Registration of Images to Unorganized 3D Point Clouds Using Contour Cues

Alba Pujol-Miro, Javier Ruiz-Hidalgo and Josep R. Casas
Universitat Politècnica de Catalunya
Image Processing Group - Signal Theory and Communications
Barcelona, Spain

***Abstract*—Low resolution commercial 3D sensors contribute to computer vision tasks even better when the analysis is carried out in a combination with higher resolution image data. This requires registration of 2D images to unorganized 3D point clouds. In this paper we present a framework for 2D-3D data fusion to obtain directly the camera pose of a 2D color image in relation to a 3D point cloud. It includes a novel multiscale intensity feature detection algorithm and a modified ICP procedure based on point-to-line distances. The framework is generic for several data types (such as CAD designs or LiDAR data without photometric information), and results show that performance is comparable to the state of the art, while avoiding manual markers or specific patterns on the data.**

***Index Terms*—feature extraction, image registration, iterative closest point algorithm, stereo vision**

I. Introduction

With the emergence of 3D acquisition systems -like Kinect and LiDAR scanners-, the availability of 3D data has recently increased for computer vision tasks. However, affordable 3D acquisition sensors provide a much lower capture quality than their counterparts in 2D imaging. A potential strategy for analysis focuses on the early fusion by registration of different data types to obtain a richer representation of the available information.

Multimodal registration has been developed in several areas and for multiple data types. In aerial imaging, for instance, image and LiDAR data [1] are registered by algorithms exploiting visual cues [2], information theory [3], [4], or surface reconstruction [5], among others. Developments have also been presented in the medical field [6]. In this case, the registration is applied to voxellized images. For the registration of several images to a point cloud [7], some algorithms start from an initial 3D reconstruction from the images by Structure-from-Motion (SfM) [8], [9], [10]. Registration algorithms for multi-head sensors –for example, Kinect and RGB camera– using calibration patterns have also been presented [11].

The method proposed in this paper aims to introduce a registration between a single 2D color image and an unorganized 3D point cloud. In this case SfM techniques cannot be applied, since there is only a single image instance available. The proposed procedure consists in detecting a set of relevant features on both sets and computing a direct matching between them using a modified version of the Iterative Closest Point algorithm [12]. The detection approach is similar to [13], without constraining the point cloud to: a) an RGBD image [13], b) a special pattern [11] or c) for the scene to have certain geometrical features [3].

This research was supported by the Spanish Ministry of Science and Innovation via a doctoral grant to the first author (FPU2014), and developed in the framework of projects TEC2013-43935-R and TEC2016-75976-R, financed by the Ministerio de Economía, Industria y Competitividad and the European Regional Development Fund (ERDF).

II. Methodology

The proposed algorithm to register an image to a 3D point cloud can be divided in three basic steps (shown in Figure 1):

Cloud representation Given the internal camera parameters, the image is mapped to a plane in 3D space representing the set of *projection vectors* that relate the 3D scene with the projected image.

Feature detection Points with a steep variation on intensity gradient or surface normal –relevant points– are detected on both the image (intensity variations only) and the point cloud (intensity -if available- and normal variations). The output relevant feature points will serve as baseline for the alignment process.

Matching The matching algorithm consists in a modification of the ICP algorithm [12] that minimizes the distance between each point and the *projection vectors* mentioned above. The final output is the rigid transformation matrix that maps the projected point cloud into the image.

This pipeline can be applied to register a single image to several 3D input data types: CAD designs, range sensor captures, etc. The only requisite is that such data must be represented as a set of points in 3-D space.

A. Cloud representation

In order to establish a direct relationship between the image and the 3D point cloud, we should represent both data types in the same space. This representation allows for the introduction of camera information -distortion, optical center, focal distance- into the system. The image is expressed as a set of 3D projection vectors, each projection vector representing the 3D line that would link the optical center with the corresponding image point in the optical plane of an ideal pinhole camera. This 3D line corresponds to all the possible locations of the given pixel in 3D space.

