Estimating Image Segmentation Difficulty

Dingding Liu [†] Yingen Xiong [‡] Kari Pulli [‡] Linda Shapiro [†]

[†] Dept. of Electrical Engineering, University of Washington, Seattle, WA, USA
[‡] Nokia Research Center, Palo Alto, CA, USA

Abstract. The heavy use of camera phones and other mobile devices all over the world has produced a market for mobile image analysis, including image segmentation to separate out objects of interest. Automatic image segmentation algorithms, when employed by many different users for multiple applications, cannot guarantee high quality results. Yet interactive algorithms require human effort that may become quite tedious. To reduce human effort and achieve better results, it is worthwhile to know in advance which images are difficult to segment and may require further user interaction or alternate processing. For this purpose, we introduce a new research problem: how to estimate the image segmentation difficulty level without actually performing image segmentation. We propose to formulate it as an estimation problem, and we develop a linear regression model using image features to predict segmentation difficulty level. Different image features, including graytone, color, gradient, and texture features are tested as the predictive variables, and the segmentation algorithm performance measure is the response variable. We use the benchmark images of the Berkeley segmentation dataset with corresponding F-measures to fit, test, and choose the optimal model. Additional standard image datasets are used to further verify the model's applicability to a variety of images. A new feature that combines information from the log histogram of log gradient and the local binary pattern histogram is a good predictor and provides the best balance of predictive performance and model complexity.

Keywords: Image segmentation difficulty, linear regression, mobile image processing

1 Introduction

Image segmentation is an important and unsolved research area. In computer vision, automatic segmentation algorithms aim to divide an image into meaningful regions for applications such as tracking or recognition [12]. They often involve supervised training to adjust the parameters for a particular application. Even with all these efforts, they are not guaranteed to work well on all images and, in general, do not perform as well as humans [9]. To achieve more reliable results, interactive segmentation algorithms [4, 8] have been developed and applied to applications such as image editing, but they require much more user interaction.

Recently, the growing number of photos on the internet and those taken by camera phones has posed new challenges. First, it is time-consuming to interactively segment a large quantity of images or examine the segmentation results one by one. Although co-segmentation has been proposed to segment two or a group of related images [11, 3], it cannot be applied to a vast amount of unrelated images. Second, the widely used camera phones have comparatively limited computational resources, and image processing on camera phones is still in the research and development stage [2]. Difficult images should be sent to the server for processing, while easier ones may be segmented on the camera phone. Third, even if the camera phones have powerful processors, their small screens prevent users from examining multiple image segmentation results at once, as could be done on bigger screens. Knowing which images are difficult to segment and require more attention would save time for users.

Motivated by the above problems, and in contrast to previous research efforts, we address the following questions. Can machines differentiate the images that are difficult to segment from those that are easy to segment? Assuming each image has a certain level of segmentation difficulty, can an algorithm quantitatively predict the difficulty prior to segmentation? Which features are most predictive?

To answer these questions, we first need suitable ground truth images with an indication of how difficult they are for machines to segment. However, defining evaluation methods for segmentation quality is itself an active research area [13, 14]. To avoid complicating the problem, we use the Berkeley benchmark suite of 100 color images [9], which are well accepted in the computer vision research community, as our ground truth dataset. Berkeley researchers tested several different algorithms on those images and ranked them according to the best F-measure representing quality of segmentation. As Figure 1 shows, different images have different F-measures [1], and the smaller the value, the more difficult is an image for a machine to segment.

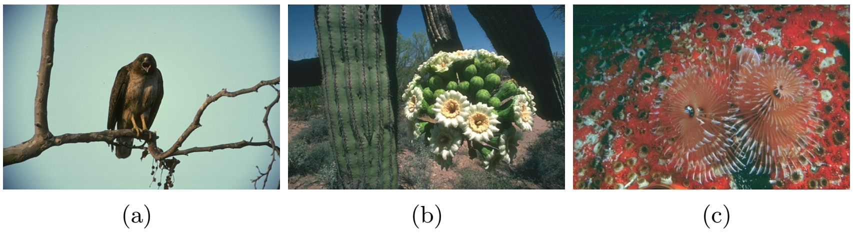

(a) (b) (c)

Fig. 1. Can an algorithm estimate the machine segmentation difficulty levels? (a) Easy (F-Measure 0.91), (b) Fair (F-Measure 0.76), (c) Hard (F-Measure 0.44).

