

A DISTANCE-WEIGHTED GRAPH-CUT METHOD FOR THE SEGMENTATION OF LASER POINT CLOUDS

Avishek Dutta, Johannes Engels and Michael Hahn

Stuttgart University of Applied Sciences
 Schellingstr. 24
 D-70174 Stuttgart, Germany
 johannes.engels@hft-stuttgart.de

Commission III, WG III/2

KEY WORDS: graph, cut, segmentation, laser point cloud

ABSTRACT:

Normalized Cut according to (Shi and Malik 2000) is a well-established divisive image segmentation method. Here we use Normalized Cut for the segmentation of laser point clouds in urban areas. In particular we propose an edge weight measure which takes local plane parameters, RGB values and eigenvalues of the covariance matrices of the local point distribution into account. Due to its target function, Normalized Cut favours cuts with “small cut lines / surfaces”, which appears to be a drawback for our application. We therefore modify the target function, weighting the similarity measures with distant-depending weights. We call the induced minimization problem “*Distance-weighted Cut*” (DWCut). The new target function leads to a slightly more complicated generalized eigenvalue problem than in case of the Normalized Cut; on the other hand, the new target function is easier to interpret and avoids the just-mentioned drawback. DWCut can be beneficially combined with an aggregation in order to reduce the computational effort and to avoid shortcomings due to insufficient plane parameters.

Finally we present examples for the successful application of the Distance-weighted Cut principle. The method was implemented as a plugin into the free and open source geographic information system SAGA; for preprocessing steps the proprietary SAGA-based LiDAR software LIS was applied.

1 INTRODUCTION

Segmentation denotes the task of partitioning a set (e.g. the pixels of an image or the points of a laser point cloud) into disjoint sets, whose elements share certain properties or exhibit similarities with respect to certain attributes. In the case of images, mostly low-level features like intensity, hue or vicinity are employed as segmentation criteria; for laser point clouds also geometric attributes like local plane parameters, point densities etc. come into consideration.

There are roughly speaking two groups of segmentation methods, see e.g. (Gonzalez and Woods 2002): The first category is based on *similarity*. Starting with elements having locally extremal values of a distinguishing attribute, elements with similar values are successively *aggregated* by a *region growing*. The second category of methods is based on *discontinuity*. Abrupt changes in the criterion function are detected, often by evaluating gradients, in order to determine the borders between two adjacent subsets. While such borders are not necessarily closed curves / surfaces, the segmentation turns out to be much simpler if this holds true; any enclosed region can then be immediately associated with a segment. The requirement is automatically fulfilled for divisive algorithms, which subdivide the image / point cloud successively into smaller pieces.

2 NORMALIZED CUT - BASIC DEFINITIONS

Normalized Cut (Shi and Malik 2000) is an established divisive segmentation method. Its applications reach from low-level tasks like image compression to semantic interpretation, e.g. the interpretation of medical images, (Carballido-Gamio et al. 2004). While it was originally conceived for perceptual grouping in raster

images, it has been successfully applied for the segmentation of laser point clouds as well, see e.g. (Reitberger 2010). The basic model of *Normalized Cut* is an undirected weighted graph $G = (V, E)$, featuring the elements of a set V , e.g. pixels or laser points, as nodes (vertices). E denotes the set of edges, i.e. of all pairs of non-identical nodes (u, v) , $u, v \in V$. To each edge (u, v) a non-negative “edge weight” $w(u, v)$ is assigned which represents a similarity measure comparing the connected nodes u, v . The graph G is successively subdivided into smaller sub-graphs: “In grouping we seek to partition the set of vertices into disjoint sets V_1, V_2, \dots, V_m , where by some measure the similarity among the vertices in a set V_i is high and, across different sets V_i, V_j is low.” (Shi and Malik 2000). This is achieved by minimizing in each decomposition step a cost function, which is essentially determined by the edge weights of the cut edges.

The choice of the cost function is decisive for the segmentation result. Cost functions are expressed in terms of the *association* of two arbitrary sets A, B , which is defined by

$$assoc(A, B) := \sum_{u \in A} \sum_{v \in B} w(u, v) \quad (1)$$

Nodes are serially numbered; we shall mostly write the weights in index notation W_{ij} instead of $w(u, v)$ where $i, j \in \{1, 2, \dots, N\}$ and N is the number of nodes in V . If we require the sets A, B to be just two disjoint subsets whose union equals the set V , *assoc* itself represents a possible cost function, which is called *cut* in this case:

$$cut(A, B) := assoc(A, B) \quad (2)$$

Cut as a cost function was proposed by (Wu and Leahy 1993). As the authors point out themselves, their *minimum cut criterion* suffers from the shortcoming that it favours unbalanced cuts, i.e.

cuts where one of the resulting subsets is small: $cut(\mathbf{A}, \mathbf{B})$ tends to be small if the number of edges to be cut is small. For this reason, (Shi and Malik 2000) introduced a modified cut criterion, called *Normalized Cut*, which avoids the disadvantages of the minimum cut criterion:

$$Ncut(\mathbf{A}, \mathbf{B}) := \frac{cut(\mathbf{A}, \mathbf{B})}{assoc(\mathbf{A}, \mathbf{V})} + \frac{cut(\mathbf{A}, \mathbf{B})}{assoc(\mathbf{B}, \mathbf{V})} \quad (3)$$

The authors point out that "the cut that partitions out small isolated points will no longer have small $Ncut$ value, since the cut value will almost certainly be a large percentage of the total connection from that small set to all other nodes" (ibidem). That means, if e.g. the subset \mathbf{A} is small, not only $cut(\mathbf{A}, \mathbf{B})$, but also $assoc(\mathbf{A}, \mathbf{V})$ will be small, so that $Ncut$ will not automatically assume small values. Compared to other cost functions the minimization of the $Ncut$ criterion can be achieved with relatively low computational effort.

