COLOR-BASED SEGMENTATION OF POINT CLOUDS

Qingming Zhan[a, b, *], Yubin Liang[b, c], Yinghui Xiao[a, b]

[a] School of Urban Design, Wuhan University, Wuhan 430072, China
[b] Research Center for Digital City, Wuhan University, Wuhan 430072, China
[c] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
qmzhan@whu.edu.cn; lyb.whu@gmail.com; yhxiaoitc@126.com

Commission III, WG III/2

KEY WORDS: Laser scanning, Point cloud segmentation, Reverse engineering, Building reconstruction

ABSTRACT:

Segmentation is one of the most fundamental procedures for the automation of point cloud processing. The methods based on geometrical derivatives such as curvature and normals often lead to over-segmentation and even failure when used to segment point clouds of geometrically-complex architectures. In this paper we present a point cloud segmentation algorithm based on colorimetrical similarity and spatial proximity. The algorithm contains region growing, region merging and refinement processes. The region growing process uses kd-tree to search the k-nearest neighbors of each seed point. The resulting regions are then merged and finally refined on the basis of colorimetrical and spatial relation. In each of the process, we developed different criteria corresponding to the different tasks to carry on the segmentation. The algorithm requires a small number of manually set parameters which are used to keep balance between under- and over-segmentation. The experiments of the presented algorithm on a point cloud of Chinese ancient architecture show its effectiveness. The segmented regions can be used to reconstruct 3D models of different parts of the architectures.

1. INTRODUTION

1.1 Problem statement

Segmentation is the points labeling process for point cloud data and after the process points sharing similar properties are grouped together. Often it is fundamental work for 3D reconstruction of scanned objects. For the problem of architectural reconstruction the unstructured point cloud data is often acquired using terrestrial laser scanners in consideration of more façade details. Nowadays laser scanners can simultaneously record 3D coordinates and RGB color values of each scanned point. Much research has been done to explore the geometrical properties of the scanned objects using the acquired 3D coordinates. But the color values are insufficiently explored compared with the exploration of 3D coordinates and also with its use in color image processing. In this paper we perceive point cloud of an object as certain spatial distribution pattern of finite colorful points. We notice that different artificial objects often come with different color distribution patterns mainly because of their different functionalities (materials) and people’s aesthetic requirements. Based on these observations we combined the color segmentation method from image processing with the region growing method from point cloud segmentation and devised the presented algorithm to segment point clouds and applied in a complicated architectural building.

1.2 Related works

Many algorithms have been devised to segment point clouds and images. Point cloud segmentation methods mainly consider geometric properties of scanned objects and exploit geometrical derivatives such as curvature and normals to group points. Color segmentation of images use colorimetrical difference measurements to separate pixels and often follow a partitioning-merging process with a consideration of 2D spatial proximity (Tremeau and Borel 1997).

Segmentation methods of unstructured point clouds can roughly be divided into two categories: indirect approaches that firstly define some criteria like spatial proximity and geometric derivatives then group points based on them in a gradual progress and direct approaches that directly estimate the geometric parameters using Generalized Hough Transformation (GHT). For a historical reason, many algorithms arise from range image segmentation.

Hoover et al. (1996) made detailed comparisons of four plane-based segmentation algorithms presented separately by four research groups. Vosselman et al. (2004) gave an overview over different techniques for the extraction of surfaces from point clouds including scanline segmentation, surface growing and connected components in voxel space and present some variation of these methods to find planar surfaces. Pu and Vosselman (2006) adopted the planar surface growing algorithm to segment building façade and automatically extract building features based on feature constraints such as size, direction and topology derived from convex hulls of segmented regions. Rabbani et al. (2006) concluded the range image segmentation algorithms into three main approaches: edge-based segmentation, surface-based segmentation, scanline-based segmentation and discussed them separately. A region growing algorithm based on smoothness constraint is developed to find connected smooth surfaces in industrial installations. The algorithm estimate surface normals for each point in unstructured point clouds and use k nearest neighbors (KNN) or fixed distance neighbors (FDN) to do normal estimation as well as to select the candidate points to grow. Residuals of plane fitting and normal variation are defined as similarity measurements which are used to grow regions. The thresholds can be adjusted to achieve a balance between under- and over-segmentation.

