Report on Research Project on the PathFinder image segmentation algorithm

Author: Fabio Drucker

Faculty advisor: Professor John MacCormick

July 2009

We investigate the performance of the PathFinder algorithm for segmenting images into visually appealing superpixels. PathFinder is compared to the Efficient Graph-Based Image Segmentation (EGBIS)[1], which is, to our knowledge, the leading algorithm in the field. We compare the two algorithms in terms of the time efficiency, and the Mean Accuracy and Explained Variation, as explained in "Superpixel Lattices"[2].

We find that while PathFinder is somewhat less precise than EGBIS, especially in explaining reflecting color variation in different regions, it is almost as accurate in determining object boundaries. On the other hand, it is much faster than EGBIS, and therefore the accuracy lost may be more than justified in real time applications or other applications in which processing speed is key.

Edge detection

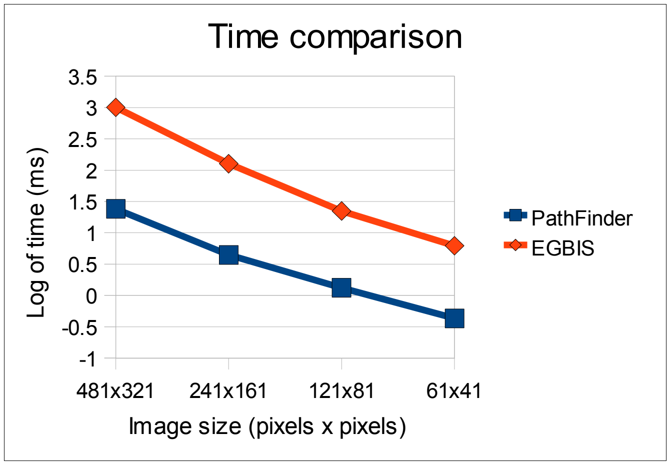
An important step towards dividing an image into superpixels is to determine where edges are likely to occur. The original implementation of the PathFinder algorithm used a directed step operator. That is, it separately determined the likelihood of a vertical and a horizontal edge being at a given point. We also tried two undirected step operators: (i) the root-mean-square of the horizontal and vertical likelihoods, and (ii) BEL[3]. We found minor improvements using the BEL edge detection algorithm, but due to time considerations (it takes about a minute to process each image, compared to a fraction of a second for the entire PathFinder algorithm with a step operator), we concluded it is not an overall improvement over step operators.

Time comparison

PathFinder seeks to provide a lower computational cost alternative to the leading algorithms, such as EGBIS. Therefore, it is EGBIS in Java, and used that implementation to run another set of experiments on the difference in time efficiency between both algorithms.

To provide a wider array of data, we ran time comparisons for different sized images, ranging from 481x321 to 61x41. As shown in the table and chart below, we found that the PathFinder algorithm is much faster than EGBIS, around 15 to 40 times, depending on the size of the image. This difference increases with image size. Timing experiments were performed in a Dell Latitude D820 with an Intel Centrino Duo CPU @ 2GHz, running Ubuntu Linux 9.04. As the accompanying table and graph show, PathFinder is at least 15 times faster than EGBIS in processing images. Moreover, the difference quickly and significantly increased as image sizes increase.

Image size	481x321	241x161	121x81	61x41
PathFinder	24.1ms	4.43ms	1.32ms	0.43ms
EGBIS	1005.37ms	126.47ms	22.26ms	6.20ms
Ratio	41.72	28.53	16.85	14.46



Explained Variation

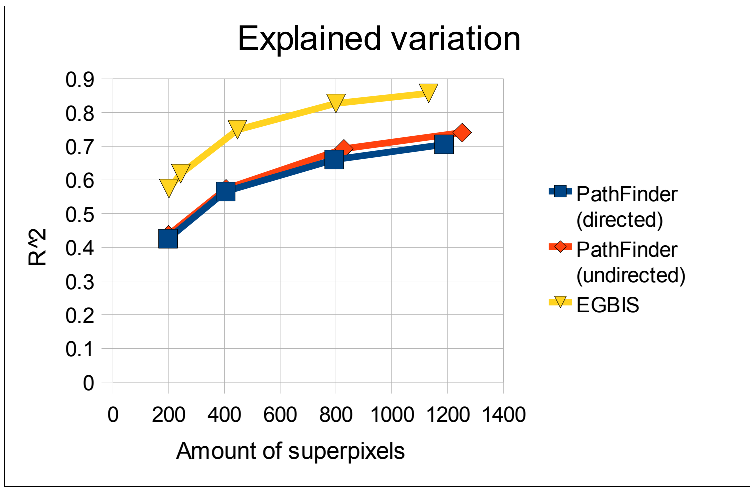
This measure seeks to explain the extent to which variations in color in the image are explained by variations in color from region to region, as opposed to variations within regions. It is therefore defined as:

$$R^2=\frac{\sum_{R,G,B}[\sum_i(\mu_i-\mu^2)]}{\sum_{R,G,B}[\sum_i(x_i-\mu)^2]}$$