For a constructive mathematical formulation, the edge weights are collected in the weight matrix $\mathbf{W} := [W_{ij}]$; \mathbf{W} is symmetric. Furthermore a diagonal matrix \mathbf{D} is defined according to

$$D_{ii} = \sum_{k \in \mathbf{V}} W_{ik} \quad (4)$$

\mathbf{D} contains for each node the sum of the weights of all incident edges; it is therefore called *total connection matrix*. The matrix $\mathbf{L} := \mathbf{D} - \mathbf{W}$ is called *Laplace matrix*.

The subdivision of the graph is conveniently expressed by an indicator vector \mathbf{x} of dimension N . The i -th element of \mathbf{x} is 1 if node i belongs to subset \mathbf{A} and -1 if node i belongs to \mathbf{B} . With the definition of a second type of indicator vectors

$$\tilde{\mathbf{y}} := assoc(\mathbf{B}, \mathbf{V})(1 + \mathbf{x}) - assoc(\mathbf{A}, \mathbf{V})(1 - \mathbf{x}) \quad (5)$$

the following equalities are easily obtained:

$$\tilde{\mathbf{y}}^T (\mathbf{D} - \mathbf{W}) \tilde{\mathbf{y}} = 4 assoc(\mathbf{A}, \mathbf{B}) assoc^2(\mathbf{V}, \mathbf{V})$$

$$\tilde{\mathbf{y}}^T \mathbf{D} \tilde{\mathbf{y}} = 4 assoc(\mathbf{A}, \mathbf{V}) assoc(\mathbf{B}, \mathbf{V}) assoc(\mathbf{V}, \mathbf{V})$$

and therefore

$$\frac{\tilde{\mathbf{y}}^T (\mathbf{D} - \mathbf{W}) \tilde{\mathbf{y}}}{\tilde{\mathbf{y}}^T \mathbf{D} \tilde{\mathbf{y}}} = Ncut(\mathbf{A}, \mathbf{B}) \quad (6)$$

The given definition of $\tilde{\mathbf{y}}$, is slightly different from the one in (Shi and Malik 2000), however leads to the same results. Unconstrained minimization of the ratio (6) yields the following generalized eigenvalue problem:

$$(\mathbf{D} - \mathbf{W}) \tilde{\mathbf{y}} = \lambda \mathbf{D} \tilde{\mathbf{y}} \quad (7)$$

The eigenvalues λ of (7) represent the $Ncut$ values of the decompositions which originate from the corresponding eigenvectors as indicator vectors. Therefore it seems that the eigenvector corresponding to the smallest eigenvector of (7) represents the solution of the minimization problem (6). The solution of the original problem, however, has to fulfil some constraints:

1. A particular combination of the \tilde{y}_i should vanish:

$$\mathbf{1}^T \mathbf{D} \tilde{\mathbf{y}} = 0 \quad (8)$$

2. The elements of \mathbf{x} may assume only values 1 or -1, therefore according to the definition (5), the elements of $\tilde{\mathbf{y}}$ are

also constrained:

$$\tilde{y}_i \in \{2 assoc(\mathbf{B}, \mathbf{V}), -2 assoc(\mathbf{A}, \mathbf{V})\} \quad (9)$$

3. The sets \mathbf{A} , \mathbf{B} should not be empty, i.e. the elements of \mathbf{x} should not be all the same.

The total connection matrix \mathbf{D} is assumed to be positive definite, otherwise there is at least one node with no incident non-zero edges and the graph \mathbf{G} cannot be connected. Furthermore, the Laplace matrix $\mathbf{D} - \mathbf{W}$ can be shown to be positive-semidefinite. The smallest eigenvalue of the system (7) is $\lambda = 0$ and the corresponding eigenvector $\tilde{\mathbf{y}}_1 = \mathbf{1}/\sqrt{N}$. This eigenvector clearly corresponds to the (undesired) case that all nodes are associated to the same subset, which is to be avoided according to the third constraint. Therefore, rather the eigenvector of (7) corresponding to the second-smallest eigenvalue has to be used.

It is easy to show that the first constraint is automatically fulfilled by all other eigenvectors. On the other hand, solutions of (7) will in general not fulfill the second constraint. Therefore the minimization of the $Ncut$ according to (6) under the given constraints is not equivalent with the computation of the eigenvector which belongs to the second smallest eigenvalue of the system (7). This problem seems to be unsolved. (Shi and Malik 2000) proposed the following pragmatic approach: Compute the eigenvector of (7) corresponding to the second smallest eigenvalue. Select a threshold t and associate the nodes with $\tilde{y}_i \leq t$ to subset \mathbf{A} and the nodes with $\tilde{y}_i > t$ to subset \mathbf{B} . This approach seems to be sufficient and effective for all practical computations. Regarding the numerical solution, (Shi and Malik 2000) proposed to solve this transformed system by means of the *Lanczos method*, see e.g. (Golub and Van Loan 2013).

3 EDGE WEIGHT FUNCTIONS FOR LASER POINT CLOUDS

(Shi and Malik 2000) give examples for the segmentation of images. They use a weight function, which is a multiplicative combination of a distance-depending part and a part comparing the greyvalues of the pixels:

$$W_{ij} = \exp\left(-\frac{\|F_i - F_j\|^2}{\sigma_I^2}\right) * \begin{cases} \exp\left(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{\sigma_X^2}\right) & \text{if } \|\mathbf{X}_i - \mathbf{X}_j\| < r \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Here F_i, F_j denote the greyvalues of the pixels i, j . σ_I, σ_X denote two scale factors, r a threshold for the consideration of the weight W_{ij} . Obviously the distance between two pixels directly effects their "similarity".

