

Point Cloud Merging for Complete 3D Surface Reconstruction

V. Matiukas, D. Miniotas

Department of Electronic Systems, Vilnius Gediminas Technical University,
Naugarduko str. LT-03227 Vilnius, Lithuania, phone: +37068641949, e-mail: vilius.matiukas@el.vgtu.lt

crossref <http://dx.doi.org/10.5755/j01.eee.113.7.616>

Introduction

Extracting a point cloud of a real object from 3D surface measurement is a common task in various applications of computer vision, computer graphics, reverse engineering and etc. Usually, a single view is not enough to describe the whole object and multiple views of the surface are necessary. Typically the views are obtained from multiple 3D scanners or from a single scanner positioned at different locations and orientation. Each view is represented as partially overlapping dense point cloud. A problem of surface registration arises, i.e. the different views have to be located in a same coordinate system. In other words, the objective is to determine a rigid transformation for each partial view, in order to find the optimal alignment among all of them. In real cases, the point-to-point correspondences for the overlapping regions are calculated.

When only two-views are present, we are required to find the transformation that minimizes the distance between the two clouds. Two views registration is a well studied problem in the literature. The Iterative Closest Point (ICP) algorithm is the most common solution [1]. It iteratively revises the transformation, usually composed of scale, rotation and translation, needed to minimize the distance between the points of the two clouds. The point correspondences are extracted considering the closest point in the other view.

In most cases ICP and other (i.e. [2]) algorithms require specifying three or four corresponding points on each of the two views. To obtain a quality results corresponding points has to be selected with high accuracy, this is the main disadvantage of such kind of algorithms.

An automatic registration method was proposed in [3]. The algorithm works well on partially overlapping point clouds, but it fails in most of the cases when the point clouds of different views are similar (i.e. two views of a cylinder parts).

In this paper we present an automatic method of point clouds merging for complete surface reconstruction. This technique is applied during scanning procedure, as an output we get pre-aligned views of the point cloud. Afterwards fine registration is applied. Later results will

show that this technique works great as on largely overlapping point clouds as on partly overlapping point clouds.

Our method does not have any special requirements for point cloud acquisition hardware; stereo vision system [4, 5] can be used instead of 3d laser scanner.

3D laser scanning system

In point cloud acquisition process we used our own implementation of 3d scanning system presented in [6], however some minor modifications had to be made in order to meet our requirements.

Basic idea of the 3d scanning system (Fig. 1.) is quite simple. Laser ray is expanded to a line by cylindrical lens. If we make an assumption that laser source is a single point in 3D space, then we can state, that this point and laser line projected on some priori known reference geometry forms a laser plane. Point cloud of the 3D object is computed by calculating intersection of the laser plane with a boundary of the object.

The main problem is that we do not hold any priori information neither on a location of the laser plane, neither on a location of the object or camera.

Relationship between camera and laser plane can be obtained during calibration process.

First step in whole system calibration is to calculate intrinsic parameters of the camera. Camera calibration is

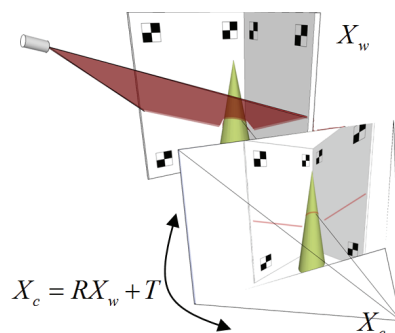


Fig. 1. Scanning system and its main components, like laser source, world (reference) coordinate system – X_w , camera coordinate system – X_c , and relationship between them – R, T

done using the ideal pinhole camera model [7] and some calibration pattern which geometry is known in advanced, for example chessboard pattern.

Intrinsic parameters of the camera depend only on optics of the camera and the camera itself. Once intrinsic parameters are calculated they do not change unless, focal length or aspect ratio is changed.

Next step is to determine cameras extrinsic parameters. Extrinsic parameters are rotation matrix (R) and translation vector (T). Extrinsic parameters define relationship between camera coordinate system and world (reference) coordinate system

$$X_c = RX_w + T, \quad (1)$$

here X_c – point coordinates in camera’s coordinate system, X_w – point coordinates in world (reference) coordinate system, R and T – rotation and translation between camera’s and world’s coordinate systems.

Computing matrix R and vector T , two calibration planes (patterns) aligned to form a right angle were used (Fig. 1).

