known, that maps \mathcal{M} from its *canonical pose* in \mathcal{M} to its initial known pose in \mathcal{S} . The value of A_0 may be known a priori, provided interactively, or established automatically albeit relatively expensively with a pose determination method.

The motion of \mathcal{M} between two successive frames $t-1$ and t is a rigid transformation B_t^{t-1} . We assume that the magnitude of B_t^{t-1} is bounded:

$$B_t^{t-1} \in \Theta^d, \quad \|\Theta^d\| \leq K \quad (1)$$

where Θ^d is a d -dimensional transformation space. The magnitude of K depends upon the physical limitations of the object motion and the sensor acquisition rate. There are no restrictions placed on the possible direction of the components of B_t^{t-1} within Θ^d . The object may therefore change direction between frames arbitrarily, as long as the magnitude of the inter-frame displacement remains bounded.

During tracking at frame t , \mathbf{P}_t is first mapped into \mathcal{M} by applying to it the previous frame's estimate:

$${}^{\mathcal{M}}\mathbf{P}_t = A_{t-1}\mathbf{P}_t \quad (2)$$

By default, a lack of superscript on the data will indicate the sensor coordinate system, i.e., $\mathbf{P}_t = {}^{\mathcal{S}}\mathbf{P}_t$.

${}^{\mathcal{M}}\mathbf{P}_t$ denotes the sensed points within \mathcal{M} for a pose B_t^{t-1} that is perturbed by a bounded amount from the canonical pose. The value of this perturbation is the same as the relative transformation between the current and the previous frames within the sensor coordinate system \mathcal{S} :

$$B_t^{t-1} = A_t A_{t-1}^{-1} \quad (3)$$

From the estimated value \hat{B}_t^{t-1} and the previous frame's pose estimate \hat{A}_{t-1} , a current pose estimate \hat{A}_t is determined as:

$$\hat{A}_t = (\hat{A}_{t-1} \hat{B}_t^{t-1}) \hat{A}_{t-1} = \hat{B}_t^{t-1} \hat{A}_{t-1} \quad (4)$$

The process iterates with the acquisition of fresh data at a new frame, and the current estimate takes the role of the previous estimate.

3 Solution Approach: Bounded Hough Transform

To estimate the value of the perturbation \hat{B}_t^{t-1} , we apply a variation of the General Hough Transform (GHT). The

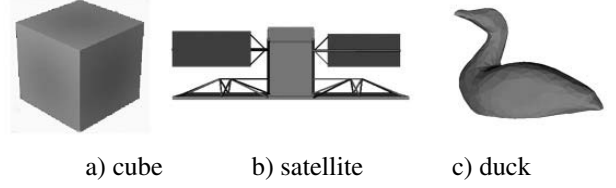


Figure 1: Test Objects

Hough Transform functions by mapping sets of image entities (e.g., points) from image space into a parameter space. The cardinality of the image space sets and the dimensionality of parameter space depend upon the characteristics of both the entities and the features under estimation. For example, if the entities comprise edge points within a 2D image, then line extraction requires individual points to be mapped into a sinusoidal curve in a 2D parameter space. For circle extraction with the same image data, sets of 3 points are mapped into a 3D parameter space. Intersections of distinct mappings are accumulated in discretized parameter space bins, and peaks therein provide evidence for the existence of features, which are then estimated by the inverse mappings of the peak bin values.

The GHT was introduced by Ballard [10] and allows the extraction of arbitrary nonparametric features. A limitation of the GHT is that the size of the parameter space grows exponentially with dimensionality d . A straightforward implementation becomes impractically inefficient in both space and time when Θ^d is too large, the typical limit being $d \leq 3$. In our case, we are dealing with rigid transformations of 3D objects, so that $d = 6$, which for a standard GHT would be prohibitively expensive.

As a remedy, the bounds on the magnitude of B_t^{t-1} can be exploited to reduce the size of the parameters space. At each frame the relative transformation B_t^{t-1} will lie within a small neighborhood of the complete pose space Θ^d , centered around the canonical pose. It is only necessary to search in this small neighborhood to estimate the current

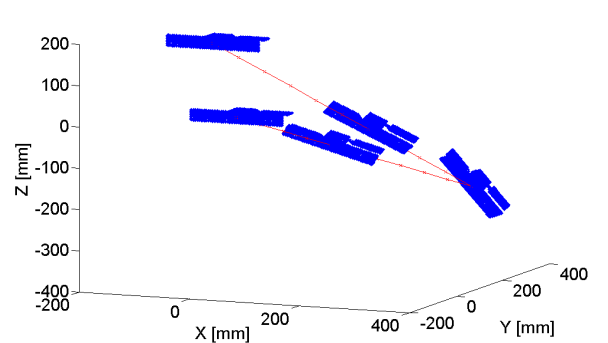


Figure 2: Satellite Trajectory

transformation.