While the above weight function for the segmentation of pan-chromatic images appears "natural", the choice of a weight function for the segmentation of laser point clouds is not so straightforward; it may depend upon the registration type (airborne, mobile, single pulse, full waveform etc.), the targeted objects (forests, buildings etc.), available additional information (colour enrichment, local plane parameters) etc. Here we deal mostly with mobile laser data of urban areas for the purpose of building modelling. In this domain, segmentation is to separate different planar segments of building facades. Therefore the most important criterion for the similarity of two points is the question if

they are located on a common planar segment. By a plane fitting in a preprocessing step the point cloud may be augmented with local plane parameters as additional attributes. Of course such plane parameters are meaningless for line-like or spatially isotropic point distributions, e.g. along power lines or on a rough vegetation surface. For such cases the eigenvalues of the covariance matrices of the local point distributions represent valuable information about the type and the spatial extension of the registered objects, upon which information a similarity measure may be based, see e.g. (Gross and Thoennessen 2006), (Jutzi and Gross 2009).

The experimental investigations for the present study were performed in the framework of the free and open-source geographic information system SAGA; for preprocessing steps the proprietary SAGA-based LiDAR software LIS was used. SAGA / LIS provide modules for local plane-fitting, colour enrichment and for the eigenvalue computation of covariance matrices of local point distributions. If no colour information is available, intensity values of the laser reflections may be used. In our proposal for a weight function for LiDAR point clouds we therefore assume that for each point of the point cloud either local plane parameters or eigenvalues of the local point distribution are available.

We propose the following weight function between two LiDAR points i, j :

$$W_{ij} = \begin{cases} W_{ij}^{Dist} [c_P W_{ij}^N * W_{ij}^O + c_{RGB}^{(1)} W_{ij}^{RGB}] & \text{if both points have valid plane parameters} \\ W_{ij}^{Dist} [c_E W_{ij}^E + c_{RGB}^{(2)} W_{ij}^{RGB}] & \text{if none of the points has valid plane parameters} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Here $c_P, c_{RGB}^{(1)}, c_{RGB}^{(2)}, c_E$ are user-selected constant coefficients, the weight contributions $W_{ij}^m, m \in \{N, O, RGB, E\}$ are explained in the following. This weight function obviously implies that points with valid plane parameters and points without such parameters are always grouped into disjoint segments. The constant coefficients are normalized according to

$$c_P + c_{RGB}^{(1)} = 1, \quad c_{RGB}^{(2)} + c_E = 1 \quad (12)$$

Those coefficients control the relative impact of the weight contributions $W_{ij}^N * W_{ij}^O, W_{ij}^{RGB}$ or W_{ij}^E, W_{ij}^{RGB} , respectively. As all weight contributions $W_{ij}^m, m \in \{N, O, RGB, E\}$ vary between 0 and 1, the range of W_{ij} is therefore also $[0, 1]$.

3-1 Distance-dependent Weight Contribution

For the distance-dependent weight contribution W_{ij}^{Dist} we propose an exponential decay similar as in (10):

$$W_{ij}^{Dist} := \begin{cases} \exp \left[-\frac{s_{ij}^2}{(c_d(i)c_d(j) + \alpha_d^2)\sigma_d^2} \right] & \text{if } s_{ij} < r \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Here $s_{ij} := \|\mathbf{X}_i - \mathbf{X}_j\|$ denotes the Euclidean distance between the points i, j ; σ_d^2 is a dimensionless scale factor. Apart from σ_d^2 we introduce a point-dependent scale factor c_d : The local point density of mobile laser data may vary considerably depending on the distance of the scanner to the targeted surface and the incidence angle of the laser rays. That means, these variations may

be characteristic for the measuring process rather than for the targeted object and do not necessarily correspond to properties of the material surface. If they are not compensated, the *Normalized Cut* principle favours cuts along surfaces where the local point density is low, because there are relatively few connections to be cut, so the corresponding $cut(\mathbf{A}, \mathbf{B})$ is relatively small.

3-2 Similarity Measures based on Plane Parameters, RGB values and Eigenvalues of the Covariance Matrix

The terms W_{ij}^N, W_{ij}^O in (11) are to quantify how well the plane parameters of the points i, j are compatible and how close each of the points is located to the local plane of the other point. We use the common implicit plane representation

$$\langle \mathbf{N}, \mathbf{X} \rangle - d = 0 \quad (14)$$

where \mathbf{N} denotes the unit normal vector of the plane and d its distance of from the origin. The weight contribution W_{ij}^N measures the similarity between the normal vectors of the points i, j :

$$W_{ij}^N = \exp \left[-\frac{\|\mathbf{N}_i - \mathbf{N}_j\|^2}{\sigma_N^2} \right] \quad (15)$$

This measure is not sufficient for the comparison of planes, as parallel planes with an offset cannot be distinguished by (15). An analogous expression for the similarity of the distance d is not invariant against translational motions of the coordinate system. We therefore prefer to use the distances of a point from the best fitting plane of the other point instead:

$$W_{ij}^O = \exp \left[-\frac{(\langle \mathbf{N}_i, \mathbf{X}_j \rangle - d_i)^2 + (\langle \mathbf{N}_j, \mathbf{X}_i \rangle - d_j)^2}{\sigma_O^2} \right] \quad (16)$$

σ_N^2, σ_O^2 are global scale factors. The similarity contributions (15), (16) are not redundant. Simple geometric configurations can be cited for which one of these measures reaches its maximum while the other remains small. For a good coincidence of the two local planes we require high values for both criteria; therefore their contributions are combined in a multiplicative way.

