Finding Semantic Structures in Image Hierarchies Using Laplacian Graph Energy

Yi-Zhe Song[1], Pablo Arbelaez[2], Peter Hall[1], Chuan Li[1], and Anupriya Balikai[1]

[1] MTRC, University of Bath, Bath, UK, BA2 7AY
[2] University of California at Berkeley - Berkeley, CA 94720
{yzs20,pmh,cl249,ab368}@cs.bath.ac.uk, arbelaez@eecs.berkeley.edu

Abstract. Many segmentation algorithms describe images in terms of a hierarchy of regions. Although such hierarchies can produce state of the art segmentations and have many applications, they often contain more data than is required for an efficient description. This paper shows Laplacian graph energy is a generic measure that can be used to identify semantic structures within hierarchies, independently of the algorithm that produces them. Quantitative experimental validation using hierarchies from two state of art algorithms show we can reduce the number of levels and regions in a hierarchy by an order of magnitude with little or no loss in performance when compared against human produced ground truth. We provide a tracking application that illustrates the value of reduced hierarchies.

1 Introduction

Hierarchical descriptions of images have long been recognized as being valuable to computer vision, the literature on how to build them and use them is vast. Ideally, hierarchies reflect assemblies that comprise real world objects, but in practice they can often be very large and complex. There are significant practical advantages to be had by simplifying hierarchical descriptions, for example we can expect gains in memory efficiency, speed and the hierarchies might be more semantically meaningful. Yet these advantages will be conferred only if the quintessential character of the object is retained by the simplification process. This paper provides a general purpose method to filter complex hierarchies into simpler ones, independent of the way in which the hierarchies are formed, with little or no loss in performance when benchmarked against ground truth data.

There are many reasons for making hierarchal descriptions and many ways to make them; the literature is vast, making a full review impossible here. In any case, we emphasize, *this paper is not about segmentation per se, nor it is about making hierarchies — it is about filtering hierarchies.* Since our purpose is extracting semantic structures from hierarchies rather than proposing algorithms for constructing new ones, we bypass the large literature on hierarchical segmentation and review only a few representatives of successful approaches.

Sieves [4] are a well established example. They are built using morphological operators to generate a tree rooted around gray level extrema in an image. Sieves

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 694–707, 2010.

are related to maximally stable extremal regions (MSER) which are made by filtering a hierarchy comprising binary regions in which each level is indexed by a gray level threshold [16]. The filtering criterion is stability, which is defined as the rate at which a region changes area with respect to the control parameter (threshold). Sieve trees are very complex, MSER trees are simpler by comparison; yet both have found applications and both address the important issue of segmentation, which is a major theme in this paper.

Mean-shift [5] is amongst the best known of the recent segmentation algorithms. A recent interesting development from Paris and Durand [18] observes that thresholds in feature space density lead directly to image space segmentations, and uses the notion of stability in feature space to produce a hierarchal description. Their definition of stability differs from that used to build MSER trees, but there is a common spirit of persistence as control variables change.

Normalized cuts [21] is another of the most widely used and influential approaches to segmentation. This approach is principled, resting as it does on spectral graph theory. Yet, it tends to produce arbitrary divisions across coherent regions in ways that are not intuitive to humans, breaking large areas such as the sky, for example. In response, there is now a sizable literature on various additions and modifications to suit specific circumstances. These include the popular multi-scale graph decompositions [6] which are directly related to hierarchical descriptions because smaller objects are children to larger ones.

The connected segmentation tree (CST) [2], which it has its roots in the early work by Ahuja [1], is specifically designed to yield semantically meaningful hierarchies. The CST takes into account the photometric properties, spatial organization, and structure of objects. It is very successful in identifying taxonomies amongst objects are therefore demonstrates the value of simple hierarchical descriptions.

The most successful boundary detectors to date are rooted in the probability of boundary (Pb) maps introduced by Martin *et al* [14]. The Pb maps compared very well against human produced ground truth using the Berkeley Segmentation Dataset (BSDS) [15], and recent improvements include multiscale analysis [19] and the use of global image information [17]. The latter are of particular interest because global-Pb lead to state of the art region hierarchies [3].

We simplify a hierarchy solely by the removal of levels, typically reducing their number by one or even two orders of magnitude. Others also simplify hierarchies: MSER simplifies a hierarchy in which thresholds make levels [16]; Kokkonis and Yuille [11] use a heuristic that estimates the cost of completing a given graph to reach a goal graph within an A^* search for structure coarsening; computational geometry and computer graphics offer many examples related to mesh simplification.

The **contribution** of this paper is to generically filter hierarchical descriptions with little or no loss of descriptive power compared to human ground truth, and with the exceptions of MSER and CST all the above hierarchies are typical in being large and complex. In particular, our contributions are three-fold:

- The extension of the notion of Laplacian energy from spectral graph theory to non-connected graphs.
- Its application as a measure of graph complexity, to finding meaningful segmentations.
- Extensive quantitative and qualitative evaluation proving that our approach preserves the semantic quality of the input hierarchy while reducing considerably its complexity.

