

# Supplementary Material Accompanying 'Geometry Driven Semantic Labeling of Indoor Scenes'

Salman H. Khan<sup>1</sup>, Mohammed Bennamoun<sup>1</sup>, Ferdous Sohel<sup>1</sup> and Roberto Togneri<sup>2</sup>

School of CSSE<sup>1</sup>, School of EECE<sup>2</sup>  
The University of Western Australia,  
35 Stirling Highway, Crawley, WA 6009, Australia

## 1 Disintegration of Higher Order Energies

In this appendix, we will show how the higher order energy potentials can be minimized using graph cuts. Since, graph cuts can efficiently minimize submodular functions, we will transform our higher order energy function (Eq. 9) to a submodular second order energy function. For the case of both  $\alpha\beta$ -swap and  $\alpha$ -expansion move making algorithms, we will explain this transformation and the process of optimal moves computation<sup>1</sup>. All of the previously defined notations are used in the same context and all of the newly introduced symbols are defined in this section. The function that accounts for the number of disagreeing nodes in a clique is defined as:

$$n_\ell(\mathbf{y}_c) = \sum_{i \in \mathbf{c}} w_i^\ell \mathbf{1}_{y_i = \ell}$$

The function  $\mathbf{1}_{y_i = \ell}$  is a zero-one indicator function that returns a unit value when  $y_i = \ell$ . We suppose here that weights are symmetric for all labels  $\ell \in \mathcal{L}$  i.e.,  $w_i^\ell = w_i$ . Further, for our implementation we set  $w_i = 1 \quad \forall i \in \mathbf{c}$ . This setting satisfies the required constraints for these parameters, i.e.,

$$w_i^\ell \geq 0 \quad \text{and} \quad \sum_{i \in \mathbf{c}} w_i^\ell = \#\mathbf{c} \quad \forall \ell \in \mathcal{L}.$$

We define a summation function that adds the weights for a subset  $\mathbf{s}$  of  $\mathbf{c}$ ,

$$W(\mathbf{s}) = \sum_{i \in \mathbf{s}} w_i^\ell = \#\mathbf{s} \quad \forall \ell \in \mathcal{L}.$$

### 1.1 Disintegration of Higher Order Energies to Second Order Sub-Modular Energies for Swap Moves

Suppose, in a clique 'c', the locations of the active nodes is represented by a set of indices  $\mathbf{c}_a$ . The nodes which remain inactive during the move making process

---

<sup>1</sup> The development of this section is similar to [13]. We also used the same notation - wherever possible - to allow the reader to easily sort out differences and commonalities.

are termed as *passive* nodes. Their locations are denoted by  $\bar{\mathbf{c}}_a = \{\mathbf{c} \setminus \forall c_i \in \mathbf{c}_a\}$ . The corresponding set of available moves to the swap move making algorithm are encoded in the form of a vector  $\mathbf{t}_{c_a}$ . For the sake of simple demonstration, let us focus on the two class labeling problem i.e.,  $\ell \in \{0, 1\}$ . The induced labeling is the combination of the old labeling for the inactive nodes and the new labeling for the active nodes i.e.,  $\mathbf{y}_c^n = \mathbf{y}_{\bar{c}_a}^o \cup T_{\alpha\beta}(\mathbf{y}_{c_a}^o, \mathbf{t}_{c_a})$ . If  $\mathbf{y}_c^n$  denotes the new labeling induced by move  $\mathbf{t}_{c_a}$  and  $\mathbf{y}_c^o$  denotes the old labeling, we can define the energy of move for an  $\alpha\beta$  swap as:

$$\begin{aligned} \psi_{\mathbf{c}}^m(\mathbf{t}_{c_a}) &= \psi_{\mathbf{c}}(\mathbf{y}_c^n) = \psi_{\mathbf{c}}(\mathbf{y}_{\bar{c}_a}^o \cup T_{\alpha\beta}(\mathbf{y}_{c_a}^o, \mathbf{t}_{c_a})) \\ &= \min_{\ell \in \mathcal{L}} \left\{ \lambda_{max} - (\lambda_{max} - \lambda_\ell) \exp\left(-\frac{W(\mathbf{c}) - n_\ell(\mathbf{y}_{\bar{c}_a}^o \cup T_{\alpha\beta}(\mathbf{y}_{c_a}^o, \mathbf{t}_{c_a}))}{Q_\ell}\right) \right\} \\ &= \min_{\ell \in \mathcal{L}} \left\{ \lambda_{max} - (\lambda_{max} - \lambda_\alpha) \exp\left(-\frac{W(\mathbf{c}) - n_0^m(\mathbf{t}_{c_a})}{Q_\alpha}\right), \right. \\ &\quad \left. \lambda_{max} - (\lambda_{max} - \lambda_\beta) \exp\left(-\frac{W(\mathbf{c} - \mathbf{c}_a) + n_0^m(\mathbf{t}_{c_a})}{Q_\beta}\right) \right\}, \end{aligned}$$

