

Image Segmentation by Weighted Aggregation with Gradient Orientation Histograms

Rory Neilson

Bashan Naidoo
Member, SAIEE

Stephen McDonald
Member, IEEE

Department of Engineering, University of KwaZulu-Natal
School of Electrical Electronic and Computer Engineering
Durban, South Africa
{neilson bnaido mcdonalds}@ukzn.ac.za

Abstract—This paper reviews the Segmentation by Weighted Aggregation (SWA) image segmentation scheme developed in [1-3] and proposes the use of gradient orientation histograms (GOH) to further differentiate between segment textures.

The SWA approach uses an Algebraic Multi-Grid inspired technique to approximate regions that correspond to a minimum in a normalised cuts cost function. The SWA scheme is a multi-scale approach, decomposing an image into a hierarchical pyramid structure, allowing the detection of segments of all scales. The SWA algorithm allows the incorporation of a multitude of aggregate properties to better discriminate segments. This paper outlines the use of GOH's in this respect as opposed to the traditional shape moments.

Index Terms—Image Segmentation, Normalised Cuts Segmentation, Salient Feature Extraction, Segmentation by Weighted Aggregation,

I. INTRODUCTION

The task of image segmentation is the grouping of pixels into regions that are distinct in some way from their immediate surroundings. There are many ways to define this distinctiveness. A popular approach is to define distinct or salient regions as those that correspond to a minimum in a normalised cuts [4] cost function. The normalised cuts cost of a region may be informally defined as the ratio of the *similarity* between adjacent pixels across its boundaries to the total *similarity* between adjacent pixels within its interior. This approach produces regions whose properties vary smoothly within their interior but bounded by relatively sharp changes in these properties.

An approach known as Segmentation by Weighted Aggregation (SWA), pioneered by *Sharon, Brandt* and *Basri* in [2] and further developed in [1-3], provides an efficient method to approximate such regions.. The SWA algorithm is hierarchical in nature, inspired by Algebraic Multi-Grid (AMG) solvers used for the numerical simulation of physical systems, it adaptively decomposes an image into an irregular pyramid structure. This hierarchical structure allows the incorporation of not only the fine level differences between individual pixels but of higher-level aggregate statistics to further distinguish between regions. This makes the SWA algorithm an intrinsically multi-scale

approach. However unlike solely filter based multi-scale approaches where statistics mix across region boundaries, SWA is adaptive, confining the calculation of statistics to within segments. These statistics may include any number of aggregate properties. Properties such as mean intensity, intensity variance or more complex filter responses may be used to characterise region texture. Region shape may be captured with the use of various shape moments. Since these statistics are averages they may be efficiently computed recursively, accumulating from level to level. The possibility for a variety of types of aggregate properties makes SWA a versatile framework that may be tailored to specific applications with the choice of a particular set of properties. The particular SWA implementation described in this paper makes use of a set of aggregate properties that include intensity mean and variance as suggested in [3] however instead of using low order shape moments [3] a histogram of gradient orientations within the aggregate is used. This choice of a Gradient Orientation Histogram (GOH) was inspired by the success of *David Lowe's* SIFT descriptor [5] and the related GLOH descriptor [6].

This paper reviews the structure of the SWA algorithm as developed in [3], and investigates the use of gradient orientation histograms as higher level statistics as opposed to shape moments [3] in measuring the similarity between regions.

II. PROBLEM DEFINITION

Graph theory provides a particularly useful framework to formulate segmentation problems as segmentation deals with both pixels (nodes) and the relation between them (edges). Within this framework an image is viewed as a large undirected edge weighted graph, $G=(V,E)$, where nodes, V , correspond to pixels and edges, E , connect neighbouring pixels with weights w_{ij} , corresponding to a similarity measure between adjacent pixels, i and j .

If a region is defined by state vector \mathbf{u} where $u_i=1$ if pixel i is part of the region and $u_i=0$ if it is not, the normalised cuts cost function [3] can be expressed as follows:

$$\Gamma(\mathbf{u}) = \frac{\sum_{i>j} w_{ij} (u_i - u_j)^2}{\sum_{i>j} w_{ij} u_i u_j} = \frac{\mathbf{u}^T \mathbf{L} \mathbf{u}}{\frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u}} \quad (1)$$

Where W is a sparse symmetric matrix with elements w_{ij} , known as the graph's similarity matrix and $L = D - W$

is the graph's Laplacian matrix where D is the diagonal degree matrix with $\text{diag}(D) = W\mathbf{1}$.