The chosen implementation assumes the optical camera center at the origin of the world $[0, 0, 0]$. Each image point

ISBN 978-0-9928626-7-1

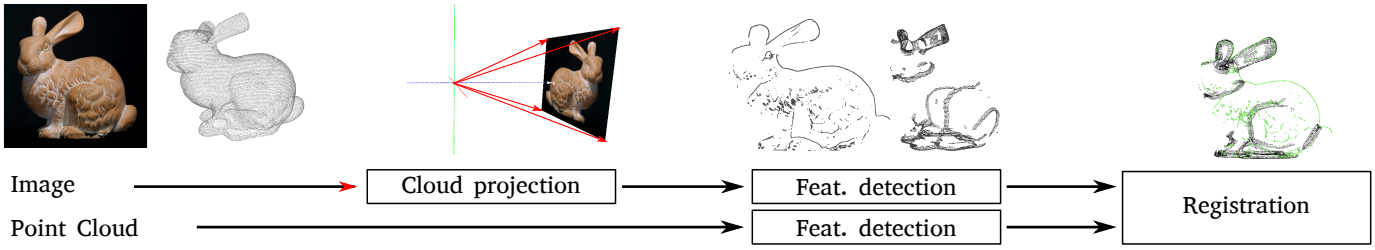

Fig. 1. Proposed method block diagram.

is represented as a 3D point located at the plane $z = 1$. Therefore, the procedure consists in the following steps:

1) Undistort the input image using the camera calibration parameters.
2) Project each undistorted image pixel onto the plane $z = 1$. The coordinates [x,y] are computed as follows:

$$[x, y] = \frac{[m, n] - [c_x, c_y]}{[f_x, f_y]} \quad (1)$$

where $[m, n]$ are the image coordinates, $[c_x, c_y]$ is the location of the optical center and $[f_x, f_y]$ the focal distance of the camera for each axis.

B. Feature detection

Once both data are in a 3D framework, the next step is to obtain a set of relevant features for the matching. These features are detected by means of two algorithms:

Gradient features The curvature between neighboring point clouds is detected using the multiscale feature detection presented by Pauly et al. [14]. This algorithm computes the covariance matrix with increasing K-neighbors, and classifies the points according to the obtained eigenvalues.

Intensity features The intensity changes between neighboring points are measured by comparing the averaged geometrical distance to the 'center of mass' of neighboring pixels. The intensity is used as the 'mass' for each pixel.

Intensity and gradient features are detected for each pixel i on the image, obtaining a $w_{g,i}$ score for the gradient detection and a $w_{y,i}$ score for the intensity detection. These two scores are combined as shown in Equation 2.

$$w_i = max\left(w_{g,i}, w_{y,i}\right) \quad (2)$$

In order to obtain a set of relevant points, the score w_i is thresholded using an hysteresis algorithm parameterized by a pair of threshold values th_h, th_l. The algorithm selects all pixels with $w_i > th_h$ and assigns them as valid pixel outputs. For each valid pixel, a small neighborhood k_h is acquired. Pixels with $w_i > th_l$ are also assigned to the output and their neighborhood added to be searched for relevant points.

Intensity feature detection - This algorithm detects intensity features as steep intensity changes in a dominant direction on a small neighborhood. In images, this procedure consists in comparing the pixel to its 4 / 8 connectivity neighborhood, depending on the method, and establishing a score based on the difference between the intensity value on the pixel and its neighbors. In 3D point clouds, however, the neighborhood is not well defined, and this procedure can not be straightforwardly applied.

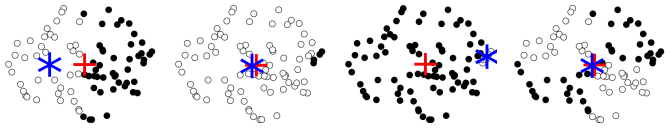

Fig. 2. Test cases for the intensity feature detection algorithm. Black and white points: luminance values in the analysis neighborhood. Red cross: geometrical center. Blue star: center of mass. Left: Centered contour. Middle-left: High luminance patch with side contour. Middle-right: Black patch with side contour. Right: Saddle point in luminance.