As an initial step, we formulated this problem as a linear regression problem and tested different image features on the 100 Berkeley benchmark images [9] to derive the model. The reasons behind this choice are twofold. First, linear regression models outperform more complex nonlinear models for prediction purposes, especially in situations with small numbers of training cases, low signal-to-noise ratio, or sparse data [6]. Second, they can be generalized to nonlinear versions by using basis-function methods [6].

The main contributions of this work are (i) introduction of the new problem of estimating image segmentation difficulty levels without performing prior segmentation; (ii) determination of relevant and effective features, including a new feature combining information from the log histogram of log gradient and the local binary pattern (LBP) texture histogram; and (iii) selection of a model that has a good generalization ability and low model complexity.

Section 2 describes the mathematical background used in this paper. Section 3 explains our approach and algorithm in detail, including the feature extraction and modeling process. Section 4 compares the experimental results of different models and justifies our selection of the most promising model with experiments on both labeled and unlabeled data. Section 5 concludes the paper and briefly describes future work.

2 Mathematical Background

A linear regression model with the form

$$y = X\beta + \varepsilon, \tag{1}$$

models the approximate relationship between the dependent variable y and regressors X. When the goal is prediction, linear regression can be used to fit a predictive model to an observed data set of X and y values. After developing such a model, if an additional value of X is given, the fitted model can be used to make a prediction of the value of y.

There are many different regression algorithms. Our method uses the Gram-Schmidt procedure for multiple regression and Principal Component Regression (PCR), which uses Principal Component Analysis (PCA) to estimate regression coefficients. Instead of regressing the independent variables directly on the data, a subset of the principal components Z_m obtained from the eigenvalue decomposition of a data covariance matrix is used, providing a regularized estimation. For more details, please refer to [6].

Cross-validation estimates the prediction error by the extra-sample error. K-fold cross-validation first divides the data randomly into K roughly equal-sized partitions. Then $K-1$ partitions are used to fit the model, and the other partition to test it.

3 Details of the Approach

Our method consists of three major parts: (1) transform difficulty measures; (2) extract image features; and (3) model the relationship between them and select an optimal linear regression model among all possible choices. When the raw data of the difficulty measures do not meet the assumption of linear regression in terms of normal distributions, a transformation is needed. These transformed difficulty measures are used as the dependent variables y to fit and test possible models, using different sets of features as X in Equation 1. The extracted image features are functions of basic features commonly used in image segmentation algorithms.

In the modeling process, the key is to find out how much weight each feature should have and which features should be left out entirely. The naive way to select a subset of the features is to perform a brute force search, but this is too costly in high-dimensional feature spaces. We propose the following efficient algorithm to find the features that will contribute to the optimal model.

3.1 Transformation of Difficulty Measures

If the histogram of segmentation difficulty measures does not resemble a normal distribution or is skewed, several data transformations are applied to test whether the transformed data would be better than the original data to use as the dependent variables y in the linear regression model. Common transformations including the log transformation and the square-root transformation were tested in this work. A Quantile-Quantile plot (Q-Q plot) that graphically compares the distribution of a given variable to the normal distribution represented by a straight line, was used as the tool to choose the best transformation, if one is needed.

3.2 Feature Extraction

To build a complete model, image features were extracted and statistics computed. Besides simple statistics like mean and variance of color and intensity, the variance and entropy of the grayscale, color, and texture histograms and the log histogram of log gradients were investigated as well. The LBP [10] operator was chosen as the texture measure due to its reported good results in texture analysis for pattern classification. The log histogram of log gradients was previously used in blind motion deblurring [7]. In addition, several new features were extracted from these two histograms as shown in Table 1. According to the different sources from which the features are computed, the resultant 29 features were divided into four groups.

