

# An approach to stereo-point cloud registration using image homographies

Stephen D. Fox<sup>a</sup> and Damian M. Lyons<sup>a</sup>

<sup>a</sup>Fordham University, Robotics and Computer Vision Laboratory, Bronx, NY

## ABSTRACT

A mobile robot equipped with a stereo camera can measure both the video image of a scene and the visual disparity in the scene. The disparity image can be used to generate a collection of points, each representing the location of a surface in the visual scene as a 3D point with respect to the location of the stereo camera: a point cloud. If the stereo camera is moving, e.g., mounted on a moving robot, aligning these scans becomes a difficult, and computationally expensive problem. Many finely tuned versions of the iterative closest point algorithm (ICP) have been used throughout robotics for registration of these sets of scans. However, ICP relies on theoretical convergence to the nearest local minimum of the dynamical system: there is no guarantee that ICP will accurately align the scans. In order to address two problems with ICP, convergence time and accuracy of convergence, we have developed an improvement by using salient keypoints from successive video images to calculate an affine transformation estimate of the camera location. This transformation, when applied to the target point cloud, provides ICP an initial guess to reduce the computational time required for point cloud registration and improve the quality of registration. We report ICP convergence times with and without image information for a set of stereo data point clouds to demonstrate the effectiveness of the approach.

**Keywords:** registration, point cloud, alignment, scan-matching, iterative closest point, mobile robot, stereo vision, computer vision, homography, egomotion

## 1. INTRODUCTION

Consider the problem of aligning scans of a 3D surface into a single surface. This is known as the ‘registration’ problem. Point clouds have become an increasingly popular way to represent a robot’s environment in mobile robotics. There are a variety of range-sensing hardware devices capable generating point clouds now available including industry-standard stereo camera models, such as the Point Grey BumbleBee2, laser range-finders, the Microsoft Kinect, and even smart phones like the HTC EVO 3D. Mobile robots use various range sensors to construct representations of their surrounding environment in order to make safe navigation decisions, to build useful maps and to acquire models from the environment. Stereo cameras provide visual information about the environment. Two overlapping images (i.e., ‘stereo’) allows us to measure the visual disparity of corresponding features in the scene, from which we can construct a depth map, or range image, of the visible surface in the environment. These surfaces are represented as a collection of discrete 3D points with respect to the location of the stereo camera, often referred to as a ‘point cloud.’

Because a point cloud computed from a single range scan can only measure the visible surface of objects in the scene, multiple scans often need to be ‘stitched’ together in order to produce a useful representation of the environment. This process of stitching point clouds together is known as registration. Since a mobile robot is not able to measure its exact transformation as it moves from one location to the next, the transformation of these point clouds also becomes uncertain. This error compounds, as a robot’s wheels slip, there is random error in the sensor readings themselves, and there is error in a robot’s odometry sensors. There is a rich literature on probabilistic mapping and localization techniques to deal with these problem.<sup>1</sup> The direct consequence for the registration problem, however, is aligning multiple scans becomes a difficult, and computationally expensive problem.

With bad or wrong transformations between scans, the composition of these clouds can become very noisy. Point clouds of the same surface from different perspectives will not align properly, so the structure of the point cloud will not accurately represent the surface it is modeling. In this way, several composed point cloud scans can

become useless, even dangerous, to an autonomous platform if they are relied on for navigation, localization or mapping.

We introduce an approach to improving registration via the iterative closest point (ICP) algorithm by first estimating the transformation between stereo scans using image homographies. We will use an image registration method presented in Lyons' previous work,<sup>2</sup> and utilize the Point Cloud Library, an open-source library of computational geometry for manipulation of point clouds for our experiment. The remainder of the paper is organized as follows. We first review the ICP method of scan alignment of 3D point clouds in Section 2. Section 3 describes our approach to computing an estimate of the camera's egomotion via image homographies. This estimate of the robot's motion between scans serves as an initial transformation estimate for registration with ICP. In Section 4, we give the resulting performance data which demonstrates the effectiveness of our homography-based egomotion guess compared to a raw odometry guess using ICP. Finally, Section 5 contains suggestions for further investigation, as well as some proposed applications of our method.

