Hierarchical Planar Correlation Clustering for Cell Segmentation

Julian Yarkony[1], Chong Zhang[2], and Charless C. Fowlkes[3]

[1] Experian Data Lab, San Diego, CA
julian.e.yarkony@gmail.com

[2] CellNetworks, University of Hiedelberg, Germany
chong.zhang@iwr.uni-heidelberg.de

[3] Department of Computer Science
University of California, Irvine
fowlkes@ics.uci.edu

Abstract. We introduce a novel algorithm for hierarchical clustering on planar graphs we call "Hierarchical Greedy Planar Correlation Clustering" (HGPCC). We formulate hierarchical image segmentation as an ultrametric rounding problem on a superpixel graph where there are edges between superpixels that are adjacent in the image. We apply coordinate descent optimization where updates are based on planar correlation clustering. Planar correlation clustering is NP hard but the efficient PlanarCC solver allows for efficient and accurate approximate inference. We demonstrate HGPCC on problems in segmenting images of cells.

1 Introduction

We approach the problem of image segmentation in the framework of hierarchical segmentation where the goal is to group the pixels into a hierarchical structure where contiguous groups of pixels are divided and further subdivided. At the coarsest level of the hierarchy, all pixels are in the same region. At the finest level of the hierarchy each pixel is its own region. Each boundary that is present at a given level of the hierarchy is present in each finer level of the hierarchy.

Hierarchical segmentation can be understood as assigning confidence to various boundaries where boundaries present in coarser levels of the hierarchy are estimated to be more reliable. Hierarchical segmentation has been done primarily using agglomerative clustering, with the Ultrametric Contour Maps Algorithm being the state of the art (Arbelaez et al., 2011). Here we frame hierarchical segmentation as an ultrametric rounding problem (Ailon and Charikar, 2005; Yarkony, 2012).

Model the data to be clustered as the nodes of a graph where each pair of nodes is connected with an edge e that is associated with a real valued weight X_e. For any real value α let $Y_e^\alpha := [X_e \geq \alpha]$ where [ ] is the indicator function. Now consider the unweighted graph G^α with edges connecting nodes only if $Y_e^\alpha = 0$. If X is an ultrametric then for all α and e, $Y_e^\alpha = 0$ if and only if the pair of nodes connected by e are in the same component of G^α. Ultrametrics define a natural model of hierarchical grouping where the threshold α specifies the level of the hierarchy. If α is large then G^α has few regions while if α is small it has many regions. Edges present in G^α are present in $G^{\alpha+v}$ for all $v > 0$.

Given an initial graph and set of (sparse) edges where each edge e is associated with a real valued target T_e, the objective of ultrametric rounding is to assign a new set of values $\{X_e\}$ to the edges which satisfies the property of being an ultrametric and is minimally distorted from the targets (either in an L^1 or L^2 sense). In our application nodes correspond to superpixels and edges indicate adjacency. Superpixels (Ren and Malik, 2003) are small compact groups of pixels which can be produced by various approaches. Superpixels are the most elementary unit in our hierarchical segmentation approach. We connect neighboring superpixels with an edge and associated score T_e that defines how strong the image boundary is locally between the two superpixels. Large T_e are associated with stronger visual indications of an edge between the superpixels connected by edge e. The goal of finding the closest ultrametric X to T can thus be interpreted as finding a hierarchical segmentation which is consistent with the local evidence encoded in T. Edges only connect nodes whose corresponding superpixels are immediately adjacent in the image. Thus our graph is planar which allows for many computational advantages which is the focus of this paper.

We focus on the application of segmenting cells in biological images. Cell segmentation is one of the prerequisite tasks in answering many biological questions related to both basic understanding of cell function and interpretation of pathological states. Recent emerging research efforts in diverse cell lines and microscopic imaging techniques require robust and automatic algorithms for performing segmentation, particularly in high-throughput experiments. While cell imaging with fluorescent labels or other chemical staining can provide contrast on objects of interest for easy segmentation, it is not ideal for studying cells under natural conditions. Without such dyes, cells are much harder to segment. Cells in brightfield or phase contrast images are only distinguishable by their outer membrane. Other major challenges of segmenting cells from these images are: touching cells, weak or broken boundaries, large variations on boundary pattern, and false boundaries due to artifacts or other sub-cellular structures.