1. Statistics from image data

Mean and variance from grayscale, color, LBP texture, and gradient information (a)+(b).

2. Statistics from histograms

Variance and entropy from the four corresponding histograms calculated from each image (c)+(d).

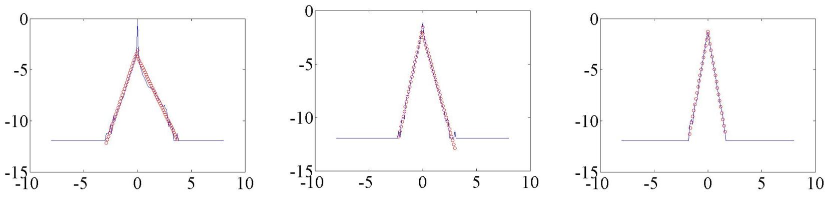
3. Shape features from log histogram of log gradients

The log histogram of log gradients is represented by shape approximation features (e), which are the slopes and intercepts of two lines that approximate the shape of the histogram as in Figure 2.

Table 1. Dimensions of Extracted Features

	A. Gray-scale	**B. CIE-Lab**	**C. LBP Texture**	**D. Log Grad.**
(a) Data Mean	1	3	1	1
(b) Data Variance	1	3	1	1
(c) Hist. Variance	1	3	1	1
(d) Hist. Entropy	1	1	1	1
(e) Shape approx.				4
(f) Maximum bin			1	

4. Maximal bin-count feature from LBP histogram
The local binary pattern histogram, using a 16 pixel neighborhood [10], is represented by the count in the maximal bin (f), which is related to how much smooth area there is in one image.



(a) (b) (c)

Fig. 2. The shape of log histogram of log gradients can be approximated by two straight lines (a) Easy, (b) Fair, (c) Hard.

As shown by the red dots in Figure 2, the shape of the log histogram of log gradients of low-resolution images, can be approximated by two straight lines instead of a Laplacian distribution. Based on this observation, the slopes and intercepts of these lines are calculated as follows:

1. Calculate the log derivative sum. Letting $log0 = 0$, calculate the log gradient in x, y direction, $logGX$ and $logGY$, as the log difference of the grayscale value of immediate neighboring pixels. Then take the sum $logG = logGX + logGY$.

2. Build a histogram of all the values of $logG$ in the interval of $[-8, 8]$ with a bin size of 0.1 and take the log of the counts in each bin.

3. Find the slopes and intercepts of two straight lines that fit the shape of the above histogram.

3.3 Modeling Process

Since we model the relationship between image features and segmentation difficulty by a linear regression model, the features used to construct the predictive variables must be selected. Different combinations of image features were tested to construct a group of possible regression models, and cross-validation(CV) was used to compare them and choose the optimal one.

The testing and comparison were performed from two different directions. From the full model using all the extracted features, possible reduced models were generated using PCR. From the 'minimal' models using single feature groups as in Table 1, models using combinations of feature groups were generated. The optimal models generated from these two paths guide the search for the global optimal model in terms of predictive performance and model complexity.

From a Full Model to Reduced Models: Principal Component Regression

On a model MO_{max} that includes the maximum number of features m_{max}, which is the number of all feature dimensions, PCR is run using different numbers of principal components $m \in [1, ..., m_{max}]$. When the CV errors are compared, the model with least error $m_{minCVErr}$ will be selected and denoted by MO_{PCRm}.

When the number of features is large, different features may offset each other's effect. So the coefficients may not accurately indicate how important a feature is, but can give us some clues about which type of features are important [6]. We sort the feature coefficients β in MO_{max} in descending order. The features corresponding to the largest $m_{minCVErr}$ coefficients are selected to help determine which features will be most predictive.

From Models Using Single Feature Group to More Complex Ones: Feature Group Combination

For the 4 feature groups in Section 3.2, there are $\sum\limits_{i=1}^{4} C_4^i = 15$ different linear regression models.