Another important aspect of our method is that it filters hierarchies *after* they have been constructed. This means that we can apply our method to many different hierarchies. The reduced hierarchies we output have a sensible semantic interpretation in terms of objects and object parts.

Our method is fully explained in Section 3 , but broadly it considers each level to be a segmented partition of an image. Nodes of the graph at any level are the segmented regions which form a region adjacency graph (RAG) [23,25] by a neighboring relationship. We compute the complexity of the graph on each level using Laplacian graph energy and keep levels whose complexity is smaller than either of the neighboring levels. We make no attempt to simplify the graph within a level. The value of our filtering is demonstrated by experiment in Section 4. We continue by developing our intuition regarding Laplacian graph energy.

2 Laplacian Graph Energy as a Complexity Measure

Graph complexity can be measured in several ways [8] and is of value to applications including but not limited to embedding [20], classification [22], and the construction of prototypes [24]. The complexity measure we use is based on *Laplacian graph energy* defined by Gutman and Zhou [10]. Laplacian graph energy is attractive in the context of this paper because it favors the selection of regular graphs, and particularly favors polygonal graphs.

Let G be a unweighted graph of n vertices and m edges, i.e., a (n, m)-graph and A be its adjacency matrix. Let d_i be the degree of the ith vertex of G and D be the corresponding degree matrix, where $D(i, i) = d_i$. Then $L = D - A$ is the Laplacian, and the LE is defined [10] to be

$$\mathcal{LE}(G) = \sum_{i=1}^{n} \left| \lambda_i - \frac{2m}{n} \right| \tag{1}$$

In which: the λ_i are eigenvalues of the Laplacian matrix and $2m/n$ is the average vertex degree. Gutman and Zhu [10] prove that $\mathcal{LE}(G)$ falls into the interval

$$\mathcal{I}[G] = [2\sqrt{M}, 2M] \tag{2}$$

in which

$$M = m + \frac{1}{2} \sum_{i}^{n} \left(d_i - 2\frac{m}{n} \right)^2 \tag{3}$$

Gutman and Zhu [10] also prove that bipartite graphs are at both ends of the interval; that with $m = n/2$ at the lower bound and $m = n^2/2$ at the upper bound.

Our aim now is to characterize the behavior of LE as a graph changes, typically because of a computer vision algorithm. A graph can change in the number of nodes, arcs, and also in arc permutation. The effect of addition and deletion of arcs on graph energy is at best difficult to predict, it can go up or down [7]. We aim to show that in a set of graphs with a fixed number of nodes and arcs, the LE of the polygonal graph is likely to be a lower bound.

We begin by noticing that if we set ν to be the variance of node degree, for a unweighted graph G, then we have $M = m + n\nu/2$. The interval containing the LE for G can now be expressed as

$$\mathcal{I}[\nu; m, n] = \left[2\left(m + \frac{n\nu}{2}\right)^{1/2}, 2\left(m + \frac{n\nu}{2}\right)\right]. \tag{4}$$

which shows that the interval is parametrized by variance. The variance is zero if and only if G is a regular graph, in which case $\nu = 0$ and the interval is $\mathcal{I}[0; m, n] = [2\sqrt{m}, 2m]$; we note this interval depends only on the number of arcs. If the graph has $\nu > 0$, then the corresponding interval bounds rise and the interval widens.

Suppose a fixed number of arcs m and nodes n. If $m < n - 1$, then this graph is disconnected; but in practice we compute LE for connected components only (see Section 3). We assume $m \geq n$ from this point. For such an (n, m) graph, the variance in degree node depends solely on how the arcs are distributed and only regular graphs have zero variance. Allowing for permutations, there is at most one regular graph in the set. The LE for such a regular graph falls into the smallest interval, $\mathcal{I}[0; m, n]$, taken over the whole (n, m) family. If the variance rises then the LE is drawn from an interval with a lower bound greater than $2\sqrt{m}$ and an upper bound greater than $2m$. The left of Figure 1 provides an illustration of how variance, the bounding interval and LE relate to each other, when considering a family of graphs where $n = m = 8$. It shows that the LE for each graph (solid black) approximates the M value (solid red), and is bounded by $[2\sqrt{M}, 2M]$ (dotted red). Graphs with large variance (to the right edge of the figure) have higher, wider intervals, as indicated by the vertical line.