where,  $W(\mathbf{c}_a) = n_0^m(\mathbf{t}_{c_a}) + n_1^m(\mathbf{t}_{c_a})$ . The minimization operation in the above equation can be replaced by defining a piecewise function:

$$\psi_{\mathbf{c}}^m(\mathbf{t}_{c_a}) = \begin{cases} \lambda_{max} - (\lambda_{max} - \lambda_\alpha) \exp\left(-\frac{W(\mathbf{c}) - n_0^m(\mathbf{t}_{c_a})}{Q_\alpha}\right) & \text{if } n_0^m(\mathbf{t}_{c_a}) > \varrho_{\alpha\beta} \left( \frac{W(\mathbf{c})}{Q_\alpha} - \frac{W(\mathbf{c} - \mathbf{c}_a)}{Q_\beta} - \log\left(\frac{\lambda_{max} - \lambda_\alpha}{\lambda_{max} - \lambda_\beta}\right) \right), \\ \lambda_{max} - (\lambda_{max} - \lambda_\beta) \exp\left(-\frac{W(\mathbf{c} - \mathbf{c}_a) + n_0^m(\mathbf{t}_{c_a})}{Q_\beta}\right) & \text{if } n_0^m(\mathbf{t}_{c_a}) < \varrho_{\alpha\beta} \left( \frac{W(\mathbf{c})}{Q_\alpha} - \frac{W(\mathbf{c} - \mathbf{c}_a)}{Q_\beta} - \log\left(\frac{\lambda_{max} - \lambda_\alpha}{\lambda_{max} - \lambda_\beta}\right) \right), \end{cases}$$

where,  $\varrho_{\alpha\beta} = \frac{Q_\alpha Q_\beta}{Q_\alpha + Q_\beta}$ . The function  $n_\ell^m(\mathbf{t}_{c_a})$  is defined as:

$$n_\ell^m(\mathbf{t}_{c_a}) = \sum_{i \in \mathbf{c}_a} w_i \delta_\ell(\mathbf{t}_i).$$

From Theorem 1 in [1], the energy defined above can be transformed to the submodular quadratic pseudo-boolean function with two binary meta variables. In this form the  $\alpha\beta$ -swap algorithm can be used for minimizing the energy function.

## 1.2 Disintegration of Higher Order Energies to Second Order Sub-Modular Energies for Expansion Moves

Suppose, in a clique 'c', the location of the nodes with label  $\ell$  is represented by a set of indices  $\mathbf{c}_\ell$ . The current labeling solution is denoted by  $\mathbf{y}_c^o$ .

If the dominant label is denoted by  $d \in \mathcal{L}$  in the current labeling  $\mathbf{y}_c^o$  is,

$$\text{s.t } W(\mathbf{c}_d) > W(\mathbf{c}) - Q_d \quad \text{where } d \neq \alpha,$$

there must be one dominant label:

$$\begin{aligned}
 Q_a + Q_b &< W(\mathbf{c}) \quad \forall a \neq b \in \mathcal{L}, \\
 \psi_{\mathbf{c}}^m(t_c) &= \psi_{\mathbf{c}}(T_{\alpha}(\mathbf{y}_{\mathbf{c}}^{\circ}, t_c)) \\
 &= \min_{\ell \in \mathcal{L}} \left\{ \lambda_{max} - (\lambda_{max} - \lambda_{\alpha}) \exp\left(-\frac{\sum_{i \in \mathbf{c}} w_i t_i}{Q_{\alpha}}\right), \right. \\
 &\quad \left. \lambda_{max} - (\lambda_{max} - \lambda_d) \exp\left(-\frac{W(\mathbf{c}) - \sum_{i \in \mathbf{c}} w_i t_i}{Q_d}\right) \right\}.
 \end{aligned}$$