The k-fold cross-validation was run on all 15 models to find the one with the least CV error. These models are denoted by $M_{\alpha,\beta}$, where $\alpha \subseteq (A, B, C, D)$ and $\beta \subseteq (a, b, c, d, e, f)$ in Table 1. The optimal model at this step is recorded as $MO_{\alpha_{opt},\beta_{opt}}$.

Given α_{opt}, if β_{opt} does not include all the features in the vertical direction under each element in α_{opt}, the missing features are included in building a model $MOB_{\alpha_{opt},\beta_{opt}}$ with more features for the next step's search. For example, if $\alpha_{opt} = (C)$, $\beta_{opt} = f$, then a model using features $(a), (b), (c), (d)$ under (C) will be built for the next step's processing and comparison.

The optimal models from Sections 3.3 and 3.3 are compared. The model with the best combination of least CV error and smallest model complexity is chosen as the optimal model.

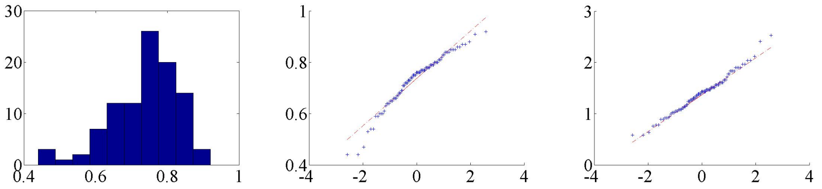
4 Experiments and Results

We use the above algorithm to find the optimal model and most predictive features from the 100 images (called labeled data) where the F-measure was available. Berkeley researchers use precision to measure the probability that a machine-generated boundary pixel is a true boundary pixel and recall to measure the probability that a true boundary pixel is detected. When they threshold the boundary map in order to compare it to the ground truth boundaries, they do so at multiple levels and record the maximum harmonic mean of precision and recall as the reported F-measure. Our difficulty measure will be the inverse of the F-measure estimated by our model.

After the model was derived and proved to be optimal numerically, it was tested on 743 additional color images. Among them, 70 were from the PASCAL dataset [5], 30 were from the MSR segmentation dataset [4] and the other 643 were from CMU-Cornell iCoseg dataset [3].

4.1 Building the Model with Labeled Data

Pre-processing of Difficulty Measures The F-measure histogram of the labeled data resembles a skewed normal distribution (Figure 3). Setting z to be the F-measure, we tested 4 transformations: $-log(1-z)$, $log(\frac{z}{1-z})$, $sqrt(z)$ and $sqrt(1-z)$. The transformation $y = -log(1-z)$, which had the best Q-Q plot, was chosen.



(a) (b) (c)

Fig. 3. (a) Histogram of F-measures, (b) Q-Q plot before the transformation, (c) Q-Q plot after the transformation.

Feature Extraction For each model, a feature matrix was constructed for which each row contains features from one image, while each column represents one of the 29 features over the set of images. All the feature data are standardized using the mean and standard deviation of each column of the feature matrix.

Modeling Process The following steps were performed to model the relationship between features and segmentation difficulty.

1. Full model and its optimal reduced model computed by PCR: The experiments were carried out as described in Section 3.3. On each model, 10-fold cross-validation was run 100 times and the average CV error recorded. The model complexity is represented by the dimension of features used as the predictive variables. Figure 4 shows that the full 29-dimensional feature model has the best performance when m (number of principal components) is 12, with prediction error at 0.005903. We denote this model by MO_{PCR12}. This result indicates that a model with a smaller set of features will perform better than one with all the features.