For the comparison of the RGB values and the eigenvalues of the covariance matrices we propose simple Euclidean distances:

$$W_{ij}^{RGB} = \exp \left(-\frac{\sum_{l \in \{R, G, B\}} (F_i^{(l)} - F_j^{(l)})^2}{\sigma_{RGB}^2} \right) \quad (17)$$

$$W_{ij}^E = \exp \left(-\frac{\sum_{l=1}^3 (\lambda_i^{(l)} - \lambda_j^{(l)})^2}{\sigma_E^2} \right)$$

$F_i^{(R)}, F_i^{(G)}, F_i^{(B)}$ denote the greyvalues of point i in the red, green and blue channel, $\lambda_i^{(1)}, \lambda_i^{(2)}, \lambda_i^{(3)}$ the eigenvalues of the covariance matrix of the point distribution in the vicinity of point i . $\sigma_{RGB}^2, \sigma_E^2$ are global scale factors.

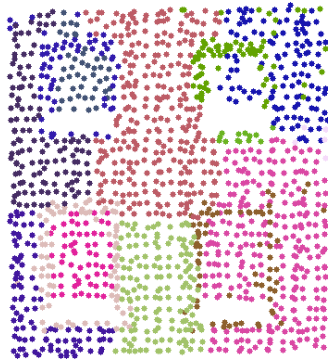
4 SEGMENTATION REQUIREMENTS AND SHORTCOMINGS OF NORMALIZED CUT

The *Ncut* minimization condition together with the afore-mentioned weight function entails some implications, which may be considered as drawbacks, depending on the particular application:

1. Slim-shaped objects tend to be cut at their narrowest section. As the weights decrease with increasing distance of the



(a) RGB colour-enriched point cloud of a facade with windows



(b) Ncut segmentation of the point cloud. Colours indicate the segment affiliation of the points

Figure 1: Ncut of a facade

nodes, at the narrowest section there are relatively few edges or connections to be cut. This implies a relatively small *cut* value, while the associations $assoc(\mathbf{A}, \mathbf{V})$, $assoc(\mathbf{B}, \mathbf{V})$ in the denominator of (3) might be big, e.g. because there are a lot of connections among the members of one and the same subset \mathbf{A} or \mathbf{B} . So the resulting *Ncut* value tends to be small. For some applications the property seems to be adequate indeed as e.g. for single tree detection from laser point clouds, see (Reitberger 2010). Here the distribution of the laser points in space essentially reflects the tree contours or silhouettes. On the other hand, in the case of terrestrial laser point clouds the local point density is somewhat random and dependent on the distance to the scanner, and it will often be undesirable to cut slim-shaped point sets at their narrowest section as the subsets may anyway form a continuous object.

2. In a recursive application of the Normalized Cut, there may appear smaller *Ncut* values on a deeper level of the recursion than on higher levels. If the *Ncut* of the higher level exceeds the selected threshold, the smaller *Ncuts* in the deeper levels are never reached.
3. An addition of “remote” nodes, i.e. nodes which are far from the expected cut line or surface, may change the *Ncut* value, as $assoc(\mathbf{A}, \mathbf{V})$, $assoc(\mathbf{B}, \mathbf{V})$ possibly change, whereas $cut(\mathbf{A}, \mathbf{B})$ may remain unaltered. The threshold for the acceptance of a cut therefore appears arbitrary or at least hard to interpret.

For the purpose of segmentation of mobile laser scanning data, we encountered in particular the first implication as a severe drawback. Figure (1) shows an example where the points of a building facade have been segmented by Normalized Cut. Obviously one of the windows is not “resolved”, on the other hand the facade is decomposed into several pieces; the borders between these pieces are mostly short lines, where only few connections had to be cut. See e.g. the upper border of the light green facade segment, which is short as it extends between two windows. This cut between the two windows is undesired, the more so as both resulting subsets exhibit very similar plane parameters and spectral properties. We propose therefore a modified minimization criterion, which avoids the afore-mentioned drawbacks.

5 DISTANCE-WEIGHTED CUT

5-1 Target Function of DWCut

As in (10) we assume again that the weights W_{ij} are of the form

$$W_{ij} = W_{ij}^{Distance} \cdot W_{ij}^{Similarity}, \quad (18)$$

i.e. that they consist of a distance-dependent factor and a factor taking into account similarity measures like similarity of plane parameters, RGB values etc. In contrast to the foregoing sections, here the term “similarity” does not include the distance between the relevant nodes any more. Let us further assume that the graph \mathbf{G} is totally connected with respect to the edge weight coefficients W_{ij} .

We propose an alternative target function, which we call – in the same way as the corresponding minimization problem – the *Distance-weighted Cut (DWCut)*:

$$DWCut(\mathbf{A}, \mathbf{B}) := \frac{cut(\mathbf{A}, \mathbf{B})}{cut_D(\mathbf{A}, \mathbf{B})} \quad (19)$$

Here $cut_D(\mathbf{A}, \mathbf{B})$ denotes the cut which results if only the distance-depending part of the weights is taken into account. Using (18), *DWCut* can be written as

$$DWCut(\mathbf{A}, \mathbf{B}) := \frac{\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{B}} W_{ij}^{Distance} \cdot W_{ij}^{Similarity}}{\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{B}} W_{ij}^{Distance}} \quad (20)$$

DWCut can be represented in terms of the indicator vector \mathbf{x} :

$$DWCut(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}}{\mathbf{x}^T (\mathbf{D}_D - \mathbf{W}_D) \mathbf{x}} = \frac{\mathbf{x}^T \tilde{\mathbf{G}} \mathbf{x}}{\mathbf{x}^T \tilde{\mathbf{H}} \mathbf{x}} \quad (21)$$

where \mathbf{W}_D , \mathbf{D}_D denote the matrices of the coefficients $W_{ij}^{Distance}$, $D_{ij}^{Distance}$. Furthermore we use the abbreviations $\tilde{\mathbf{G}} := \mathbf{D} - \mathbf{W}$, $\tilde{\mathbf{H}} := \mathbf{D}_D - \mathbf{W}_D$.