## 2. POINT CLOUD REGISTRATION

There are many algorithms which address the problem of scan alignment and registration including polar scan matching,<sup>3</sup> virtual scan alignment<sup>4</sup> and iterative closest point (ICP).<sup>5</sup> In particular, many finely tuned versions of ICP have been used throughout mobile robotics.<sup>6</sup> The Point Cloud Library<sup>7</sup> is an open-source C++ library for processing point clouds which we have used to implement experiments in our project. The PCL has three implementations of ICP alone, in addition to other 3D feature-based registration techniques.

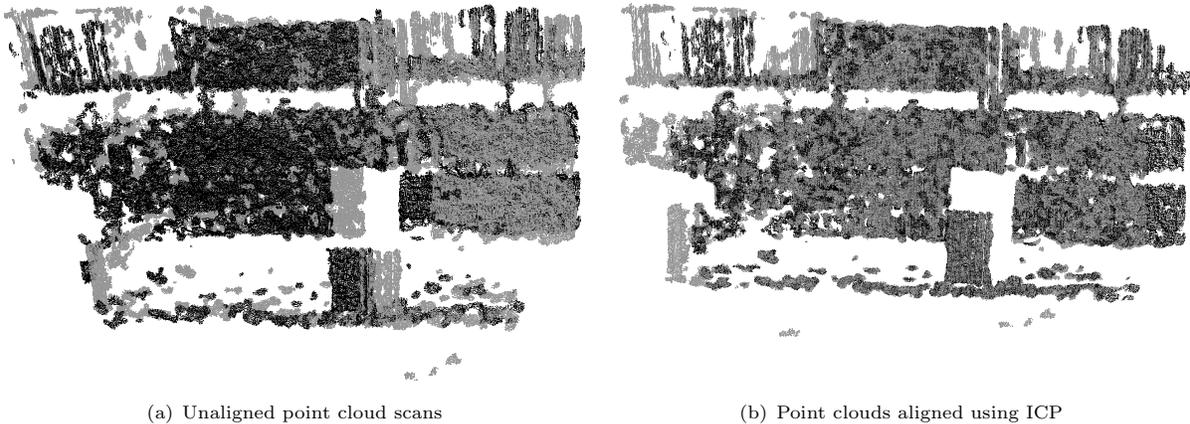


Figure 1. Registration of the two point clouds in (a) yields the aligned clouds in (b).

When used for scan matching of point clouds, ICP returns a rigid transformation that might align two clouds. To register point cloud scans, we assume they must be at least partially overlapping: a subset of each point cloud must represent some common area of the same surface.

Many variations of ICP follow a basic iterative pipeline of:

1. searching for correspondences,
2. rejecting outliers (i.e., points that are 'probably' not in the set of overlapping regions—metrics vary<sup>6,8</sup>),
3. and minimizing the distance between corresponding points.

When ICP converges, the resulting transformation that it produces is a minimization of the distance between corresponding points. Unfortunately, it is possible for the algorithm to get stuck in a local minimum, thus returning a bad registration transformation.

ICP is useful and efficient enough for registration, but one of its major drawbacks is its reliance on a theoretical convergence to the nearest *local* minimum of the system. The classical ICP algorithm is numerically stable,<sup>5</sup> but

without a ‘good enough’ initial guess of the transformation from one scan to the next, the algorithm can report convergence to an incorrect local minimum. There is no guarantee that the ICP algorithm will accurately align the scans. One bad registration in this manner can result in a dangerous map, or render the data useless for navigation or object segmentation. Furthermore, on large sets of point cloud data as stereo cameras produce (proportionally larger data sets with resolution), each iteration is very expensive to compute. To alleviate the problem of bad convergence and to reduce the total convergence time, it is customary to apply a guess of the estimated transformation from one scan to the next if known. The source of this guess can be raw odometry, gyroscope readings, GPS readings or other motion-estimating hardware. When dealing with ICP, we must not forget that the further the initial estimate is from the ‘actual’ transformation, (a) the longer the convergence time will be, and (b) the higher the likelihood of bad convergence.

### 3. REGISTRATION VIA IMAGE HOMOGRAPHIES

Our approach to registration first computes an estimate of the camera’s motion from one position to another. One advantage of using a stereo camera as a range-sensor for a mobile robot is the ability to compute depth information in addition to the video data a standard camera provides. In previous work, Lyons et. al. estimates an affine transformation to synchronize synthetic and real visual imagery.<sup>9</sup>

In this paper, we use his 2D image registration technique to estimating an affine transform as an initial estimate for 3D registration of stereo point clouds with ICP. In this section, we first review the method for 2D image registration and estimation of the affine transformation. We then describe its experimental application as the initial guess for ICP.