The algorithm operates in a discretized mapping of 3D space, wherein each continuous point \vec{p}_i maps injectively to a distinct voxel $\vec{v}_i \in \mathbf{V}^3$. The resolution ρ of voxel space \mathbf{V}^3 (i.e., the size of each voxel) is related to the maximum inter-frame motion bound $\|\Theta^d\|$, as well as the maximum sensor noise ϵ . All of these quantities can be expressed as Euclidean distances, and ρ is set conservatively to be no less than the Nyquist limit, which is twice the sum of these quantities:

$$\rho/2 \geq \max\|\Theta^d\| + \max\|\epsilon\|. \quad (5)$$

In our case individual \vec{p}_i are mapped into a surface manifold in a d -dimensional parameter space, where d is the dimension of Θ^d . Thus, if Θ^d describes either purely translational or purely rotational motions, then $d=3$, whereas for general rigid transformations, $d=6$. Each axis of the parameter space respectively represents a basis vector of Θ^d , so that the parameter space is a pose space. Each \vec{p}_i maps to the set of all transformations that could give rise to \vec{p}_i , i.e., the set $\{B_i\}_1^{N_B}$ of all N_B (discrete) relative transformations that, when applied to the canonical pose, would cause some surface of \mathbf{M} to intersect with \vec{p}_i . Whereas in general the cardinality N_B of this set could be large, because Θ^d is bounded and effectively discrete, N_B becomes quite small and manageable. Indeed, as is described following, B_i^{t-1} can be estimated by considering only $3^d = 729$ bins.

3.1 Preprocessing

In preprocessing a set $\{V_i\}_{i=1}^{N_B}$ of *exemplars* is generated, one for each discrete transformation, denoted as \bar{B}_i . Each V_i is a template of the voxel occupancy that results within \mathbf{V}^3 when \mathbf{M} is transformed by \bar{B}_i . The exemplars are generated by calculating the voxels that intersect with the surface of $\bar{B}_i\mathbf{M}$, and each V_i is stored as a 3D binary array. The complete surface model of \mathbf{M} is used when generating the V_i , without any consideration for self-occlusions that can result from a specific sensor vantage.

3.2 Runtime

During runtime at frame t , \mathbf{P}_t is transformed into the object frame \mathcal{M} by applying the inverse of the previous frame's pose estimate (Eq.2). The pose B_t^{t-1} can then be estimated as one of the \bar{B}_i , and the intersection of ${}^{\mathcal{M}}\mathbf{P}_t$ with \mathbf{V}^3 will correlate highest with the V_i corresponding to \bar{B}_i . The highest correlating V_i can be identified by casting votes in an accumulator space. For each voxel $\vec{v}_j \in \mathbf{V}^3$ that is *surface-valued* (i.e., intersects with a surface of ${}^{\mathcal{M}}\mathbf{P}_t$), we enumerate the exemplars in which \vec{v}_j is also surface-valued. This operation is efficient because the V_i have the same 3D array structure as \mathbf{V}^3 , so that the indices of each \vec{v}_j need only be calculated once. Also, the number N_B of V_i is small, so that the complete set can be stored in memory.

For each V_i for which \vec{v}_j is surface-valued, the identity i is incremented in parameter space, which is a discrete rep-

resentation of the neighborhood of Θ^d around the canonical pose. Each \vec{p}_k therefore votes for a surface manifold in Θ^d . Once all votes have been cast, the peak value i_{\max} in the parameter space signifies the best transformation estimate $\bar{B}_{i_{\max}}$, and $B_t^{t-1} \leftarrow \bar{B}_{i_{\max}}$. The voting procedure could alternately be substituted with a template-set matching scheme [11], whereby each V_i is correlated with \mathbf{V}^3 , and the highest correlation identifies the pose $\bar{B}_{i_{\max}}$.