2 Related Work

2.1 Related Work on Clustering

Hierarchical clustering has been considered since early machine learning. Agglomerative clustering is the primary way in which this has been approached in the domain of computer vision. The seminal ultrametric contour maps algorithm (UCM) (Arbelaez et al., 2011) is the clearest application of this approach in image segmentation. UCM associates with each pair of superpixels i, j a distance metric D_{ij}. D_{ij} is initially a function of image features. UCM initializes each superpixel as an independent region. UCM proceeds by merging the pair of adjacent regions whose average distance metric between superpixels across the boundary is minimal. Usually this is average weighted by the length l_{ij} of the boundary between the superpixels. Let $B(Q_1, Q_2)$ be the set of edges between the superpixels making up region Q_1 and Q_2. The weighted average distance is computed as:

$$\bar{D}(Q_1, Q_2) = \frac{\sum_{[i,j] \in B(Q_1,Q_2)} l_{ij} D_{ij}}{\sum_{[ij] \in B(Q_1,Q_2)} l_{ij}} \tag{1}$$

$$\tag{2}$$

In the UCM algorithm, when two regions Q_1 and Q_2 are merged, each edge between the superpixels spanning across the two regions is set to the average value $\bar{D}(Q_1, Q_2)$. This assure that the resulting set of distances forms an ultrametric. UCM continues grouping the pair of regions whose average distance is minimal until all superpixels are in the same region. UCM is a fast greedy method which is quite successful but does not claim to minimize the ultrametric distortion.

Ultrametric rounding for image segmentation has been explored in a regime in which each X_e may only take on a set of fixed discrete values (Yarkony, 2012) using the formulation of (Ailon and Charikar, 2005). Our work significantly departs from this line as it does not restrict X to take on a set of discrete values.

2.2 Related Work on Cell Segmentation

Many efforts have been devoted very recently in cell segmentation based on boundaries. In (Liu et al., 2014) cell segments are selected from a UCM-based hierarchical segmentation region candidates through an integer linear programming (ILP) formulation. Each region candidate has a score predicted from SVM classifier, that takes part of its input from a cell contour shape model. This technique tries to find the best segmented cells from multiple hierarchical layers. However, the dependency on a common cell shape may not likely to apply this technique on cells evolve or deform such as the fibroblast cells in (Wu et al., 2012). However, the fact that in (Wu et al., 2012) the segmentation is formulated as a partial matching problem between cell boundaries obtained from consecutive frames in time-lapse images limits its applicability to static images. An interactive cell segmentation approach to correct erroneous segmentation is proposed very recently (Su et al., 2014). It uses an augmented affinity graph to efficiently incorporate and propagate corrected labels for an updated partitioning of the superpixels. But this method explicitly uses phase retardation features (Su et al., 2013) to generate superpixels so as to enable efficient corrections on superpixel level. Yet another method in (Zhang et al., 2014a) combines detection of cell centers and clustering cell boundary points in an ILP fashion. But this method is primarily designed for cells with convex shapes with similar sizes.

3 Ultrametric Rounding

We start by formulating ultrametric rounding as an optimization problem. Consider a graph G with edges indexed by e. G is often a sparse graph meaning that most pairs of nodes are not connected. We denote the desired ultrametric as X which is indexed by e. Here we assume X_e is a real valued in the range $[0, 1]$. For X to be an ultrametric it must be the case that if we remove the set of edges for which X_e is greater than any given value α we do not remove any edges within a connected component of the resulting graph. This can be enforced by the constraint that for any cycle C in our graph containing an edge e between adjacent superpixels separated by a boundary (a pair where $X_e \geq \alpha$), at least one other boundary is present along every cycle C connecting them. We write this as:

$$\sum_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall C \in \text{Cycles} : \hat{e} \in C \tag{3}$$

where [ ] to denotes the indicator function whose value is 1 if the condition is true and otherwise outputs a zero. An equivalent definition is that for an edge in a cycle there must be at least one other edge in the cycle whose value is as large or larger.