According to (20) the distance-weighted cut can be interpreted as a weighted average of the similarity measure over the cut edges ij , where the similarity measure is given by $W_{ij}^{Similarity}$ and the weight by $W_{ij}^{Distance}$. If a distance-depending weight function analogous to (10) is used and the distance threshold r is selected sufficiently small, the distance-depending weight will be zero for most of the edges; only sufficiently short edges are then taken into account. *DWCut* is particularly easy to interpret, if we content ourselves with the following form of the distance-depending

weights:

$$W_{ij}^{Distance} = \begin{cases} 1 & \text{if } \|\mathbf{X}_i - \mathbf{X}_j\| < r \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

In this case, *DWCut* simply equals the mean similarity measure over the cut edges (where only the edges with a length smaller than r are taken into account).

It is easy to prove that the drawbacks of the *Ncut*, which we have pointed out in section 4, are all avoided by *DWCut*. However, just like the solution of *Normalized Cut*, the solution of *DWCut* is not necessarily unique.

The distance-weighted cut has to be regularized, because otherwise its value would be undefined if all nodes are associated to the same set, i.e. if one of the sets \mathbf{A} , \mathbf{B} is empty. In this case the indicator vector \mathbf{x} is $\mathbf{x} = \pm \mathbf{1}$, but $(\mathbf{D} - \mathbf{W})\mathbf{1} = \mathbf{0}$, $(\mathbf{D}_D - \mathbf{W}_D)\mathbf{1} = \mathbf{0}$, so that both numerator and denominator of the target function (21) vanish.

Regularization of the denominator with a matrix \mathbf{R} according to $\mathbf{H} := \mathbf{D}_D - \mathbf{W}_D + \mathbf{R}$ leads to the modified minimization problem

$$DWCut(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{x}^T \tilde{\mathbf{G}} \mathbf{x}}{\mathbf{x}^T \mathbf{H} \mathbf{x}} \stackrel{!}{=} \text{Minimum} \quad (23)$$

under the constraints

- a) $x_i \in \{1, -1\} \quad \forall i \in \{1 \dots N\}$
- b) The elements of \mathbf{x} must not be all the same.

Neglecting constraint a), (23) leads to the following generalized eigenvalue problem:

$$\tilde{\mathbf{G}} \mathbf{x} = \tilde{\lambda} \mathbf{H} \mathbf{x} \quad (24)$$

Two different regularization matrices suggest themselves:

- a) $\mathbf{R} = \mathbf{x}_1 \mathbf{x}_1^T$, with $\mathbf{x}_1 := \mathbf{1}/\sqrt{N}$.
- b) $\mathbf{R} = \mathbf{I}$.

It can be shown that for both regularizations the eigenvalues $\tilde{\lambda}_k$ and eigenvectors \mathbf{x}_k , $k = 2 \dots N$ of the problem (24) are not impaired, i.e.

$$\mathbf{H} \mathbf{x}_k = \tilde{\mathbf{H}} \mathbf{x}_k \quad (25)$$

5-2 Numerical Computation of the Distance-Weighted Cut

For the solution of the problem (23), we follow the approach of (Shi and Malik 2000) for the *Normalized Cut*: We initially neglect constraint a) of (23). As the smallest eigenvalue $\tilde{\lambda}_1$ of (23) corresponds to the eigenvector $\mathbf{x}_1 = \mathbf{1}/\sqrt{N}$, i.e. to a decomposition where all nodes go into one of the subsets \mathbf{A} or \mathbf{B} , we search the eigenvector corresponding to the second smallest eigenvalue. Then we select a threshold t and associate the nodes with $x_i \leq t$ to subset \mathbf{A} and the nodes with $x_i > t$ to subset \mathbf{B} .

(24) is a slightly more complicated eigenvalue problem than (7), as the matrix \mathbf{H} on the right hand side is not diagonal, in contrast to the matrix \mathbf{D} in (7). Therefore a transformation to a standard eigenvalue problem requires a higher numerical effort.

It is in general easier to find the smallest eigenvector than the second smallest. Therefore we prefer to *shift* the eigenvalue $\tilde{\lambda}_1 = 0$ to a higher value. This can be achieved by a modification of the matrix $\tilde{\mathbf{G}}$ according to

$$\mathbf{G} := \tilde{\mathbf{G}} + \mathbf{x}_1 \mathbf{x}_1^T \quad (26)$$

It can be proved that $\mathbf{x}_1, \mathbf{x}_k$ with $k = 2 \dots N$ are also eigenvectors of the modified problem

$$\mathbf{G} \mathbf{x} = \lambda \mathbf{H} \mathbf{x} \quad (27)$$

with $\lambda_1 = 1$, $\lambda_k = \tilde{\lambda}_k$; instead of computing the *second* smallest eigenvalue of the system (24) and its corresponding eigenvector, we may equivalently compute the smallest eigenvalue of the system (27) and its eigenvector.

For the solution of the system (27) or the computation of its eigenvector corresponding to the smallest eigenvalue, respectively, we apply a variant of the *Arnoldi algorithm* which was proposed by (Golub and Ye 2002).

5-3 Preprocessing Steps

In a first step local plane parameters for all points of the point cloud are calculated. Then the distance weights W_{ij}^{Dist} and the similarity weights $W_{ij}^{Similarity}$ are computed as described in section (3) and stored in a sparse matrix format. A convenient and efficient format is the *compressed-column representation* (CCR) of a sparse matrix, see e.g. (Golub and Van Loan 2013) p. 598 ff.