3.3 Full Dimensionality

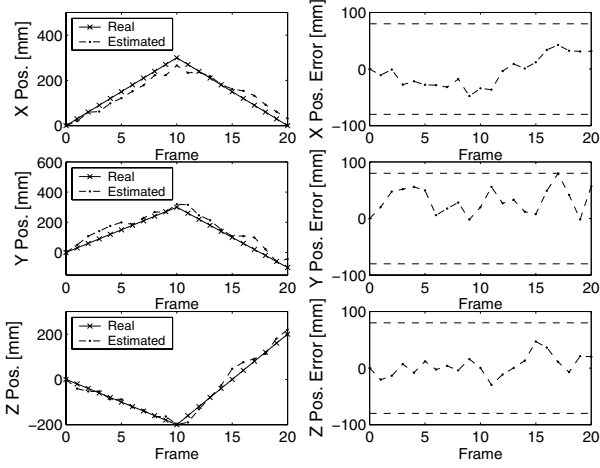
When Θ^d is the space of general rigid transformations, then $d=6$. In the most straightforward realization of the method, the transformation space is d -dimensional, with each axis of Θ^d representing one of 3 translations or 3 rotations. The set $\{B_i\}_1^{N_B}$ of discrete transformations enumerates all states adjacent to the canonical pose in the quantized Θ^d . There are therefore a total of $N_B = 3^6 = 729$ states, and a corresponding set $\{V_i\}_1^{N_B}$ of exemplars.

3.4 Dimensional Projection

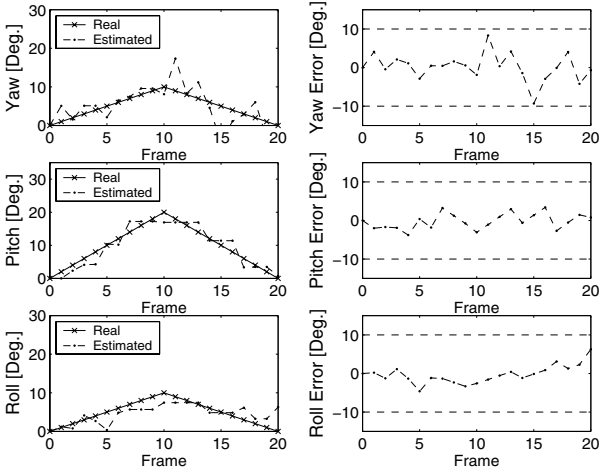
An alternative is to partition Θ^d into disjoint subspaces. The method can then be applied to each subspace sequentially, with the partial solution determined at each stage projected into the subsequent subspace by applying it to the current estimate \hat{B}_t^{t-1} .

One possibility is to solve the translational subspace first, followed by the rotational subspace. Each state in the first stage represents the projection of the 3 dimensional rotational subspace onto a coordinate in the translational subspace. Each element of the set $\{\bar{B}_i\}_1^{B_T}$ therefore represents the union of all possible adjacent rotations at a particular translation. An estimate \hat{T}_t^{t-1} of the translation is determined using the procedure outlined above and is applied to the inverse of the previous frame's estimate. The transformation $\hat{T}_t^{t-1}\hat{A}_{t-1}^{-1}$ therefore maps the data into a state that is offset from the canonical pose by a pure rotational increment \hat{R}_t^{t-1} . This rotation is next resolved by applying the method using the pure rotational exemplars, and the complete transformation is composed as $\hat{B}_t^{t-1} = \hat{R}_t^{t-1}\hat{T}_t^{t-1}$.

The main benefit of Dimensional Projection is improved runtime efficiency. Each of the two subspaces is 3-dimensional, and their combination yields $2 \times N_B = 2 \times 3^3 = 54$ states, which is less than $1/10^{\text{th}}$ of the number of states needed for the Full Dimensional case. This improved efficiency comes at the cost of a potential loss of reliability. The projection of the complete rotational subspace onto each coordinate of the translational subspace results in a merging of the peaks of the voting space. The merging of two peaks may cause a shift of the detected peak if their contributions cannot be distinguished in the projection. This is of particular concern when the data is very sparse, or when the object has a very compact symmetrical shape. Despite this possibility, this occurrence may in practise be quite rare, as is demonstrated by our experimentation in Sec. 4.