$$\max_{e \in C - \hat{e}} X_e \geq X_{\hat{e}} \quad \forall C \in \text{Cycles} : \quad \hat{e} \in C \tag{4}$$

We call the above inequalities "ultrametric inequalities". Each edge e is associated with a target value $T_e \in [0, 1]$. Finding the ultrametric X closest to T in an L^p sense (p is 1 or 2 depending on the desired norm) is the objective of ultrametric rounding. We write the optimization problem below.

$$\min_X \sum_e |X_e - T_e|^p \tag{5}$$

$$\text{s.t.} \max_{e \in C - \hat{e}} X_e \geq X_{\hat{e}} \quad \forall \{C \in \ \text{Cycles} : \ \hat{e} \in C\} \tag{6}$$

3.1 Correlation Clustering

When constructing our solver for minimizing ultrametric distortion we rely heavily on repeated calls to a solver for correlation clustering on a planar graph. Thus we now briefly discuss correlation clustering (Bansal et al., 2002; Kim et al., 2011; Yarkony et al., 2012; Bagon and Galun, 2011; Andres et al., 2012, 2013, 2011). Correlation clustering is a powerful clustering criteria in which each pair of nodes (in our case adjacent superpixels) is associated with a real valued term θ_e where e indexes the edge between the two nodes. Correlation clustering groups the nodes into regions so as to minimize the sum of the θ_e terms of edges spanning the boundary. We define the presence of a boundary using binary indicator vector Y which is indexed by e. Here $Y_e = 1$ if and only if there is a boundary on edge e. Notice that if $\theta_e > 0$ then it is desirable to set $Y_e = 0$ and if $\theta_e < 0$ it is desirable to set $Y_e = 1$. However Y has to be set so that a clustering is produced. This means that no Y_e can be set to 1 in the middle of a region. These constraints are called cycle inequalities and they are the discrete binary analog of the ultrametric inequalities in Eq 3, 4. The form of cycle inequalities are written below.

$$\sum_{e \in C-\hat{e}} Y_e \geq Y_{\hat{e}} \quad \forall \{C \in \text{Cycles},\ \hat{e} \in C\} \tag{7}$$

Correlation clustering is a natural clustering criteria because the number of regions is not a user defined hyper-parameter that must be hand tuned for each problem. Instead it is a function of the potentials θ themselves. Notice that if θ is exclusively positive then all superpixels are in the same region in the optimal solution. Also notice that if all θ terms are negative then each superpixel is in its own region in the optimal solution.

Solving the correlation clustering problem is NP hard even for planar graphs (Bachrach et al., 2011). However for many problems in computer vision the PlanarCC algorithm (Yarkony et al., 2012) can solve them exactly usually in seconds or fractions of seconds. PlanarCC is a dual column generation algorithm operating only on planar graphs. PlanarCC provides upper and lower bounds on the optimal value of the objective. The upper bound is associated with a partition Y that achieves this value. In practice the upper and lower bounds are identical or nearly identical for problems in the domain of image segmentation (Yarkony et al., 2012) meaning that the solution is verified to be the global optima. PlanarCC provides fast performance for image segmentation problems in computer vision notably on the benchmark Berkeley Segmentation Data Set (BSDS)(Martin et al., 2001).

4 The Hierarchical Greedy Planar Correlation Clustering Algorithm (HGPCC)

We now consider the problem of minimizing ultrametric distortion as in Eq 5. We employ a coordinate descent approach in which at each step we identify optimal setting of X in a particular space that includes that current solution. We alternate between three unique coordinate descent steps which are described below. When we apply an update we denote the current setting of our solution as X^0, and the output as X^1. We initialize X^0 to be the zero vector. At all times during our algorithm our solution describes an ultrametric. Two out of the three coordinate updates use the PlanarCC algorithm which requires planarity of the graph in order to work. To satisfy planarity in our application we have edges between each adjacent pair of superpixels and no other edges.

