Consistent normals for large point clouds

Global optimality for streaming orientation of point cloud normals

Nico Schertler, Bogdan Savchynskyy, Stefan Gumhold

TU Dresden, Germany

TECHNISCHE UNIVERSITÄT DRESDEN

Computer Graphics and Visualization

Introduction

Consistent normal orientations are essential for renderings and many processing algorithms.

Given a point cloud with unoriented normals, we re-orient the normals in a consistent way.

Key points of our work:

- Consolidation of previous work into a *Markov Random Field* model
- Globally optimal solution of the MRF
- Out-of-core framework for large point clouds

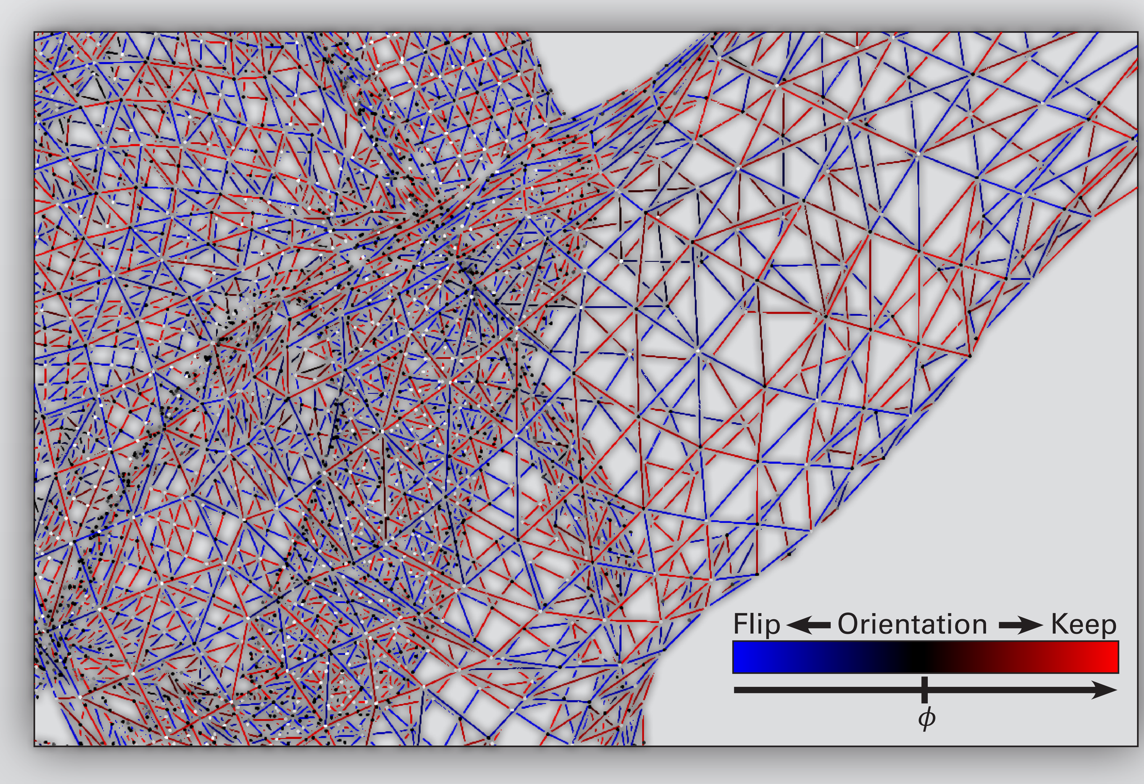
Problem formalization

Every point p_i with normal n_i is assigned a label $l_i \in \{-1, 1\}$, such that the adapted normals $n'_i := l_i n_i$ are most consistent.