a) translational dimensions



b) rotational dimensions

Figure 3: Tracking Accuracy, Full Dimensionality

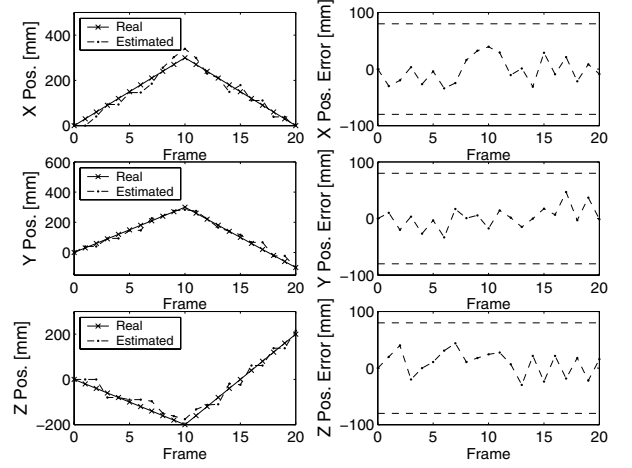
It is possible to partition the transformation space differently, yielding even greater efficiencies. For example, 5 of the 6 dimensions could be projected onto the remaining dimension. Once this dimension is determined, 4 of the 5 remaining unresolved dimensions could then be projected to the other remaining unresolved dimension, etc. This scheme would result in the minimal number of states of all possible projections, with only $N_B = 6 \times 3$ possible states. The likelihood of ambiguous peaks, however, increases accordingly.

4 Experimental Results

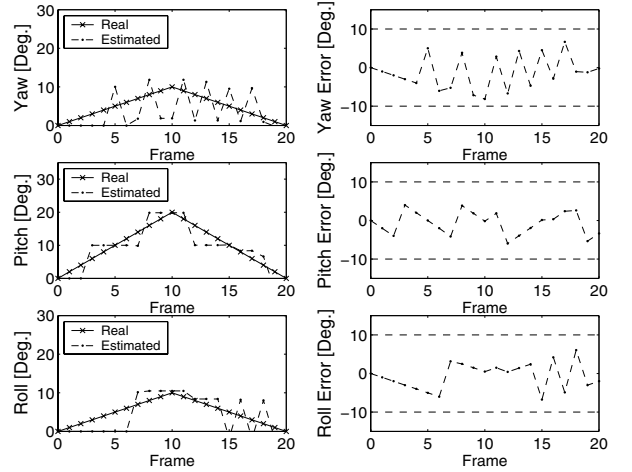
We have implemented both the Full Dimensional and Dimensional Projection methods, and have tested them on both simulated and real data sets.

4.1 Simulated Data

The method was tested using simulated range data sequences for 3 objects: a cube, a satellite, and a duck, illustrated in Fig.1. For each object, a surface model was



a) translational dimensions



b) rotational dimensions

Figure 4: Tracking Accuracy, Dimensional Projection

produced in the canonical pose, and the exemplar sets were generated in preprocessing. A set of simulated range data sequences were then generated by transforming each model through a motion trajectory, and collecting 20 frames at regular intervals along the trajectory. The transformations between successive frames contained motion components in all 6 directions, and were limited by known bounds. Each image sequence started with the model in its known canonical position, so that $A_0 = \mathbf{I}$. For each frame, the pose of the model was estimated and the estimate was compared against the known true pose values.

A representative set of frames from a test sequence of the satellite is illustrated in Fig.2. The object starts in its canonical pose at $t=0$, and is transformed for the first 10 frames with an inter-frame translational and rotational velocity of $\{30, 30, -20\}$ mm/frame and $\{1, 2, 1\}$ degs./frame, respectively. At frame 11, the velocity instantaneously changes to $\{-30, -40, 40\}$ mm/frame and $\{-1, -2, -1\}$ degs./frame,

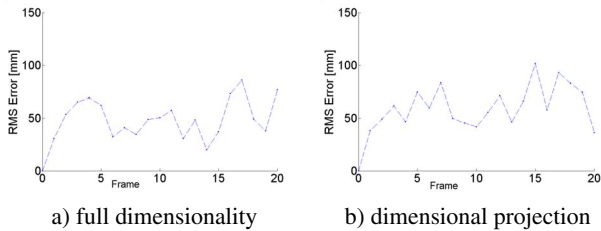


Figure 5: Tracking Accuracy, RMS Error

and the motion continues with this velocity to frame 20.

Full Dimensionality

For the Full Dimensional method, $N_B = 3^6 = 729$, and the resolution of \mathbf{V}^3 was set at $\rho = 80$ mm. For the objects under consideration, this value of ρ was greater than the minimum bound identified by Eq.5.

The values of the true and estimated pose parameters at each frame for the satellite object are plotted in Fig.3, with the true and estimated values of each of the 6 distinct dimensions plotted on separate graphs. The errors, i.e. the differences between the true and estimated values, are plotted in adjacent graphs, with the error bounds indicated by dashed lines. It can be seen that the estimates closely track the true values in each dimension, even when the trajectory abruptly changes direction at frame 11. The magnitude of the errors are less than their respective inter-frame bounds of ± 80 mm and $\pm 10^\circ$. This indicates that the method succeeded at tracking the pose to within the accuracy bounds of the discrete transformation space.