#### 3.1 Computing the Image Registration

To compute an estimate of the camera’s transformation from one scan position to another, a pair of images is analyzed for salient ‘keypoints,’ which we identify as corners. For detection of keypoints, our implementation uses the Trajkovic and Hedley *Fast Corner Detection* library.<sup>10</sup> Once the keypoints are identified in each image, we use the RANSAC algorithm to estimate an affine transformation between corresponding keypoints in the image. If we have two images,  $I_0$  and  $I_1$  from *different* stereo scans (i.e. there has been some translation or rotation of the camera), we can compute a  $2 \times 2$  rotation matrix  $\mathbf{A}$  and translation vector  $\mathbf{b}$ , such that for each pixel location  $p_0 = (x_0, y_0)$  in  $I_0$  and  $p_1 = (x_1, y_1)$  in  $I_1$ ,

$$p_1 = \mathbf{A}p_0 + \mathbf{b} . \tag{1}$$

Parameters  $\mathbf{A}$  and  $\mathbf{b}$  can be estimated using the RANSAC algorithm. In affine space, this gives us a  $3 \times 3$  homography matrix,

$$\mathbf{H}_e = \begin{pmatrix} A_{11} & A_{12} & b_1 \\ A_{21} & A_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} . \tag{2}$$



(a) Initial Image with Corners (b) Image with corners after 5 degree Rotation of Robot (c) 2(a) warped to 2(b)

Figure 2. Corners in (a) are matched to those in (b) to compute an image homography (Eq. 2) that ‘warps’ (a) to (b), resulting in (c).

Using the intrinsic parameters of the camera, focal length  $f$ , and principal point  $(c_x, c_y)$ , we construct the camera’s intrinsic matrix in the following way:

$$\mathbf{K} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

From this, we can compute an estimation of the translation and rotation of the camera from one scan to the next. This requires the multiplication,

$$\mathbf{R} = \mathbf{K}^{-1} \mathbf{H}_e \mathbf{K}, \quad (4)$$

resulting in the rigid transformaiton  $\mathbf{R}$ . At this point, the matrix  $\mathbf{R}$  must be expressed in the form of a  $4 \times 4$  matrix: a rotation matrix augmented by translation in affine space. For purposes of this experiment, we assumed no translation of the camera. More details on this method can be found in Lyons’ recent work.<sup>2,9</sup>

### 3.2 Applying as a Rigid Transformation

We use the reference image of a BumbleBee2 stereo camera to compute an estimate of the camera’s egomotion from one scan position to the next. The matrix  $\mathbf{R}$  computed in Eq. 4 is a rigid transformation representing the estimation of the camera’s motion between its position from the initial image to the final image. This becomes our ‘guess’ for registration of the two point clouds.

## 4. EXPERIMENT AND RESULTS

We compare three methods of registration using the Point Cloud Library’s implementation of nonlinear iterative closest point (ICP). The most naive use of ICP, which we refer to as Method 1, has no guess for the initial registration. Method 2 uses the raw encoder odometry of a Pioneer 3 AT mobile robot as a guess for the ICP registration. Method 3 uses the transformation computed from the image homograhpy in Eq. 4 between references images of the stereo camera before and after the robot’s rotation about its center. In this section, we first describe our sample stereo data sets and the steps we took to preprocess the point clouds. We then report the performance of each method, measured by the average convergence time to the proper registration. Since PCL’s

Figure 3. ICP Convergence times for the indoor data samples.

ICP Convergence Time:  
Indoor Samples (1-6)