If the requirement that \mathbf{u} be a binary vector is relaxed to allow real values, the problem of finding these regions can be formulated as a generalised Eigen problem.

$$L\mathbf{u} = \lambda W\mathbf{u} \quad (2)$$

Unfortunately the computational cost of a direct solution to this problem increases exponentially with image size [4], precluding its direct application to all but the smallest images. However an approach originally developed for uses within solvers for physical simulations, known as the Algebraic Multi-Grid (AMG) allows for an efficient approximation to such solutions. The AMG approach recursively approximates the problem with coarser and coarser graphs that may be solved and interpolated back as an approximate solution to the original problem.

III. WEIGHTED AGGREGATION

The AMG approach uses a technique known as weighted aggregation to successively coarsen the graph so producing ever-simpler approximations to the initial graph.

Given an initial graph of N nodes, indexed $\{i_1, i_2, \dots, i_N\}$ the coarsening process begins by selecting a subset of representative nodes, known as seeds, $C = \{c_1, c_2, \dots, c_{\hat{N}}\}$, where $\hat{N} \approx N/2$. These seeds are selected such that every remaining node is strongly *coupled* to at least one seed. These seeds become the nodes of a new coarser graph. With a suitable selection of seeds it is possible to approximate a state vector $\mathbf{u} = (u_1, u_2, \dots, u_N)^T$ at the fine level by a state vector $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{\hat{N}})^T$ at the coarser level through a sparse $N \times \hat{N}$ interpolation matrix P .

$$\mathbf{u}; P\hat{\mathbf{u}} \quad (3)$$

This interpolation relation allows for the transfer of information between the fine graph and its course approximation. There are a number of approaches to defining this interpolation matrix P . A popular choice is:

$$p_{ij} = \begin{cases} 1 & i = c_j \\ w_{ij} / \sum_{k \in C} w_{ik} & i \notin C \\ 0 & i \in C, i \neq c_j \end{cases} \quad (4)$$

This interpolation matrix essentially defines non-seed nodes, which have no direct counterpart in the coarse graph, as a weighted combination of their neighbouring seeds. This may be viewed as a weighted aggregation of nodes where p_{ij} denotes the probability that node i in the fine graph, belongs to the aggregate represented by node j in the coarser graph.

By substituting the interpolation relation (3) into the cost function (1) you get the following approximate normalised cuts cost function.

$$\Gamma(\mathbf{u}); \hat{\Gamma}(\hat{\mathbf{u}}) = \frac{\hat{\mathbf{u}}^T P^T L P \hat{\mathbf{u}}}{\frac{1}{2} \hat{\mathbf{u}}^T P^T W P \hat{\mathbf{u}}} \quad (5)$$

From the denominator of this expression it follows that the edge weights of course problem may be approximated as

follows:

$$\hat{W} = P^T W P \quad (6)$$

The numerator may be similarly approximated in terms of the new coarse edge weights. Note that unlike W , the diagonal elements of \hat{W} will in general be non-zero and correspond to the internal coupling within the aggregates.

This coarsening is repeated recursively, each time halving the number of nodes until a single node remains that represents the entire image. This process results in an irregular pyramid structure of graphs.

IV. THE ALGORITHM

The SWA algorithm can be separated into two processes, the bottom up construction of the pyramid of graphs followed by top down interpolation of salient nodes through the pyramid to delineate salient regions.

