EAI Endorsed Transactions
on Context-aware Systems and Applications
Research Article ICST.ORG

Fourier Volume Registration based Dense 3D Mapping

Luke Lincoln[1,*], Ruben Gonzalez[1]

[1]Institute for Integrated and Intelligent Systems (IIS), Griffith University, Gold Coast, Australia

Abstract

In image processing phase correlation has been shown to outperform feature matching in several contexts. In this paper, a novel volume registration technique is proposed for solving the simultaneous localization and mapping (SLAM) problem. Unlike existing methods which rely on iterative feature matching, the proposed method utilises 3D phase correlation. This method provides high noise robustness, even in the presence of moving objects within the scene which are problematic for SLAM systems. Furthermore, a novel projection method is proposed which performs Fourier based volume registration 3 times faster. Quantitative and qualitative experimental results are presented, evaluating the proposed method's the noise sensitivity, performance, reconstruction quality and robustness in the context of moving objects.

Received on 16 May 2016; accepted on 27 June 2016; published on 12 September 2016

Keywords: SLAM, volume registration, phase correlation, Fourier transform

doi:10.4108/eai.12-9-2016.151675

1. Introduction

Simultaneous localization and mapping (SLAM) has many applications in robotics, architecture and engineering, business and science. Its objective is to produce a map (2D birds-eye-view, or 3D) of an environment using image and other sensory data. This is typically performed by computing local features, iteratively matching them across frames and solving for the camera pose and location. This feature matching approach is dependent on finding a sufficient number of matches. When this is true the approach is able to cope with local matching disparities by treating them as outliers. This technique is not robust when features are not stable or when feature confusion occurs.

To alleviate these shortcomings we propose using 3D phase correlation based volume registration to solve the SLAM problem. Given two volumes, volume registration algorithms find a geometrical transformation in which to align the data within the volumes. Fourier based registration is known to be fast, robust to noisy data and scales naturally to parallel processing [8]. It has also been shown to be able to outperform local feature based methods in 2D image processing [10]. This approach is capable of real-time SLAM if used in a parallel programming context. However, this approach is still computationally complex for most practical volume sizes on multiprogramming systems.

In the context of SLAM, scale is often not required in registering volumes. Therefore we propose a novel technique to speed up Fourier based volume registration for estimating translation and rotation parameters. This is achieved by applying a novel projection transform called a spherical-map transform. Along with orthogonal projection, this method allows 2D phase correlation to be used in place of the 3D counterpart. The result is a 3D phase correlation method which is 3 times faster than the original. Applied to SLAM this performance increase allows larger volumes to be processed for the same complexity or correspondingly greater accuracy for a given amount of computation time.

2. Previous Work

[*] Luke Lincoln. Email: luke.lincoln@griffithuni.edu.au

2.1. Monocular Camera Feature Based Systems

Monocular Feature based SLAM systems use feature matches to estimate camera pose and location changes across frames [4]. Variations of this method use different features including: corners and lines [14], image patches [25] and exemplar feature matching [3]. SIFT features are used most often in SLAM [1, 7, 13, 22], in addition FAST features have been explored [16–19]. Beall et al [1] made use of both SIFT and SURF features in their underwater SLAM system. Real-time monocular SLAM systems based on this approach have also been proposed [3, 22]. RANSAC is often used in monocular SLAM [7, 16–18, 24] to remove outliers which cause incorrect camera parameter estimates. Bundle adjustment is also used as an additional step to refine camera parameter estimation [7].

2.2. Stereo Camera Feature Based Systems

Stereo based SLAM systems also use features to estimate camera parameters. However, stereo based systems are capable of generating dense depth data more easily using stereo algorithms. Miro et al [20] proposed a stereo based method which uses SIFT and the extended Kalman filter. The method by Van Gool et al [23] works with un-calibrated stereo pairs. It uses Harris corner features and a multi-view stereo algorithm. Sim et al [26] and Gil et al [9] both presented stereo based SLAM systems which use SIFT.

2.3. RGB-D Sensor Feature Based Systems

RGB-D SLAM systems use both depth and image data and are capable of generating dense 3D reconstructions. Many of these methods rely on feature matching techniques [5, 6, 11]. RANSAC is often used to filter outliers for the estimation of camera parameters[5, 6, 11]. Another method which has also been used extensively in the area is Iterative Closest Point (ICP) [2, 6, 11, 12, 21, 27]. ICP iteratively registers point cloud data, and is used to refine camera parameter estimates. A method named KinectFusion was proposed by Newcombe et al [21] which uses RANSAC and a GPU implementation of IPC. Whelan et al [29] extended this method allowing it to map larger areas using Fast Odometry From Vision (FOVIS) over ICP. Bylow et al [2] improved the ICP approach by registering data using a signed distance function.