We can extend our intuition further by considering graphs of fixed n but an increasing number of edges, summarized in the right of Figure 1. It shows the output from a simulation in which arcs were randomly placed over a graph of eight nodes by thresholding a symmetric random matrix. Threshold values were chosen so that one new arc was added at each step, starting with isolated nodes and building to a complete graph. The Figure shows an LE trajectory, in red, having one main peak. This occurs on the most irregular graph. The black curve plots our modified LE, defined in Section 3. It shows the effect of allowing for graphs comprising many connected components. The first peak corresponds to graphs with many small components. Notice that when the graph comprises a single component, our modified energy corresponds to the standard LE, seen where and the two curves coincide. The graph with the lowest energy is to the

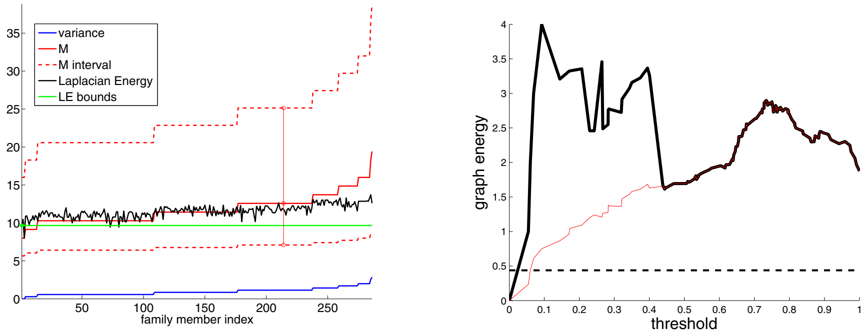

Fig. 1. Left: The relation between variance, the bounding interval and LE for graphs randomly drawn from an (m, n) family, where $(m = 8, n = 8)$. The family is ordered into sets of equal variance. The contour of LE through the (m, n) polygon is drawn in green. **Right:** The modified LE for random graphs (see Section 3) picked from $n = 8$ family, formed by adding one edge to each previous member. Note the two main peaks, the right peak corresponds to a single connected component. The standard LE is shown in red.

far left of this region, it is closest to a (m, n) polygon. The complete graph is the rightmost. This is empirical evidence that LE is minimal for polygonal graphs.

In summary, Laplacian graph energy is a broad measure of graph complexity. Regular structures which tend to be visually meaningful, such as polygons, exhibit lower Laplacian graph energy than structures comprising randomly selected arcs.

3 Using Laplacian Graph Energy to Filter Hierarchies

We suppose a full hierarchical description comprises a collection of N distinct levels, our problem is to determine $M << N$ levels needed in a reduced hierarchy. These M levels must preserve the semantic content of the full hierarchy. We begin our account by being more concrete about the hierarchies we have in mind. Image primitives, which are connected regions, reside at the bottom level of the hierarchy and partition the input image into a RAG — so nodes are synonymous with regions and only neighboring regions can be adjacent. A combination of primitives makes a parent region that is larger in size and which resides on the level directly above its children. The union of all regions at any level partition the image and also constitute a RAG, but in addition we have links between levels that specify child-parent relations. We constrain the RAG so that only children of a common parent can be adjacent, similar to the CST [2]. We assume that parent regions can be combined in recursive fashion, thus generating new levels. Such combination continues until the production algorithm halts; the halting criterion is algorithmic dependent but a level comprising a single region which covers the whole image provides a universal terminating case. It follows that

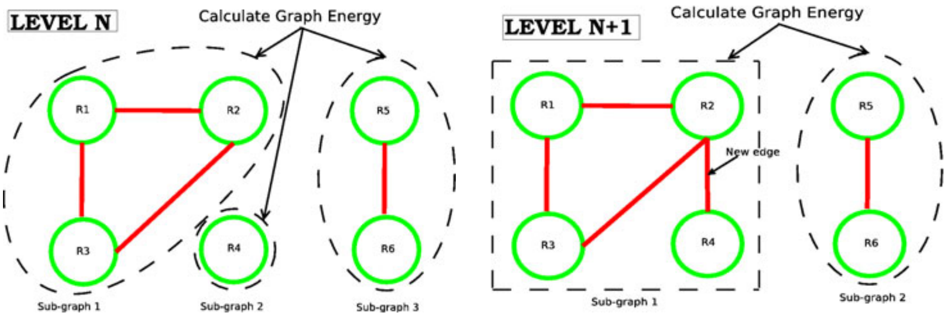

Fig. 2. Illustrating how graph energy is calculated on two levels of the hierarchy. Each node R_i corresponds to a region primitive. When the level increases, sub-graphs merge to create larger ones while the number of connected components falls.

we can represent any such hierarchy by a collection of levels, each one being a distinct partition of the input image, and each region in each level will bottom out into a distinct collection of region primitives. Moreover, each region the full hierarchy is partitioned by its children, its grand-children and eventually by its ancestral image primitives. Notice that we can represent such a hierarchy via an image map and therefore all arcs are implicitly specified.

