

Use of point clouds for video surveillance system cover zone imitation

Aleksandr Mezhenin¹[0000-0002-7150-9811], Vera Izvozchikova²[0000-0002-8707-9510] and
Viktoriiia Ivanova¹[0000-0001-8247-7344]

¹ITMO University, Kronverksky Ave. 49, St. Petersburg, Russia
a.v.mezhenin@gmail.com, vikaaiiv17@gmail.com

²Orenburg State University, Ave. Pobedy 13, Orenburg, Russia
viza-8.11@mail.ru

Abstract. The article is devoted to assessing the coverage areas of projected video surveillance systems, determining the optimal number of cameras, identifying the so-called “blind zones” at the design stage. For this purpose, it is proposed to use a three-dimensional modeling system in which a polygonal model of a video surveillance scene is created. The resulting model is converted from polygons to a cloud of points. Using different camera locations, we get different coverage areas. During the experiments, the objects of observation and surveillance cameras, whose location changed, were modeled. The resulting 2.5-d surfaces, represented as a cloud of points, are transferred to a common coordinate system and compared. The comparison results are shown as a heat map. Experimental modeling was carried out using specially created objects in the environment of Autodesk 3ds Max and CloudCompare.

Keywords: Video surveillance systems, Designing, Virtual modeling, Point clouds, Thermal maps.

1 Introduction

The main purpose of video surveillance systems is to provide a visual inspection of an object equipped with them. When designing, it is necessary to take into account the combination of factors that increase the efficiency of the developed video surveillance systems. One of the most important factors is the coverage zone of the observed area with cameras and the absence of so-called “blind zones”. Their appearance depends on many factors: the characteristics of the cameras being installed, their location, the number, used data formats [16].

Obviously, the more cameras, the easier it is to place them so as to get the most complete coverage of the observed scene, but this increases the cost of the

developed system. The proposed computer simulation of the coverage area in the form of a cloud of points of the observed surfaces of the 3D space, according to the authors, will allow obtaining preliminary estimates of the effectiveness of the developed video surveillance systems [16] at the design stage, which will improve design solutions. A mathematical model is proposed, as well as algorithms for modeling a cloud of points and obtaining heat maps for comparing coverage areas.

2 Analog overview

As analogues, we consider special software designed for the development of video surveillance systems. The first analogue is a program for the design of video surveillance systems-JVSG [13], which allows you to set the number and location of video surveillance cameras, to calculate the developed system, to determine the viewing area, to place the camera on an existing or created from scratch plan of the premises, to place obstacles in three-dimensional space to identify blind spots. As a result of calculation there is a project that includes drawings of zones of video cameras, the block diagram and the explanatory note.

The second equivalent is the DlinkSurveillanceFloorPlannerPRO [14] software. This program also helps to create a project for deploying cameras for video surveillance systems. It allows you to select cameras, place them on a floor plan and visually displays coverage areas.

Both of the reviewed programs allow to adjust the camera settings, import a floor plan into a program, determine some camera visibility areas. Thus, they allow to evaluate the effectiveness using such system capabilities as the choice of cameras with specific settings, their location and cost. However, coverage areas are presented in a form of plans, which, according to the authors, does not allow to evaluate them fully.

Currently, widespread method of representing objects of space in the form of clouds of points. A point cloud is a collection of vertices in three-dimensional or two-dimensional space, obtained from an object and representing the surface of this object. The data for displaying a cloud of points can be obtained by scanning 3D objects with special devices or by using computer simulations. In addition to coordinates, points may contain additional information, such as colors or normal's.

There are several packages available for visualizing point clouds: MeshLab, CloudCompare and Point Cloud Library (PCL), which can be integrated into OpenCV. At this stage, the program CloudCompare [17] was used, which allows

editing and processing 3D point clouds. CloudCompare supports import and export of files with the extension OBJ, E57, ASCII, PLY, etc. Also, the program allows you to create cameras, place them on the workspace and display visibility areas for them.

3 Mathematical envisioning of a surface comparison problem

For a three-dimensional one-sided surface, constructed from the point cloud S , which contains N points, there exists a coordinate system $Oxyz$, which is constructed in such a way, that the point cloud point $S = \{(x^i, y^i, z^i)\}_{i=1}^N$ can be seen as the function $z = f(x, y)$, built on the discrete point range $\{(x^i, y^i)\}_{i=1}^N$.

Here is an example. Two one-sided surfaces, S_1 and S_2 are formed respectively from the point clouds $\{(x_i^1, y_i^1, z_i^1)\}_{i=1}^{N_1}$, $\{(x_i^2, y_i^2, z_i^2)\}_{i=1}^{N_1}$ in the $Oxyz$ coordinate system so that S_1 and S_2 will be projected onto the surface Oxy in E^3 . It is required to find a measure for comparing the surfaces S_1 and S_2 , that will satisfy the semi metric axioms and calculate it [1].