Many man-made objects can be described by regular shapes like planes, cylinders and spheres. Because these shapes can be described by only a few parameters, it is feasible to directly extract shapes from point clouds. Direct approaches fall into this category which accomplish segmentation and recognize surface types at the same time (Marshall et al. 2001). These approaches exploit the duality of the object space and parameter space and map all the points to parameter space. The detection of expected geometric shape can be transformed to a voting procedure in a discrete parameter space. The selected parameters can be used to reconstruct shapes in object space. Maas and Vosselman (1999) used Plane-based segmentation method to detect planar roof faces from raw laser altimetry data. Vosselman and Dijkman (2001) used the same method and combined it with ground plans to reconstruct 3D models of building roofs. Rabbani and van den Heuvel (2005) presented a cylinder detection algorithm which broke up cylinder-detecting GHT into two steps: orientation estimation and position and radius estimation. The sequential breakdown reduced the high computational space and time complexity while remained robust. Rabbani et al. (2007) integrated modeling and global registration of point clouds. The approach firstly modeled each scan using GHT then used the corresponding models to register the scans. When the point clouds of the industrial installations contain enough geometric information for automatic detection and fitting the parameterized shapes, this approach simultaneously modelled and registered different scans while avoiding the accumulation of errors.

1.3 Problems with existing methods

Many point clouds segmentation methods are developed to segment industrial installation scanning data whose real world counterparts usually have relatively regular and simple geometric shapes. But these geometrical derivatives-based methods easily lead to over-segmentation when used to segment point clouds of geometrically-complex architectures. Although several algorithms have been developed to find planes in point clouds, it is not enough to group points using coplanar constraints because geometrically-complex parts of architectures (e.g. flying eaves) often consist of many spatially-connected planar patches which are not coplanar as a whole. And the situation is complex to segment point clouds using curved regions constraints. Even if the patches have been successfully segmented, much manual editing work is still needed to assemble them to reconstruct 3D model of an architectural part. In addition, it is also hard to separate doors from a piece of wall using plane-based segmentation method when they are almost coplanar. Furthermore, usually we cannot provide enough geometric information to segment architectural point clouds because of unavoidable occlusion and this makes it difficult to segment data using geometry-based methods.

While images preserve the color information of objects, they lose depth information in the projection process from 3D to 2D. The region growing process based on 2D spatial proximity may fail to include the right pixels because images cannot always ensure that neighboring pixels in images correspond to the neighboring objects in real world. Although many global clustering methods have been used to solve this problem, they easily lead to false segmentation in a complex environment.

1.4 Motivation

On the basis of the above observations, we developed the presented algorithm out of the following motivation:

1. We assume a raw unstructured 3D point cloud as the input to the algorithm which has only 3D coordinates and RGB values.
2. We use only colorimetrical difference as discriminator to segment the data as it has a really clear meaning.
3. The algorithm should use as few parameters as possible and allow users to adjust the parameters to achieve a balance between under- and over-segmentation.
4. We assume the segment should be locally connected and the algorithm should segment the connected and colorimetrically similar parts as a whole no matter what their shapes are e.g. colorimetrically similar facades of a building perpendicular to each other at the corners.

2. SEGMENTATION ALGORITHM

The segmentation method has three steps: region growing, region merging and refinement. The algorithm1 details region growing process and algorithm 2 details region merging and refinement process and the two algorithms are explained below.

2.1 Region growing

The region growing process starts from growing the first point of the input point cloud. When the process meets an unlabeled point, a new region is initiated, the current unlabeled point is added to the region and pushed into a points-to-grow stack *Points*. Then each time *Points* pops one point and finds its k nearest neighbors within a given distance *TD* and grows the current region on the basis of colorimetrical similarity. The colorimetrically similar points are added to the current region and added to *Points* and the current growing process terminates when *Points* is empty. The procedure iterates reading in an unlabeled point, grows a new region and terminates until all the points are labeled. The output of this process is roughly segmented regions to be used in region merging and refinement.

Algorithm 1 Region growing
Inputs: Point cloud={**P**}, neighbor searching function KNN(.), colorimetrical difference calculation function $CD(C_1,C_2)$, point-point colorimetrical similarity threshold TPP, TNN which indicates the number of neighbors to search when using KNN(.) on a point and threshold TD for distance between a point in stack Points and its neighbors.
Initialize: Region array {**R**}←Φ
while there is any point in {**P**} that hasn't been labeled
  push an unlabeled point into stack Points
  initialize a new region R_c and add current point to R_c
  while stack Points is not empty
    pop Points' top element Tpoint
    for each point p in {KNN_{TNN}(Tpoint)}
      if p is labelled or Dis(p,Tpoint)>TD
        continue
      if CD(Tpoint,p)<TPP
        push p into Points
        add it to R_c and update the statistics of R_c
      end if
    end for
  end while
  add R_c to {**R**}
end while
Return {**R**}

The neighbor searching technique and the definition of colorimetrical difference measurement are explained below.
K nearest neighbors (KNN) We use KNN instead of fixed distance neighbors (FDN) because it's an adaptive method which automatically adjusts the size of neighborhood of a point according to the local point density and avoids degenerate cases (e.g. no neighbors are found within a given distance from a point when using FDN). This ensures the local connectivity when the segmentation proceeds. We will also use KNN to search neighbors of a region in region merging and refinement process. We use Euclidean distance metric because it reflects the positional relationship of two objects in 3D space. Search for KNN can be implemented using kd-tree (Bentley 1975; Press et al. 2007).