Knowing cameras intrinsic and extrinsic parameters, point cloud can be calculated by finding intersection between laser plane and camera ray (Eq. 2). Such intersection gives us 3D coordinates of the object points in world coordinate system

$$P_w = \Lambda_c(x, y, 1) \cap \Pi_{laser}(t), \quad (2)$$

here P_w – object’s point coordinates in world coordinate system, Λ_c – camera ray, Π_{laser} – laser plane and t notes that position of laser plane changes in time.

Camera ray is defined as a ray from origin of the camera’s coordinate system passing thru point of laser line projection in image plain (Eq. 3). Laser line’s structure in image is detected with sub-pixel accuracy using shifted Gaussian kernels approach presented in [8]

$$\Lambda_c = p_0 + \lambda \vec{v}, \quad (3)$$

here Λ_c – camera ray, p_0 – camera’s center, coordinates, \vec{v} – direction vector of the ray, λ – positive scale factor.

Laser plain (Fig. 2.) can be calculated from its projection on image plane. Laser plane is defines by one of its points p_0 and a normal vector. Any point p belongs to the laser plane if and only if vectors $(p - p_0)$ and \vec{n} are orthogonal, such that

$$\Pi_{laser} = \{ \vec{n}^t (p - p_0) = 0 \}, \quad (4)$$

here, Π_{laser} – laser plane, \vec{n} – normal vector of the plane, p_0 – reference point of the plane, p – any other point on the plane.

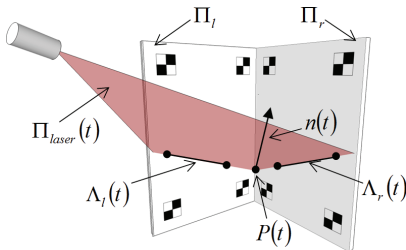


Fig. 2. Laser plane in world (reference) coordinate system

To find the point which belongs to the laser plane we calculate an intersection of two laser lines projected on

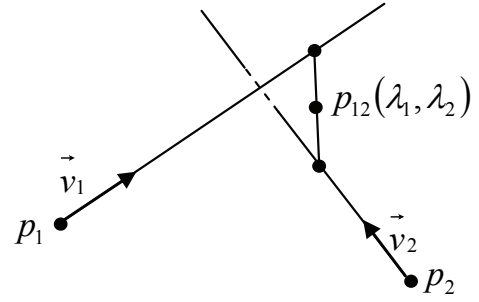


Fig. 3. The case when two lines are not intersecting. Approximate intersection as the point p which minimizes the sum of the square distances to both lines is calculated

known geometry of the background (the same which was used in cameras extrinsic calibration):

$$P(t) = \Lambda_r(t) \cap \Lambda_l(t) \quad (5)$$

and

$$n(t) = \Lambda_r(t) \otimes \Lambda_l(t), \quad (6)$$

here $P(t)$ – intersection point of two lines, $n(t)$ – normal vector, $\Lambda_r(t)$, $\Lambda_l(t)$ – laser lines projected on right and left reference planes.

Equations of lines can be found applying Hough transform for line detection and taking two points lying on each of the lines. At this point lines might not intersect at all (Fig. 3.), so we are trying to define approximate intersection as the point p which minimizes the sum of the square distances to both lines

$$f(p, \lambda_1, \lambda_2) = \|p_1 + \lambda_1 \vec{v}_1 - p\|^2 + \|p_2 + \lambda_2 \vec{v}_2 - p\|^2. \quad (7)$$

Main disadvantage of describes scanning system is that we cannot get the point cloud for complete surface representation. Instead we get multiply point clouds of the same object in world coordinate system with different angles of view which are not priori known. Merging such point clouds in one unique representation is a challenging task.

Point cloud merging using marker patterns

Most of the point cloud or 3D surface merging and registration algorithms require manually specify 3 or 4 corresponding points on each part of the different point cloud representing the same object. After that in most cases iterative closest points algorithm or its modifications are applied to calculate final point cloud representation.

This approach works fine if point clouds overlap largely. In our case point clouds should overlap only at their boundary regions.

First step in point cloud merging process is to find initial rotation and translation between different point cloud views of the same object. Our proposed method could handle scaling as well, but in current implementation this is left to fine registration step.

The solution used in our method is close to one used

in augmented reality applications [9], where marker tracking is used for real-time pose estimation.