As a quantity that will show the difference between the clouds we will use $d(x,y)$. This is the Euclidean distance in point clouds between two points, which is calculated by using the following formula:

$$d(x, y) = \sqrt{\sum_{i=1}^N (x_i, y_i)}. \quad (1)$$

We will denote the range of points in both clouds, S_1 and S_2 , that are closest to each other - $\{(s_i^1, s_i^2)\}_{i=1}^N$. E is the error margin. Then, E is lowered gradually to the required amount.

$$E = \frac{1}{N} \sum_{i=1}^N d(s_i^1, s_i^2) \rightarrow \min \quad (2)$$

4 Algorithm for comparing object surfaces

The iterative closest point (ICP), algorithm is the most fitting one for comparing two different point clouds of the same given object. This algorithm is based on the minimization of the average distance between two point clouds. A special procedure is used to minimize the distance between the two point clouds in question. The algorithm can be used to compare point clouds based on the same object, which has had some changes made to it. Also it works if the object was scanned from a different angle. In either case, the point clouds must have similar areas, that cover each other. This algorithm moves and adjusts these

corresponding areas until they overlap to the extent denoted by E – the margin of error.

For the algorithm to start functioning an initial evaluation of the similarity between the clouds is required, which, over time, becomes more accurate, because of the repetition of the cycle. The two point clouds - $\mathfrak{n} S_1$ and $\mathfrak{n} S_2$ will gradually become one cloud. The maximal accuracy will be achieved if the position of the points in space was similar initially and the similar areas of the clouds were large. The pairs of closest points, which are located in the similar areas of the clouds, must be picked in a certain quantity, which should not exceed a set limit. If the given similar areas of the clouds contain too many pairs of points, then the accuracy of the algorithm's work becomes much lower.

During the process, the error margin should be lowered as much as possible:

$$E \rightarrow \min \quad (3)$$

The offered algorithm of solving the given problem includes the following steps:

1. Find the corresponding closest point (s_i^1, s_i^2) , such that

$$S_2(s^2) = |s_i^2| \quad (4)$$

$$(s^2) = \underset{s_i^1 \in S_1, s_i^2 \in S_2}{\operatorname{argmin}} S(s^2) \quad (5)$$

s_i denotes the given point in the space \mathbb{R}^3 , for which we must find the shortest possible distance s_i^1 to the range S_1 . This means that we need to find such a point, s_i^1 , that will make the functional small as possible (5).

2. Find the most optimal way to transform the first point cloud, by rotating it or changing the points' position, using the least squares method. This step should be repeated until the error margin E is satisfied. We shall get t – the vector of transformation and the rotation matrix R , which will deliver minimum function:

$$J(R, t) = \sum_{i=1}^N |(Rs_i^1 + t) - s_i^2|_2^2 \quad (6)$$

$$(R, t) = \underset{R \in SO(3), t \in \mathbb{R}^3}{\operatorname{argmin}} J(R, t), \quad (7)$$

the matrix R is an objective orthogonal 3×3 matrix whose determinant equals 1, therefore it belongs to the orthogonal rotation group $S(3)$;

the vector t displays the translation of the range of points along the vector $[x, y, z]^T$;

$\|\cdot\|_2^2$ is the square of the normal l_1 or the square of the Euclidean distance between two given points.

3. Adjustment of the point cloud being transformed by using the rotation matrix and translation vector that were found in the above mentioned steps. Thus we will get a new point cloud.

$$s_i^1 = R s_i^1 + t \quad (8)$$

4. Repeat steps 1 through three.

The work of the algorithm will stop only then, when the amount of similarity, imposed by E – the error margin is met. The amount of difference between the positions of the points in clouds S_1 and S_2 will be the quantity we sought to find. This amount can be also displayed as a percentage of the original distance between the points.

5 Experimental results

Figure 1, (a) shows the point cloud S_1 , which was constructed from a three-dimensional scene in the cloud processing software, CloudCompare. Figure 1 (b) shows another point cloud, which imitates a scan of the same object from a different angle.

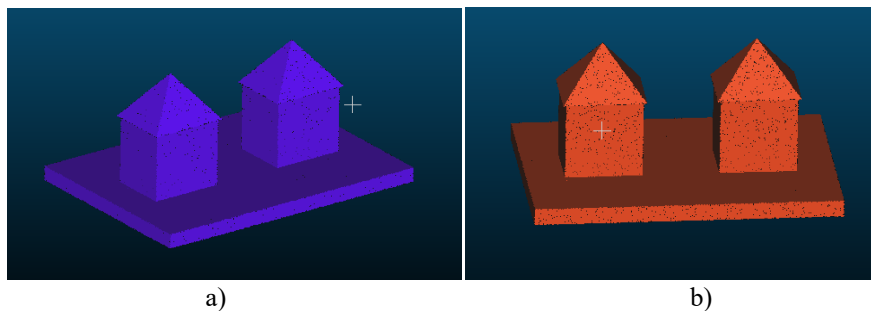


Fig.1. Point cloud S_1 , point cloud S_2 .