The intensity feature detection algorithm we propose can be applied to point clouds which have an intensity value assigned to each point. This algorithm is based on the computation of the geometrical center $m_{\vec{g},K,i}$ and the center of mass $m_{\vec{c},K,i}$ for each pixel/3D point i at position $[x_i, y_i, z_i]$, intensity I_i and neighborhood level K. Equation 4 shows the computation of the two centers. The layer score $w'_{y,K,i}$ corresponds to the distance between the two centers (Equation 5).

$$m_{\vec{g},K,i} = \frac{1}{K}\sum_{k=1}^{K}[x_k, y_k, z_k] \quad (3)$$

$$m_{\vec{c},K,i} = \frac{1}{\sum_{k=1}^{K} I_k}\sum_{k=1}^{K} I_k[x_k, y_k, z_k] \quad (4)$$

$$w'_{y,K,i} = ||m_{\vec{g},K,i} - m_{\vec{c},K,i}|| \quad (5)$$

Figure 2 shows the location of the geometrical center $m_{\vec{g},K,i}$ (red cross) and the center of mass $m_{\vec{c},K,i}$ (blue star) in several cases. When a sharp contour is present close to the center of the neighborhood under analysis, there is a shift between the geometrical center and the mass center (left). However, when this contour is located on the neighborhood side, the behavior of the mass center changes depending on the predominant luminance level (middle left, middle right). It should also be noted that the presented algorithm does not have a strong response on saddle points (right). However, in the use case presented in this document this fact has no significant impact on the result.

ISBN 978-0-9928626-7-1 © EURASIP 2017

To balance the response for different intensity patterns an additional center of mass is computed inverting the intensities. The score is updated as shown in Equation 7, selecting the center of mass with less distance to the geometrical center. This dual computation allows to have a symmetrical response on the contours, and to select only those points that have a steep intensity change on their close neighborhood, avoiding thick contours when increasing the analysis neighborhood.

$$\vec{m'_{c,K,i}} = \frac{1}{\sum_{k=1}^{K}(1-I_k)} \sum_{k=1}^{K}(1-I_k)[x_k, y_k, z_k] \tag{6}$$

$$w_{y,K,i} = min\left(||\vec{m_{g,K,i}} - \vec{m_{c,K,i}}||, ||\vec{m_{g,K,i}} - \vec{m'_{c,K,i}}||\right) \tag{7}$$

This score is computed in increasing K levels, similarly to Pauly algorithm [14]. The final weight is incremented each time that $w_{y,k,i}$ is higher than a certain threshold th. However, the absolute distance between the two centers also depends on the search radius. To get a robust estimate of the search radius, $w_{y,k,i}$ is normalized with the average distance of the K neighbors to the query point $\sum_{k=1}^{K} d_k$. Therefore, the final score is computed as shown in Equation 8.

$$w_{y,i} = \frac{1}{K_{max} - K_{min}} \sum_{K=K_{min}}^{K_{max}} \left(\frac{w_{y,K,i}}{\sum_{k=1}^{K} d_k} > th \right) \tag{8}$$

The presented multiscale analysis allows to compute in a single pass both gradient contours –with Pauly's multiscale feature detection [14]– and intensity contours –using the proposed feature detection method. The presented multiscale analysis allows to compute in a single pass both gradient and intensity contours using Pauly's multiscale feature detection [14] and the proposed feature detection method, respectively.

C. Image-to-cloud ICP

The features detected in the previous step are used to directly register an image plane to a 3D point cloud. This alignment is done using a modification of the ICP algorithm. A high-level summary of the ICP procedure is as follows:

1) Establish correspondences between point clouds using the Euclidean distance between points and selecting the nearest neighbor.
2) Compute the rigid (rotation-translation) transformation that minimizes the Euclidean distance between points.
3) Iterate until convergence.

In the scenario presented in this paper, no depth information is available for the image data. Therefore, the 3D location of image pixels is defined by a projection vector that represents all the possible locations of the image pixel in 3D space.

If the features have been correctly detected, when the image and the point cloud are registered each pixel of the image, represented as a line in the 3D space, should have minimal distance to a relevant point in the 3D point cloud.

Therefore, the ICP algorithm has been modified to use the Euclidean point-to-line distance instead of the Euclidean distance to match the 3D point cloud and the 2D image pixels.

Fig. 3. Input data. Left: Capture with a Kinect sensor. Middle: Stanford bunny color image. Right: Stanford bunny mesh reconstruction.

The main changes to the ICP algorithm are follows:

- Establish the correspondence pairs that minimize the point-to-line distance between the projection vector $\vec{v_i}$ and the query point p_i:

$$d_i = \frac{|p_i \times (p_i - \vec{v_i})|}{|\vec{v_i}|} \tag{9}$$

- Compute the rigid transformation that minimizes the global Euclidean point-to-line distance between pairs. This minimization is implemented using the Levenberg-Marquadt algorithm to solve the non-linear minimization.