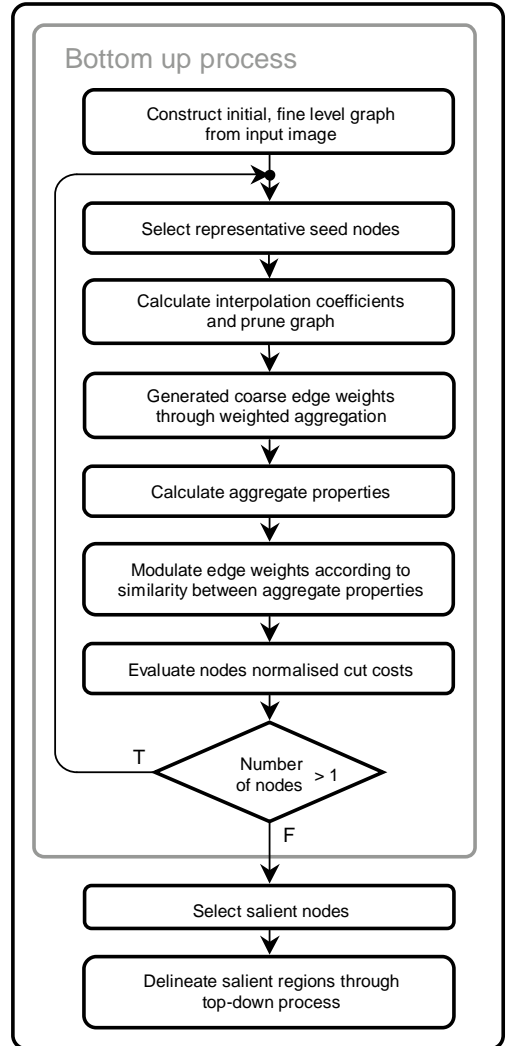


Fig. 1. Flow chart of SWA algorithm structure

The number of elements in the W and P matrices scale quadratically with image size. This makes the explicit storage of every element impractical. However W and P are sparse with non-zero elements only occurring in positions corresponding to edges. It is thus computationally useful to formulate the algorithm in terms of nodes and

their neighbours. It is convenient to introduce *neighbourhood* notation where $N_i = \{j \neq i : w_{ij} \neq 0\}$ denotes the set of nodes connected to node i . Similarly $C_i = N_i \cap C_i$ denotes the set of seeds connected to node i .

A. Bottom-up Construction process

The bottom-up construction process begins with the construction of an initial base graph directly from the input. Where each pixel corresponds to a node and each node is connected to the four nodes corresponding to neighbouring pixels. The initial edge weight between adjacent nodes, i and j , is defined as :

$$w_{ij} = e^{-\alpha |I_i - I_j|} \quad (7)$$

where I_k is the image intensity associated with pixel k and α is a globally set positive constant. The base aggregate properties are also initialised. This initial graph is then recursively coarsened and properties aggregated to successively form the levels of an irregular pyramid. This coarsening procedure may be divided into the following steps:

- Seed selection
- Interpolation coefficient calculation and graph pruning
- Coarse edge weight generation
- Aggregate property calculation
- Coarse edge weight modulation
- Normalized cuts cost calculation

The following sub-sections describe in detail each of these processes.

1) Seed Selection

The seeds are required to be selected in such a way that all remaining nodes are strongly coupled to them. Thus the coupling of non-seed nodes to seeds must be an appreciable portion of their total coupling.

The selection of seeds is done in a sequential manner. The nodes are first sorted according to their total coupling, both internal and external, which for node i , corresponds simply to $\sum_{j \in N_i} w_{ij} + w_{ii}$. Each node is successively tested against the following condition:

$$\sum_{j \in N_i} w_{ij} > \beta \sum_{j \in C_i} w_{ij} \quad (8)$$

where β is a positive constant. If the condition is satisfied the node is made a seed, $C \leftarrow C \cup \{i\}$, if not $C \leftarrow C$. If insufficient seeds are selected on the first pass this process may be repeated on the remaining non-seeds with incrementally greater β until roughly half the nodes are selected as seeds.

2) Calculating interpolation coefficients

The interpolation coefficients are computed as in (4). However, only non-zero elements need to be taken into account. Thus (4) may be computed as follows:

$$p_{ij} = \begin{cases} 1 & i = c_j \\ w_{ij} / \sum_{k \in C_i} w_{ik} & i \notin C \\ 0 & i \in C, i \neq c_j \end{cases} \quad (9)$$

For computational efficiency it is beneficial to keep W and P as sparse as possible. Thus edges that contribute a

negligible amount to the aggregates are pruned from the graph. Edges between seeds and non-seeds are pruned if their associated interpolation coefficient is only a small fraction of the average for that node. This may be expressed with the following condition:

$$p_{ij} < \frac{\epsilon}{N(C_i)} \quad (10)$$

where $N(C_i)$ is the number of neighbouring seeds and ϵ is a positive constant (usually on the order of 0.1). Once pruned the interpolation coefficients are re-normalised.

3) Generating Coarse Edge Weights

The coarse edged weights are defined by the matrix product in (6) may be efficiently computed in terms of node neighbours.

$$\hat{w}_{ij} = \sum_{m \in C_i \cup \{c_i\}} P_{mi} \sum_{n \in C_j \cup \{c_j\}} w_{mn} P_{nj} \quad (11)$$

This expression allows an intuitive interpretation of course weights as simply a weighted average of the edges coupling the nodes of both aggregates, i and j .

4) Calculating Node Properties

The SWA algorithm allows for a multitude of aggregate properties to be used provided they can be calculated recursively. The SWA implementation described within this paper makes use of the following aggregate properties

- Size (number of pixels)
- Mean intensity
- Variance of sub-aggregates mean intensities
- Mean gradient
- Variance of sub-aggregates mean gradient magnitude
- Eight bin Gradient Orientation Histogram (GOH)

The intensity mean allows for the distinction between solid regions where as the gradient measures are more suited to discriminating between textures. The intensity variance and gradient magnitude variance also characterise texture, and also allow the normalisation of intensity and gradient when comparing properties.