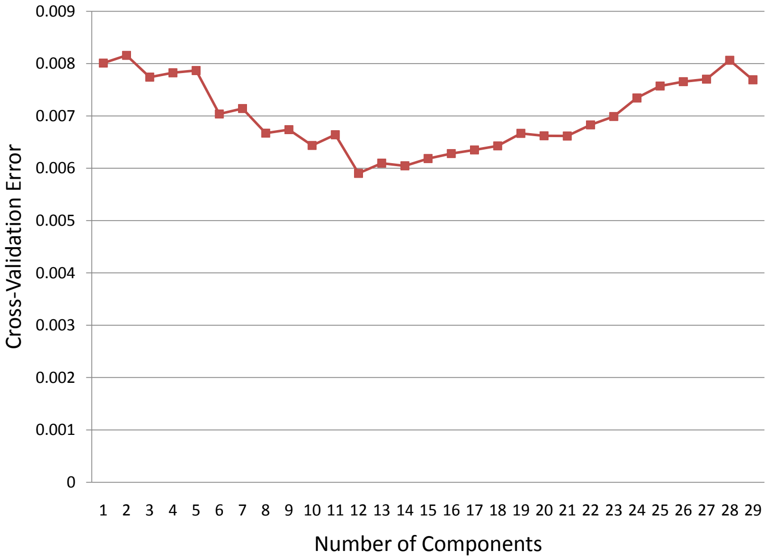


Fig. 4. Principal component regression of the full model.

The coefficients β for features in the full model can give us clues about which features might be most predictive. After the absolute values of the coefficients are sorted in descending order, the top 12 are the following features in Table 1: Cc, Cd, Dd, the lightness mean in Ba, Dc, Cf, Aa, Dc and De. It is clear that features from the LBP texture and Log gradient information are very important.

2. Optimal model from feature group combinations: Figure 5 shows a plot of the CV error vs. the number of predictive variables for the 15 models in Section 3.3. The specific groups of features corresponding to Table 1 are also indicated on the horizontal axis with parentheses.

The optimal model $MO_{\alpha_{opt},\beta_{opt}}$ is the one using feature groups 3 and 4, with 5 predictive variables and an error of 0.006372. The error is slightly larger than that of the model obtained by adding feature group 1 to it, where the error is 0.006148. However, that model's complexity is 17, much bigger than 5. It can also be seen that combining feature groups 3 and 4 produces a much better result than using either group 3 or group 4 individually.

Since feature groups 3 and 4 are related to LBP texture and log gradient information, the other statistics extracted from those two feature groups, row (a-d), column (C,D) in Table 1, could be added to make additional models

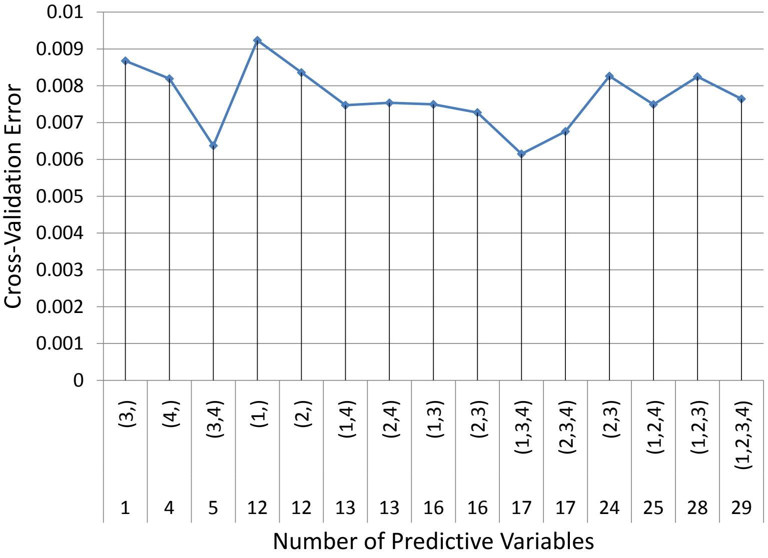


Fig. 5. 15 subset models regression.

$MOB_{\alpha,\beta}$. From Figure 5, adding feature group 1 to groups 3 and 4 yields far better results than adding feature group 2. Thus we only test the addition of $\alpha \subseteq (C, D)$ and $\beta \subseteq (a, b)$. These addtions result in another $\sum_{i=1}^{4} C_4^i = 15$ different models for which we plot the cross-validation test results in Figure 6. It turns out that adding just the variance of image LBP texture produces the best result among all of the 30 models. This model, denoted by MO_{set6}, has a model complexity of 6 and prediction error of 0.005877.