2.4. Non-Feature Based Methods

Several RGB-D SLAM systems are also non-feature based [12, 15, 28]. Weikersdorfer et al [28] presented a novel sensor system named D-eDVS along with an event based SLAM algorithm. The D-eDVS sensor combines depth and event driven contrast detection. Rather than using features, it uses all detected data for registration. Kerl et al [15] proposed a dense RGB-D SLAM system which uses a probabilistic camera parameter estimation procedure. It uses the entire image rather than features to perform SLAM.

2.5. Summary

As is evident from the current literature, SLAM typically relies on feature matching and RANSAC. However, these approaches fail when there are too few features, when feature confusion occurs or, when features are non-stationary due to object motion. As the extent of random feature displacement becomes more global the effectiveness of these approaches diminishes. Feature matching also dominates in image registration. However, Fourier based methods have been shown to work well under larger rotations and scales [10] whilst being closed form, insensitive to object motion and scaling naturally to GPU implementations. Accordingly, we propose a novel, closed form Fourier based SLAM method.

3. Method

The proposed SLAM method consists of various steps. First each frame f_i that is captured, consisting of a colour and depth image pair is projected into 3D space, forming colour point cloud $points_i$ and re-sampled into a volume V_i. Then, the transform parameters between pairs of volumes V_i and V_{i+1} are estimated using $VolumeRegister_{\theta\varphi t_x t_y t_z}$ shortened to $VR_{\theta\varphi t_x t_y t_z}$. These parameters are used to update transformation matrix M. The points corresponding to f_2 ($points_1$) are then transformed using the updated M matrix and added to the cumulative $PointCloud$ database. Two lists, $Cameras$ and $Poses$, are also updated to track camera pose and location per frame. This basic procedure is given in listings 1 and elaborated upon in subsequent subsections.

3.1. Sensor Input

The input to our method is a color and depth image pair, $f(u, v)$ and $g(u, v)$ obtained using an Asus Xtion PRO LIVE sensor at a resolution of 640 × 480. Each pixel is projected into 3D space using $X_{u,v} = \frac{(u-c_x)Z_{u,v}}{f}$, $Y_{u,v} = \frac{(v-c_y)Z_{u,v}}{f}$ and $Z_{u,v} = g(u, v)$. Here, $[c_x c_y]^T$ represent the center of the image whilst f represents the focal length, defined as 525.0. The point clouds generated by projecting these images are then quantized into image volumes. Results reported in this paper were obtained using volumes of 384^3 voxels in size.

3.2. Volume Registration

Figure 1 shows a functional block diagram of our method. The input data are two 3D volumes

Listing 1. Phase Correlation Based SLAM Algorithm

```
$f_1$ = ReadFrame();
$PointCloud$ = project($f_1$);
$M$ = IdentityMatrix();
$Camera$ = $[0,0,0]^T$;
$Pose$ = $[0,0,1]^T$;
$Cameras$ = $[Camera]$, $Poses$ = $[Pose]$;
while(more frames){
        $f_2$ = ReadFrame();
        $points_1$ = project($f_2$);
        $points_2$ = project($f_1$);
        $V_1$ = ResampleVolume($points_1$);
        $V_2$ = ResampleVolume($points_2$);
        $(\theta, \varphi, t_x, t_y, t_z) = VR_{\theta\varphi t_x t_y t_z}(V_1, V_2)$;
        $M = M \times$TransformMat($(\theta, \varphi, t_x, t_y, t_z)$);
        $points_1$ = Transform($points_1$, $M$);
        $PointCloud$ = $PointCloud \cup points_1$;
        $Camera$ = $M^{-1} \times Camera$;
        $Pose$ = $M^{-1} \times Pose$;
        $Cameras.add(Camera)$;
        $Poses.add\left(\frac{Pose-Camera}{|Pose-Camera|}\right)$;
        $f_1$ = $f_2$;
}
```

($Volume_1$ and $Volume_2$) and the output is the transformation matrix required to register the two volumes. The volumes are first Hanning windowed. Next, a translation independent representation is obtained for each by taking the magnitude of their 3D FFTs. Then a log function is applied to the resulting magnitude values, improving scale and rotation estimation [10]. Following a log-spherical transformation, 3D phase correlation is performed to find the global rotation and scale relationship between $Volume_1$ and $Volume_2$. $Volume_1$ is then inversely transformed by the rotation and scale parameters, leaving only the translation to be resolved. This is found by applying phase correlation again between the transformed $Volume_1$ and $Volume_2$.