Our principle in solving the above problem is to choose those levels that are lower in complexity than their neighbors, which follows the intuition developed in Section 2. We measure complexity via Laplacian graph energy, as defined above in Section 2. Note that rather than simplifying the hierarchy as a whole [9], we select levels of the hierarchy that exhibit lower Laplacian graph energy. Two modifications to the standard definition of Laplacian graph energy were proposed. First, we propose to use a weighted matrix A in which the element in row i and column j is given by

$$a_{ij} = \exp(-\frac{w_{ij}}{w_{max}}) \tag{5}$$

where w_{ij} is the average boundary strength between region i and region j, and w_{max} is a decay factor, set to the maximum over all w_{ij}. Thus our adjacency matrix is akin to the similarity matrix used in Normalized cuts [21].

Secondly, we introduce an extension to the standard definition of Laplacian graph energy that we call the *component-wise* Laplacian graph energy (cLGE). Such extension is motivated by the fact that we consider a scene to comprise a set of independent objects; within a hierarchy, these are defined by child-parent relationships. For a graph with K disconnected components, we define the cLGE to be

$$\xi = k \sum_{i=1}^{K} \frac{\mathcal{LE}(G_i)}{|n_i|} \tag{6}$$

in which G_i is ith connected component of $|n_i|$ nodes and k is the number of nodes in the whole graph. This is, at root, the sum of individual component

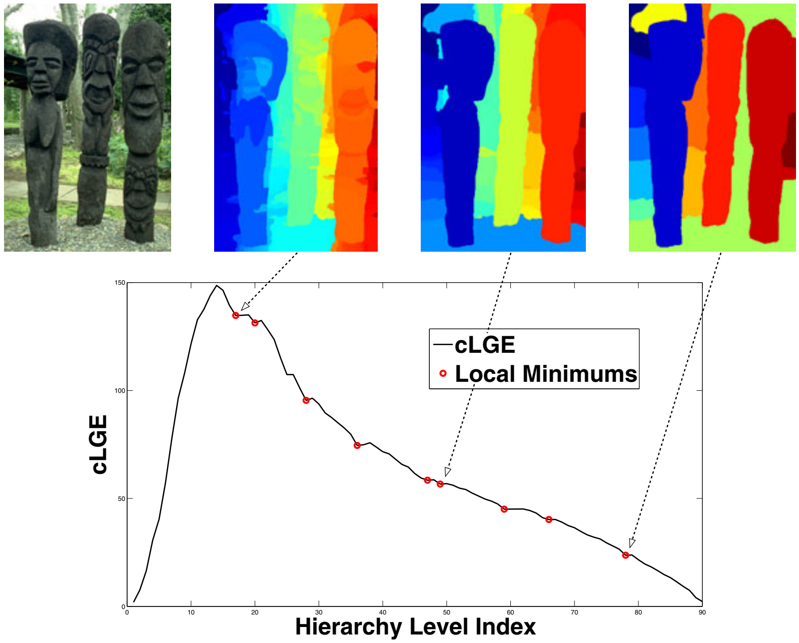

Fig. 3. Graph energy (equation 6) as a function of level index. Local minimum correspond to levels that are less complex compared to neighboring levels.

energies, but in which each is normalized by the number of nodes it contains. The scale factor k is used so that in the case of a single connected component our expression returns the original graph energy exactly.

We compute the cLGE at every level in the hierarchy independently using graphs built from the primitives at the lowest level; hence k in Equation 6 is the total number of image primitives. At the bottom level of the hierarchy, each primitive is a 1-node sub-graph on its own, whereas the top level forms a single connected graph. At intermediate levels, as segmentations become coarser, subgraphs are merged to create larger ones, and so the number of disconnected components will fall. Figure 2 illustrates how we compute cLGE over two levels with a simple graph of 6 nodes, each of which represents a region primitive. In this way, we numerically construct the function $\xi(z)$ where z is the level index. As z rises the number of regions falls, and each region covers a larger number of primitives.

As seen in Figure 3, cLGE for the level as a whole can rise or fall, depending on the way these primitives are connected. Following the intuition developed above (Section 2), $\xi(z)$ falls as individual connected components tend towards regular graphs which have minimal cLGE, so we keep those levels at which cLGE is locally minimal (circled in red in Figure 3). In the same figure, segmentations corresponding to selected local minimums are also shown, where finer visual details are retained in lower levels and semantic objects emerge at a higher level.

The different shapes of the plots in Figures 1 and 3 is explained by the number of primitives and edges, and the fact that the former figure uses un-weighted graphs whereas the latter uses weighted graphs

4 Results

This section presents both quantitative and qualitative results, beginning with **quantitative results** in Tables 1 and 2.