This procedure is iterated until the difference of the transformation between two successive iterations is lower than a certain threshold or a maximum number of iterations has been reached.

III. Results and Analysis

This section contains a qualitative and quantitative evaluation of the main contributions presented in this paper. In the first part, the intensity feature detection algorithm is evaluated. The second part contains an evaluation of the full registration pipeline. The implementation of this algorithm has been done using the Point Cloud Library (PCL) [15] and it is currently available online [16].

Figure 3 contains a representation of the data used in the testing process. This data can be summarized as follows:

- Scene capture with a Kinect sensor. Several RGBD and RGB captures of the same scene from different viewpoints.
- Color image of the Stanford bunny.
- Mesh reconstruction of the Stanford bunny. The points in the cloud are the vertices of the reconstruction without color/luminance information.

A. Intensity feature detection

The performance of the intensity feature detection presented in Section II-B is qualitatively evaluated with the results shown in Figure 4. Detected edges contain the valuable intensity transitions present in the scene, provided with good localization accuracy (thin contours) without any additional post-processing, both for strong and weak transitions. This is achieved thanks to the dual computation of the mass center in Equation 8.

The quantitative evaluation proposed consists in the precision of the contour detection as the ratio of wrongly detected pixels or 3D points over all detected pixels or 3D points.

ISBN 978-0-9928626-7-1 © EURASIP 2017

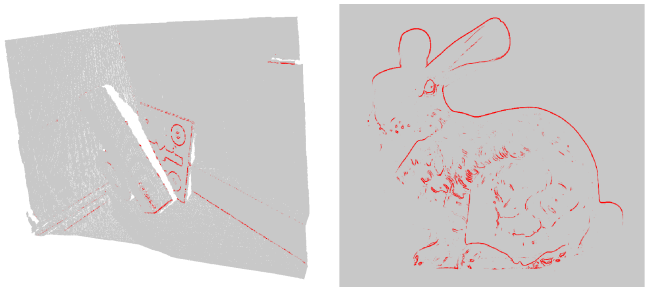

Fig. 4. Intensity gradient detection applied to Left: Point cloud captured using the Kinect sensor and Right: Stanford bunny color image.

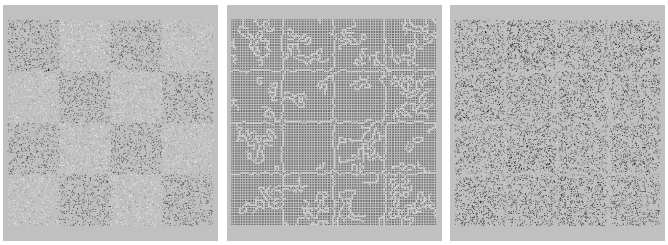

Fig. 5. Test performed using a synthetic chessboard. Left: chessboard with noise. Center: contours detected using Canny. Right: contours detected using the intensity feature detection.

The setup is based on a synthetic checkerboard image of 4x4 squares with a resolution of 128x128 pixels, as shown in Figure 5.

In the first test (top of Figure 6), zero mean Gaussian noise in the intensity values has been added. The contours are detected using three algorithms: the intensity feature detection algorithm presented in this paper, Canny edge detection and Canny edge detection with a previous Sobel filtering, a classical approach for contour detection.

The results obtained show that the proposed algorithm has a good response to large noise levels, being able to avoid false detections. The algorithm outperforms both Canny and Sobel+Canny without the need of any noise pre-processing or line thinning post-processing.

The second test (bottom of Figure 6) shows the performance of the algorithm with localization noise in $[x, y, z]$ coordinates. For this test, the checkerboard has been randomly generated by means of a uniform sampling of the surface. Additive Gaussian noise is then added to all axes on the 3D location. Even though the positional noise affects the algorithm, since the location of the query points is shifted alongside with the noise, the proposed algorithm is robust against small amounts of noise.

B. Point-to-line ICP registration

This section shows the evaluation results obtained with the full pipeline with several set-ups. First of all, a qualitative evaluation is presented with two sets of data in Figure 7. In these images, the black points represent the image cloud, the red points the initial cloud projection and the green points the final cloud projection.