That is, we sum over all three color channels for all pixels *i*, where x_i is the value of the pixel in that color, μ_i is the mean value of the color over that pixel's superpixel, and μ is the global mean of that color in the image. The value of R^2 ranges between 0 (when there is only one superpixel and therefore μ_i is always equal to μ) and 1 (when each superpixel has only one pixel). It should be noted that, for color images, this formula differs slightly from that used in “Superpixel lattices”, since that formula does not sum over all three color channels. We do, however, use the same formula for gray-scale images that have only one color channel (the “Superpixel lattices” paper does not specifically state how color images are dealt with).

In this experiment, we recorded the Explained Variation for both EGBIS and PathFinder over 50 images chosen from The Berkeley Segmentation Dataset and Benchmark[4]. For each image, we obtained the results with different amount of superpixels per image since that, as previously explained, has a great effect in the Explained Variation of the algorithm. For the following table slight differences in the amount of superpixels produced with each algorithm are rounded up to fall in the four categories presented for ease of comparison, but the exact data points are used in the following graph.

Superpixels	200	400	800	1200
PathFinder (directed operator)	0.43	0.57	0.66	0.71
PathFinder (undirected operator)	0.44	0.57	0.69	0.74
PathFinder (BEL)	0.49	0.57	0.70	0.74
EGBIS	0.57	0.75	0.83	0.86



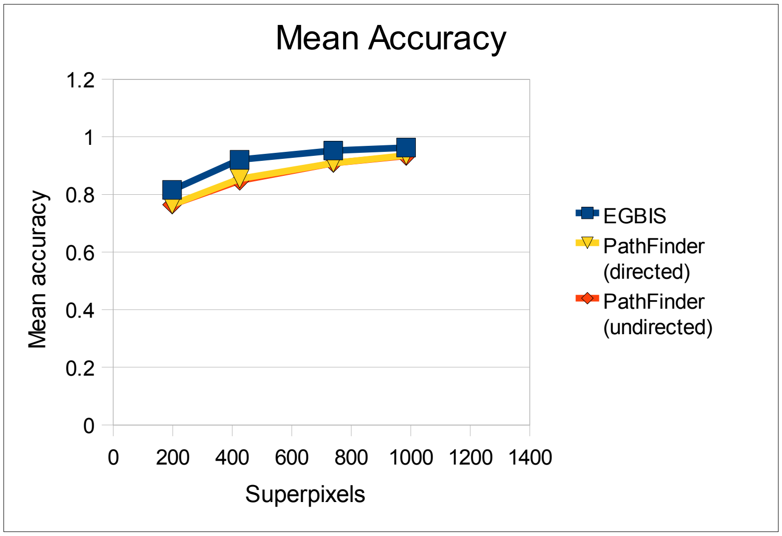
This clearly shows than in this measure there is a significant advantage in favor of EGBIS. It is however worth considering that EGBIS considers only color similarities among contiguous pixels in determining its regions, and therefore it is to be expected that a measure that focuses on color variation would play to EGBIS' strong points and therefore it would be surprising for PathFinder to approach EGBIS success in this measure.

Mean Accuracy

The purpose of this metric is to determine the extent to which true object boundaries are found by the PathFinder algorithm, where true boundaries are defined as region boundaries determined by human subjects. For each of the superpixels found by our algorithm, we determined to which true region most of its pixels belong to, and then determine what percentage of pixels actually are in that region. The overall Mean Accuracy is the overall proportion of pixels in an image that were placed in the correct region as previously defined.

In this measure, no appreciable difference was discovered when using directed versus undirected step operators (as can be appreciated in the accompanying graph), therefore we make no distinction between them in the following table and graph.

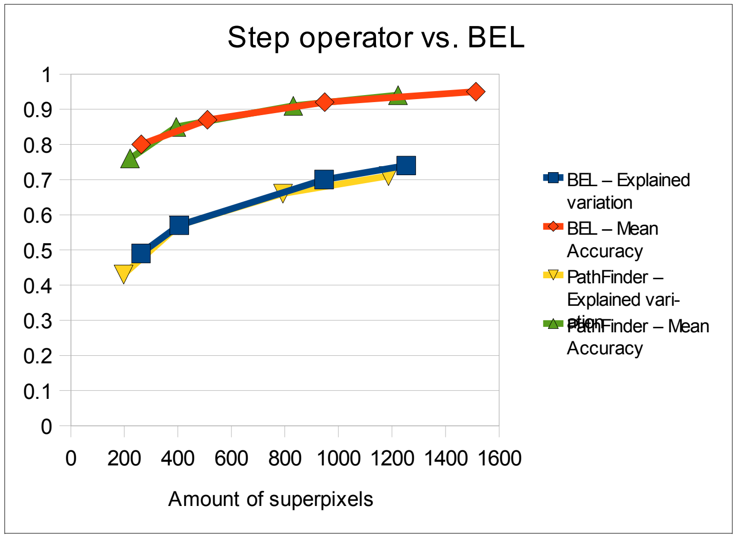
Superpixels	200	400	800	1200
PathFinder (step)	0.76	0.85	0.91	0.94
PathFinder (BEL)	0.80	0.87	0.92	0.95
EGBIS	0.82	0.92	0.95	0.96