Both tables were constructed by evaluating both full and reduced hierarchies, we used the Berkeley Segmentation Dataset (BSDS) [15] as a foil against which to assess the retention of semantic information. We obtained benchmarks against not only the boundary models of images introduced in the original BSDS [15], but also against that of regions as well. Two state of the art segmentation hierarchies were benchmarked. One is due to [3], which is premised on global-Pb (gPb) edge maps, oriented watershed transform (owt) and ultrametric contour maps (ucm) which offers a convenient duality between boundary maps and hierarchical image segmentations. We refer to their algorithm as gPb-owt-ucm. As a comparison basis, we also include benchmark results of quad-trees with 8 levels [12], denoted as quad-tree-8.

The other algorithm is a topological approach to mean-shift authored by Paris and Durand [18], but with both owt and ucm implemented over its edge map representation, here referred to as Paris-owt-ucm. It is worth noting that Paris and Durand [18] obtained a F-measure of 0.61 on the original BSDS boundary benchmark, after applying the *-owt-ucm algorithm from Arbeláez *et al*, we observe an increase in performance signified by a F-measure of 0.63. Each of these algorithms yields hierarchies with hundreds of levels, yet experiments show that

Table 1. Boundary Benchmarks on the BSDS. Four new algorithms were benchmarked together with gPb-owt-ucm which is State-of-the Art. Results show little or no downgrade on F-measures of the cLGE filtered hierarchies (denoted *-cLGE) when compared to the originals, gPb-owt-ucm and Paris-owt-ucm [18]. Benchmark scores of a randomly filtered hierarchy (gPb-owt-ucm-M) are also given where a clear decrease on F-measures against gPb-owt-ucm-cLGE can be seen. Results of benchmarking quad-trees [12] with 8 levels are also included as a direct comparison basis.

Method	ODS	OIS	AP
human	0.79	0.79	-
gPb-owt-ucm	0.71	0.74	0.77
gPb-owt-ucm-cLGE	**0.71**	**0.72**	**0.77**
gPb-owt-ucm-M	0.67	0.57	0.69
Paris-owt-ucm	0.63	0.66	0.71
Paris-owt-ucm-cLGE	0.63	0.66	0.71
Paris-owt-ucm-M	0.61	0.65	0.48
quad-tree-8	0.37	0.39	0.26

Table 2. Region Benchmarks on the BSDS. We follow [3] to obtain region benchmarks for each of the four algorithms in Table 1. Again, *-cLGE delivered on-par benchmark scores against the originals on region covering criteria (leftmost three columns), Probabilistic Rand Index (PRI) and Variation of Information (VI). The right most columns shows the average number of nodes (AN) and levels (AL), demonstrating an order of magnitude improvement in nearly all cases.

Method	ODS	OIS	Best	PRI	VI	AN (AL)
human	0.73	0.73	-	0.87	1.16	-
gPb-owt-ucm	0.58	0.64	0.74	0.81	1.68	16267 (80)
gPb-owt-ucm-cLGE	**0.58**	**0.60**	**0.66**	**0.79**	**1.78**	**829 (8)**
gPb-owt-ucm-M	0.53	0.58	0.64	0.77	2.04	1349 (8)
Paris-owt-ucm	0.52	0.60	0.69	0.78	2.12	28448 (124)
Paris-owt-ucm-cLGE	0.52	0.60	0.68	0.78	2.03	5870 (28)
Paris-owt-ucm-M	0.50	0.60	0.68	0.77	2.07	6339 (28)
quad-tree-8	0.33	0.39	0.47	0.71	2.34	21845 (8)

each provides a high quality segmentation at some level within their representation, when compared to human segmented ground truth. Unfortunately, neither of them provide any method by which to choose these optimal level or levels: a method such as ours, which automatically picks semantic levels, is therefore potentially very useful.

In each case we create a full hierarchy of levels by thresholding the ucm output by the particular algorithm. We aim to demonstrate that our filtering technique is able to reduce the number of levels in full hierarchies which is usually in their hundreds down to only tens, yet retain semantic information. Columns of Tables 1 and 2 (boundary benchmarks and region benchmarks respectively) are exactly the same as these used by [3] apart from an extra column in Table 2; ODS, OIS and AP stand for Optimal Dataset Scale (best scale for the entire dataset), Optimal Image Scale (best scale per image) and Average Precision respectively, whereas Best (Best Covering Criteria), PRI (Probabilistic Rand Index) and VI (Variation of Information) are three different measures common in the literature to measure region segmentation quality instead of boundaries.