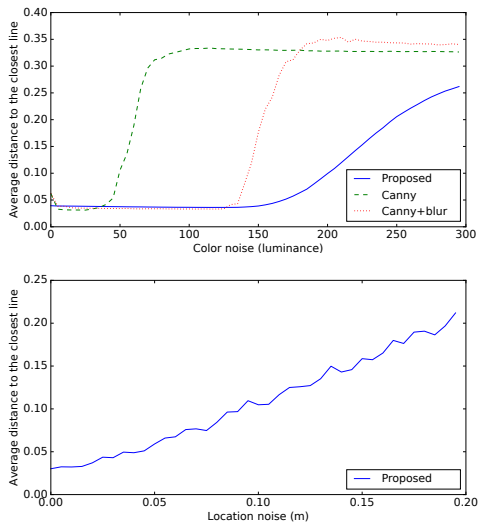

Fig. 6. Quantitative evaluation of the intensity feature detection algorithm. Top: evolution of the algorithm with different color noise levels compared with Canny and Sobel filtering + Canny. Bottom: evolution of the algorithm precision with increasing levels of noise in the coordinate location.

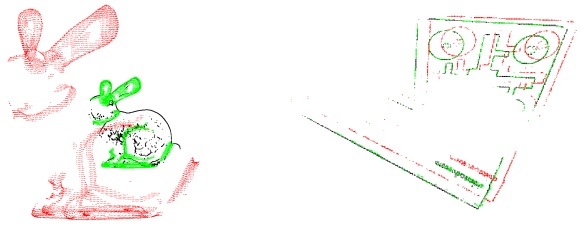

Fig. 7. Registering examples. Left: Bunny 3D model and bunny picture. Right: Kinect range capture and color image. The image contours are shown in black. The red contours are the initial location of the cloud projection, and the green contours are the final location of the cloud projection.

The algorithm is able to correctly register the detected features on the image and the point cloud, albeit not detecting the same features in both data. In the left image, the initial bunny –in red, larger– shows strong detection on the ears, feet, neck and tail, but the strong contour on the back of the bunny (see Figure 4 for intensity gradient detection on the image) is not present. However, since the detected features provide a good alignment between the two data types, the transformation that registers the data can be successfully obtained. In this case, the transformation consists of a translation to move the cloud away from the image and center the projection, and a rotation to align the data.

In the right image, the detection result for the original image in Figure 3 left is shown. In this case, the color information on both the image and the point cloud plays an important role. The circles and drawings present on the board provide strong contours on both sets that allow the registration of both images with a very small error.

The performance of the presented algorithm is compared against the calibration system between image cameras and range sensors based on checkerboard detection and plane

ISBN 978-0-9928626-7-1 © EURASIP 2017

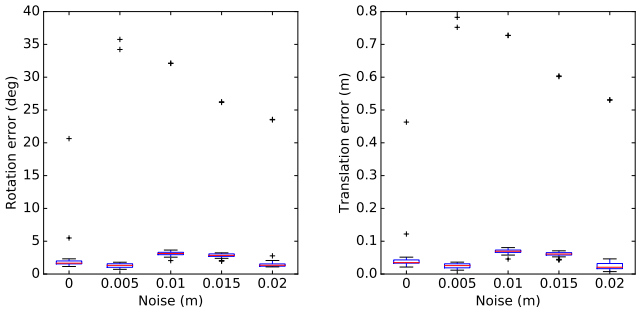

Fig. 8. Evaluation of the registering algorithm emulating the camera setup presented in [11]. Rotation (left) and translation (right) error is shown with varying degrees of Gaussian noise present in both the image and the point cloud. On each box, the central mark is the median, the edges extend to the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

fitting presented in [11]. Their evaluation consists of the rotation error e_r -distance between computed and ground truth translation vectors- and translation error e_t -angle between computed and ground truth rotation axis-with different Kinect/camera setups. The setups have been emulated by using a calibrated image-point cloud pair with the data shown in Figure 3. The point cloud has been rotated to the specified disparities on the paper and noise on the location/rotation is added using the specified tolerances to emulate the 29 captures used in their evaluation. Additionally, Gaussian noise has been added to perform each test.