Unlike the SWA approach described in [3], this implementation makes use of a histogram of gradient orientations rather than shape moments. The choice of a Gradient Orientation Histogram (GOH) was inspired by the success of *David Lowe's* SIFT descriptor [5] and the related GLOH descriptor [6]. Both of these descriptors are constructed from a set of GOH's arranged in a spatial grid. These descriptors have been shown to be very effective at distinctively characterising image regions [6]. The idea is that the GOH will incorporate some of the distinctiveness associated with these GOH based descriptors into the process to better distinguish between image segments.

In this research an eight-bin histogram was used with each sample weighted by the square of its magnitude. To avoid discontinuities associated with bin boundaries each sample contributes to its two nearest bins, weighted according to the sample's distance from their respective centres. This weighting is controlled by a smooth window function $f_w(d)$ defined such that $f_w(0) = 1$, $f_w'(d) \leq 0$, $f_w(d) + f_w(1-d) = 1$ for $0 \leq d < 1$ and $f_w(d) = 0$ for $d \geq 1$. The simplest choice being a hat function, another

possible choice is:

$$f_w(d) = \begin{cases} 1-3d^2-2|d|^3 & |d| \leq 1 \\ 0 & |d| > 1 \end{cases} \quad (12)$$

Given a gradient with polar coordinates (ρ, ϕ) it will contribute to bin b of histogram \mathbf{h} with B bins as follows

$$h_b = f_w\left(\frac{\Delta\phi_b}{2\pi B}\right)\rho^2 \quad (13)$$

Where $\Delta\phi_b$ is the smallest absolute angle between ϕ and the centre of bin b , $\phi_b = 2\pi b/B$. The following relation is one method of computing this differential angle.

$$\Delta\phi_b = |\text{mod}(\phi_b - \phi, 2\pi) - 2\text{mod}(\phi_b - \phi, \pi)| \quad (14)$$

Where $\text{mod}(a, b) = a - b \lfloor a/b \rfloor$.

All these properties are calculated recursively as weighted averages of sub-aggregate properties. Coarse aggregate sizes $\hat{\mathbf{s}} = (\hat{s}_1, \hat{s}_2, \mathbf{K}, \hat{s}_N)^T$, may be computed in terms of fine node sizes $\mathbf{s} = (s_1, s_2, \mathbf{K}, s_N)^T$ as follows:

$$\hat{\mathbf{s}} = \mathbf{P}^T \mathbf{s} \quad (15)$$

In terms of node neighbours (15) becomes:

$$\hat{s}_i = \sum_{k \in N_i \cup \{i\}} p_{ik} s_i \quad (16)$$

The mean intensity and gradient constitute first order properties. If these properties are packaged into a matrix A where rows correspond to nodes and columns to properties the coarse properties \hat{A} are:

$$\hat{A} = \frac{\mathbf{P}^T A}{\mathbf{P}^T \mathbf{1}} \quad (17)$$

which may be computed as follows

$$\hat{a}_{ij} = \sum_{k \in N_i \cup \{i\}} p_{ik} a_{ij} / \sum_{k \in N_i \cup \{i\}} p_{ik} \quad (18)$$

The second order properties which include the variances and the GOH may be similarly calculated, in a multi-scale way, in terms of their first order counterparts.

5) Modulating Edge Weights

Edge weights are modulated according to a distance measure d_{ij} , between coarse node properties. This distance measure commonly takes the form of a normalised Euclidean distance, where components are first normalised with respect to their standard deviation as well as a user defined *importance* weighting. This distance d_{ij} is converted into relative similarity measure by a modulating function $f_m(d_{ij})$ with the property that $f_m(0) = 1$, $\lim_{d \rightarrow \infty} f_m(d) = 0$ and $f'_m(d) \leq 0$ for $d \geq 0$. One such modulating function is:

$$f_m(d_{ij}) = \frac{1}{1 + \gamma d_{ij}^2} \quad (19)$$

where γ is a positive constant. The coarse edge weights thus become $\hat{w}_{ij} \leftarrow f(d_{ij}) \hat{w}_{ij}$.

6) Calculating node Cost

Within the SWA algorithm only the cost of individual nodes is considered. The normalised cuts cost for a single node i corresponds to simply the ratio between the sum of off diagonal elements in the i^{th} column of \hat{W} to the on diagonal element, w_{ii} .

$$\hat{\Gamma}(\mathbf{u}_i) = \frac{2}{\hat{w}_{ii}} \sum_{j \in N_i} \hat{w}_{ij} \quad (20)$$

Costs are calculated for all nodes within the pyramid with the exception of the finest base nodes since they have no internal coupling and so infinite cost.