3.3. Phase Correlation

Given a volume V_1 and a spatially shifted version of it V_2, the offset can be recovered using $PhaseCorrelation$ (Eq. 1). This function takes two volumes as input and returns the translation between them.

$$(x, y, z) = PhaseCorrelation(V_m, V_n) \tag{1}$$

The $PhaseCorrelation$ function first applies 3D FFTs to volumes, V_1 and V_2, converting them into the frequency domain, i.e. $F_{1_{x,y,z}} = FFT(V_1)$ and $F_{2_{x,y,z}} = FFT(V_2)$. Taking the normalised cross power spectrum using Eq. 2 completes the Phase correlation function.

$$F_{3_{x,y,z}} = \frac{F_{1_{x,y,z}} \circ F^*_{2_{x,y,z}}}{|F_{1_{x,y,z}} \circ F^*_{2_{x,y,z}}|} \tag{2}$$

Here, $\circ$ is an element-wise multiplication and $|x|$ is the magnitude function. Taking the inverse FFT of F_3, gives the phase correlation volume V_3 ($V_3 = FFT^{-1}(F_3)$). The location of the peak value in V_3, (x_1, y_1, z_1) gives the shift between the V_1 and V_2. The phase correlation volume is typically noisy making the peak difficult to locate.

3.4. Recovering Scale, Rotation and Translation Parameters

If V_1 and V_2 are instead rotated and scaled versions of the same volume, such that they are related by some translation (t_x, t_y, t_z), y-axis rotation θ, and scale φ. Further action is required to recover translation parameters. The first step, given two volumes V_1 and V_2 of size N^3 is to apply a Hanning windowing function (Eq. 3).

$$HW_{x,y,z} = \frac{1}{2}\left(1 - cos\left(\frac{2\pi\left(\sqrt{\left(\frac{N}{2}\right)^3} - \sqrt{\left(x-\frac{N}{2}\right)^2 + \left(y-\frac{N}{2}\right)^2 + \left(z-\frac{N}{2}\right)^2}\right)}{2\sqrt{\left(\frac{N}{2}\right)^3} - 1}\right)\right) \tag{3}$$

The rotation and scale factors are recovered first using a translation independent representation of the volumes using the Fourier shift theory. For this, the magnitude of the FFT of the volumes is taken, $M_1 = |FFT(V_1)|$, $M_2 = |FFT(V_2)|$. The zero-frequency of both M_1 and M_2 is shifted to the center of the volume and the log of the result is taken $M_1' = Log(M_1)$, $M_2' = Log(M_2)$ which reduces noise on the phase correlation volume. A log-spherical transform is then used to turn rotation and scaling into translation for both M_1' and M_2'. Eq. 4 shows the corresponding log-spherical space coordinate $(X_{log-spherical}, Y_{log-spherical}, Z_{log-spherical})$ for a given (x, y, z) euclidean space coordinate.

$$\begin{aligned}
X_{log-spherical} &= \frac{atan\left(\left(\frac{x-\frac{N}{2}}{\sqrt{x^2+y^2+z^2}}\right)\left(\frac{y-\frac{N}{2}}{\sqrt{x^2+y^2+z^2}}\right)^{-1}\right)N}{360} \\
Y_{log-spherical} &= \frac{acos\left(\frac{y}{\sqrt{x^2+y^2+z^2}}\right)N}{180} \\
Z_{log-spherical} &= \frac{log\left(\sqrt{x^2+y^2+z^2}\right)N}{log\left(\frac{N}{2.56}\right)}
\end{aligned} \tag{4}$$

The log-spherical transforms of M_1' and M_2' are then phase correlated to find the shift between them, $(x_{M'}, y_{M'}, z_{M'}) = PhaseCorrelation(M_1', M_2')$. The rotation θ and scale φ factors between V_1 and V_2 can