The right-most column in Table 2 provides the average number of nodes (AN) and the average number of levels (AL) for each of the benchmarked hierarchies across all of the 100 BSDS testing images. We refer to our reduced hierarchy by a graph energy suffix, *-cLGE. To introduce a control measure we also filtered by picking $M << N$ levels at random, with M determined via cLGE; we refer to these cases with the suffix *-M. Over the 100 testing images in BSDS, gPb-owt-ucm hierarchies contain an average of **80 levels**, whereas our reduced gPb-owt-ucm-cLGE only contains an average of **8 levels** which is an order of magnitude better. Similarly, Paris-owt-ucm has **124 average levels** whereas Paris-owt-ucm-cLGE reduce that to **28 levels**, again an order of magnitude improvement. The average number of regions is reduced from **16267** to **829** for gPb-owt-ucm when cLGE is used to select levels, and to **1349** when the M levels are randomly selected. Again we see an order of magnitude improvement, and we

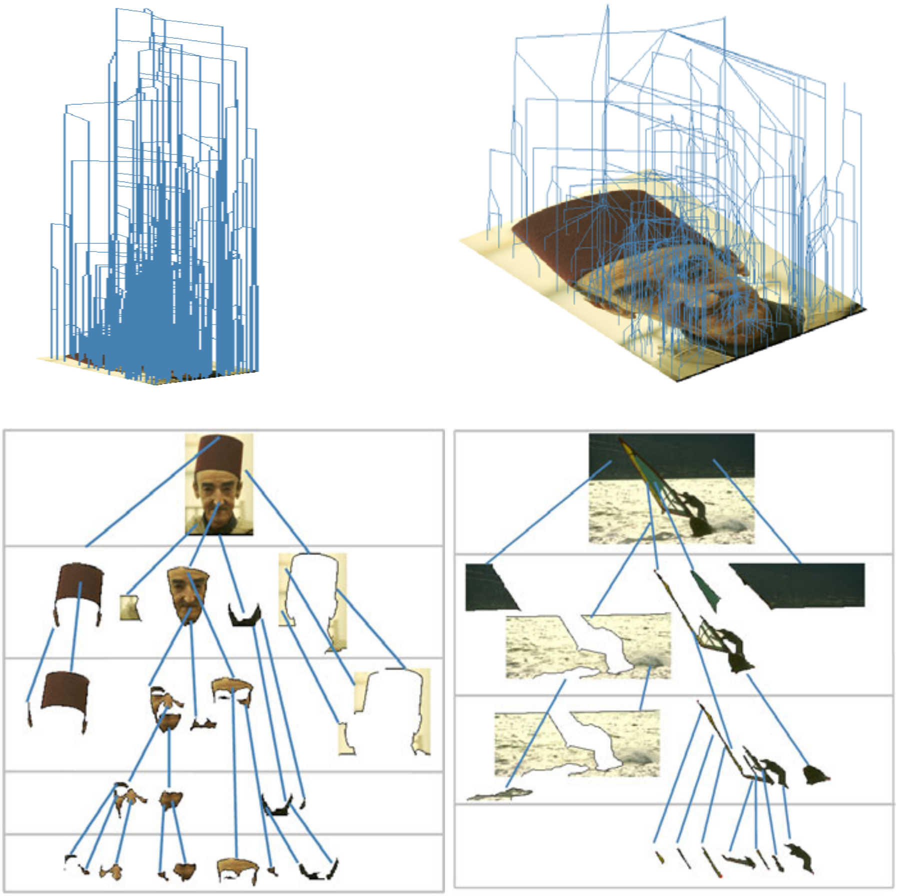

Fig. 4. The qualitative value of simplifying a hierarchy by removing levels. The top row visualizes merging regions as a tree, left over all levels of a gPb-owt-ucm hierarchy, right over the levels that remain after our filtering; bottom shows how objects are broken into useful parts (the original images are included for visualization purpose only).

conclude cLGE provides a non-random selection of levels. For Paris-owt-ucm we reduce the number of regions by about 1/5.

Despite many fewer levels and nodes, the table of boundary benchmarks, Table 1, shows cLGE filtered hierarchies retain the F-measures of the original, a similar story can be told in the region benchmark table, Table 2, with identical scores on ODS and fairly close ones on other measurements. In all cases, cLGE out performs our control of random selection. Overall, we see that cLGE retains benchmark scores of the original, while only keeping a small subset of its content.

In the rest of this section, we provide some **qualitative** results of the reduced gPb-owt-ucm hierarchies (gPb-owt-ucm-cLGE). In Figure 4, we offer visualizations of the original gPb-owt-ucm hierarchy and that of the reduced hierarchy (gPb-owt-ucm-cLGE), where a dramatic decrease in the number of levels is