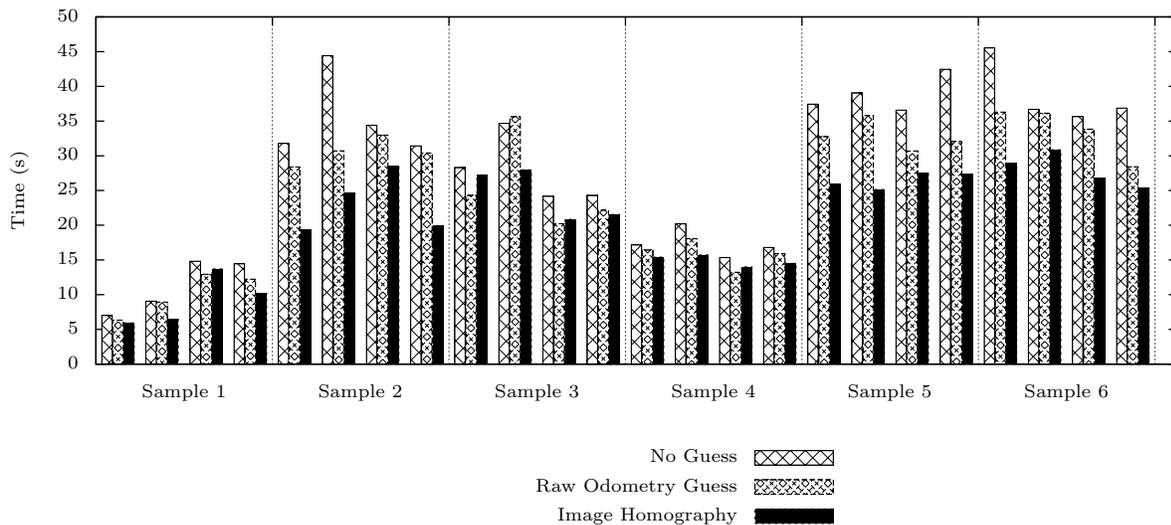
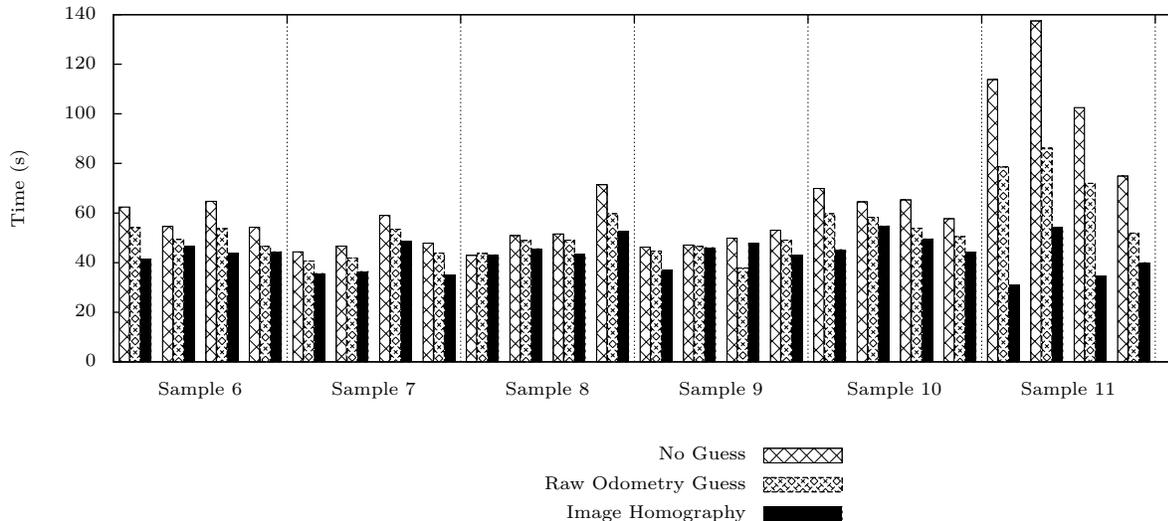


Figure 4. ICP convergence times for outdoor data samples.

ICP Convergence Time:  
Outdoor Samples (6-12)



implementation of ICP, as well as our image homography estimator relies on a stochastic RANSAC routine, we average the results of thirty independent registration trials for each method in order to measure convergence time.

#### 4.1 Sample Stereo Data Sets

Twelve stereo data sets were collected using a Point Grey BumbleBee2 camera at  $800 \times 600$  resolution. Each set of scans contains five stereo scans, taken at approximately five degree intervals of rotation about the center of the mobile robot. The rotation was measured using the raw encoder odometry of the Pioneer 3 AT mobile robot. To simplify computation, the stereo camera was mounted in the center of the mobile robot platform.

Sample sets come equally from indoor (Samples 1-6) and outdoor (Samples 7-12) locations with approximately uniform depth. Since the outdoor scenes tend to have more light and more texture, point clouds from these samples are considerably more dense. All stereo processing, including calibration of the camera, rectification of the stereo image pairs, and conversion to a point cloud was done using Point Grey’s Triclops SDK (version 3.1).