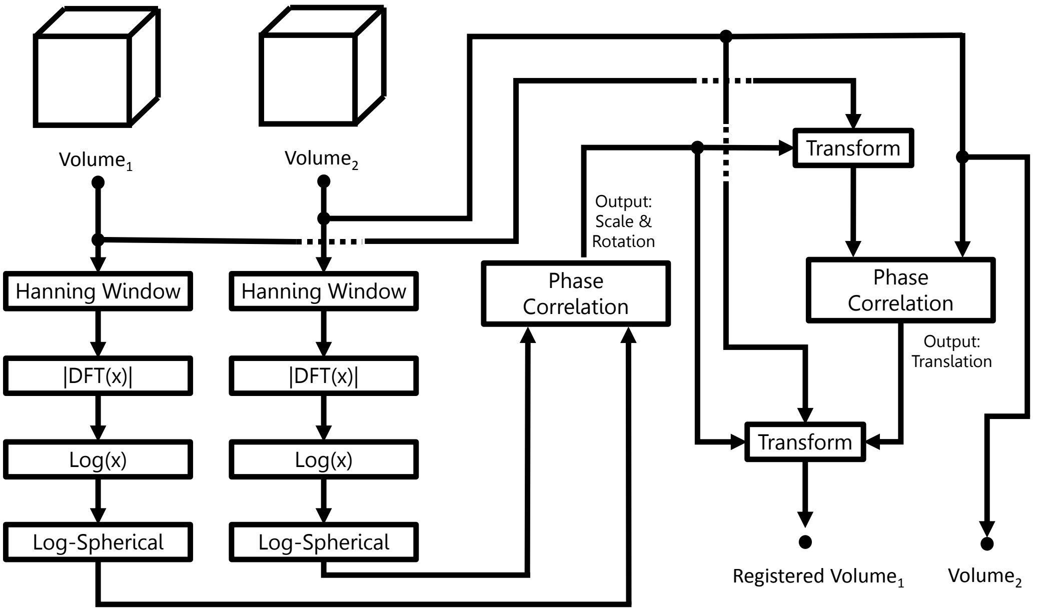


Figure 1. System Diagram for Registration Process

then be found from the shift parameters using Eq. 5 .

$$\theta = \frac{-360x_{M'}}{N}$$
$$\varphi = e^{-\left(2.56^{-1}N\right)z_{M'}N^{-1}} \tag{5}$$

Using θ and φ, V_1 can now be inverse transformed (using $(\frac{N}{2}, \frac{N}{2}, \frac{N}{2})$ as the origin). This aligns V_1 and V_2 with respect to scale and y-axis rotation. The translation parameters (t_x, t_y, t_z) can then be found using phase correlation as given in Eq. 6.

$$(t_x, t_y, t_z) = PhaseCorrelation(scale(rotate(V_1, \theta), \varphi), V_2) \tag{6}$$

The complete function to recover translation, rotation and scaling, combining equations 2-6 as is denoted in 1 is 7.

$$(\theta, \varphi, t_x, t_y, t_z) = PhaseCorrelation_{\theta\varphi t_x t_y t_z}(V_m, V_n) \tag{7}$$

4. Improving Performance

To reduce complexity we focus on areas which require the most computation time. In the earlier defined Fourier based reconstruction technique, this occurs in the two 3D phase correlations which need to be computed. We describe the method of reduction here; a block diagram for this technique is given in figure 2. We refer to this method as fast volume registration (FVR) in reference general volume registration (VR). The speedup begins by computing the 3D DFT of both input volumes and taking the magnitude of these. Rather than directly performing a 3D log-spherical transform and a 3D phase correlation operation on these volumes, we use a novel transform we call a spherical-map transform (details in 4.1).

This transform converts rotation into translation whilst simultaneously unfolding the 3D space down to 2-dimensions. After this, a 2D phase correlation that requires significantly less processing compared with the 3D counterpart is used to compute the rotation. Next, having obtained the rotation parameter, the rotation is eliminated from the transformation by rotating the first volume by this parameter. The two volumes are then passed through two orthogonal projection mapping functions. This also converts the volumes to 2D space. We use two transforms for both volumes, one projection along the x-axis, another along the z-axis. Once the x-axis projections of both volumes are complete, we can do another 2D phase correlation to give us the z-translation. The 2D phase correlation of the z-axis projections gives us the x and y axis translations separating the original volumes. The final output of this method gives the rotation and translational shifts between the input volumes. The projections add little complexity to the overall algorithm and since 2D phase correlation operations are used in place of 3D ones, much computation time is reduced.

4.1. Spherical-map transform

The spherical map transform both reduces the 3D volume to a 2D image, and any rotation about the y-axis becomes x-axis translation in the output image.