B. Top-down Region Delineation

Salient nodes are identified as having a normalised cuts cost well below the average cost of all nodes (excluding the base nodes) within the pyramid. However the actual region that corresponds to salient nodes still needs to be determined. This is done for every salient node with a top-down process that interpolates the salient node's characteristic state from the level where it's found down the pyramid, using the interpolation relation in (3), to the base level where the boundary may be recovered. However, direct interpolation produces blurry regions. To remedy this, the state vector is sharpened at every level before interpolation to nudge it towards a binary vector. This may be done by applying a non-linear transform to state components such as:

$$T(u) = 3u^2 - 2u^3 \quad (21)$$

This increases the contrast of the region at every interpolation step. At the finest level pixels are assigned to the region corresponding to the salient node that produced the greatest state value at their location.

V. RESULTS

The following figure illustrates the results of apply the SWA implementation to a set of challenging natural images.

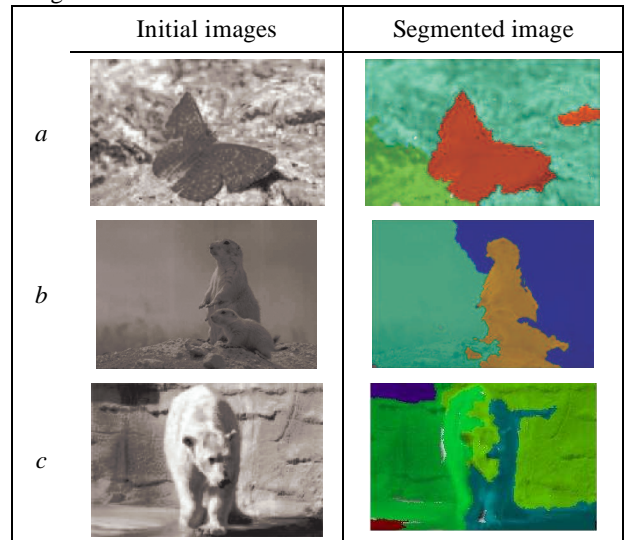




Fig. 2 The result of the application of the segmentation scheme to a set of challenging natural scenes. Images a-c, and d, correspond to the test scenes used in [3] and [2] respectively.

Since the SWA algorithm produces an entire hierarchy of salient regions, only a subset of these regions are shown in Fig. 2. The segments shown in Fig. 2 correspond to the larger scale regions, near the top of the pyramid. The analysis of segmentation results is inherently application specific, however generally results are usually judged against a human's intuitive segmentation of the scene. In this respect Fig. 2.a is segmented relatively well as the butterfly is clearly distinguish from the background. Fig. 2.b is also a relatively good result despite the sky encroaching on some of the foreground regions. However the algorithm failed to isolate the polar bear in Fig. 2.c. The scene in Fig. 2.d shows good segmentation between foreground and background however the algorithm failed to separate the foreground objects.

VI. CONCLUSION

This paper has described a particular implementation of the SWA algorithm that makes use of Orientation Histograms to aid the segmentation of strongly textured regions.

The SWA algorithm in general however is extremely versatile allowing natural integration of application specific statistic within the segmentation process. This coupled with the fact that the regions produced are hierarchical in nature makes SWA a promising candidate as a basis for future robust object detection systems.

REFERENCES

- [1] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," presented at International. Conference of Computer Vision, 2003.
- [2] E. Sharon, A. Brandt, and R. Basriy, "Fast Multiscale Image Segmentation," presented at IEEE Conference for Computer Vision and Pattern Recognition, 2000.
- [3] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," *Nature Letters*, vol. 442, 2006.
- [4] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 888–905, 2000.
- [5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Journal of Computer Vision*, 2003.
- [6] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, 2005.

Rory T. Neilson received his BSc. Electronic Engineering degree from the University of KwaZulu-Natal in 2005. He is currently pursuing his MSc at the same institution.

Bashan Naidoo received his BSc. Engineering degree from the University of Natal in 1992 and his MSc. Engineering degree from the University of Natal, Durban in 2003. He is currently a senior lecturer in the School of Electrical, Electronic and Computer Engineering at the University of KwaZulu-Natal, Durban.

Stephen McDonald received his BSc. And MSc. engineering degrees from the University of Natal, Durban in 1987 and 1989 respectively. He was active in industry for 5 years and is currently a senior lecturer and Director of the Electronic Engineering Program at the University of KwaZulu-Natal, Durban.