Colorimetrical difference measurement

We use Euclidean distance to measure colorimetrical difference in RGB space. The colorimetrical difference function is defined as:

$$CD(C_1, C_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \tag{1}$$

Where C1, C2 represent color vectors of two objects which can be points and regions. RGB values of points can be read directly from raw data while RGB values of regions are from statistics of regions defined in region structure as below:

```
Structure Region {
int label; // label of the region
vector<int> ptList; // point list of the region
        double r, g, b, σ; // statistics of the region
};
```

Where r, g, b are average RGB values of the points belonging to the region.

2.2 Region merging and refinement

This process merges and refines the roughly segmented regions output from the region growing process. We use region-region threshold *TRR* for colorimetrical similarity if two neighboring regions are colorimetrically similar we consider that the two regions are homogeneous and record them in a homogeneous region lists set {**H**} which records homogeneous regions in the same list.
The region merging process is analogous to the region growing process. For each region R_i, the process firstly searches it in {**H**} and ensures that the region has been in some list in {**H**}. Then search the neighbors of R_i. The searching can be realized by doing *KNN(.)* on the points in R_i. To ensure the correctness of neighbor searching, we use a distance constraint *TD2* and number constraint *TNN2*. The searched neighboring regions are compared with R_i on the basis of colorimetrical similarity. The similar neighbors are added to the list which contains R_i. After the merging process, all the regions in {**R**} have been classified into different lists in {**H**}, we merge the regions in the same lists in {**H**} and get merged regions {R'}. Finally, we refine {**R**'} using a threshold *Min* for the minimal size of an acceptable region. The regions which has less than *Min* points dare merged to their neighboring regions. The merging process is carried out in the following way. For each point P_i in the region, find its nearest neighboring point which belongs to another region R_j and add P_i to R_j.

Algorithm 2 Region merging and refinement
Inputs: Roughly segmented regions {**R**} from Algorithm 1, region-region colorimetrical similarity threshold TRR, minimal size threshold for a region Min, distance threshold TD2 and number threshold TNN2 for neighbor searching using KNN(.) on the points of a region, colorimetrical difference calculation function $CD(C_1,C_2)$.
Initialize: Homogeneous region lists set {**H**}←Φ
for each region R_i in {**R**}
  if R_i is not in {**H**}
    create a new list to record R_i
  end if
  for each region R_j in $\{KNN_{TNN2,TD2}(R_i)\}$
    if $CD(R_i,R_j)<TRR$
      if R_j is in {**H**}
        continue
      else
        add R_j to the list which contains R_i
      end if-else
    end if
  end for
end for
merge all the regions in the same list in {**H**} and get {**R**'}
for each region R_i in {**R**'}
  if sizeof(R_i)<Min
    merge R_i to its nearest neighbors
  end if
end for
Return the merged and refined {**R**'}