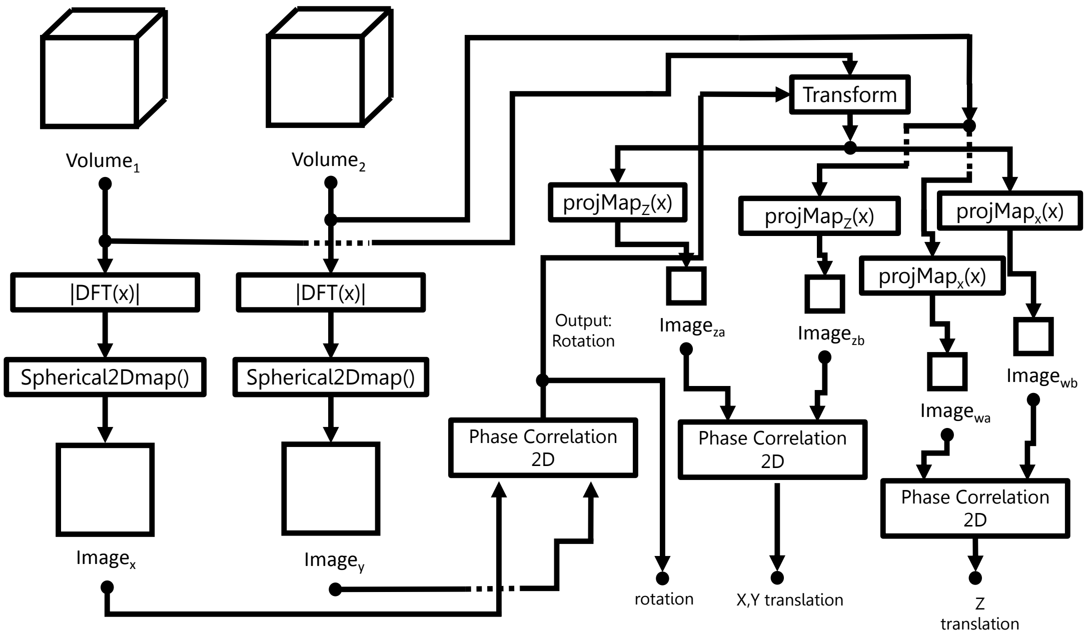


Figure 2. System Diagram for Fast Volume Registration

An example of the bunny model and the spherical-map transform of this model is given in figure 3, the mathematics are defined in equations 8 and 9. Given a coordinate in 2D Cartesian space x,y, we compute the ray $[Ray_x Ray_y Ray_z]^T$ from the volume center and sum up the voxel values along the ray (equation 9).

$$\begin{aligned} Ray_x(x,y) &= cos\left(\frac{360x}{N}\right) sin\left(\frac{180y}{N}\right) + \frac{N}{2} \\ Ray_y(y) &= cos\left(\frac{180y}{N}\right) + \frac{N}{2} \\ Ray_z(x,y) &= sin\left(\frac{360x}{N}\right) sin\left(\frac{180y}{N}\right) + \frac{N}{2} \end{aligned} \tag{8}$$

$$Im_{x,y} = \sum_{r=1}^{(2^{-1}N)^{1.5}} Vol(Ray_x(x,y)r, Ray_y(y)r, Ray_z(x,y)r) \tag{9}$$

This is process essentially sums up the values along a given ray defined by scaling spherical coordinates and adding up the values intersecting the ray. The resulting image, maps 3D y-axis rotation to 2D x-axis translation.

4.2. Projection-map transform

The projection map transform is similar to an orthogonal projection of the volume along some given axis. For the projection map transform, given an output image Im_a and an input volume Vol_a, each pixel in Im_a is defined mathematically as the summation of values along a particular axis given the image coordinates. The x-axis transform and the z-axis transform are defined in equations 10 and 11 respectively.

$$Im(z,y) = \sum_{x=0}^{N} Vol_a(x,y,z) \tag{10}$$

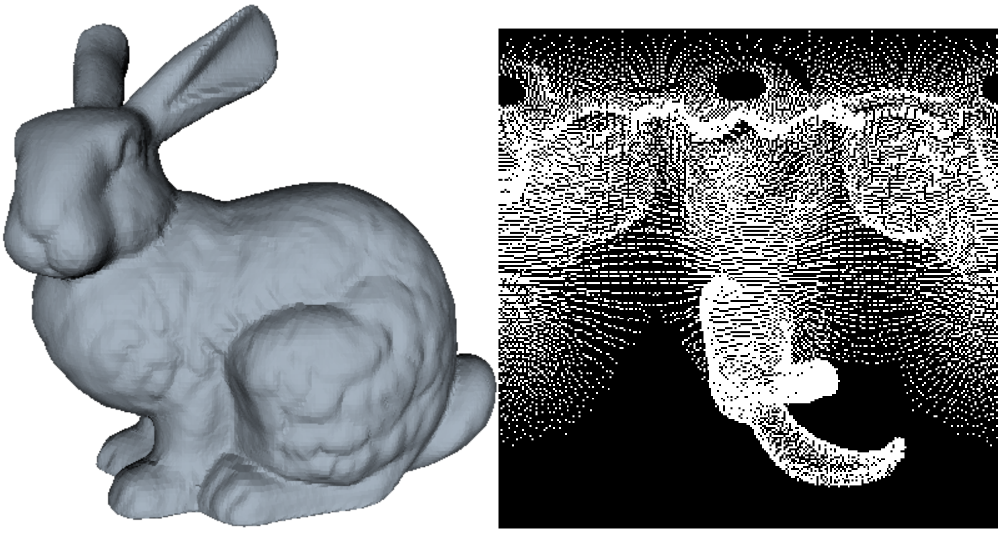
$$Im(x,y) = \sum_{z=0}^{N} Vol_a(x,y,z) \tag{11}$$

The process defined by equation 10 maps 3D z-axis translation to 2D x-axis translation, whilst equation 11 maps 3D x-axis and y-axis translation into 2D x-axis and y-axis translation.