When we introduced the target function of *DWCut* in section (5-1), we assumed that the graph is totally connected with respect to the distance-depending edge weights W_{ij}^{Dist} , otherwise there were, apart from \mathbf{x}_1 , further eigenvectors of the generalized eigenvalue problem (24), for which the Rayleigh quotient in (23) was indeterminate. Therefore we have to decompose the graph into subgraphs that are totally connected. Each subgraph is represented by a list of points; the list is initialized with an arbitrary point which has not been associated to a subgraph so far. Successively all neighboring points of the points in the list are added (with the term “neighboring point” here we denote a point, whose edge with the current point features a nonzero weight).

In the following step each subgraph is recursively subdivided by *DWCut* into smaller segments, until the segments fall below a certain size or until the *DWCut* exceeds the selected threshold. The proposed solution method by (Golub and Ye 2002) yields local minima of the target function instead of the global minimum; the result may depend on the initial approximation of the indicator vector. It is therefore important to start from several initial approximations which have to be carefully selected. Appropriate approximations can be found by an algorithm similar to the one just described for the determination of subgraphs, while here a higher similarity than zero is required; the sets of nodes connected with this “minimal similarity” can be considered as approximate partitions.

6 A HYBRID SEGMENTATION METHOD: COMBINING DW CUT AND AGGREGATION

While *DWCut* is a promising segmentation method with pleasant properties, there are drawbacks. One drawback is the slow performance and high memory consumption (which however does not considerably exceed the requirements of *Normalized Cut*). Another disadvantage consists in partly insufficient partitionings of laser point clouds due to “unsharp” plane parameters. Consider the laser profile of a protrusion in figure 2. Let the offset d between the left and the right plane be just sufficiently big to resolve the planes, i.e. to decompose the points on the two planes into two different segments. In order to determine the plane parameters of

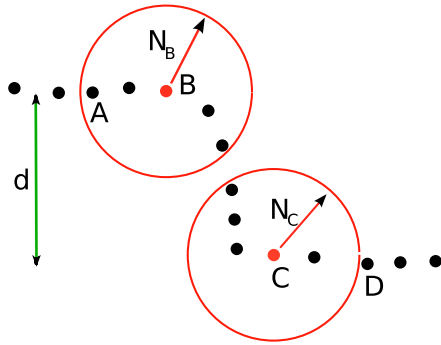


Figure 2: Blurred plane parameters cause insufficient partitioning (see text for explanation)

an arbitrary laser point, the neighbouring points within a sphere of a certain radius are selected; a plane is fitted to those points by an adjustment. As e.g. the spheres around the points B and C in the figure already contain points of the orthogonal connection between the planes, the local normal vectors of the points B and C are deflected to the right. Accordingly the plane offset between B and C appears much smaller than the true value d , and therefore the decomposition of the point set fails: The plane parameters appear “blurred” across the edge of the protrusion, and the edge weight between the points B and C appears higher than it should be. Therefore a cut between B and C may be rejected.

This kind of blurring appears quite frequently and may seriously deteriorate the results of the segmentation; it is not a peculiarity of *DWCut*, but may appear with other graph cut methods as well, if weight contributions of the form (15) and (16) are applied. Since the drawback is rather due to the plane fitting than due to graph cut itself, it is hard to avoid without a preceding segmentation: If the affiliation of the points to planar segments is already known in the beginning, the plane fitting to an arbitrary point could rely on the points within the same plane only. But the segmentation on its own is based on the plane parameters, so the snake bites its tail.

As a way out of this dilemma, we propose to compute the plane parameters as described, but perform a thinning of the point cloud before the subsequent segmentation. The thinning should be done in such a way, that in each voxel the point with the highest planarity is retained. In the following we call the retained points also “seed points”. In figure 2 most probably not the points B and C, but e.g. the points A and D would be retained, as the spheres around these points contain only neighbouring points which are situated on the related planes, therefore the planarity will be high and the plane parameters are not affected by “outliers”. Then the thinned point cloud is segmented by *DWCut*. The cut which separates the points A and D will probably be accepted, because the plane parameters of these two points are sufficiently different. Finally the residual points are aggregated to the seed points. As an aggregation criterion the similarity measures analogous to (16) and (17) can be used for the weight contributions W_{ij}^O , W_{ij}^{RGB} and W_{ij}^E . However, in the case of W_{ij}^O we recommend an asymmetric criterion: Only the plane parameters of the seed point should be taken into account, as the plane parameters of the residual point are not as reliable. That means, W_{ij}^O is essentially based on the distance between the local seed point plane and the residual point. In this way also such points can be aggregated to a planar segment, which do not have valid plane parameters themselves.

The aggregation benefits from the preceding segmentation of the thinned point cloud: Without that segmentation a complicated merging procedure would be necessary, which was to combine

planar seed point segments with similar plane parameters. In the proposed workflow the merging is unnecessary, as the seed points themselves are already grouped by the segmentation. On the other hand, due to the preceding thinning, the segmentation proves to be easier and more reliable: Thanks to the relative high planarity of the points of the thinned point cloud the segmentation is more robust. Furthermore the computational effort of *DWCut* is, by the reduction of the number of points, greatly reduced.

7 EXPERIMENTAL INVESTIGATIONS

As pointed out in section 3, we propose to define the similarity between two laser points with valid local plane parameters as a combined measure of the distances between the points in normal vector space, RGB space and the mutual distances between one point and the local plane of the other point. The success of the algorithm is obviously dependent on how the parameters of the similarity function are defined; these parameters essentially control the decay of the similarity measure with increasing disagreement in one of the mentioned criteria and also the relative impact of the individual constituents. Furthermore the distance-dependent weight, which defines the relative impact of an edge similarity within the averaged cut, is of crucial importance. In general, choosing a relaxed value for a similarity criterion may group together semantically unrelated objects, while the choice of a strict value may produce undesired cuts. This trade-off is important to realize and the choice of the parameters must be well guided by the nature of the application.

In the following we give some results of the Distance-weighted Cut algorithm, which also illustrate the influence of the similarity and weight parameters on the resulting segmentation.