The point cloud S_2 was generated by applying rotating and translation operations, based on the matrix, illustrated below, to the original point cloud S_1 .

$$\begin{pmatrix} 0.86 & 0.5 & 0 & 60 \\ -0.5 & 0.86 & 0 & -10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This transformation matrix can be seen to contain two parts: a rotation matrix

$$R = \begin{pmatrix} 0.86 & 0.5 & 0 \\ -0.5 & 0.86 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

And a translation vector

$$t = \begin{pmatrix} 60 \\ -10 \\ 0 \end{pmatrix}.$$

On Figure 2 we can see the relative position of clouds S1 and S2, both for a view from a random angle (see Fig. 2, (a)) and for a view from the top (see Fig. 2, (b)).

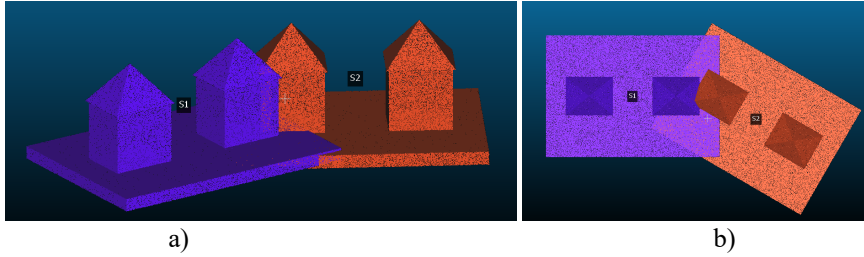


Fig.2. Point clouds S1 and S2, view from a random angle (a), view from the top (b).

Figure 3 displays a graphic rendering of the distance of point cloud S1 relative to point cloud S2. This distance is calculated as the distance between corresponding points, both for random angle view (see Fig. 3, (a)) and view from the top (see Fig. 3, (b)). The areas that are colored blue correspond to the minimal distance between similar points on the two surfaces, while areas colored red denote the maximal distance. The ratio of distance to color is shown on the scale in the right part of the figure.

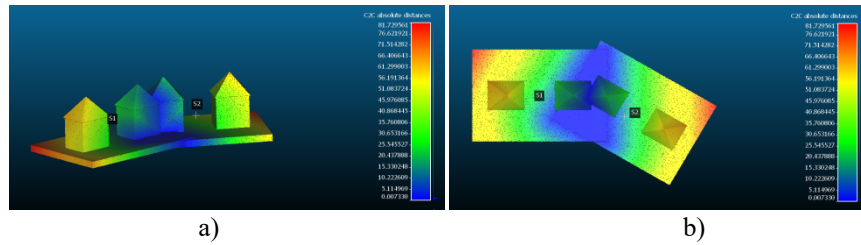


Fig.3. Distance between point clouds S1 and S2, random angle view (a), top view (b).

Initial average distance between the points of cloud S1 and cloud S2 was 26.231, the standard difference was 21.597; S2 relative to S1 – 26.894, standard difference – 22.061.

Shown on Figure 4 are the results of applying the point cloud registration operation, that utilizes the ICP algorithm in the cloud processing program CloudCompare for an arbitrary view (see Fig. 4, (a)) and view from the front (see Fig. 4, (b)).

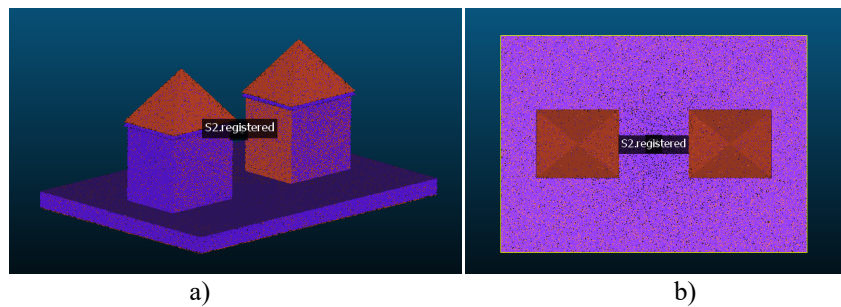


Fig.4. The result of applying the ICP algorithm for point clouds S_1 and S_2, arbitrary view (a), top view (b).

The calculated distance of the point cloud S1 relative to the point cloud S2 and from point cloud S2 relative to point cloud S1 is displayed on Figure 5, both from an arbitrary view (see Fig. 5, (a)) and for a top view (see Fig. 5, (b)).