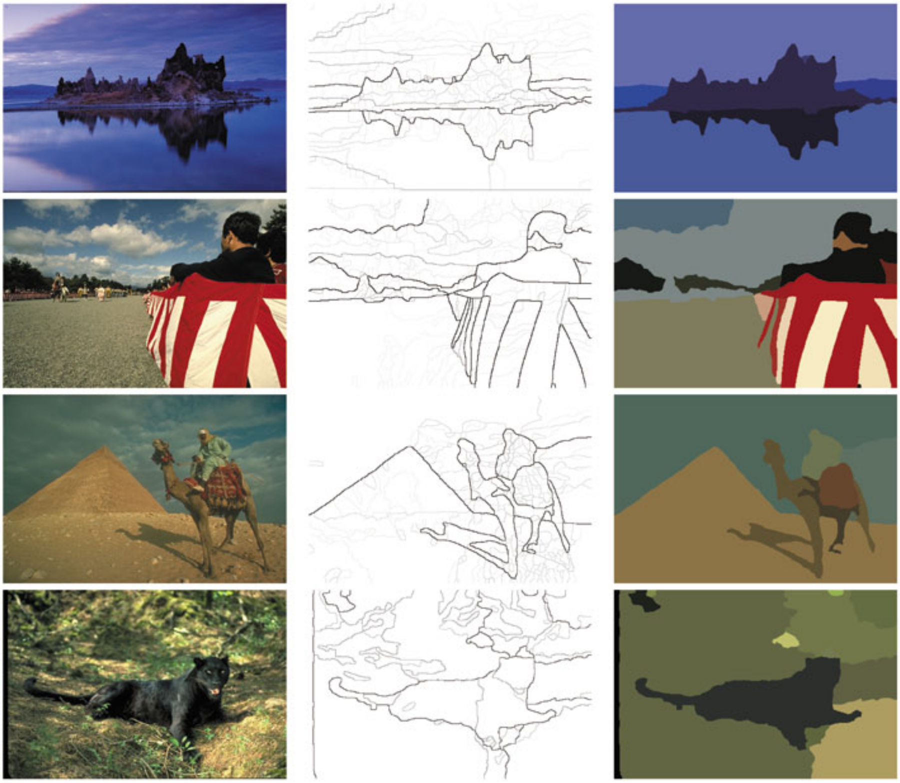

Fig. 5. Object-level image segmentations can be obtained independent of ground truth data. Left column: Original color images; Middle column: Reduced hierarchies represented as Ultrametric Contour Maps; Right column: Object-level segmentations chosen using the last local minimum on graph energy, which are the top-levels of the reduced hierarchies.

visible which in-turn made reasonable visualization possible. The bottom of the same figure shows gPb-owt-ucm-cLGE in terms of how nodes are broken down on two images. It is worth noting that because we only filter hierarchies, the segmentation results will depend on the quality of original hierarchies.

Finally, in Figure 5, we illustrate gPb-owt-ucm-cLGE as Ultrametric Contour Maps and show how a single object-level segmentation can be automatically chosen without the use of a threshold or by appeal to human ground truth data. For instance, [3] relies on ground-truth data to obtain thresholds. Although the threshold corresponding to ODS can be generalized to other images, the best results are obtained by OIS which is image-dependent. To yield a single object-level segmentation for a given image, we simply choose the level of the hierarchy corresponding to the last local minimum on graph energy, that is the top level in our gPb-owt-ucm-cLGE hierarchies. Such segmentations are the coarsest in the hierarchy.

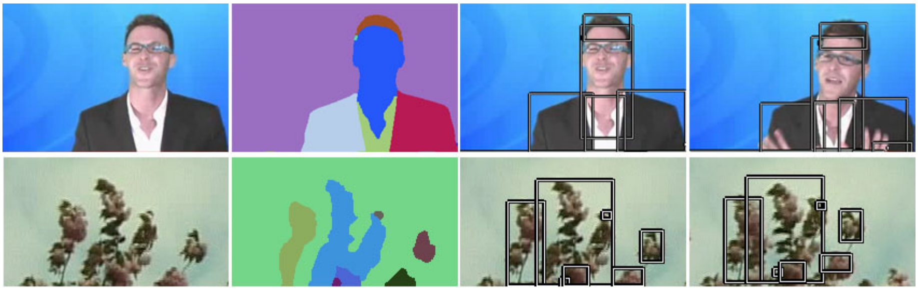

Fig. 6. Two examples of stable tracking using our reduced hierarchy over Berkeley's UCM maps (gPb-owt-ucm); see supplementary material for the full set of videos. Left to right: frame one, a segmentation map, typical frames from the video tracking regions.

With regards to runtime, because we work on regions rather than pixels, our graphs are relatively small in size and sparse in nature. This in turn made Eigenvalue decomposition less of a problem. In practice, our current Matlab implementation takes around 20 seconds per image on a Intel Core2Duo 2.6GHz machine with 4GB of RAM. Code will also be made available on-line[1].

5 Application to Tracking

Here we show the value of reduced hierarchies to tracking, in particular memory and complexity is improved, with marginal gain in accuracy. To describe a video, a hierarchy should be stable across the entire sequence. We evaluate a hierarchy's stability by the temporal stability of its regions. If a region is stable, it changes little over time and can be tracked more easily.