Consistency is measured upon a neighbor graph with edges $\mathcal{E}$ and the flip criterion $\phi : \mathcal{E} \to \mathbb{R}$ (positive for consistent edges, negative for non-consistent edges). Hoppe's flip criterion [1] can be expressed as:

$$\phi_{Hoppe}(i,j) = \langle n_i, n_j \rangle$$

Example graph:

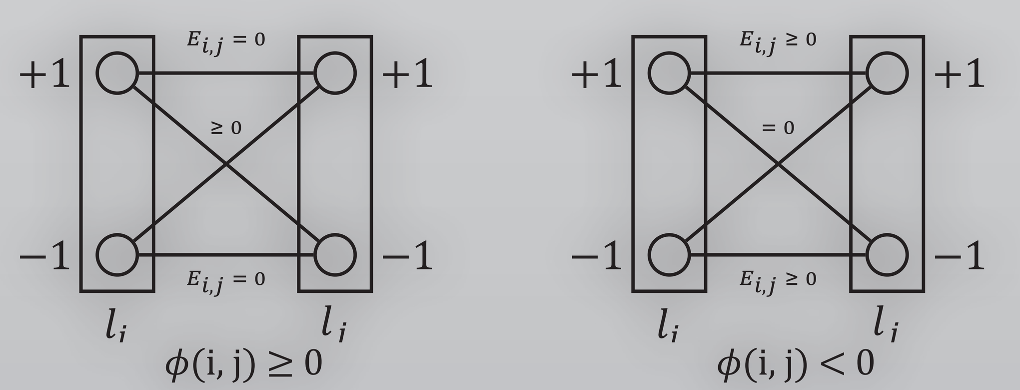


The following potentials are derived from this flip criterion:

$$E_{i,j}(l_i, l_j) = \begin{cases} |\phi(i,j)| \cdot \omega(p_i, p_j) & \phi(i,j) \geq 0 \oplus l_i = l_j \\ 0 & \text{otherwise} \end{cases}$$

$$\omega(p_i, p_j) = 1 - \frac{distance(p_i, p_j)^2}{r^2}$$

This potential definition can be visualized as follows:



The sum of potentials forms a labeling's energy. The optimal orientation is the energy's minimizer:

$$E(L) = \sum_{\{i,j\} \in \mathcal{E}} E_{i,j}(L_i, L_j) \qquad (1)$$

$$L^* = \arg\min_L E(L)$$

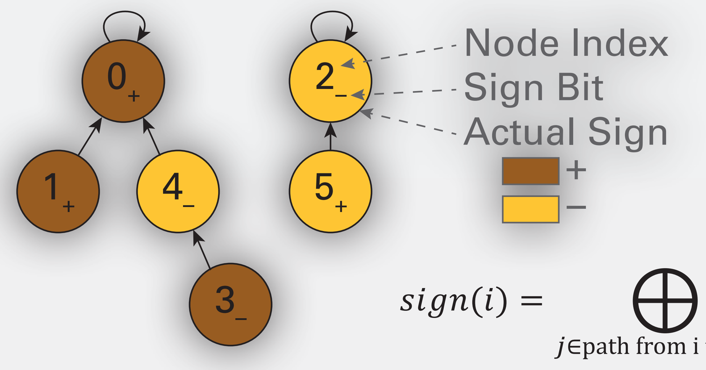
The nearby sections give an overview of various ways to solve this equation.

Signed Union Find

An approximate solution of (1) can be found by propagating the orientation along the minimum spanning tree [1].

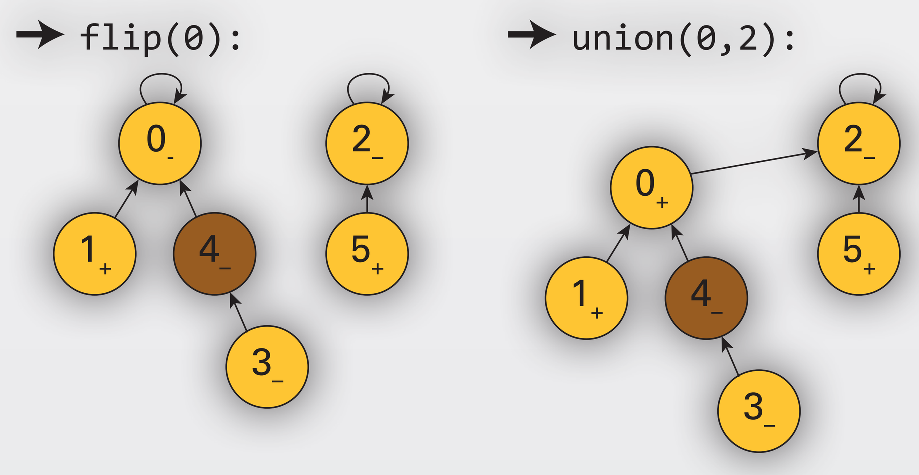
The MST can be calculated with Kruskal's algorithm and the Union Find data structure [2].