3. RESULTS AND DISCUSSIONS

The presented algorithm was applied to point cloud data acquired from an ancient Chinese architecture which contains 498,890 points. The data was collected during one scan. The raw data set is displayed in Figure 1. The global segmentation results using plane-based segmentation algorithm and presented algorithm are separately shown in Figure 2a and Figure 2b respectively. The central idea of plane-based segmentation algorithm comes from Rabbani et al. (2006). The algorithm first does local plane fitting using KNN method and calculates fitting residuals for each point. Then a region growing process is carried out in consideration of both local connectivity and surface smoothness. The thresholds in this plane-based segmentation experiment are set as: plane fitting residual threshold r=0.05m, number of neighbors to search k=30, angle variation threshold θ=25 degree, and finally we get 1582 segmented regions from this process. To get a balance between under- and over-segmentation with a preference to over-segmentation, the thresholds for our presented algorithm are set as: *TNN*=30, *TPP*=35, *TD*=0.3m, *TRR*=10, *TNN2*=100, *TD2*=0.5m, *Min*=10. The segmentation results 120 different regions. From the comparison between Figure 2a and Figure 2b, we can see that although local smoothness in terms of angle variation threshold and relatively large plane fitting residual threshold have been taken into account the plane-based algorithm tends to over-segment the geometrically complex parts such as eaves and pillars. While our algorithm segments the colorimetrically similar objects as a whole and the details are shown in Figure 3, Figure 4a and Figure 4b. In Figure 3 we see that the pillars, walls, doors, facades of the foundation platform and some non-architectural objects are successfully segmented. The walls and doors are basically segmented as a whole. And the pillars are distinguished from the roof above and the foundation platform beneath and the walls behind as shown in Figure 3. There are two urns in the scan and they are symmetrically standing beside the steps. The segmented urns are shown in Figure 4a and Figure 4b. In Figure 4a we see that the urn (yellow) is well scanned and the point density is relatively high while the other urn (blue) in Figure 4b is also well segmented although it has spare point data. The steps are over-segmented and there are some objects (flowers) on the steps are mis-segmented. This is mainly because the presented algorithm doesn’t take into account complex lighting and shading factors from the environment and the thresholds are adjusted to relative large values to get a balanced result. The mis-segmentation can be solved using even smaller thresholds but this will lead to more over-segmented regions for the global segmentation. As for each segmented region, it is possible to further adjust the thresholds to improve the segmentation effect based on the global result. In Figure 5, the number of neighbors to search is increased to 70 and keeps other parameters unchanged, finally we get the result in which the roof and pillars are segmented as a whole.

Figure 1: The raw data of the testing building

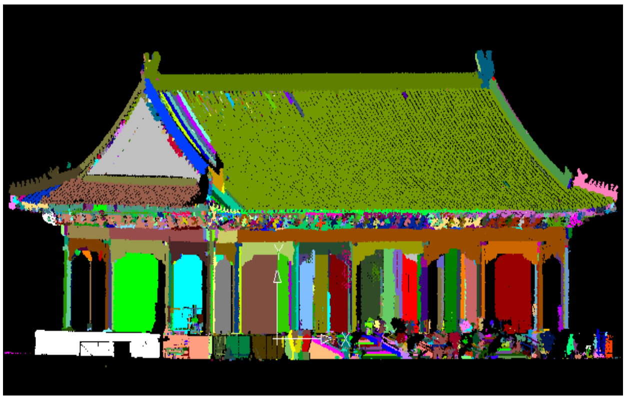

Figure 2a: Segmented result using plane-based algorithm

Figure 2b: Segmented result using presented algorithm

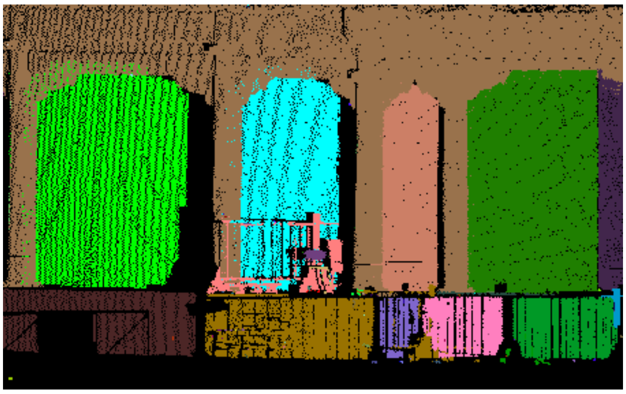

Figure 3: The segmented pillars, walls, facades of the foundation platform and a shelf standing on it