4.1 Update One: Shifting the Values in the Ultrametric While Preserving their Order

Consider optimizing over X subject to the constraint that the ordering of X does not change. We frame this as an optimization problem which is potentially a linear or quadratic program depending on the norm applied on the ultrametric.

$$\min_X \sum_e |T_e - X_e|^p \tag{8}$$
$$\text{s.t. } X_e \geq X_{\hat{e}} \qquad \forall e,\ \hat{e}\ :\ X^0_e \geq X^0_{\hat{e}}$$

Let Λ_b be the set of edges that take on the b'th smallest unique value specified by X^0. Our goal is to find new values λ_b to assign to each set of edges Λ_b. Let $|\lambda|$ denote the the number of unique values in X^0 and $|\Lambda_b|$ the cardinality of Λ_b.

$$\min_\lambda \sum_b \sum_{e \in \Lambda_b} |T_e - \lambda_b|^p = \min_\lambda \sum_b |\Lambda_b||T_e - \lambda_b|^p \tag{9}$$
$$\text{s.t. } \lambda_b \leq \lambda_{b+1}$$

In addition to solving the optimization above as a linear/quadratic program we can approach it as a dynamic program on a chain structured Markov random field. For each variable λ_b we create a node that has cost to take on each possible value α_b of $Z_b(\alpha_b)$ which is defined below.

$$Z_b(\alpha_b) = \sum_{e \in \Lambda_b} |T_e - \alpha_b|^p = |\Lambda_b||T_e - \alpha_b|^p \tag{10}$$

We also have a pairwise potential over each pair of adjacent b values $Z_{b,b+1}(\alpha_b, \alpha_{b+1})$ which is defined below.

$$Z_{b,b+1}(\alpha_b, \alpha_{b+1}) = \infty[\alpha_b > \alpha_{b+1}] \tag{11}$$

This pairwise potential simply enforces that the ordering of the values of b remains constant. We discretize the space of possible values for λ terms making sure to include all unique values in X^0. For example we can include 1000 uniformly distributed points between $\min(T)$ and $\max(T)$ in addition to all unique values of X^0. We denote the set of all such values as Ω.

Computing the optimal λ in the above graphical model can be done using dynamic programming in time $O(|\Omega||\lambda|)$. Once we solve for λ we simply set each index of X to its associated value in λ. Thus X^1_e is set to λ_b $\forall b, \forall e \in \Lambda_b$.

4.2 Update Two: Raising the Values of X to α in Large Groups

We now introduce a coordinate update that raises the values in X in large groups over long ranges of value while preserving the ultrametric property of X. This is a coordinate update parameterized by a randomly chosen value α on the range of $[\min(T), \max(T)]$. Here α is different every time this update is done.

During this update we optimize X over the space of ultrametrics subject to the constraint that $X_e \in \{X^0_e, \max(X^0_e, \alpha)\}$ for all e. We denote this space as $\hat{S}(X^0, \alpha)$ and the super-space that does not enforce the ultrametric property as $S(X^0, \alpha)$. We now write the objective of this update formally.

$$X^1 = \arg \min_{X \in \hat{S}(X^0, \alpha)} \sum_e |T_e - X_e|^p \tag{12}$$

Notice that in the space $S(X^0, \alpha)$ the only possible violations to the ultrametric property come in the form of ultrametric inequalities of pairs of cycle C, and edge $\hat{e}$ such that: $\forall e \in C,\ \ X^0_e < \alpha$ and $X_{\hat{e}} = \alpha$.

Using this we write a version of the ultrametric inequalities needed to ensure that any $X \in S(X^0, \alpha)$ also lies in $\hat{S}(X^0, \alpha)$.

$$\max_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall \{C \in \text{ Cycles }, \hat{e} \in C\} \tag{13}$$

Notice that we can replace the max in the above equation with a $\sum$. This is because each of the inequalities can be only violated if all terms under the sum/max are zero.

$$\sum_{e \in C - \hat{e}} [X_e \geq \alpha] \geq [X_{\hat{e}} \geq \alpha] \quad \forall \{C \in \text{ Cycles } \hat{e} \in C\} \tag{14}$$

We write our coordinate update as an instance of correlation clustering. We use binary indicator Y_e as an indicator for $[X_e \geq \alpha]$ and edge potentials θ given by:

$$\theta_e = \begin{cases} |\alpha - T_e|^p - |X^0_e - T_e|^p & \forall\ e\ s.t.\ X^0_e < \alpha \\ -\infty & o.w. \end{cases}$$

where edges with potential $-\infty$ are required to be active in the final solution.