The results obtained are shown in Figure 8. As a reference, [11] presents an average median translation error of $\sim$ 5cm with extremes up to 2.5m, and an average median rotation error of $\sim 10^\circ$ with extremes up to 150°. The proposed algorithm outperforms these error measures showing an average median translation error of 4.2cm with extremes up to 0.8m, and median rotation error of 2.05° with extremes up to 35°. Therefore, the proposed algorithm offers similar performance without the need of using any special pattern. Regarding the noise level, the performance is similar on both algorithms, as they are both able to handle small amounts of noise. On higher values, experiments have shown that the error of the presented algorithm has a highly correlated behavior with the accuracy of the edge detection shown in Figure 6.

IV. Conclusions

In this document a framework to directly register an image to a 3D point cloud has been presented. The proposed algorithm allows to obtain the transformation that aligns the projection of the image to the point cloud without needing manual markers or specific patterns.

The main contributions are: a) the proposal of a novel multiscale intensity feature detection algorithm and b) the modification of the ICP algorithm to exploit point-to-line distances in a 2D color image and 3D point cloud registration.

The proposed registration algorithm can be applied to several data types, from range scans with color information to reconstructed meshes, CAD designs or scans without color or luminance information. The approach consists in detecting points with strong curvature and/or color changes, which will be later exploited by the ICP algorithm.

The performance of the presented registration algorithm is comparable to state-of-the-art methods [11] without the need of using specific pattern in the images. The registration obtained can be used for both increasing the information of the target point cloud or to add depth information to the color image. This registration can be obtained without a specific pattern (checkerboard) and may allow camera viewpoint localization in a virtual environment.

References

[1] R. Mishra and Y. Zhang, "A review of optical imagery and airborne lidar data registration methods," *The Open Remote Sensing Journal*, vol. 5, no. 1, pp. 54–63, 2012.

[2] Z. Kato and L. Tamas, "Relative Pose Estimation and Fusion of 2d Spectral and 3d Lidar Images," in *Computational Color Imaging*, ser. Lecture Notes in Computer Science, A. Trémeau, R. Schettini, and S. Tominaga, Eds. Springer International Publishing, Mar. 2015, no. 9016, pp. 33–42, dOI: 10.1007/978-3-319-15979-9_4.

[3] I. Stamos and P. E. Allen, "3-D model construction using range and image data," in *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings*, vol. 1, 2000, pp. 531–536 vol.1.

[4] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of LIDAR and optical images of urban scenes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2639–2646.

[5] B. O. Abayowa, A. Yilmaz, and R. C. Hardie, "Automatic registration of optical aerial imagery to a LiDAR point cloud for generation of city models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 106, pp. 68–81, Aug. 2015.

[6] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3d/2d registration methods for image-guided interventions," *Medical Image Analysis*, vol. 16, no. 3, pp. 642–661, Apr. 2012.

[7] I. Stamos, "Automated registration of 3d-range with 2d-color images: an overview," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2010, pp. 1–6.

[8] R. Pintus and E. Gobbetti, "A Fast and Robust Framework for Semi-automatic and Automatic Registration of Photographs to 3d Geometry," *Journal on Computing and Cultural Heritage*, vol. 7, no. 4, pp. 1–23, Feb. 2015.

[9] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, and R. Scopigno, "Fully Automatic Registration of Image Sets on Approximate Geometry," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 91–111, Mar. 2013. [Online]. Available: http://link.springer.com/10.1007/s11263-012-0552-5

[10] H. Kim and A. Hilton, "Influence of Colour and Feature Geometry on Multi-modal 3d Point Clouds Data Registration," in *2014 2nd International Conference on 3D Vision*, vol. 1, Dec. 2014, pp. 202–209.

[11] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 3936–3943.

[12] P. J. Besl and H. D. McKay, "A method for registration of 3-d shapes," vol. 14, no. 2, pp. 239–256.

[13] C. Choi, A. J. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1568–1575.

[14] M. Pauly, R. Keiser, and M. Gross, "Multi-scale Feature Extraction on Point-Sampled Surfaces," *Computer Graphics Forum*, vol. 22, no. 3, pp. 281–289, Sep. 2003.

[15] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[16] A. Pujol-Miro, J. Ruiz-Hidalgo, and J. Casas, "Multimodal registratioun software," https://imatge.upc.edu/web/resources/registration-images- unorganized-3d- point-clouds-using-contour-cues, 2017.