Each scan produces a point cloud with approximately 300,000 points, but much of this is noise that should be filtered. Before applying the registration routines, we clean the point clouds by following these steps:

1. Remove all points less than 3 centimeters above the ground plane
2. Uniformly downsample the point cloud using voxels of  $1 \text{ mm}^3$
3. Filter for statistical outliers using clusters of 500 points and rejection  $\sigma$  of 0.25 to remove noise distant from the main part of the cloud
4. Filter for statistical outliers using clusters of 100 points and rejection  $\sigma$  of 0.25 for local sensor noise.

This pipeline extracts a point cloud with a structure reasonably similar to the target in the scene. The structure is good enough for registration of stereo-generated point cloud scans.

#### 4.2 Results

Once the stereo point clouds were pre-processed, we compared the convergence time of ICP using three methods. ‘No Guess’ indicates the point clouds were not transformed before beginning ICP. ‘Raw Odometry Guess’

indicates the point cloud was transformed based on the raw encoder odometry of the Pioneer 3 AT mobile robot before registration with ICP. Finally, the ‘Image Homography’ method uses our initial guess computed in Eq. (4) to transform the point cloud before beginning ICP. Our ‘Image Homography’ transformation was computed between the reference\* images in the stereo camera.

The data in Figures 3 and 4 represent pairwise scans at approximately five degree rotations. Each sample contains five scans, with four pairwise point clouds rotated roughly five degrees. For each pair of point clouds, each method was run thirty times. The data in the charts represent the average convergence time for each method on each pair of point clouds. The data suggests that this is a useful approach to improving registration compared to estimates given by raw encoder odometry.

## 5. SUMMARY

In this paper, we introduced an approach to improving stereo point cloud registration by processing the 2D visual information. The objective is to compute an image homography to register the images of the reference camera to provide the ICP algorithm an initial guess of the camera’s transformation between scans. We tested the effectiveness of this approach against raw encoder odometry as the initial registration guess. Experimental results show improved convergence time for stereo-generated point cloud registration.

This approach works when the visual scene has enough corner points that an affine transformation can be estimated. This is not an unreasonable constraint with stereo point clouds, since stereo cameras rely on texture in images to produce good depth information. This approach could be automated for model extraction, mapping, and localization. Furthermore, for mobile robots who do not have hardware for sensing its change in position accurately, this method could be used to get an initial estimate of the translation and rotation, improved by

## ACKNOWLEDGMENTS

This work was initially funded by a Fordham College undergraduate research grant in the 2010-2011 academic year. Support for Stephen Fox’s current graduate study at Fordham University comes in part from Defense Threat Reduction Agency, Basic Research Award # HDTRA1-11-1-0038, to Georgia Tech and Fordham University.

## REFERENCES

1. Thrun, S., Burgard, W., and Fox, D., [*Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*], Intelligent robotics and autonomous agents, The MIT Press (August 2005).
2. Lyons, D. M. and Benjamin, D. P., “Locating and tracking objects by efficient comparison of real and predicted synthetic video imagery,” *SPIE Conference on Intelligent Robots and Computer Vision XXVI: Algorithms and Techniques* (January 2009).
3. Diosi, A. and Kleeman, L., “Fast Laser Scan Matching using Polar Coordinates,” *The International Journal of Robotics Research* **26**(10), 1125–1153 (2007).
4. Lakaemper, R. and Adluru, N., “Using virtual scans for improved mapping and evaluation,” *Auton. Robots* **27**, 431–448 (November 2009).
5. Besl, P. and McKay, N., “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 239–256 (1992).
6. Rusinkiewicz, S. and Levoy, M., “Efficient variants of the ICP algorithm,” *Third International Conference on 3D Digital Imaging and Modeling (3DIM)* (June 2001).
7. Rusu, R. B. and Cousins, S., “3D is here: Point Cloud Library (PCL),” *IEEE International Conference on Robotics and Automation (ICRA)* (May 9-13 2011).
8. Segal, A., Haehnel, D., and Thrun, S., “Generalized-ICP,” *Proceedings of Robotics: Science and Systems* (June 2009).
9. Lyons, D. M., Chaudhry, S., and Benjamin, D. P., “Synchronizing real and predicted synthetic video imagery for localization of a robot to a 3d environment,” *SPIE Conference on Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques* (January 2010).
10. Trajkovic, M. and Hedley, M., “Fast corner detection,” *Image Vision Comput.* **16**(2), 75–87 (1998).

---

\*The reference image of a stereo camera is the image in which the origin of the point cloud coincides.