Figure 3: Segmentation result for the point cloud of figure 1 by *DWCut*

Figure 1 was to demonstrate the crucial drawback of the Normalized Cut, which gave reason to a modification of the target function. Figure 3 now shows the segmentation of the same point cloud by *DWCut*. Apart from some segments with very few points, which may be considered as outliers, the facade appears as one segment; in particular the undesired cuts of figure 1 along the facade are avoided. We would like to emphasize that *this improvement was not achieved by a mere suppression of cuts by a tightening of the stop criterion of the algorithm, but rather by the modified target function*. This becomes evident by the fact that now all windows are “resolved”. As in the case of the Normalized Cut, the points on the facade and on the window panes have valid plane parameters, while the points on the embrasures

mostly do not as the embrasures are too narrow: the plane offset between the facade and the window panes amounts to ca. 30 cm. Sometimes facade points and points on the window panes are very close to each other; this implies that the distinction between the window segments and the facade segment is achieved by regular cuts and not by a separation of unconnected subgraphs. The cuts between facade and window panes were mostly effected by the constituent W_{ij}^O in the similarity function. The example demonstrates that the proposed similarity function is appropriate to distinguish different planar segments. Although the “blurring” of the plane parameters as described in figure 2 actually appeared in the embrasures of the windows, for this example it did not affect the result, the more so as we selected a small neighbourhood radius in the computation of the local plane parameters. However, as we shall see, such blurring effects can not always be avoided. Very small segments (size < 5 points) seldom signify anything of importance in the point cloud. We therefore applied a merging algorithm as a post-processing step in order to incorporate the small segments into bigger ones. The merging algorithm may choose to assign points of the small segments to bigger segments by checking which neighbouring segment’s local plane is closest to the point. If no neighbouring segment features a valid plane, the segment belonging to the closest neighbour wins the point. Figure 4 shows the result after the merging step.

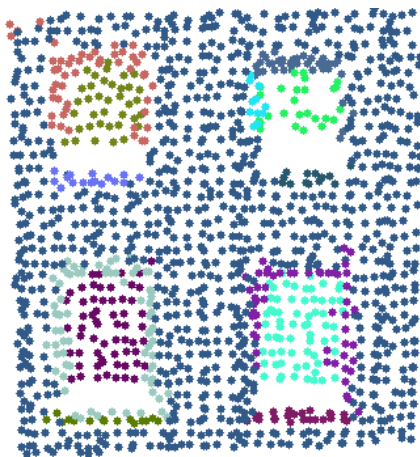


Figure 4: Segmentation result by DWCut for the point cloud of figure 3 after merging of small segments

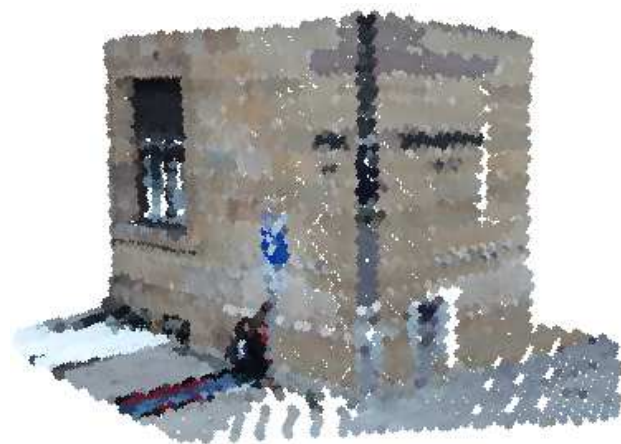


Figure 5: Colour-enriched point cloud containing a building edge

Figure 5 depicts a more complicated point cloud; most important objective is to see if the algorithms are able to reproduce the sharp building edge. Obviously with DWCut this failed as can be

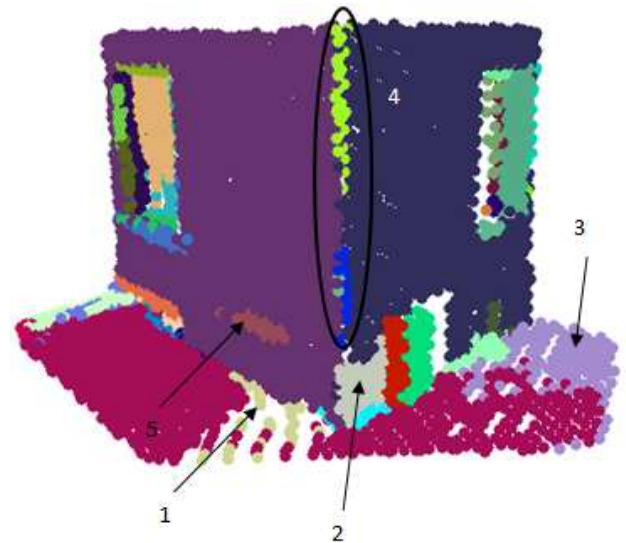


Figure 6: Segmentation result for the point cloud of figure 5 with DWCut

seen in figure 6, region (4): Along the edge there are two (undesired) elongated segments available. These segments appear since the local plane parameters, in particular the normal vectors, are “blurred” for the points near the edge. The neighbourhood of an arbitrary point close to the building edge may contain a considerable number of points of the adjacent facade, particularly if the chosen radius of the neighbourhood is relatively big. Therefore the edge is blurred in such a way that the normal vector is changing gradually and this may give rise to additional segments. It might even occur that the plane fitting fails and the points do not feature valid plane parameters. These effects can be mitigated or even avoided, if for each point only a very small neighbourhood is employed for the computation of the plane parameters. The downside of this strategy is that the computed plane parameters are unreliable and unstable. This becomes apparent in region (1) of figure 6. Here the laser scan lines have a big distance to each other – bigger than the selected neighbourhood radius. Therefore the plane parameters in that region are mostly computed with points of one scan line at a time, which yields insignificant normal vectors and therefore the apparent superfluous segment on the ground. The superfluous segment in region (3) of figure 6 is also for similar reasons.