The minimization operator in the above function can be replaced by a piecewise function:

$$\psi_{\mathbf{c}}^m(\mathbf{t}_c, \mathbf{t}_{c_d}) = \begin{cases} \lambda_{max} - (\lambda_{max} - \lambda_{\alpha}) \exp\left(-\frac{n_0^m(\mathbf{t}_c)}{Q_{\alpha}}\right) & \\ \quad \text{if } n_0^m(\mathbf{t}_c) > \varrho_{\alpha d} \left( \frac{W(\mathbf{c})}{Q_{\alpha}} \right. & \\ \quad \quad \left. - \log\left(\frac{\lambda_{max} - \lambda_{\alpha}}{\lambda_{max} - \lambda_d}\right) \right), & \\ \lambda_{max} - (\lambda_{max} - \lambda_d) \exp\left(-\frac{W(\mathbf{c}) - n_0^m(\mathbf{t}_{c_d})}{Q_d}\right) & \\ \quad \text{if } n_0^m(\mathbf{t}_c) < \varrho_{\alpha d} \left( \frac{W(\mathbf{c})}{Q_{\alpha}} \right. & \\ \quad \quad \left. - \log\left(\frac{\lambda_{max} - \lambda_{\alpha}}{\lambda_{max} - \lambda_d}\right) \right), & \end{cases}$$

where,  $\varrho_{\alpha d} = \frac{Q_{\alpha} Q_d}{Q_{\alpha} + Q_d}$  and function  $n_{\ell}^m(\mathbf{t}_c)$  is defined as:

$$n_{\ell}^m(\mathbf{t}_c) = \sum_{i \in \mathbf{c}} w_i \delta_{\ell}(\mathbf{t}_i).$$

From Theorem 2 in [1], the energy defined above can be transformed to the submodular quadratic pseudo-boolean function with two binary meta variables. In this form the  $\alpha$ -expansion algorithm can be used for minimizing the energy function.

## 2 Plane Detection Algorithm

Algorithm 1 shows our region growing algorithm for depth based segmentation. Note that, point normals are found by local plane fitting and Singular Value Decomposition (SVD). In step 2,  $k_{sm}$  is the smoothing kernel and  $\mathbf{N}_{sm}$  is the smooth version of local normal image. In step 23, we also check if  $\mathbf{n}_{r-id_x}$  already been replaced by another region or not. If  $\mathbf{n}_{r-id_x}$  is found to be previously replaced, we use the updated value. In the following section, we list some example results of our algorithm.

### 2.1 More Example Results

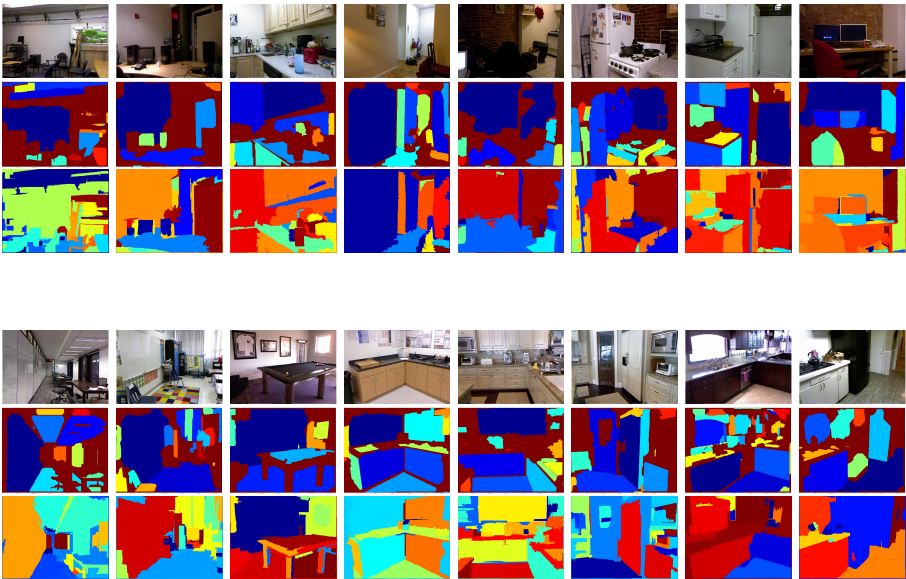
---

**Algorithm 1** Region Growing Algorithm for Depth Based Segmentation

**Input:** Point cloud =  $\{\mathbf{P}\}$ , Depth map =  $\{\mathbf{D}\}$ , RGB image =  $\{\mathbf{I}\}$ , Edge matching threshold  $e_{th}$ , Normalized boundary matching threshold  $b_{th}$