Dimensional Projection

The above experiment was repeated for the same motion trajectory using the Dimensional Projection method. The resulting estimates and errors for each dimension are plotted in Fig.4, which show the errors to be within the error bounds.

Fig.5. presents another plot of the tracking errors for both the Full Dimensional and Dimensional Projection methods. At each frame, the position of each range data point at the estimated pose value is compared against its corresponding point at the known true pose. The average of square of the Euclidean distance of each such point, the *rms* error, is calculated at each frame.

Robustness

In the preceding tests, the simulated range data was generated by sampling the surface of the model in a given pose from the sensor vantage point. Self-occluded data were filtered out, so that the images were $2\frac{1}{2}$ -D, as are typically acquired by conventional range sensors. Each datum did,

however, measure an exact error-free sample of the model surface in its given pose. An attractive aspect of the BHT is that it accumulates evidence from each point independently, and therefore has the potential of being effective for noisy and sparse data sets.

To evaluate noise-robustness, the data quality was degraded with simulated Gaussian noise and outliers. Random additive Gaussian noise, which simulates measurement error, was added to each image point. The Gaussian noise was zero mean, and the standard deviation varied between 0% and 200% of ρ . For each noise level, tracking was executed for the same motion trajectory. The *rms* error was calculated for each trial and plotted in Fig.6a). For each noise level the min, max, and average *rms* error for all frames is displayed. It can be seen that the accuracy degrades fairly gracefully, with the *rms* error doubling at about the 75% noise level. Once the noise level exceeds 100%, Eq.5 is violated and the tracking can no longer be guaranteed.

In a second test, spurious data points (*outliers*) were randomly added to each data set. The outliers were generated to lie within the bounding box of each data set, with the number of outliers varying from 0% to 200% of the number of data points N_t in each frame t . The min, max, and average *rms* for all frames is plotted in Fig.6b) with respect to the percentage of outliers. The method demonstrates a high level of robustness to outliers, as the average *rms* value is barely affected at the 200% level, where there are twice as many outliers as true data points. We further tested the effects of the data sparseness by randomly removing a certain percentage of the data points at each frame, and re-executing the tracking on each sparse sequence. The results of this tests, plotted in Fig.6c), indicate a significant ability to function correctly for sparse data. Indeed, visual inspection of the results confirmed that the tracking algorithm worked correctly for data sets with only ~ 20 points per frame.

In addition to the satellite object tests, we ran similar tests on the cube and the duck objects. In each test, the estimated transformation values tracked the true values to within the inter-frame motion bounds. The method does not depend upon any extractable features or surface regularities, and the satellite object has a more complicated shape than the cube, although it is still mostly polyhedral. Alternately, the duck is a freeform object with no planar surfaces, except for the flat underside which was not acquired in our sequences. The duck therefore represents the most general of all rigid tracking scenarios.

4.2 Real Data

Lidar Satellite Data

The algorithm was tested with real data collected from a time-of-flight (i.e., *lidar*) range sensor. A $1/5^{th}$ scale model