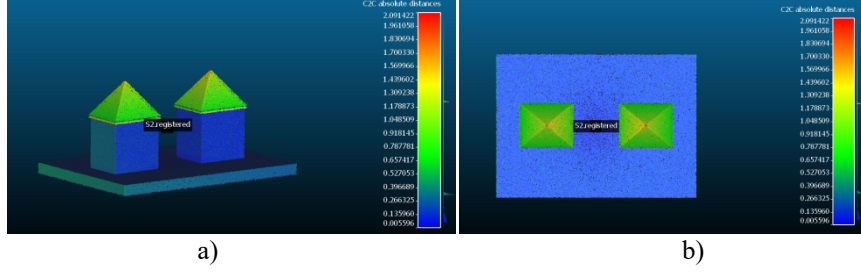


Fig.5. Distances between point clouds S1 and S2 after registration by using the ICP algorithm, rendered from a random angle view (a), and top view (b).

The average distance from point cloud S1 to point cloud S2 is 0.315, standard difference is 0.32; S2 relative to S1 – 0.322, standard difference is 0.325.

As a result of the execution of the ICP algorithm on the point cloud S_2 the following matrix was used:

$$R = \begin{pmatrix} 0.865 & -0.502 & 0.000 \\ 0.502 & 0.865 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{pmatrix}$$

And this vector was applied:

$$t = \begin{pmatrix} -56.917 \\ -21.375 \\ 0.878 \end{pmatrix}$$

The results fit the allowed margin of error.

6 Conclusion

The proposed approach to the modeling of coverage areas and their visualization will allow obtaining more effective design solutions for video surveillance systems. The possibility of using universal 3D modeling and data processing systems for solving a specific applied problem is shown. The results of the comparison, presented in the form of a heat map, make it possible to estimate the degree of coverage of the observation zone and reveal the presence of dead zones. These provisions are the basis for further ongoing research in this area.

Acknowledgments. The research has been supported by the RFBR, according to the research projects No. 17-07-00700 A.

References

1. Dyshkant, N.: Measures for Surface Comparison on Unstructured Grids with Different Density. *Lecture Notes in Computer Science: Discrete Geometry for Computer Imagery*, Vol. 6607, pp. 501–512 (2011).
2. Dyshkant, N.: Disparity Measure Construction for Comparison of 3D Objects' Surfaces. *Inproceedings of the Workshop IMTA, Lisbon, Portugal: INSTICC Press*, pp. 43–52 (2009).
3. Dyshkant, N.: An algorithm for calculating the similarity measures of surfaces represented as point clouds. *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, vol.20, No. 4, pp. 495–504 (2010).
4. Tomaka, A.: The application of 3d surfaces scanning in the facial features analysis. *Journal of edical Informatics and Technologies*, pp. 233–240 (2005).
5. Szymczak, A., Rossignac, J., King, D.: Piecewise regular meshes: Construction and compression. *Graphical Models*, vol. 64, No. 3-4, pp. 183–198 (2002).
6. Stepanyants, D.G., Knyaz, V.A.: PC-Based Digital CloseRange Photogrammetric System for Rapid 3D Data Input in CAD Systems. *International Archives of Photogrammetry and Remote Sensing*, vol. 33, part B5, pp. 756–763 (2000).
7. Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3D geometry. In *proceedings ACM SIGGRAPH*, pp. 560–568 (2006).
8. Liu, Y., Rodrigues, M.A.: Geometrical analysis of two sets of 3D correspondence data patterns for the registration of free-form shapes. *J. Int. and Rob. Systems*, vol. 33, pp. 409–436 (2002).
9. Knyaz, V.A., Zheltov, S.Yu.: Photogrammetric Techniques for Dentistry Analysis, Planning and Visualisation. In *proceedings ISPRS Congress Beijing 2008, Proceedings of Commission V*, pp. 783–788 (2008).
10. Koidis, P., Patias, P., Tsioukas, V.: 3D Visualization of Dental Data for Virtual Treatment Planning. In *proceedings ISPRS Congress Istanbul 2004, Proceedings of Commission V*, pp. 996–1001 (2004).
11. MathWorks - Makers of MATLAB and Simulink, URL: <https://www.mathworks.com/>.
12. Unity, URL: <https://unity3d.com/ru>.
13. JVSG: CCTV Design Software, URL: <http://www.jvsg.com/>.
14. D-Link Surveillance Floor Planner, URL: <http://tools.dlink.com/intro/sfp/>.
15. CloudCompare - Open Source project, URL: <https://www.danielgm.net/cc/>.
16. Shardakov, V.M., Parfenov, D.I., Zaporozhko, V.V., Izvozchikova, V.V. Development of an Adaptive Module for Visualization of the Surrounding Space for Cloud Educational Environment In: *11th International Conference Management of Large-Scale System Development, MLSD* (2018).