**Output:** Labeled planar regions =  $\{\mathbf{R}\}$

- 1: Calculate point normals:  $\{\mathbf{N}\} \leftarrow \mathcal{F}_{normal}(\mathbf{D})$
  - 2: Remove inconsistencies by low-pass filtering:  $\{\mathbf{N}_{sm}\} \leftarrow \mathbf{N} * k_{sm}$
  - 3: Cluster 3D points with similar normal orientations:  $\{\mathbf{N}_{clu}\} \leftarrow \mathcal{F}_{k-means}(\mathbf{N}_{sm})$
  - 4: Initialize:  $\mathbf{R} \leftarrow \mathbf{N}_{clu}$
  - 5: Line segment detector:  $\{\mathbf{L}\} \leftarrow \mathcal{F}_{LSD}(\mathbf{I})$
  - 6: Diffused line map:  $\{\mathbf{L}_{sm}\} \leftarrow \mathbf{L} * k'_{sm}$
  - 7: Identify planar regions with missing depth values:  $\{\mathbf{M}\} \leftarrow \mathcal{F}_{holes}(\mathbf{N}_{clu}, \mathbf{D})$
  - 8: Find adjacency relationships for each cluster in  $\mathbf{N}_{clu}$ :  $\mathbf{A}_{clu}$
  - 9: Identify all unique neighbors of clusters in  $\mathbf{M}$ :  $\mathbf{U}_{nb}$
  - 10: From  $\mathbf{U}_{nb}$ , separate correct and faulty clusters into  $\mathbf{N}_{cor}$  and  $\mathbf{N}_{inc}$  respectively
  - 11: Initialize available cluster list:  $\mathbf{L}_{avl} \leftarrow \mathbf{N}_{cor}$
  - 12: Initialize label propagation list:  $\mathbf{L}_{prp} \leftarrow \emptyset$
  - 13: **while** list  $\mathbf{L}_{avl}$  is not empty **do**
  - 14:     Randomly draw a cluster from available  $\mathbf{N}_{cor}$ :  $\mathbf{r}_{idx}$
  - 15:     Identify  $\mathbf{r}_{idx}$  neighbors ( $\mathbf{N}_{r-idx}$ ) with faulty depth values using  $\mathbf{A}_{clu}$  and  $\mathbf{M}$
  - 16:     **for** each neighbor  $\mathbf{n}_{r-idx}$  in  $\mathbf{N}_{r-idx}$  **do**
  - 17:         Find mutual boundary ( $\mathbf{b}_m$ ) of  $\mathbf{r}_{idx}$  and  $\mathbf{n}_{r-idx}$
  - 18:         Calculate edge strength at  $\mathbf{b}_m$  using  $\mathbf{L}_{sm}$ :  $e_{str}$
  - 19:         Calculate normalized boundary matching cost:  $b_{str} = \mathbf{b}_m / \text{Area of } \mathbf{n}_{r-idx}$
  - 20:         **if**  $e_{str} < e_{th} \wedge b_{str} > b_{th}$  **then**
  - 21:              $\mathbf{n}_{r-idx} \xrightarrow{\text{add}} \mathbf{N}_{cor}$ ,  $\mathbf{n}_{r-idx} \xrightarrow{\text{add}} \mathbf{L}_{avl}$
  - 22:              $\mathbf{r}_{idx} \xrightarrow{\text{rem}} \mathbf{L}_{avl}$ ,  $\mathbf{n}_{r-idx} \xrightarrow{\text{rem}} \mathbf{N}_{inc}$
  - 23:             Update  $\mathbf{L}_{prp}$  with  $\mathbf{r}_{idx}$  and  $\mathbf{n}_{r-idx}$
  - 24:          $\mathbf{r}_{idx} \xrightarrow{\text{rem}} \mathbf{L}_{avl}$
  - 25:     **for** any leftover clusters in  $\mathbf{N}_{inc}$  **do**
  - 26:         Randomly draw a cluster from available  $\mathbf{N}_{inc}$ :  $\mathbf{r}'_{idx}$
  - 27:         Execute similar steps (from line 15 to 24) for  $\mathbf{r}'_{idx}$
  - 28: Update  $\mathbf{R}$  according to  $\mathbf{L}_{prp}$
  - 29: **return**  $\{\mathbf{R}\}$
-



**Fig. 1.** Comparison of our algorithm (*last row*) with [2] (*middle row*) is shown. Note that the *maroon* color in middle row shows *non-planar* regions (but not in the last row where all regions are approximated by planes). The *last row* shows detected planes averaged over super-pixels.

# Bibliography

- [1] Kohli, P., Torr, P.H., et al.: Robust higher order potentials for enforcing label consistency. *IJCV* 82(3), 302–324 (2009)
- [2] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: *ECCV*, pp. 746–760. Springer (2012)