Given a video, we build a hierarchy from the first frame, and track every single region using the standard KLT tracker [13]. Figure 6 shows identifiable objects are tracked over time, these regions have come from a reduced hierarchy and qualitatively demonstrate the regions are semantic. The filtered hierarchy only consumes about one-tenth of the memory (which is the fraction of total regions in the reduced hierarchy compared to the full) and takes about one-tenth of the tracking time. We ran the experiment on several scenes and find the filtered hierarchies keep good regions that can be stably tracked. Stable regions can be seen in both Figure 6 and the supplementary material.

6 Conclusion

In this paper, we have introduced *component-wise Laplacian graph energy*, cLGE, as a complexity measure useful to filter image description hierarchies. cLGE is a measure of graph complexity that is simple to compute. We showed that

[1] http://www.cs.bath.ac.uk/Song

- cLGE operates over two state of the art image hierarchies, which lends support to our claim of algorithmic independence;
- we reduce the number of levels by an order or magnitude with little or no effect on the semantic quality of the result.
- the reduction in data leads to a description that benefits applications, as demonstrated by our tracking example.

We measured the semantic quality of an hierarchy using the widely used Berkeley Segmentation Dataset (BSDS) [15]; apart from the original boundary benchmarks, experiments were also conducted on a new extension on regions. Both experiments show little or no loss of semantic quality of the graph energy filtered hierarchies when compared to the originals.

Despite the good filtering performance of cLGE, the quality of the end result will depend on the quality of the original hierarchies. We have, though, shown that the filtered gPb-owt-ucm hierarchies, largely retain their performance; in addition they provide a solid basis for tracking because they are stable over time, and visualizations are reminiscent of CSTs [2].

Future work includes accessing how graph energy can be used to generate hierarchies of a more semantic fashion, possibly by recursively applying it to individual regions rather than the whole image. The question of whether an overall objective function can be optimized across the layers can also be a future research direction. Applications such as more efficient and accurate object classification and matching are being considered too.

References

1. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. TPAMI 18(12), 1211–1235 (1996)
2. Ahuja, N., Todorovic, S.: Connected segmentation tree - a joint representation of region layout and hierarchy. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (June 2008)
3. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empitical evaluation. In: Computer Vision and Pattern Recognition (2009)
4. Bangham, J.A., Harvey, R.W., Ling, P.D., Aldridge, R.V.: Morphological scale-space preserving transforms in many dimensions. Journal of Electronic Imaging 5, 283–299 (1996)
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. TPAMI 24(5), 603–614 (2002)
6. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 1124–1131 (June 2005)
7. Day, J., So, W.: Singular value inequality and graph energy. Electronic Journal of Linear Algebra 16, 291–299 (2007)
8. Escolano, F., Hancock, E.R., Lozano, M.A.: Polytopal graph complexity, matrix permanentsm, and embedding. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 237–246. Springer, Heidelberg (2008)

9. Fisher, D.: Iterative optimization and simplification of hierarchical clusterings. J. Artif. Int. Res. 4(1), 147–179 (1996)
10. Gutman, I., Zhou, B.: Laplacian energy of a graph. Linear Algebra and its applications 414, 29–37 (2006)
11. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 802–809 (June 2009)
12. Liu, J., Yang, Y.: Multiresolution color image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 689–700 (1994)
13. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 674–679 (April 1981)
14. Martin, D., Fowlkes, C., Malik, K.: Learning to detect natural image boundaries using local brightness, color and texture cues. TPAMI 26(5), 530–549 (2004)
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision., vol. 2, pp. 416–423 (2001)
16. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide basline stereo from maximally stable extremal regions. In: BMVC, pp. 384–393 (2002)
17. Maire, M.: Arbeláez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize juncations in natural images. In: CVPR (2008)
18. Paris, S., Durand, F.: A topological approach to hierarchical segmentation using mean shift. In: CVPR, pp. 1–8. IEEE Computer Society Press, Los Alamitos (2007)
19. Ren, X.: Multi-scale improves boundary detection in natural images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 533–545. Springer, Heidelberg (2008)
20. Robles-Kelly, A., Hancock, E.R.: A riemannian approach to graph embedding. Pattern Recognition 40, 1042–1056 (2007)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI, 888–905 (2000)
22. Shokoufandeh, A., Dickinson, S.J., Siddiqi, K., Zucker, S.W.: Indexing using a spectral encoding of topological structure. In: International Conference on Pattern Recognition, pp. 491–497 (1999)
23. Tu, P., Saxena, T., Hartley, R.: Recognizing objects using color-annotated adjacency graphs. In: Shape, Contour and Grouping in Computer Vision, pp. 246–263 (1999)
24. White, D., Wilson, R.: Mixing spectral representations of graphs. In: International Conference on Pattern Recognition (2006)
25. Worthington, P., Hancock, E.: Region-based object recognition using shape-from-shading. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 455–471. Springer, Heidelberg (2000)