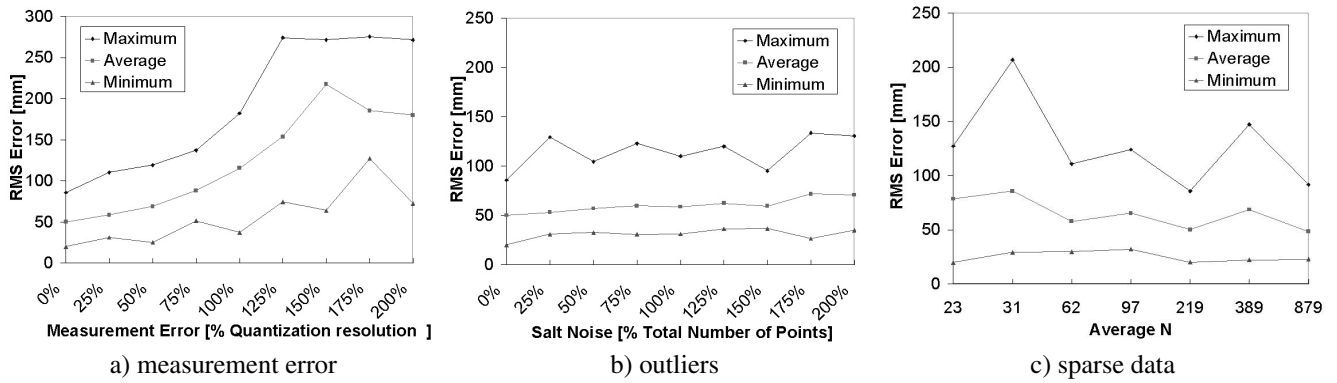


Figure 6: Effects of Data Degradation



Figure 7: Robot Mounted Satellite Model

of a Radarsat satellite was mounted on a 6 dof articulated robotic manipulator as illustrated in Fig.7. Starting from its canonical position, the robot positioned the satellite through a motion trajectory, and 40 image frames were acquired at regular time intervals. The robot joint encoder readings and inverse kinematic solution provided ground truth measurements of the pose of the satellite at each frame. Unfortunately, the poor accuracy of the robot calibration made it difficult to identify the translational dimensions of the motion to a meaningful accuracy. The rotational measurements, however, were based upon a single joint reading, and were quite accurate and therefore useful for evaluating tracking accuracy.

As the satellite model was relatively large, and due to limitations of the reachable workspace of the robot, it was difficult to obtain meaningful trajectories containing all 6 dimensions of motion. The motion trajectory therefore contained 1 translational (z) and 2 rotational (yaw and roll) dofs. Over 40 frames, the yaw oscillated between 0° and -100° by increments of $\pm 10^\circ$ /frame, and the roll incremented by 10° at 3 frames. The estimated trajectory is illustrated in Fig.8. Each image contained 50,000 points, the

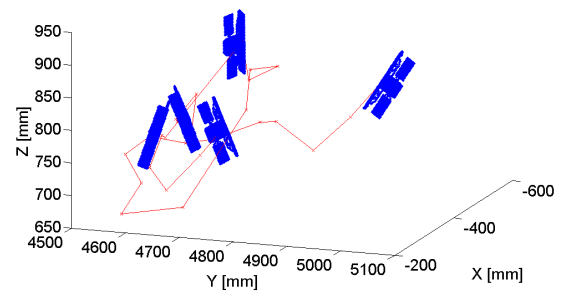


Figure 8: Satellite Trajectory, Lidar Data

majority of which fell on the surface of the robot and the background and were therefore outliers. To demonstrate the effectiveness of the method at tracking in sparse as well as cluttered data, a set of 600 points were randomly selected from each image, and only these points were used.

The 3 tracked rotational dimensions and tracking errors for both the Full Dimensional and Dimensional Projection methods are illustrated in Fig.10. The estimated values follow the ground truth values closely, particularly for the Full Dimensional case. Starting at frame 20, there were 3 frames where the error exceeded the bound in the yaw dimension. An examination of the data showed that the satellite was in a particularly difficult pose at these frames, where most of the points on the solar panel and bottom of the satellites are dropouts. In all cases where the bound was exceeded under both methods, it was subsequently recovered within a few frames.

The 3 tracked translational dimensions for both methods are illustrated in Fig.9. Although there was no ground truth data to compare against, the estimates can be seen to closely coincide in each dimension. The Full Dimensional method produced smoother estimates, which indicates a greater accuracy.

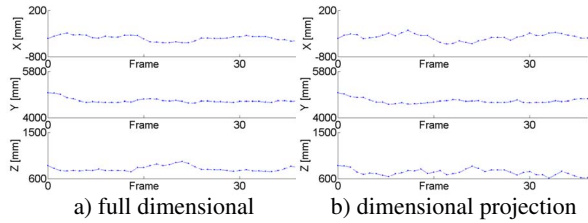


Figure 9: Satellite Translations, Lidar Data

Stereovision Cube Data

In another experiment, a range image sequence of a cube was captured using a stereovision system. The cube was manually repositioned through a motion sequence that included all 6 dimensions, and 100 data frames were captured at $\sim 10\text{Hz}$. One of the image pairs and the associated disparity map at frames 1, 40, and 100 are illustrated in Fig.11. The estimated motion trajectory calculated using the Full Dimensional method is illustrated in Fig.12. As the cube was hand-held, there was no ground truth measurement against which to evaluate the accuracies of the estimates. Qualitatively, the tracking was judged to work well, maintaining a lock on the cube throughout a wide range of motions, speeds, and arbitrary changes of direction.

5 Discussion

The BHT is a more efficient alternative to ICP for tracking in sparse range data. At frame t , each of the N_t data points votes by checking membership in each of the N_B exemplars, for a total of $N_B \times N_t$ operations. Peak detection loops through the discrete transformation space requiring another N_B comparisons. The value of N_B is constant and small, with $N_B = 729$ for the Full Dimensional case, and $N_B = 54$ for Dimensional Projection. The complexity expression for the runtime algorithm is therefore only $O(N_t)$ per frame, with small constants. In contrast, ICP requires a nearest neighbor computation for each point at each iteration, at an expense of $O(\log N_t)$ per point. For k iterations per frame, the complexity expression of ICP is therefore $O(k N_t \log N_t)$ per frame.