We augment this structure with a sign bit, which enables fast flips of entire connected components. Thus, MST propagation can be executed on-the-fly without explicit MST calculation.



$$sign(i) = \bigoplus_{j \in \text{path from } i \text{ to root}} s_j$$

Components can be flipped by inverting their root's sign bit. Merging components requires the sign bit update $s_i \leftarrow s_i \oplus s_j$.



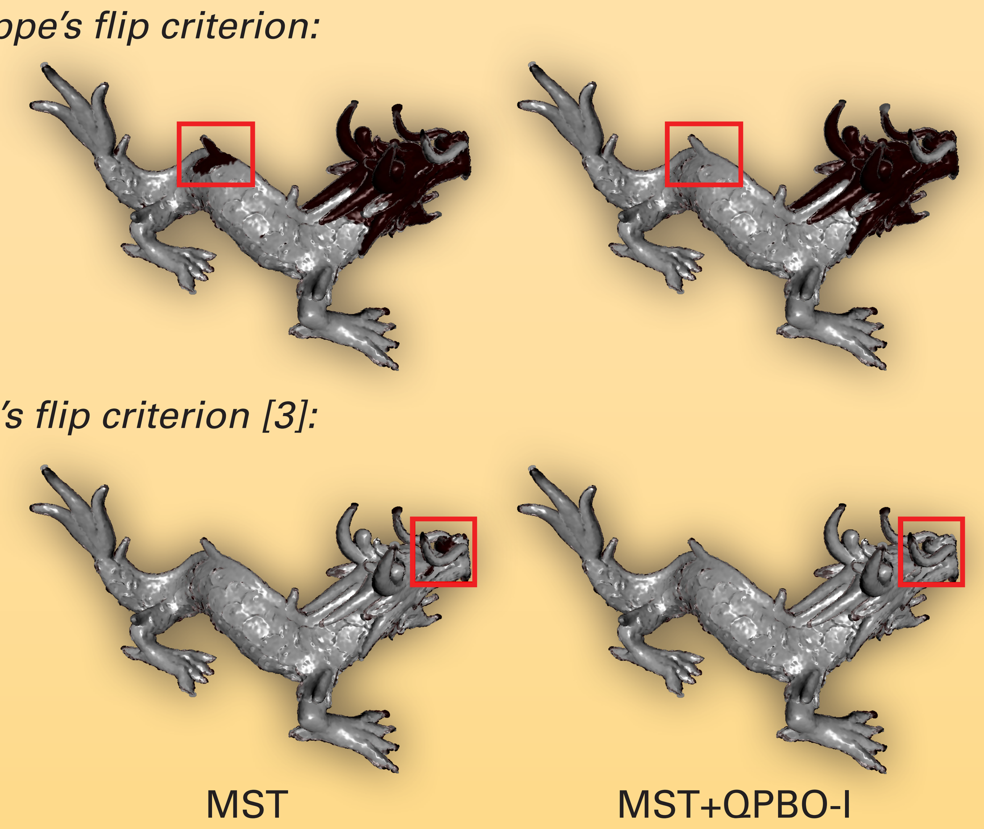
Similar updates exist for *path compression*. Sign bits are unaffected by *union-by-rank*.

Results

Using a global optimizer yields more consistent results than the greedy MST approach:

Hoppe's flip criterion:

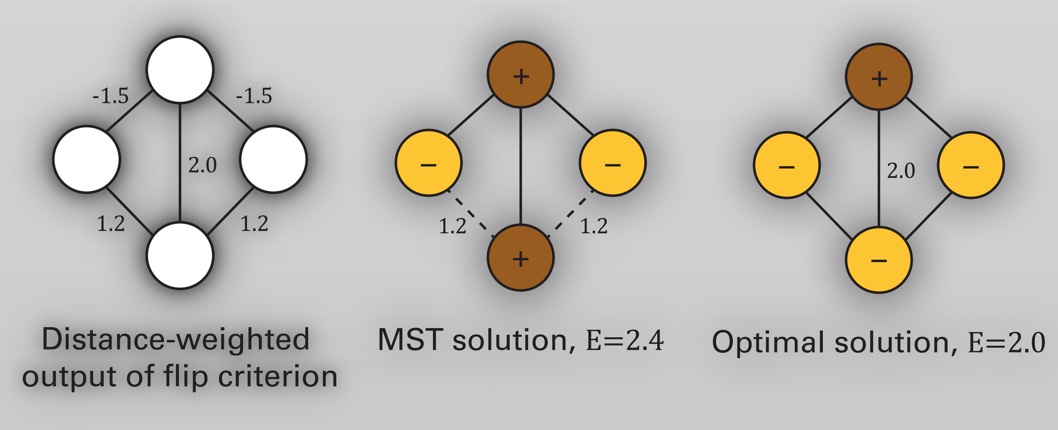
Xie's flip criterion [3]:



Our out-of-core approach allows orientation of large data sets and higher performance even for medium-sized data sets.

Globally optimal solution with QPBO

Greedy spanning tree-based solutions fail in cases with contra dicting edges like in this simple example:



Instead, we use QPBO (quadratic pseudo-boolean optimization [4]) to find a globally optimal solution.

QPBO outputs a labeling with $l_i \in \{-1, 1, \emptyset\}$, where $\emptyset$ denotes an unlabeled node. Every labeled node is part of a globally optimal solution (partial optimality property).

In order to remove the solution ambiguity, we force the first element of every connected component to the label +1 by altering the energy as follows:

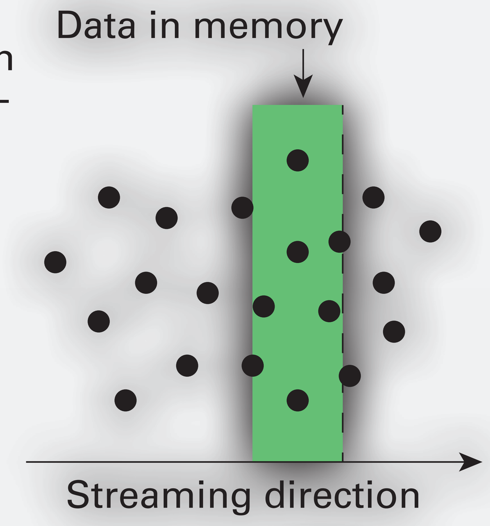
$$E_{QPBO} := E + \sum_{c \in C} \begin{cases} 0 & l_{c_0} = +1 \\ \zeta & \text{otherwise} \end{cases}$$

for the set of connected components C and an arbitrary positive number ζ.

We use the variant QPBO-I (QPBO Improve) to find labels for formerly unlabeled nodes. In this process, we successively fix random nodes to their according MST solution and re-solve QPBO until all nodes are labeled. We refer to this method as *MST+QPBO-I*.

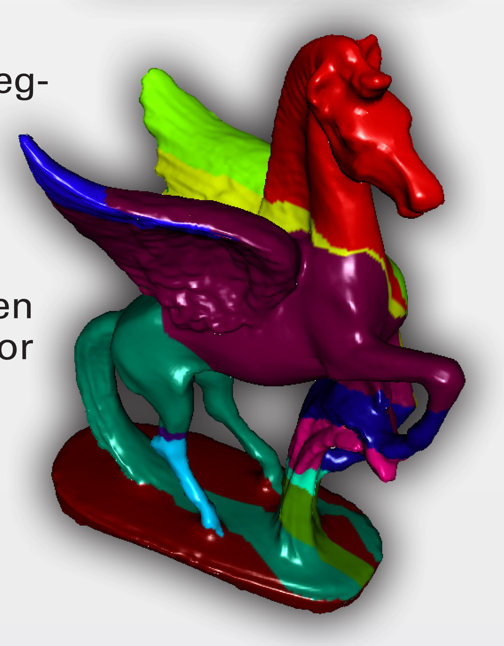
Orientation of large point clouds

Big data sets are first sorted and then streamed along the x-direction [5]. This allows out-of-core processing with a small slice of data in memory.



In a first step, the point cloud is segmented into locally orientable patches, i.e. there are no contradicting edges within a single patch.

