Global Features and Energy-Based Models for Estimating and Improving the Quality of Scene Segmentations

Kenton McHenry, Svetlana Lazebnik, Jean Ponce

Abstract. *Local information cannot capture all of the image/scene constraints available for image segmentation. Imposing global constraints such as context and shape priors are known to often improve segmentation results. In this paper we consider two things. First we propose using global information, in the form of global features over a segmentation, to derive a means of estimating the quality of that segmentation. Second we attempt to use these quality estimates to improve an initial segmentation, in effect imposing learned global constraints. Given images that are initially over-segmented into regions of nearly uniform color and texture we use a set of global features on these regions and their class assignments to learn an energy function. This energy-based model is trained so as to assign lower energies to segmentations that have a larger percentage of correctly labeled pixels. The resulting energy function is then used to refine a given segmentation constructed from local features of the initial (over-segmented) regions. We demonstrate our approach with quantitative and qualitative results.*

1 Introduction

Image segmentation may be one of the most difficult tasks in computer vision. Learning has been used to tackle this problem through the use of local properties such as materials [1–3], modeling region similarity [4], and more recently geometry from visual properties [5]. These local methods can be improved upon by incorporating non-local information such as context [6–9], symmetry [10], and shape [11–14]. In Rabinovich et al. [8] context in the form of class co-occurrences are used. Within a CRF framework they attempt to maximize object label agreement within a segmentation. In Rousson et al. [14] a template shape is used to a guide contour evolution such that curves matching the given shape are favored. The template is allowed to undergo affine transformations thus allowing some variability of the shape within the image. The parameters of the affine transformation, which come from the current level set function (i.e., labels), are what makes this a global function. Hoiem et al. [15] have proposed using their scene segmentation method [5] as a guide to an external object classifier. Specifically they have used the segmentation of an outdoor scene into ground/vertical/sky to determine a likely ground plane and camera position. This information is then used to guide a pedestrian or vehicle classifier by effectively limiting the position and scale of positives to ones consistent with viewing perspective. Though Hoiem et al. use this global information obtained from their segmentation to aid a separate classifier, this needs not be the case. For example in the case of indoor scene segmentation, knowledge of the ground plane would be useful in labeling regions such as chair, since chairs should be on the floor.

We attempt to use non-local information in the form of global features to estimate the quality of a given segmentation. Segmentations that are more correct (i.e. have more of their pixels labeled correctly) should match a series of global constraints specific to their particular scene type (in our case indoor scenes). Given this quality estimate we then attempt to improve a segmentation by choosing changes in class assignments that optimize the estimated quality. Segmenting indoor scenes is a challenging task since most objects present (chairs, desks, etc.) do not have very characteristic textures or colors. With regards to less local information such as context these scenes are also challenging as nearly all images contain all of the classes making class co-occurrences which are prevalent in datasets like the MSRC dataset of little use. Given a set of global features that attempt to capture various global scene properties we learn an energy-based model [16] such that better segmentations have lower energies. This energy function is then used to iteratively refine our initial segmentation. To our knowledge, our approach is unique in using an energy-based model trained with synthetically generated contrastive data to evaluate the quality of segmentations. Rather than working at the pixel level, we use regions from an initial over-segmentation of the image which we obtain from the graph based method of Felzenswalb et al. [17]. Regions obtained from an over-segmentation, also called super-pixels, have been shown to respect region boundaries [4, 18] and, unlike pixels, they provide us with some spatial support in order to calculate local features.

In Section 2, we define our set of local features and describe the local model used to construct our initial segmentations. In Section 3, we define our set of global features, describe the energy-based model and how it is used to improve a given segmentation. In addition we describe the problem of error amplification that occurs as one tries to iteratively refine an image and propose a means to minimize it's effects. Finally we present our results in Section 4, and our conclusions in Section 5.

2 Local Model

For our initial segmentation we use a method similar to that of Hoiem et al [5], providing a large set of local features within each super-pixel and allowing feature selection during the learning process to determine which features are best. A probabilistic classifier is trained on these features for each class and later combined using the active region method of Paragios et al [2].

2.1 Features

The features listed in Table 1 attempt to capture material properties, spatial properties, shape characteristics, and some geometric properties.

To model the materials making up the classes, we use color and texture. For color, we convert to the LAB color space so that Euclidean distance captures the similarity between colors and store the mean and standard deviation of each channel within each region. For texture, we use the MR filter bank of [19] which has the property of being rotationally invariant. Each image is convolved with a filter bank consisting of spot, edge, and bar filters at two scales ($\sigma = 0.5, 1$).