The BHT requires that an estimate of the motion bound exists so that a correct value of ρ can be selected. The ICP also requires a motion bound, so that the object lies within the minimum potential well space across adjacent frames. In practise, it is rare to encounter a tracking scenario in which an explicit motion bound does not exist, due to physical and system constraints.

Whereas the BHT resolves the pose only to within a bounded precision, ICP can continue to iterate until a desired precision is met, limited only by the measurement fidelity. The BHT is essentially trading off reduced precision for increased efficiency. For certain applications, such as robotic grasping, the location of the tracked object is only

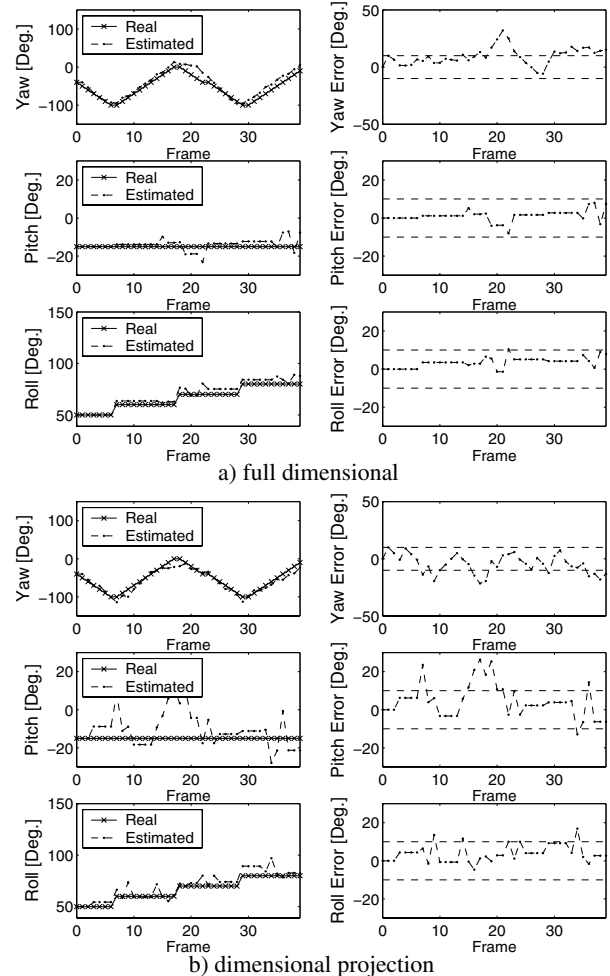


Figure 10: Satellite Rotations, Lidar Data

required to a high precision at the completion of the tracking sequence, so long as limited precision tracking is maintained throughout the sequence. In robotic space operations in particular, computational resources are at a premium, and the tradeoff between precision and computational efficiency becomes attractive [12].

Lidar data is relatively expensive to collect, and the low frame rates can result in motion skew within a frame. If this skew is the dominant noise component, then more frequent estimation using fewer points may actually improve the accuracy of BHT as compared to ICP. Similarly, the computational expense of the Dimensional Projection method is over 10 times ($729/54$) less than that of Full Dimensionality. If the rate limiting factor is the processing rather than the acquisition, then Dimensional Projection should be provided with data at rates 10 higher than Full Dimensionality, thereby improving tracking accuracy.

Currently, the BHT does not include any predictive techniques. While predictive techniques such as Kalman filtering do not respond well to arbitrary motions, they may im-

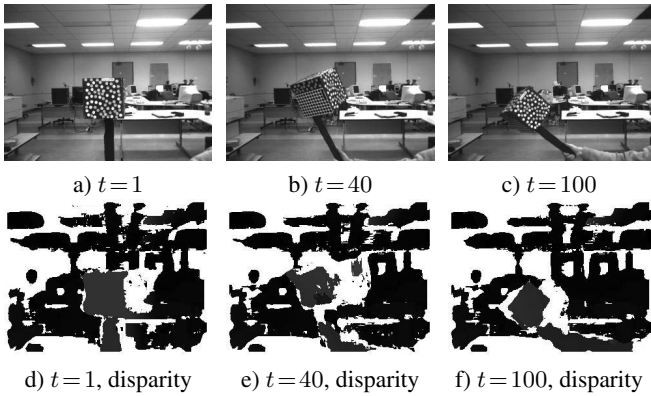


Figure 11: Stereovision Test Images

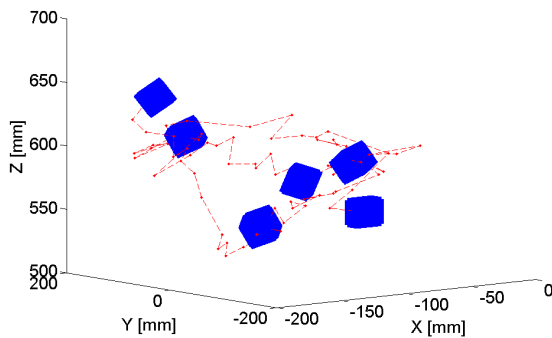


Figure 12: Cube Trajectory

prove precision [13].