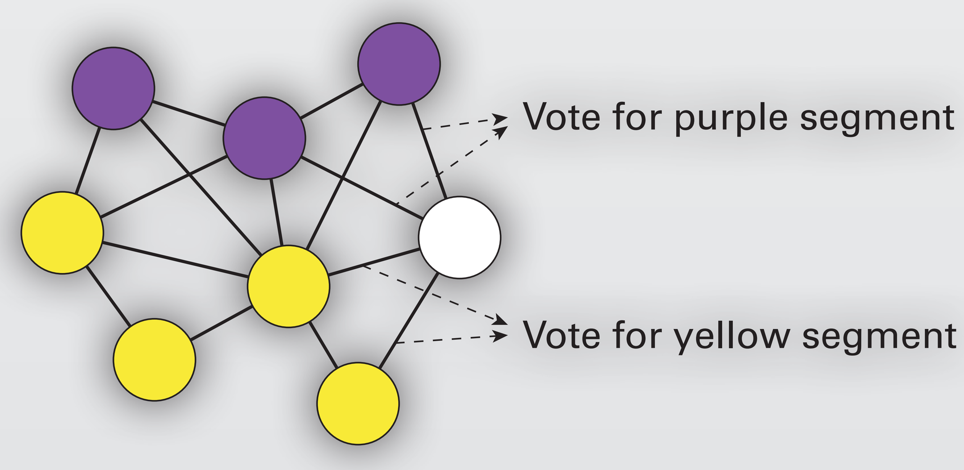
The global patch orientation is then calculated on the reduced neighbor graph, where points of the same patch have been contracted to a single node. The energy functions of edges are summed during this contraction.



Segmentation Details

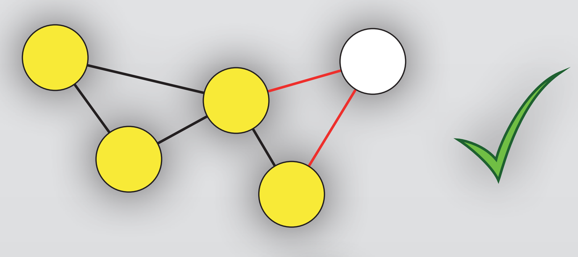
When a point is processed, all of its left neighbors are already segmented. The segments of the closest neighbors are considered as the segment for the processed node.

The sum of distance-weighted flip criterion output for each segment is the segment's vote. The segment with the greatest absolute vote wins:

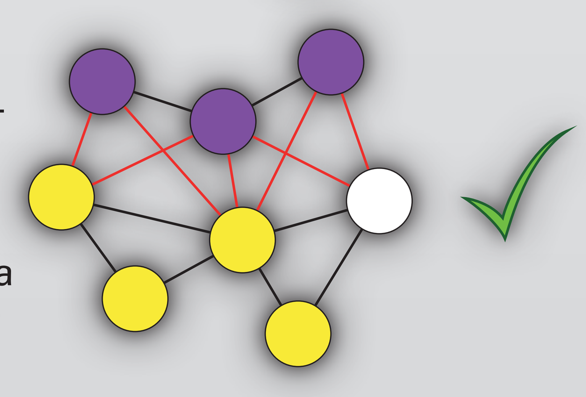


For the reduction to be energy-preserving, each considered segment must fulfill the *intra-segment criterion* and *inter-segment criterion*.

The *intra-segment criterion* constrains edges from neighbors within the considered segment. Every edge must have the same sign. We allow a small tolerance to reduce the number of segments.



The *inter-segment criterion* constrains edges from neighbors in other segments. They must have the same sign as all previous edges between the two segments. We allow a small tolerance to reduce the number of segments.



If no eligible segment exists, a new segment is created.

References

[1] Hoppe H. et al.: Surface reconstruction from unorganized points. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques (New York, NY, USA, 1992), SIGGRAPH '92, ACM, pp. 71–78.

[2] Kruskal J. B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society 7, 1 (1956), 48–50

[3] Xie H. et al.: Piecewise c1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. In Visualization, 2003. VIS 2003. IEEE, pp. 91–98

[4] Rother C. et al.: Optimizing binary MRFs via extended roof duality. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (2007), IEEE, pp. 1–8.

[5] Pajarola R.: Stream-processing points. In Visualization, 2005. VIS 05. IEEE (Oct 2005), pp. 239–246.

Acknowledgements

The Dragon data have been provided by the Stanford 3D Scanning Repository. The Pegasus model has been provided by the AIM@SHAPE-VISIONAIR Shape Repository. Data provision is thankfully acknowledged.

This work is partially funded by the European Social Fund and the Free State of Saxony (ESF project number 100226943, "ADFEX")