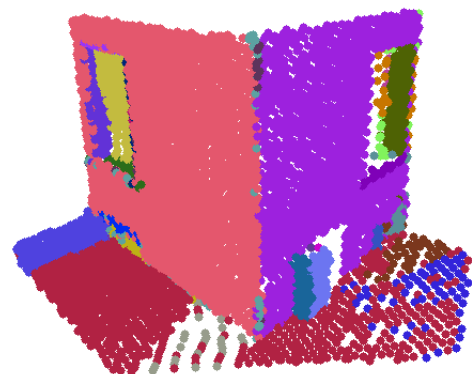


Figure 7: Segmentation result for the point cloud of figure 5 with DWCut, tuned parameters

In region (2) a proper segment of the facade itself is encountered. Actually there is a slight protrusion, starting from the height of the segment in (2) till the ground, on both faces of the facade. (5) shows a segment on the border between facade and protrusion.

For our application, we wished to suppress the protrusion, so the segment in (2) was undesired. For this purpose we relaxed the σ_N^2 parameter for the similarity of the normal vectors (0.08 to 0.75) and the corresponding σ_d^2 for the distance-depending weight (0.8 to 1.2); the impact of the RGB similarity was reduced to 0 in order to avoid the effect of shadows etc. in the segmentation. In order to avoid the segments in (4) and to obtain a sharp building edge, the plane parameters were computed from a smaller neighbourhood. Relaxing the distance parameter σ_d^2 allows for a bigger neighbourhood, which is often helpful in bridging over wide scan lines.

Figure 7 shows the improved result. It is apparent that the protrusion segments have disappeared. As expected, the superfluous ground segments are still available. Also the problems on the building edge are still visible, though somewhat mitigated.

A considerable improvement is made by means of the hybrid algorithm according to section 6: The point cloud is thinned based on the best planarity among the points in a voxel. The thinned point cloud is segmented using DWCut and subsequently an aggregation is carried out, where the segmented points of the thinned point cloud act as seed points; each aggregated point inherits the segment affiliation of the seed point to which it is aggregated.

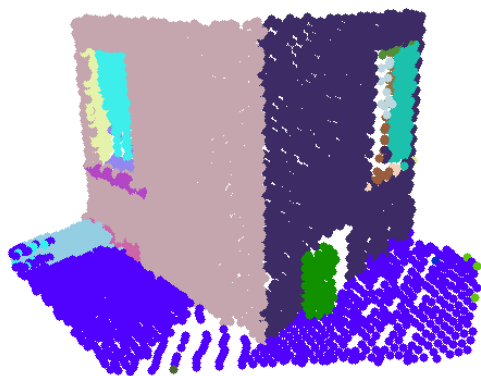


Figure 8: Segmentation result for the point cloud of figure 5 with hybrid algorithm

Figure 8 shows the segmentation result of the hybrid algorithm. Here the undesired segments of the building edge have disappeared; the edge is represented quite sharp. The ground is, apart from the stair at the left and some single points at the right, represented by one segment. The big protrusion on the right face has been grouped into one segment, although it has two distinct planar facets. This is due to the fact, that the thinning procedure failed to locate a point with considerable planarity on the missing facet.

Finally, figure 9 shows the segmentation result by the hybrid method applied to the complete facade. Although some problems remain, the result confirms the potential of the proposed algorithms.

ACKNOWLEDGEMENTS

This work was funded by the German Federal Ministry of Education and Research within the project “mms – Automatisierte Extraktion vertikaler Strukturen im städtischen Bereich aus Multisensor Mobile Mapping Daten”. The data used in this study was kindly provided by TopScan GmbH. Last not least, very helpful discussions with Volker Wichmann and Frederic Petrini of Laserdata GmbH, Innsbruck, Austria are gratefully acknowledged.



Figure 9: Segmentation result for a complete building facade with the hybrid algorithm

REFERENCES

- Carballido-Gamio J, Belongie SJ, Majumdar S 2004: Normalized cuts in 3-D for spinal MRI segmentation. *IEEE Trans Med Imaging* January 2004 23(1): 36-44.
- Gonzalez R and Woods R 2002: *Digital Image Processing*. 2nd Edition 2002. Prentice-Hall, Upper Saddle River, New Jersey.
- Golub G and Van Loan Ch 2013: *Matrix Computations*. 4th Edition 2013. John Hopkins University Press, Baltimore, Maryland.
- Golub G and Ye Q 2002: An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems. *SIAM Journal of Scientific Computing*, Vol 24 Issue 1, pp 312-334
- Gross H and Thoennessen U 2006: Extraction of Lines from Laser Point Clouds. In: W. Förstner and R. Steffen (eds), *Photogrammetric Computer Vision*, IAPRS, Vol. XXXVI Part3.
- Jutzi B and Gross H 2009: Nearest Neighbor classification on Laser point clouds to gain object structures from buildings. *ISPRS Hannover Workshop 2009: High resolution earth Imaging for geospatial Information*, Hannover, Germany. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII, Part 1-4-7/W5
- Reitberger J 2010: *3D-Segmentierung von Einzelbäumen und Baumartenklassifikation aus Daten flugzeuggetragener Full Waveform Laserscanner*. PhD Thesis Technische Universität München, Institut für Photogrammetrie und Fernerkundung
- Shi J and Malik J 2000: Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No 8, pp 888-905, August 2000
- Wu Z and Leahy R 1993: An Optimal Graph Theoretic Approach to Data Clustering: Theory and its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No 11, pp 1101-1113, November 1993