Local Feature Descriptions	**#**
Color	**6**
LAB values: mean and std	6
Texture	**6**
DOOG filters: mean response	6
Location	**10**
Mean x and y	2
Min x and y	2
Max x and y	2
# of points on top border	1
# of points on bottom border	1
# of points on left border	1
# of points on right border	1
Shape	**4**
2nd moment matrix	3
Size	1
Geometry	**36**
Longest Contour Line: mean x and y	2
Longest Contour Line: orientation	1
Longest Contour Line: length	1
Long Lines: total number in region	1
Long Lines: weighted total in region	1
Long lines: % of nearly parrallel pairs	1
Line Intersections: mean intersection	2
Line Intersections: histogram over 8 angles, entropy (near center)	9
Line Intersections: histogram over 8 angles, entropy (far from center)	9
Line Intersections: histogram over 8 angles, entropy (very far from center)	9

Table 1. *The features used to represent each region. The "#" column indicates the dimensionality of the feature.*

The edge and bar filters are generated at 6 orientations between 0 and π. To make the result rotation invariant we only keep the maximum response of the 6 orientations. Thus the result consists of 2 spot filter responses and 2 edge/bar filter responses at 2 scales for a total of 6 values. Rotationally invariant texture features are important when the goal is to distinguish materials. Consider two similar pieces of wood, one rotated so that the grain is vertical and the other rotated so that the grain is horizontal. The important texture features is the grain, not the grain's orientation.

Some classes can take advantage of discriminative spatial information. For example ceiling regions tend to be at the top of the image and floor regions at the bottom. This information is obtained by storing the mean x and y values of the pixels within each region. Since the mean position depends on the size and extent of the region we also store the minimum and maximum of the x and y values. Both doors and walls tend to occupy the top of many images but wall regions tend to cover more of the top pixels. Thus we also store the number of pixels each region has at the top, bottom, left and right borders of the image.

Many classes have discriminative shape information. For example door regions tend to be fairly large, due to being of a uniform material, and are often elongated in the vertical direction. To capture this shape information we use the 2nd moment matrix of the pixel positions within each region and store the major and minor axis length as well as the orientation of the corresponding ellipse. In addition, region sizes (i.e., the number of pixels within the region) are stored.

Most of our classes are planar or made up of several planar parts. The intersection of two planes in 3D will project onto a line in the image plane. We attempt to identify regions that are likely to correspond to planes by finding the longest line along their contour. From such a line we store the mean x and y values along with its orientation and length. Knowing the orientation of these planes in 3D would be extremely helpful. We attempt to indirectly capture this information as was done in [5] by looking at long near-parallel lines within each region and observing their intersections to get an idea of the planar surfaces vanishing line.

2.2 Probabilistic Model

Regions from our training images are represented by their 62-dimensional feature vectors x_i. In this paper, we use the logistic regression version of Adaboost ("Adaboost.L", see [20, 21]) to learn the posterior probability $P(y|x)$ that some observed feature vector x belongs to some class y. We have experimented with both SVMs and decision trees [5] as our weak classifiers, and the latter have empirically given better results. One of these boosted decision trees is constructed for each of our classes. In each case training is done in a one versus rest manner.

2.3 Segmentation

We use the *coupled geodesic active region* (CGAR) approach of Paragios et al. [2] to construct an initial image segmentation from the probabilities provided in the above local model. The CGAR attempts to label each pixel so as to maximize the probability of each class assignment while at the same time maximizing boundary smoothness. Since class labels are assigned to pixels we need to add a step to retrieve labels for our regions. Since the original over-segmentation tends to be reliable this also tends to improve things in the case where the curve evolution incorrectly eats away at a region. To do this we simply have each active region vote once for every pixel it occupies within a region. The class of the active region with the most votes is then assigned to the region.

3 Global Model

We now describe the global features, our model to estimate a segmentations quality and suggest a means of using this estimate to improve an initial segmentation.

Global Feature Descriptions	#
Compactness	**1**
Number of connected groups	1
Location	**43**
Mean position (per class)	14
Min position (per class)	14
Max position (per class)	14
Mean P(class|y-coordinate)	1
Shape	**29**
% of image covered (per class)	7
Total boundary length internal to image	1
Length of boundary internal to image (per class)	7
Length of boundary on image boundary (per class)	7
Total boundary length to area ratio (per class)	7
Neighborhood Statistics	**56**
% of each class nearby (per class)	56
Confidence	**1**
Sum of P(class|region)	1

Table 2. *The features used to represent a correctly labeled image.*

3.1 Features

The previous section only considered local information in terms of features within individual regions. However it is known that there is useful information that can not be captured locally. Consider an indoor scene with a class floor and ceiling. Floors are usually at the bottom of an image and ceilings at the top. If we see a large group of regions labeled floor at the bottom, a large group of regions labeled ceiling at the top, and one region labeled floor among the ceiling regions, then we should expect that this one region may be wrongly labeled. We attempt to capture a variety of non-local information with the global features listed in Table 2.

A correctly segmented image should be compact in the sense that there are a handful of connected groups of similarly labeled regions. For example, an image that has had its regions randomly assigned labels would not likely be compact since there would likely be small groups of floor scattered throughout the image. We capture this idea of compactness by counting the number of connected groups.