The resulting correlation clustering problem is then

$$\min_Y \theta_e Y_e \tag{15}$$

$$\text{s.t.} \sum_{e \in C - \hat{e}} Y_e \geq Y_{\hat{e}} \quad \forall (C \in \text{Cycles},\ \hat{e} \in C) \tag{16}$$

After computing Y we simply set $X^1_e \leftarrow \alpha$ iff ($Y_e = 1$ and $X^0_e < \alpha$); otherwise set $X^1_e \leftarrow X^0_e$.

Implementation Detail Since we already established that no edge e s.t. $X^0_e \geq \alpha$ is involved in any necessary ultrametric inequality in $S(X^0, \alpha)$ and their θ terms are negative then we can simply remove (ignore) those edges from the graph and set their values in X^1 to X^0. This saves us from having $-\infty$ as the value of an edge potential.

Another way of ignoring edges such that $X^0_e \geq \alpha$ is as follows. For each such edge set $\theta_e = 0$. Next then solve for Y. Finally set $X^1_e \leftarrow X^0_e$ for all such edges. We use this approach as it avoids instantiating multiple graph structures.

4.3 Update Three: Lowering the Values of X for a Subset of X

We now discuss a coordinate update that lowers the values in X for all values that take on a unique value in X so as to reduce the ultrametric distortion of X. This update parameterized by a randomly chosen value α on the range of $[\min(T), \max(X))$. Here α is different every time we perform this update. Let the set of all unique values in X^0 be denoted λ^0. Here λ^0 is sorted with λ^0_0 being the smallest and $\lambda^0_{|\lambda|}$ being the greatest. Let μ be the smallest value in λ^0 greater than α.

We optimize over the space of solutions in which each X_e such that $X_e^0 = \mu$ may take on either α or μ and all X_e such that $X_e^0 \neq \mu$ must continue to take on their current value. We denote the space of solutions that meet these properties as $V(X^0, \alpha)$ and the subset of that space corresponding to ultrametrics as $\hat{V}(X^0, \alpha)$. We now formally write the optimization over the space $\hat{V}(X^0, \alpha)$.

$$X^1 = \arg \min_{X \in \hat{V}(X^0, \alpha)} \sum_e |T_e - X_e|^p \tag{17}$$

The ultrametric inequalities needed to enforce that an $X \in V(X^0, \alpha)$ is also in $\hat{V}(X^0, \alpha)$ are written below.

$$\max_{e \in C - \hat{e}} [X_e > \alpha] \geq [X_{\hat{e}} > \alpha] \quad \forall \{C \in \text{Cycles } \hat{e} \in C\} \tag{18}$$

As in the previous subsection (see the transition from Eq 13, to Eq 14) we can replace the max with a $\sum$ allowing us to write our coordinate update as an instance of correlation clustering with Y_e as an indicator for $[X_e > \alpha]$. The correlation clustering objective is described by the potentials

$$\theta_e = \begin{cases} -|\alpha - T_e|^p + |X_e^0 - T_e|^p. \quad \forall e \text{ s.t. } X_e^0 = \mu \\ -\infty \quad \forall e \text{ s.t. } X_e^0 > \mu \\ \infty \quad \forall e \text{ s.t. } X_e^0 \leq \alpha \end{cases}$$

where the edges with negative and positive infinite weights are required to be cut or not cut respectively.

After solving the optimization above we simply set $X_e^1 \leftarrow \alpha$ iff ($Y_e = 0$ and $X_e^0 = \mu$); otherwise set $X_e^1 \leftarrow X_e^0$. Note that this operation can be performed in parallel with a unique value α chosen between each pair of adjacent μ. As in the previous section we can ignore the edges that must be boundaries in the solution meaning ($X_e^0 > \alpha$) as they are not involved in any violated cycle inequalities and furthermore must be set to 1. Ignoring them is done by setting their θ value to zero. Similarly we can merge any superpixels that are connected by an ∞ valued potential. Merging superpixels was not done in our experiments but can conceivably make inference faster.