When MO_{PCR12} is compared to MO_{set6}, the latter is a better model, because it has a smaller CV error of 0.005877 with only 6 features instead of 0.005903 with 12 features. We define the features in MO_{set6} as the new *segmentation difficulty estimation feature*, a 6-dimensional vector containing the variance of LBP texture, shape approximation of log histogram of log gradients and the count from the maximum bin in the LBP texture histogram.

4.2 Applying the Model to Additional Data

After the optimal model has been derived and proved to be optimal numerically as above, we tested it on 743 additional images from three datasets. Experiments on these unlabeled images can qualitatively indicate how well our model works on a large number of images used in various segmentation and co-segmentation tasks. In these experiments, the proposed 6-dimensional segmentation difficulty estimation features were extracted and their normalized values plugged into the optimal model derived above, producing estimation of F-measures similar to

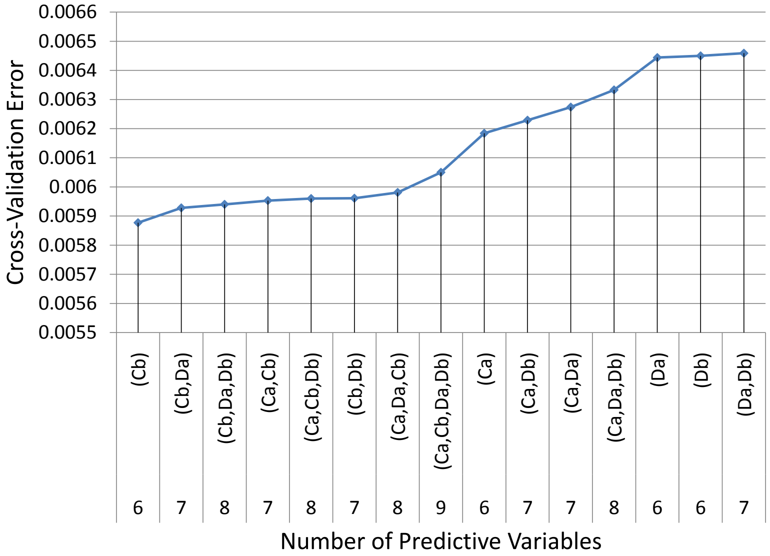


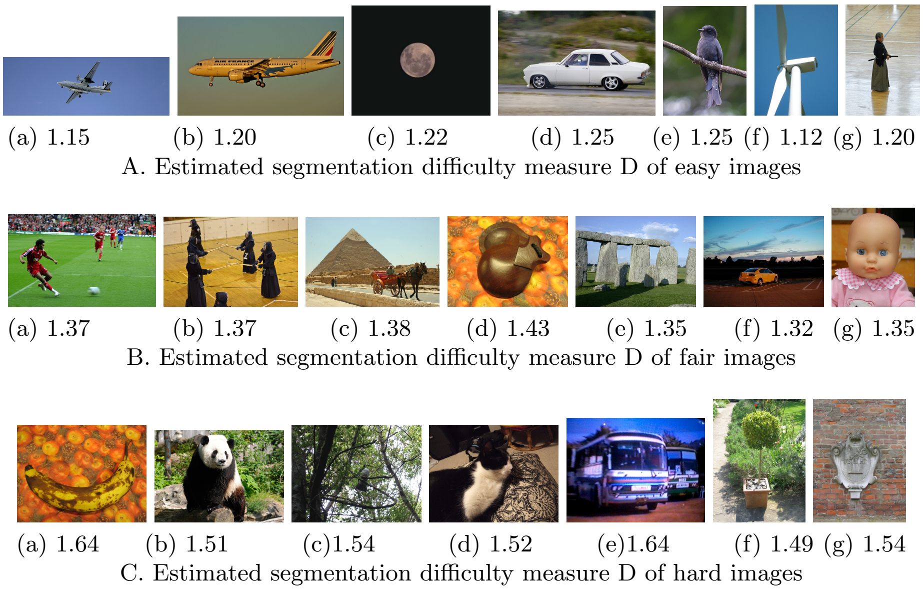
Fig. 6. Another 15 subset models regression containing feature groups 3 and 4 plus the feature(s) in Table 1 indicated on the horizontal axis in parentheses.

those reported by Berkeley researchers. To make it more intuitive, we define the segmentation difficulty measure as D=1/(estimated F-measure) for each image. Thus larger D indicates a more difficult image to segment.