5. Performance Analysis

5.1. 3D Phase Correlation Performance

To assess the performance of our method, the size of the volumes being registered is defined as N^3 whilst each frame is sampled at a resolution of $W \times H$. The projection process requires $12WH$ operations whilst re-sampling the point cloud requires $2WH$ operations. The Volume Registration process, $VolumeRegister\theta\varphi t_x t_y t_z(V_1, V_2)$ consists of 2 × Hanning windowing processes, 2 × 3D FFTs, 2 × volume-logs, 2 × log-spherical transforms, 2 × phase correlation processes and 1 × linear transformation and peak finding.



(a) original **(b)** transform

Figure 3. The Spherical Map Transform.

The Hanning windowing function requires 26 operations. The 3D FFT has complexity of $3N^3 \log N$, the log and log-spherical transform functions require 3 and 58 operations per voxel respectively. Multiplying two frequency spectra together and transforming a volume requires 15 and 30 operations per voxel respectively. Finding the peak value requires $2N^3$ operations. The complexity in terms of number of operations for the phase correlation process is given in Eq. 12 This process requires 2 × 3D FFTs, 1 × frequency spectra multiplication, and 1 × peak finding operation.

$$6N^3 \log N + 2N^3 + 15 \tag{12}$$

The total complexity can then be found by taking into account the projection and re-sampling totals as well as the total for $VolumeRegister\theta\varphi t_x t_y t_z(V_1, V_2)$. Tallying the number of operations for each process and multiplying them by number of times the process is performed gives us the number of operations as a function of W, H and N in Eq. 13.

$$6N^3 + 28WH + 18(N^3 \log N) + 230 \tag{13}$$

5.2. Analysis of Speed Improvement

To compare performance of the generic volume registration method with the speed up, we use the complexity defined in equation 13. Here, we ignore the cost of projecting the depth map. The 3D DFT has complexity $3N^3 log(N)$. This is the first step (see figure 2), the next is the spherical-map transform which is complexity $45N^3$. If processed on the GPU the performance becomes 45 operations per voxel assuming that one voxel is assigned to one unit of processing. A 3D transform is 30 operations per voxel, 2D phase correlation requires 15 operations to multiply the frequency spectra and $2N^2 log(N)$ operations to do the 2D FFT. Finally a projection map transform requires 1 operation per voxel.

In total, the proposed method requires 2× 3D FFTs, 2× spherical-map transforms, 1× 3D geometrical transformation, 3× 2D phase correlations and 4× projection map transforms. The total complexity is added up for all of these functions and given in equation 14.

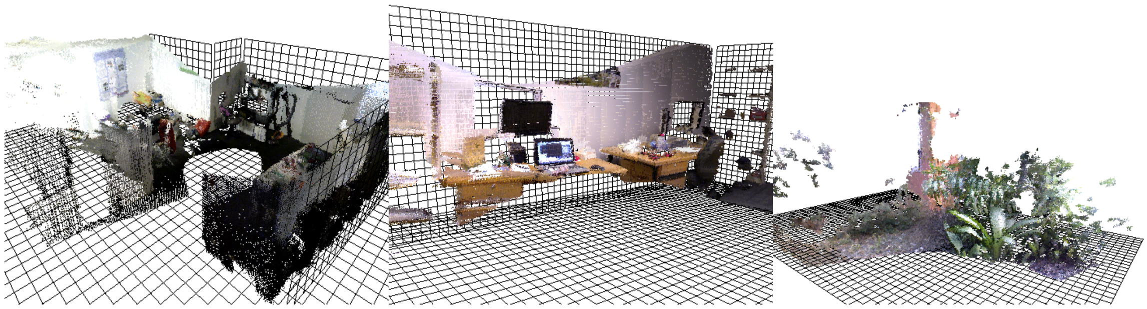
$$6log(N) \times (N^3 + N^2) + 169 \tag{14}$$

Figure 5 provides a visualization of the performance improvement which the proposed method achieves over the original Fourier volume registration approach. It is clear that the proposed method is around 3 times faster than the original Fourier based volume registration approach. This is due to the reduction in the amount of data to process afforded by the novel spherical-map transform and orthogonal projection methods.

6. Experiments

[ht]

A number of experiments were undertaken to assess the reconstruction accuracy, noise sensitivity and robustness to object motion. Experiments were performed using an ASUS Zenbook UX303LN with an Intel i7 5500u Dual Core 2.4GHz processor, 8GB of RAM and an NVIDIA GE-FORCE 840 M GPU.



(a) Apartment (b) Office (c) Garden

Figure 4. Reconstructed Scenes.