4.4 Final Procedure

Updates can be performed in any order. Furthermore one can complete multiple updates of one type in a row. For our experiments we consider one iteration to be completing updates 1,2,1,3. We repeat this iteration many times in our experiments.

4.5 Optimality in PlanarCC

For our experiments we used the PlanarCC code provided by the authors of (Yarkony et al., 2012). We operated this code unchanged. PlanarCC attacks an NP hard problem so it is conceivable that its lower and upper bounds are not tight at convergence or when a user would want an anytime solution. Thus when we terminate PlanarCC which we run for no more than a minute we take the best anytime solution generated (including the solution corresponding to the initial solution). We never saw this time limit reached.

5 Experiments

In order to evaluate the generality and robustness of our approach, we test it on datasets that differ in sample preparation and imaging equipment and conditions.

Data set one: These are bright field Diploid yeast cell images from (Zhang et al., 2014a), in which both out-of-focus and in-focus cells exist and are cluttered together. And the cells of interest are only the in-focused ones, i.e. those with least contrast on cell boundaries. Apart from this, cell boundaries can be partially missing and with diverse appearances, even in the same cell.

Data set two: These are phase-contrast HeLa cell images from (Arteta et al., 2012). It presents a high variability in cell shapes and sizes, as opposed to the ellipse like cells in data set one. These images have relatively lower resolution, where cell boundaries are disturbed by the bright halo owing to this specific imaging technique.

5.1 Producing Problem Instances

The edge probability map is predicted from a trained classifier using ilastik (Sommer et al., 2011), an open-source toolkit that relies on a family of generic nonlinear image features and random forests, to estimate the probability of belonging to a cell boundary edge for each individual pixel. We use a small labeled training data set.

To compute superpixels we use a watershed transformation then smooth the result using a gaussian filter. Finally we compute the average boundary probability along each superpixel boundary thus providing a value Pb_e for every edge e. UCM operates on this raw probability. We take the log odds ratio to convert that to an energy which is then used as the targets for HGPCC. The equation for the targets is written below.

$$T_e = -\log(\frac{1-Pb_e}{Pb_e}) \tag{19}$$

For HGPCC we experimented with $L1$ and $L2$ norms in the log odds ratio space. In Fig 1 we display the results of UCM and of HGPCC for an image in

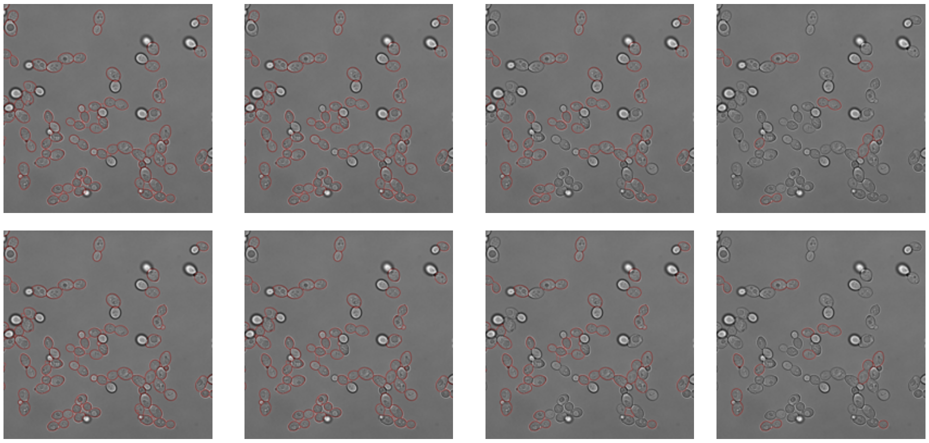

Fig. 1. Top Row: UCM segmentation thresholding X at values various thresholds. Bottom Row: HGPCC segmentation thresholding X at values at the same thresholds as UCM. We indicate boundaries in red.

the data set one. Once the nearest ultrametric to T is solved for in log odds space we convert X to a probability by a sigmoid operation.