BEL Edge Detection

Besides directed and undirected step operators, we experimented with using the BEL algorithm to detect edges before the division into superpixels. The main hurdle to actually implementing it within the PathFinder algorithm is that it takes a lot more time to process each image than the step operator we were using, therefore negating the main advantage of the algorithm. Exact time comparisons are not available due to the unavailability of BEL's source code. However, we can compare the performance of PathFinder using BEL against PathFinder using the previously examined directed step operator, both in terms of Explained Variation and Mean Accuracy.

As the accompanying graph shows, there are slight improvements when using BEL, but not significant ones. Therefore, due to the severe loss in time efficiency, we do not consider that using BEL to detect image edges in PathFinder is preferable to step operators.



Extension to video

One of the potential areas in which PathFinder's time efficiency could be important is in video, for example in problems like motion estimation, since processing multiple frames would be problematic with a slower algorithm. In order to apply this algorithm to video, we would need image segmentations to be consistent through time – that is, for the regions in sequential frames to follow the movement of the objects they capture.

We implemented a way to do this in which in each frame after the first one, each path from the previous frame is selected and the best path that is close to it is constructed. By being "close" we mean that each pixel in the new path is at most *d* pixels away, horizontally for vertical paths and vertically for horizontal paths, from the position of the equivalent pixel in the previous path. The value *d* is a parameter that can be changed. This generates a segmentation that looks somewhat consistent, but it faces several problems. For small values of *d*, we get paths that are not very meaningful, and face the possibility of objects having moved more than paths are allowed to move. However, if *d* is increased, the best successor path of several different paths is the same, so paths tend to converge to the same paths quite quickly, which is again undesirable.

Further dissemination

I intend to present my results next semester in a Math/CS chat in the CS department. My faculty advisor also hopes to assemble these findings into a paper for presentation in a conference.

Timeline

The research project proceeded according to the following approximate timeline:

- Week 1: Port the existing C++ implementation of the EGBIS algorithm into Java. Debug and ensure the outputs of both versions of the algorithm are the same.
- Week 2: Using the Java version of EGBIS, run time comparisons between EGBIS and PathFinder on images of different sizes. Write the necessary code to be able to run Explained Variation tests, and run experiments on Explained Variation with multiple images with PathFinder default settings and similar number of superpixels for EGBIS.
- Week 3: Implement a way to process ground truth data from human subjects to measure the Mean Accuracy of both algorithms, and develop a way to perform tests for Mean Accuracy for

both algorithms. Expand the tests both for this and for Explained Variation to different amounts of superpixels per image.

- Week 4: Develop a way for PathFinder to use BEL for edge detection in its segmentation. Re-run PathFinder tests using BEL for edge detection instead of the step operator. Implement a draft version of a way to adapt PathFinder to video processing and motion estimation.
- Week 5: Make the necessary changes for PathFinder to use an undirected step operator instead of a directed one. Re-run all the test to compare the performance of the algorithm when both options are used. Start work on summarizing graphs and on a final report.
- Week 6: Investigate further work on ways of assessing video performance and improving video processing. Summarize results, producing necessary tables and graphs and writing this report.

Conclusion

We found the PathFinder algorithm to be moderately less accurate than EGBIS in terms of Explained Variation, and slightly less in terms of Mean Accuracy. However, we also found it to be a very significant improvement in terms of time efficiency. Therefore, while PathFinder may not be an attractive alternative for applications where time is not a significant factor, real time uses and other applications for which fast processing is key would probably prefer it, since it achieves huge time savings with a moderate compromise of performance that may be acceptable in some applications.

[1]P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision,* 59(2), 2004
[2]A.P Moore, S.J.D. Prince, J. Warrell, U. Mohammed and G. Jones, “Superpixel Lattices”, *CVPR*, 2008

[3]Piotr Dollar, Zhuowen Tu, and Serge Belongie, “Supervised Learning of Edges and Object Boundaries”, IEEE Computer Vision and Pattern Recognition (CVPR), June, 2006

[4]*http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/*