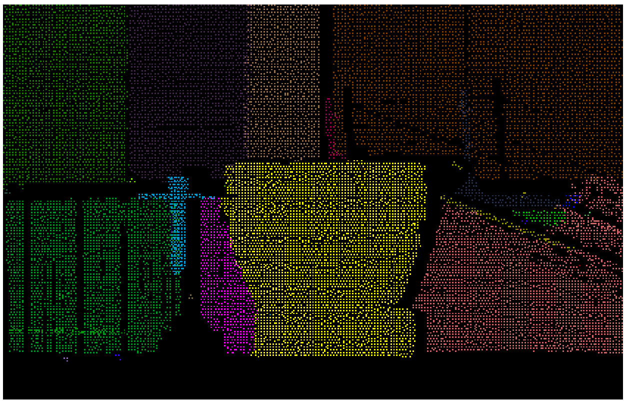

Figure 4a: One segmented urn

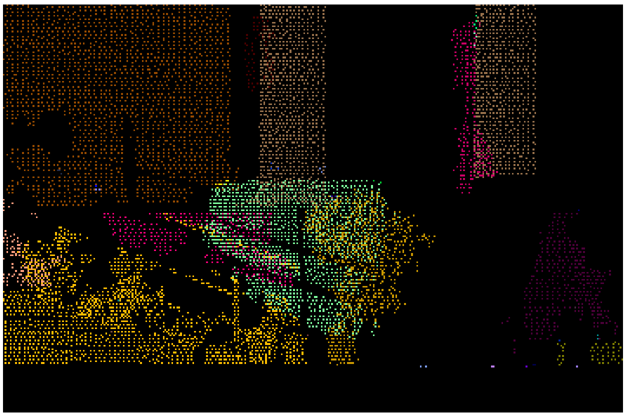

Figure 4b: Another segmented urn

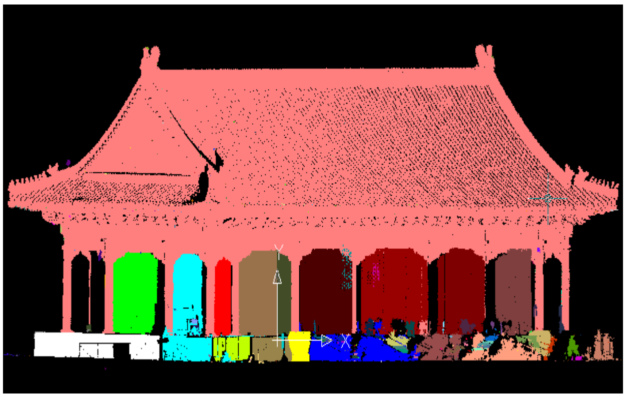

Figure 5: The segmented result with k set as 70

4. CONCLUSIONS AND FUTURE WORK

In this paper we presented our approach to segment point cloud data acquired from an ancient Chinese architecture. The algorithm uses only colorimetrical difference as a similarity measure, which is calculated in RGB color space using Euclidean distance. The roughly segmented regions are got from region growing process based on point-point colorimetrical similarity. Then the regions are merged according to the colorimetrical similarity and finally refined to smooth the noisy points. The results on an architectural point cloud were presented that show the effectiveness of the method and its ability to get balance between under- and over-segmentation. Although it is possible to segment point clouds using only RGB values, it is hard to achieve satisfactory global segmentation effect. This is because RGB color model is not in perceptive conformity with human eyes. And the similarity in RGB space doesn't necessarily reflect the perceptive color similarity because the three color components are relative and influenced by lighting and shading factors in the scene. The future work will be mainly on studying perceptively-conform color model i.e. CIELAB model and integrating possible geometrical derivatives such as normal to improve the segmentation effects.

Acknowledgment

This research is supported by the Hi-Tec Research and Development Project (863 project, No. 2006AA12Z151) and the National Natural Science Foundation of China (No. 40871211).

References

Bentley, J. L., 1975. Multidimensional binary search trees used for associative searching. Communications of the ACM 18(9), pp. 509-517.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A. and Fisher, R.B., 1996. An experimental comparison of range image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(7), pp. 673-689.

Maas, H.G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing 54(2-3), pp. 153-163.

Marshall, A.D., Lukács, G. and Martin, R.R., 2001. Robust segmentation of primitives from range data in the presence of geometric degeneracy. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3), pp. 304-314.

Press, W.H., Teukolsky, S.A., Vetterling, W. T. and Flannery, B. P., 2007. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, New York, pp. 1099-1110.

Pu, S. and Vosselman, G., 2006. Automatic extraction of building features from terrestrial laser scanning. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Dresden, Germany, 5p.

Rabbani, T., Dijkman, S., van den Heuvel, F.A. and Vosselman, G., 2007. An integrated approach for modeling and global registration of point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 61(6), pp. 355-370.

Rabbani, T., van den Heuvel, F.A. and Vosselman, G., 2006. Segmentation of point clouds using smoothness constraint. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Dresden, Germany, pp. 248-253 .

Rabbani, T. and van den Heuvel, F.A., 2005b. Efficient hough transform for automatic detection of cylinders in point clouds. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Enschede, the Netherlands, pp. 60-65.

Tremeau, A. and Borel, N., 1997. A region growing and merging algorithm to color segmentation. Pattern Recognition 30(7), pp. 1191-1203.

Vosselman, G. and Dijkman, S., 2001. 3d building model reconstruction from point clouds and ground plans. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Annapolis, USA, pp. 37-44.

Vosselman, G., Gorte, B.G.H., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Freiburg, Germany, pp. 33-38.

Jiang, X. Y., Bunke, H., 1994. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications* 7(2), pp. 115-122.