With regards to the quality of the segmentations we found no significant qualitative difference between UCM and HGPCC. That being said HGPCC has multiple advantages over UCM. First it is an energy minimization formulation which allows for structured learning and principled mathematical extensions to be used. Second HGPCC is robust to indications of no boundaries being placed on actual boundaries, which may result in the merging of these boundaries at finer positions in the hierarchy for UCM than desirable.

5.2 Experimental Comparisons: Distortion and Timing

For problems in data set one and data set two we completed 500 iterations of HGPCC. For each iteration we completed updates 1,2,1,3 in that order. We found that HGPCC converged very rapidly. Furthermore the time to complete an iteration of HGPCC decreases at first then after convergence begins to increase again. We compared against the UCM algorithm which since it is not an iterative algorithm was not timed. It is very fast compared to our approach. We found that HGPCC produces lower distortion ultrametrics than UCM very early during optimization.

When plotting the distortion we applied the following normalization scheme. All distortions including the output of UCM are normalized by subtracting off the lowest value of HGPCC for a given instance and dividing by the gap between the lowest and highest distortions of HGPCC for a given instance. All results are averaged across the data sets. All results are plotted in Fig 2.

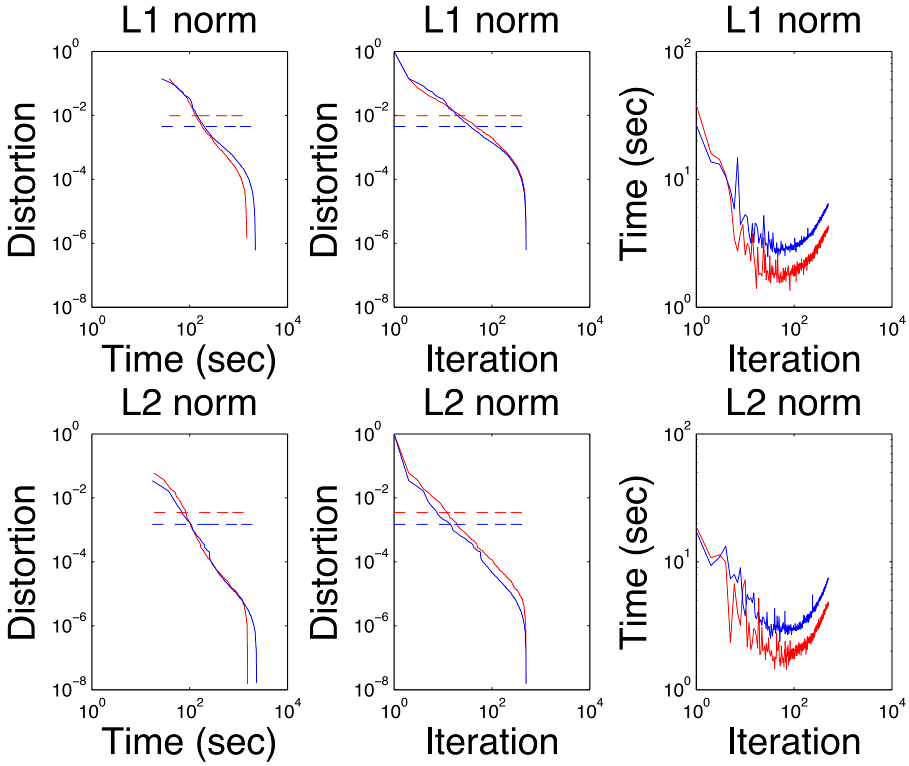

Fig. 2. We show the convergence of HGPCC as a function of time and iteration and compare it to the final result of UCM (which is not timed). We use the data set one and data set two and color their results red and blue respectively. Dotted lines correspond to UCM and solid lines to HGPCC. Left Column) Distortion as a function of time. Center Column) Distortion as a function of iteration. Right Column) Time for an iteration of HGPCC as a function of iteration.

6 Conclusion

We present a novel fast algorithm for finding low distortion ultrametrics on planar graphs. Our method exploits the fact that correlation clustering can often be done efficiently on planar graphs with very high degrees of accuracy. Our method is an analog of alpha expansion/alpha beta swap (Boykov et al., 2001) as both make large efficient moves in the space of values for their variables. This work extends the family of PlanarCC (Yarkony et al., 2012; Andres et al., 2013; Zhang et al., 2014b) methods so as to include efficient hierarchical clustering.