6 Conclusions and Future Work

We have presented a novel formulation of the Hough Transform to track objects in a range data sequence. The BHT effectively trades off localization precision for computational efficiency. The main idea is to exploit the coherency between frames that results from the relationship between the known bounds on the object's velocity and the sensor frame rate. The inter-frame motion bounds allows the transformation space to be reduced to a small size.

The BHT is both general and efficient. It works with any shape of object, including freeform surfaces, and executes in $O(N_t)$. Experimental tests have been performed on both simulated and real data, and verify the correctness of the method. An attractive aspect of the technique is that it functions well in very sparse data, possibly comprising only tens of points per frame. It has also demonstrated a high degree of robustness to measurement error and outliers.

In the future, we wish to compare the performance of efficient implementations of the BHT and ICP algorithms. We also plan to implement a hierarchical version [14] that will accommodate an increase in precision. It may also be possible to create a hybrid method that starts with the BHT and then switches to the ICP, the aim being to benefit from the

increased efficiency of the BHT as well as the high precision of the ICP. The benefits of predictive techniques, such as Kalman and particle filtering, will also be investigated.

Acknowledgements

The authors gratefully acknowledge the financial support of MDRobotics and NSERC.

References

- [1] Paul J. Besl and Neil D. McKay. A method for registration of 3d shapes. *IEEE Trans. PAMI*, 14(2):239–256, February 1992.
- [2] David A. Simon, Martial Hebert, and Takeo Kanade. Real-time 3-D pose estimation using a high-speed range sensor. In *IEEE Intl. Conf. Robotics and Automation*, pages 2235–2241, San Diego, California, May 8-13 1994.
- [3] P. Jasiobedzki, J. Talbot, and M. Abraham. Fast 3d pose estimation for on-orbit robotics. In *ISR 2000: International Symposium on Robotics*, Montreal, Canada, May 14-17 2000.
- [4] Francois Blais, Michel Picard, and Guy Godin. Recursive model optimization using icp and free moving 3d data acquisition. In *4th Intl. Conf. 3-D Im. Mod.*, pages 251–258, Oct. 2003.
- [5] W.E.L. Grimson and D.P. Huttenlocher. On the sensitivity of the hough transform for object recognition. *IEEE Trans. PAMI*, 12(3):255–274, March 1990.
- [6] J. Illingworth and J. Kittler. A survey of the hough transform. *CGVIP*, 44:87–116, 1988.
- [7] J.M. Nash, J.N. Carter, and M.S. Dixon. Dynamic feature extraction via the velocity hough transform. *Pat. Rec. Ltrs.*, 18(10):1035–1047, 1997.
- [8] Pelopidas Lappas, John N. Carter, and Robert I. Damper. Object tracking via the dynamic velocity hough transform. In *Intl. Conf. Im. Proc.*, 2001.
- [9] Luca Iocchi, Domenico Mastrantuono, and Daniele Nardi. A probabilistic approach to hough localization. In *Proc. IEEE Intl. Conf. Rob. Aut.*, pages 4250–4255, May 21-26 2001.
- [10] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pat. Rec.*, 13(2):111–122, 1981.
- [11] Michael Greenspan. Geometric probing of dense range data. *IEEE Trans. PAMI*, 3(24):495–508, March 2002.
- [12] P. Jasiobedzki, M. Greenspan, and G. Roth. Pose determination and tracking for autonomous satellite capture. In *iSairas 2001: 6th Intl. Symp. AI, Rob., Aut. in Space*, Montreal, Quebec, Canada, June 18-22 2001.
- [13] Chengping Xu and Sergio A. Velastin. The mahalanobis distance hough transform with extended kalman filter refinement. In *Intl. Sym. Cir. Sys.*, pages 5–8, 1994.
- [14] M. Atiqzaman. Multiresolution hough transform - an efficient method of detecting patterns in images. *IEEE Trans. PAMI*, 14(11):1090–1095, 1992.