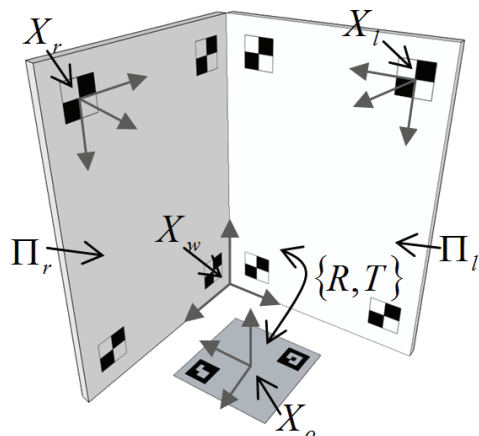


Fig. 4. Marker plane, with its own coordinate system is placed in world's coordinate system. Any point X_w from world coordinate system can be transformed to the point X_o in the markers (objects) coordinate system applying rotation and transformation

The idea is to place the object on marker plane and to transform all coordinates of the point cloud from world coordinate system, to markers plane coordinate system, which we will call objects coordinate system (Fig. 4.). Relationship (rotation R and translation T) between world and object coordinate system can be computed using POSIT algorithm [10].

During the scanning process object is rotated and translated together with the marker plane and every time we can track and evaluate these pose changes.

Our marker plane is composed of two markers, to avoid cases where one of the markers is covered by the object. Any combination of simple geometrical shapes can be used as marker as long as this combination is not rotation-invariant (Fig. 5.).

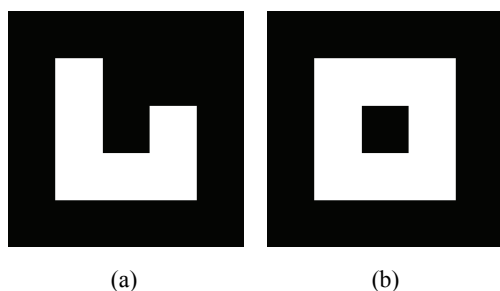


Fig. 5. An example of markers: (a) – marker which could be used for pose estimation and (b) – marker which could not be used for position tracking, because it is not rotation-invariant

As an output of such scanning process we get pre-aligned scanning views of different parts of the surface.

For fine registration step we chose algorithm presented in [11]. The algorithm iteratively minimizes a cost function considering all the views simultaneously.

Results

3D scanner and point cloud merging method was

implemented using C++ and OpenCV library. From technical point of view, web camera and line laser module are the only hardware used in described scanning system.

It is important that our laser scanner works in real time. Pre-aligned point clouds are obtained as an output of scanning process, which means, that only few iterations are needed for fine registration step.

To evaluate accuracy of the system, simple objects (cube, cylinder and pyramid) of different scales were scanned. The average error of the merged point cloud is close to 1%. The largest error rate is at the regions of sharp edges and corners of the object; here error rate varies about 2–3%.

Figure 6 shows two scans of a candle. Each point cloud is made of more than 200000 points. Figure 7 shows merged point clouds using simple ICP algorithm applied automatically and using our approach.

ICP algorithm tries to minimize point-to-point distance function. Our object has cylindrical shape, so point clouds of different object views are similar to each other and automatic ICP outputs wrong results. Our method evaluates relative position between these views and merges them in to correct representation.

Final point cloud was subsampled and filtered, to reduce level of noise, by applying octree subdivision.

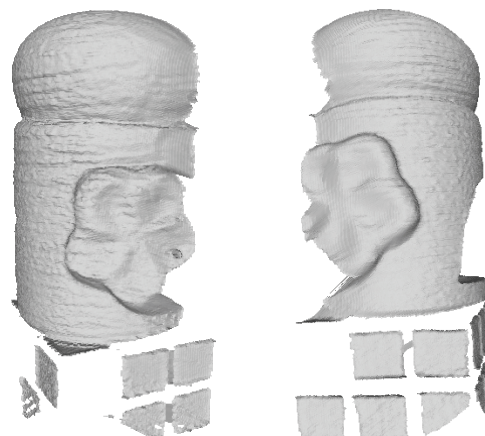


Fig. 6. Two point clouds of a candle, from different angles of view

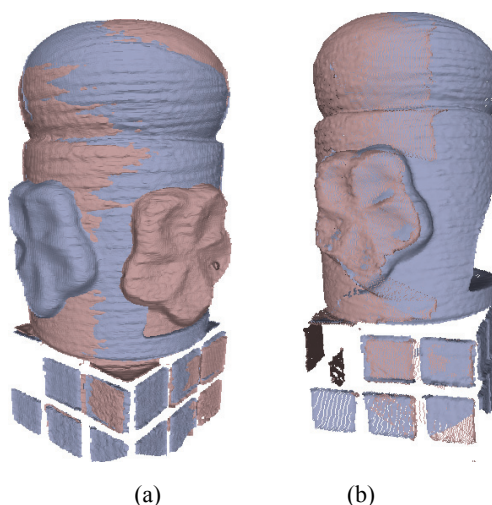


Fig. 7. Point cloud merging results: (a) – point cloud registration is done automatically applying ICP algorithm and (b) – point cloud merging is done using our approach

Conclusions

Method for point cloud merging, presented in this paper allows to calculate initial rotation and translation between point clouds of different view angles.