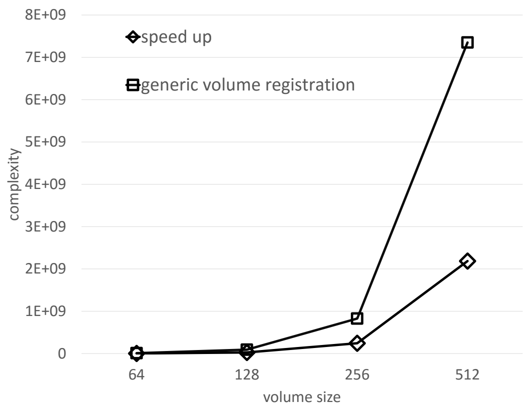


Figure 5. Comparison of performance between volume registration and the proposed speed up for different volume sizes.

translation (cm)	noise range (%)	SNR	error (cm)	error (voxel)
5cm	0	∞	0	0
5cm	10	20db	0	0
5cm	25	12db	0	0
5cm	50	6db	0	0
5cm	75	2.5db	112.28	89.83
10cm	0	∞	0	0
10cm	10	20db	0	0
10cm	25	12db	0	0
10cm	50	6db	156.65	125.32
15cm	0	∞	2.8	2.24
15cm	10	20db	2.8	2.24
15cm	25	12db	2.8	2.24
15cm	50	6db	198.55	158.84

Table 1. Translation Tracking

For volumes of 384^3, 1 × registration per second was possible. To achieve real-time performance, 1 out of every 30th frames was processed.

[ht]

6.1. Reconstruction Quality

To assess reconstruction accuracy, two indoor environments (Apartment and Office) as well as one outdoor environment (Garden) were used, these can be seen in figures 4a, 4b and 4c respectively. The Apartment reconstruction was recorded by moving through a room whilst rotating the camera. Some frames contained nothing but featureless walls, others had contrast shifts due to the camera's automatic contrast feature, yet, accurate reconstruction was achieved. The office reconstruction was generated by rotating the camera about the y-axis while moving backwards. Whilst our method is a closed form solution, its accuracy is still comparable to existing feature based SLAM methods. Typical feature based methods work well with indoor environments where local features are readily distinguishable and easy to match. They do not tend to work as well with complex outdoor scenes where feature confusion is likely. To assess performance in such outdoor scenes, a garden scene containing bushes, plants and a ground covering of bark and rocks was used. In the case of a feature matching approach this scene would likely result in feature confusion, making camera tracking difficult. The proposed method was able to produce a good quality reconstruction. Hence, our approach readily overcomes difficulties common to feature matching methods.

6.2. Noise Sensitivity

To assess robustness to noise, the estimated camera parameters are compared to ground truth data under different noise conditions. In each experiment, varying amounts of random noise were added per voxel prior to registration. This is expressed in decibels using the Signal to Noise Ratio (SNR). Each voxel value lies in the range [0-1]. Here, a noise value of 10% means random noise was added in the range [-0.05, 0.05]. Tracking error is measured in centimetres and voxel error (the error in the phase correlation volume). The first experiment evaluated noise robustness whilst the camera was translated by varying amounts (5cm, 10cm and 15cm). Results in Table 1 show that, for camera translations up to 15cm and SNR values above 6.0 our method is robust to noise. At video rates, a

rotation	noise (%)	SNR	error (θ)	error (voxel)
10°	0	∞	0.31	0
10°	10	20db	0.31	0
10°	25	12db	0.63	1
10°	30	10.5db	90.62	96
20°	0	∞	0.31	0
20°	10	20db	0.63	1
20°	15	16.5db	38.13	40
30°	0	∞	3.75	4
30°	10	20db	3.28	3
30°	15	16.5db	30	32

Table 2. Rotation Tracking

Object Size	error (cm)	error (voxel)
0.35	0	0
2.95	0	0
6.22	0	0
12.28	0	0
19.82	0	0
22.39	0	0
26.09	0	0
31.00	0	0
48.23	38.42	15
74.32	113.57	44

Table 3. Object Motion Test

displacement of 10cm per frame equates to a camera velocity of 3 m/s (about twice the normal walking speed).

Table 2 shows the results for tracking camera rotations of 10, 20 and 30 degrees per frame. At video rates, 12 degrees per frame is almost a full rotation per second. In rotations of 10 degrees, the error was less than a degree for all but a noise level of 30% and above. This base line error is due to the sampling resolution of the volume, as voxel error was in fact zero. As with pure translation, the effect of noise increases with camera disparity. At 30 degrees, little matching information is available. However, for noise levels of 10% or less, voxel distance error was as low as 4 with an angular error less than 3.8. Rotations of this magnitude are unlikely, moreover motion blur would occur.

6.3. Robustness to Object Motion

To assess robustness to object motion, experiments were conducted by moving the camera backwards along the z-axis by 5cm per frame whilst moving objects in and out of the scene so that they only appear in one of the volumes being registered. Various sized objects including stacks of CDs, large boxes, people and several pieces of furniture were used and are measured by the percentage of the frame they occupy. Results from Table 3 show the proposed method was accurate upto an object size of 31%, but failed for objects taking up over 48.23%.