The experimental results demonstrate that our model can reasonably recognize and estimate that the segmentation difficulty increases as there is less contrast and more texture in the image without any user interaction. Figure 7 shows three categories of estimated difficulty level: easy, fair, and hard.

5 Conclusion and Future Work

Looking at the challenges in segmenting a large quantity of various images from a different perspective, we introduced a new research problem: estimating the segmentation difficulty of an image without performing segmentation. We model the relationship between image features and segmentation difficulty by a linear regression model. Besides using the 100 test images of the Berkeley segmentation dataset as labeled data to build, train and test the optimal model, we tested our model on 743 color images from three other well accepted datasets to further verify and demonstrate the power of the model. A new feature using information from the LBP histogram and log histogram of log gradient of an image was discovered and proved to be very effective, despite its low complexity. The model shows good generalization ability in terms of performance and complexity.


(a) 1.15 (b) 1.20 (c) 1.22 (d) 1.25 (e) 1.25 (f) 1.12 (g) 1.20

A. Estimated segmentation difficulty measure D of easy images

(a) 1.37 (b) 1.37 (c) 1.38 (d) 1.43 (e) 1.35 (f) 1.32 (g) 1.35

B. Estimated segmentation difficulty measure D of fair images

(a) 1.64 (b) 1.51 (c)1.54 (d) 1.52 (e)1.64 (f) 1.49 (g) 1.54

C. Estimated segmentation difficulty measure D of hard images

Fig. 7. Non-related images and their estimated segmentation difficulty measure D=1/(estimated F-measure)

Being a fully automatic algorithm to estimate the image segmentation difficulty, our model is not only a useful complement to previous efforts in segmenting a group of related images, either cosegmentation or interactive cosegmentation, but can also help process a large quantity of unrelated images by separating images that are difficult to segment from those that are easy to segment. Due to these advantages, this work can lead to many interesting applications, such as grouping images on mobile phones and providing better user experiences in large-scale content-based image retrieval. In the future, we plan to estimate the segmentation difficulty levels within a single image by finding local regions that are hard to segment. We hope the exploration of this topic will give new perspectives and lead to more exciting work.

References

1. http://www.eecs.berkeley.edu/research/projects/cs/vision /bsds/bench/html/images.html.
2. A. Adams, D. Jacobs, J. Dolson, M. Tico, K. Pulli, E. Talvala, B. Ajdin, D. Vaquero, H. Lensch, M. Horowitz, et al. The Frankencamera: an experimental platform for computational photography. In *ACM SIGGRAPH 2010*, pages 1–12. ACM, 2010.
3. D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *Proc. CVPR*, pages 3169–3176. IEEE, 2010.

4. A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. *Lecture Notes in Computer Science*, pages 428–441, 2004.
5. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.
6. J. Friedman, R. Tibshirani, and T. Hastie. *The elements of statistical learning.* Springer-Verlag New York, 2001.
7. A. Levin. Blind motion deblurring using image statistics. *Advances in Neural Information Processing Systems*, (19):841–848, 2007.
8. Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.
9. D. Martin, C. Fowlkes, D. Tal, and J. Malik. 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*, volume 2, pages 416–423, July 2001.
10. T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. *Computer Vision-ECCV 2000*, pages 404–420, 2000.
11. C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Proc. CVPR*, volume 1, pages 993–1000. IEEE, 2006.
12. J. Winn. Locus: Learning object classes with unsupervised segmentation. In *Proc. ICCV*, pages 756–763, 2005.
13. H. Zhang, J. Fritts, and S. Goldman. An entropy-based objective evaluation method for image segmentation. *Proc. SPIE-Storage and Retrieval Methods and Applications for Multimedia*, 2(4), 2004.
14. H. Zhang, J. Fritts, and S. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.