By tracking marker position, scanning and point cloud registration can be done in real time with high accuracy.

References

1. **Romero L., Felix C.** A Robust Iterative Closest Point Algorithm with Augmented Features // *Advances in Artificial Intelligence*. – Springer, 2008. – Vol. 5317. – P. 605–614.
2. **Aiger D., Mitra N. J., Cohen-Or D.** 4-points Congruent Sets for Robust Surface Registration // *ACM Transactions on Graphics*. – ACM, 2008. – Vol. 27. – No. 3. – P. 1–10.
3. **Liu Y.** Automatic registration of overlapping 3D point clouds using closest points // *Image and Vision Computing*. – Citeseer, 2006. – Vol. 24. – No. 7. – P. 762–781.
4. **Surgailis T., Valinevičius A., Eidukas D.** Stereo Vision based Traffic Analysis System // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2011. – No. 1(107). – P. 15–18.
5. **Serafinavičius P., Sajauskas S.** New Generation User Interface based on Hand Tracking in 3D Space // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 3(99). – P. 69–72.
6. **Winkelbach S., Molkenstruck S., Wahl M. F.** Low-Cost Laser Range Scanner and Fast Surface Registration Approach // *Pattern Recognition*. – Springer, 2006. – Vol. 4174. – P. 718–728.
7. **Heikkila J., Silven O.** A four-step camera calibration procedure with implicit image correction // In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. – 1997. – P. 1106–1112.
8. **Matiukas V., Miniotas D.** Detection of Laser Beam's Center-line in 2D Images // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2009. – No. 7(95). – P. 67–70.
9. **Yeol J. L., Seo D. W., Rhee W. G.** Tangible authoring of 3D virtual scenes in dynamic augmented reality environment // *Computers in Industry*. – Elsevier, 2011. – Vol. 62. – No. 1. – P. 107–119.
10. **Dementhon D. F., Davis L. S.** Model-based object pose in 25 lines of code // *International Journal of Computer Vision*. – Kluwer Academic Publishers, 1995. – Vol. 15. – No. 1–2. – P. 123–141.
11. **Toldo R., Beinat A., Crosilla F.** Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework // *3DPVT 2010 Conference*, 2010.

Received 2011 04 27

V. Matiukas, D. Miniotas. Point Cloud Merging for Complete 3D Surface Reconstruction // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 7(113). – P. 73–76.

This article addresses the problem of reconstructing 3D surfaces from unorganized point sets (also known as point clouds). This issue is common to many different areas of science and engineering, including computer graphics and computer vision. Unorganized point sets can be acquired in different ways, e.g., using laser scanners, computer tomography, magnetic resonance imaging, multi-camera vision systems, etc. This paper presents a method for point cloud merging that allows calculating the initial rotation and translation between point clouds of different viewing angles. By tracking the marker's position, scanning and point cloud registration can be done in real time and with high accuracy. Ill. 7, bibl. 11 (in English; abstracts in English and Lithuanian).

V. Matiukas, D. Miniotas. Nestruktūrizuoto taškų rinkinio sujungimas vientisam paviršiui rekonstruoti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 7(113). – P. 73–76.

Šiame straipsnyje nagrinėjama trimačių paviršių rekonstravimo pagal nestruktūrizuotų taškų rinkinius problema. Ši problema būdinga daugeliui mokslo ir technikos sričių, tarp jų ir kompiuterinei grafikai ir regai. Nestruktūrizuotų taškų rinkinius galima generuoti įvairiais būdais, pavyzdžiui, naudojant lazerinius skenerius, kompiuterinę tomografiją, vaizdavimą naudojant magnetinį rezonansą, daugiakamerines regos sistemas. Straipsnyje aprašytas nestruktūrizuotų taškų rinkinio sujungimo metodas, leidžiantis apskaičiuoti pradinį sukimąsi ir slinktį tarp įvairiais stebėjimo kampais išsidėsčiusių taškų rinkinių. Sekant žymeklio padėtį, galima skenuoti objektus ir registruoti nestruktūrizuotų taškų rinkinius realiu laiku bei dideliu tikslumu. Il. 7, bibl. 11 (anglų kalba; santraukos anglų ir lietuvių k.).