7. Conclusions

In this paper, we proposed a novel non-feature based approach to SLAM, which can generate accurate 3D color reconstructions of both indoor and outdoor environments. This method is a closed form solution, scales naturally to the GPU, and is shown to be robust to global noise and object motion. We also proposed a method to speed up this method by up to 3 times.

8. Future Work

Future work in this area includes investigating a system to recover from mis-registered frames and to continue to improve performance.

Acknowledgement. Luke Lincoln would like to thank Thao Lincoln.

References

[1] Beall, C., Dellaert, F., Mahon, I. and Williams, S.B. (2011) Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys. In *OCEANS, 2011 IEEE-Spain* (IEEE): 1–6.

[2] Bylow, E., Sturm, J., Kerl, C., Kahl, F. and Cremers, D. (2013) Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS) Conference 2013*, **9**.

[3] Chekhlov, D., Pupilli, M., Mayol, W. and Calway, A. (2007) Robust real-time visual slam using scale prediction and exemplar based feature description. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (IEEE): 1–7.

[4] Davison, A.J. and Murray, D.W. (2002) Simultaneous localization and map-building using active vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(7): 865–880.

[5] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D. and Burgard, W. (2012) An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (IEEE): 1691–1696.

[6] Engelhard, N., Endres, F., Hess, J., Sturm, J. and Burgard, W. (2011) 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, Vasteras, Sweden*, **180**.

[7] Eudes, A., Lhuillier, M., Naudet-Collette, S. and Dhome, M. (2010) Fast odometry integration in local bundle adjustment-based visual slam. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (IEEE): 290–293.

[8] Foroosh, H., Zerubia, J.B. and Berthod, M. (2002) Extension of phase correlation to subpixel registration. *Image Processing, IEEE Transactions on* **11**: 188–200.

[9] Gil, A., Reinoso, O., Mozos, O.M., Stachniss, C. and Burgard, W. (2006) Improving data association in

vision-based slam. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (IEEE): 2076–2081.

[10] Gonzalez, R. (2011) Improving phase correlation for image registration. In *Image And Vision Computing New Zealand 2011 (IVCNZ2011)* (http://ieeexplore. ieee. org/xpl/conhome. jsp? punumber= 1002602).

[11] Henry, P., Krainin, M., Herbst, E., Ren, X. and Fox, D. (2010) Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER* (Citeseer).

[12] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J. *et al.* (2011) Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (ACM): 559–568.

[13] Jensfelt, P., Kragic, D., Folkesson, J. and Bjorkman, M. (2006) A framework for vision based bearing only 3d slam. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (IEEE): 1944–1950.

[14] Jeong, W.Y. and Lee, K.M. (2006) Visual slam with line and corner features. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (IEEE): 2570–2575.

[15] Kerl, C., Sturm, J. and Cremers, D. (2013) Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (IEEE): 2100–2106.

[16] Konolige, K. and Agrawal, M. (2008) Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on* **24**(5): 1066–1077.

[17] Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V. and Fua, P. (2010) View-based maps. *The International Journal of Robotics Research* .

[18] Kundu, A., Krishna, K.M. and Jawahar, C. (2010) Realtime motion segmentation based multibody visual slam. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing* (ACM): 251–258.

[19] Leelasawassuk, T. and Mayol-Cuevas, W.W. (2013) 3d from looking: using wearable gaze tracking for hands-free and feedback-free object modelling. In *Proceedings of the 17th annual international symposium on International symposium on wearable computers* (ACM): 105–112.

[20] Miro, J.V., Zhou, W. and Dissanayake, G. (2006) Towards vision based navigation in large indoor environments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (IEEE): 2096–2102.

[21] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P. *et al.* (2011) Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on* (IEEE): 127–136.

[22] Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C. *et al.* (2008) Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision* **78**(2-3): 143–167.

[23] Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J. and Koch, R. (2004) Visual modeling with a hand-held camera. *International Journal of Computer Vision* **59**(3): 207–232.

[24] Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M. and Bathiche, S. (2013) Monofusion: Real-time 3d reconstruction of small scenes with a single web camera. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on* (IEEE): 83–88.

[25] Silveira, G., Malis, E. and Rives, P. (2008) An efficient direct approach to visual slam. *Robotics, IEEE Transactions on* **24**(5): 969–979.

[26] Sim, R., Elinas, P., Griffin, M., Little, J.J. *et al.* (2005) Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, **14**: 9–16.

[27] Stückler, J. and Behnke, S. (2012) Robust real-time registration of rgb-d images using multi-resolution surfel representations. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on* (VDE): 1–4.

[28] Weikersdorfer, D., Adrian, D.B., Cremers, D. and Conradt, J. (2014) Event-based 3d slam with a depth-augmented dynamic vision sensor. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE): 359–364.

[29] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J. and McDonald, J. (2012) Kintinuous: Spatially extended kinectfusion. *MIT-CSAIL* .