7 Acknowledgement

We thank F. Huber and M. Knop from ZMBH University of Heidelberg, Germany for sharing the bright field images.

Bibliography

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 73–82, 2005.

Julian Yarkony. *MAP Inference in Planar Markov Random Fields with. Applications to Computer Vision*. PhD thesis, University of California Irvine, 2012.

Xiaofeng Ren and J. Malik. Learning a classification model for segmentation. In *Ninth IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 10–17 vol.1, Oct 2003.

Fijun Liu, Fuyong Xing, and Lin Yang. Robust muscle cell segmentation using region selection with dynamic programming. In *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pages 1381–1384, 2014.

Zheng Wu, Danna Gurari, Joyce Wong, and Margrit Betke. Hierarchical Partial Matching and Segmentation of Interacting Cells. In *Fifteenth Annual Medical Information Computing and Computer Assisted Intervention (MICCAI, 2012)*, volume 7510, pages 389–396, 2012.

Hang Su, Zhaozheng Yin, Takeo Kanade, and Seungil Hun. Interactive cell segmentation based on correction propagation. In *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pages 1267–1270, 2014.

Hang Su, Zhaozheng Yin, Seungil Hun, and Takeo Kanade. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical Image Analysis*, 17:746–765, 2013.

Chong Zhang, Florian Huber, Michael Knop, and Fred . A. Hamprecht. Yeast Cell Detection and Segmentation in Bright Field Microscopy. In *Eleventh IEEE International Symposium on Biomedical Imaging (ISBI 2014)*, pages 1267–1270, 2014a.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *Journal of Machine Learning*, pages 238–247, 2002.

Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang Dong Yoo. Higher-order correlation clustering for image segmentation. In *Advances in Neural Information Processing Systems,25*, pages 1530–1538, 2011.

Julian Yarkony, Alexander Ihler, and Charless Fowlkes. Fast planar correlation clustering for image segmentation. In *Proceedings of the 12th European Conference on Computer Vision(ECCV 2012)*, 2012.

Shai Bagon and Meirav Galun. Large scale correlation clustering 816 optimization. In *CoRR, abs/1112.2903*, 2011.

Bjoern Andres, Thorben Kroger, Kevin L. Briggman, Winfried Denk, Natalya Korogod, Graham Knott, Ullrich Kothe, and Fred. A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *Proceedings of the Twelveth International Conference on Computer Vision (ECCV-12)*, 2012.

Bjoern Andres, Julian Yarkony, B. S. Manjunath, Stephen Kirchhoff, Engin Turetken, Charless Fowlkes, and Hanspeter Pfister. Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs. In *Proceedings of the Ninth Conference on Energy Minimization in Computer Vision and Pattern Recognition (EMMCVPR-13)*, 2013.

Bjoern Andres, Joerg H. Kappes, Thorsten Beier, Ullrich Kothe, and Fred A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV-11)*, pages 2611–2618, 2011.

Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov, and Morteza Zadimoghaddam. Optimal coalition structures in graph games. *CoRR*, abs/1108.5248, 2011.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference on Computer Vision (ICCV-01)*, pages 416–423, 2001.

Carlos. Arteta, Victor. Lempitsky, J. Allison. Noble, and Andrew Zisserman. Learning to Detect Cells Using Non-overlapping Extremal Regions. In *Fifteenth Annual Medical Information Computing and Computer Assisted Intervention (MICCAI, 2012)*, volume 7510, pages 348–356, 2012.

Christoph Sommer, Christoph Straehle, Ullrich Kothe, and Fred A. Hamprecht. "ilastik: Interactive learning and segmentation toolkit". In *Eighth IEEE International Symposium on Biomedical Imaging (ISBI 2011)*, 2011.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

Chong Zhang, Julian Yarkony, and Fred A. Hamprecht. Cell detection and segmentation using correlation clustering. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2014*, volume 8673, pages 9–16. 2014b.