As in the local case, position is a useful global cue. This time, however, we are concerned with the set of regions assigned to a particular class. We again capture position information by storing the mean, standard deviation, min, and max of the x and y coordinates of each class. As an example consider the class floor, which one expects to be at the bottom of an image. If in a given image most of what is labeled floor is at the bottom except for one or two regions which are labeled near the top, the mean position of the class would be shifted upward. A more correct segmentation would not have these out of place regions and would have a lower mean position. In this case the lowering of the mean position is helpful in distinguishing one segmentation as being better than another.

Size is a useful feature as well. In most indoor images we would expect that most of what we see be either floor or wall. To capture this we store the percentage of the image covered by each class. Shape, though a usual cue, would be too complex too directly use as a feature within our setup. We attempt to indirectly capture characteristics of shape by looking at the length of the boundary of each class and also looking at the ratio of the boundary versus the area. Consider the difference between tables and chairs. Tables are larger objects with smooth boundaries and thus should have a small boundary as compared to its area. Chairs on the other hand have sharp bends at the seat, arms, and legs. Combined with their smaller area, chairs should have a higher boundary to area ratio.

Next, we include neighborhood statistics. An example of this is that chairs tend to be near tables. Another could be that chairs usually are not surrounded by wall, which might mean that the chair is flying. To capture this for each class we count the percentage of each other class along its borders. We include image boundary as a class here, giving us 8 numbers for each of the 7 classes.

Lastly, to prevent the learned global model from completely overriding the local model we add the following local confidence term:

$$\sum_i n_i P(y_i|x_i)$$

where x_i is the local feature vector for region i, y_i is the class currently assigned, n_i is the number of pixels within the region, and P is the local probability of region i being class y_i.

3.2 Energy-Based Model

We attempt to learn an energy function [16] such that better segmentations will have lower energies than worse ones. Learning to assign an absolute energy to each segmentation is difficult so we use the idea of contrastive divergence [22] to instead learn locally within the feature space which segmentations are better and should have lower energies than nearby worse segmentations. We begin with an energy function of the form:

$$E(w, z_i) = w^T z_i$$

where z_i is a global feature vector for image i and w is the vector of parameters for the energy function. We choose a linear function for it's simplicity; however more complex functions can be used. Within our training data we have a set of feature vectors z_i, which we wish to have low energy values, as they represent good segmentations (perhaps the ground truth or a slightly noisy version of it). Relative to these we have a set of contrastive examples z_i' that are nearby, in the sense that they come from segmentations that are for the most part the same as those of z_i except that some of the label assignments have been changed to incorrect values. These contrastive examples should have higher energies than the

good examples. Our goal is to set w so as to maximize the number of contrastive examples with energies higher than nearby positive examples.

We can set w by minimizing the negative log-likelihood loss described in LeCun et al. [16]:

$$L_{nll}(w, z_i) = E(w, z_i) + \log \left(\int_{z' \in z'_i} e^{-E(w,z')} \right)$$

The above loss function is derived by maximizing the probability of the good examples, given in the form of a Gibbs distribution, over their relative contrastive examples. This is done by minimizing the negative log-likelihood, which is why it is referred to as the negative log-likelihood loss function. Differentiating it with respect to the parameters, w, yields the following update equation:

$$w \leftarrow w - \eta \left(\frac{\partial E(w, z_i)}{\partial w} - \sum_{z' \in z'_i} \frac{\partial E(w, z')}{\partial w} P(z', w) \right)$$

Using gradient descent we find a w which is a local minima of the loss. The learning rate, η, is determined by a line search at each iteration of the gradient descent. Within contrastive divergence learning, $P(z', w)$ is usually estimated via MCMC. This estimation requires that we generate new contrastive examples at each iteration. For our setup this is impractical as it would require a somewhat costly process over the training set (Section 3.2). Thus, we instead estimate $P(z', w)$ through the Gibbs distribution:

$$P(z', w) = \frac{e^{-E(w,z')}}{Z}$$

where the normalizing constant Z is the sum of the energies of our training examples.

Training data The training data is generated in the following manner. Let z be the feature vector representing a ground truth segmentation s of our training set. We now select a random number of regions (super-pixels) in the image and change each of their classes to one of the other 6 possibilities, yielding a new segmentation s' with feature vector z'. The segmentation s is an improvement over s' since one or more regions of s' are mislabeled (this will be our contrastive example). We can generate our positive and contrastive training data in this manner by randomly changing regions to other classes. In addition to using ground truth segmentations as the basis for improved segmentations we also use versions of the ground truth with noise added. Noise is added by randomly selecting a number of regions and changing their class assignments. When the worse segmentation is generated for the contrastive example we make sure not to change any of these mislabeled regions back to the correct class. The idea behind this is that we will likely encounter these less than perfect segmentations.

3.3 Segmentation Improvement

Given an initial scene segmentation we attempt to improve it by enforcing the learned global constraints captured within the energy-based model above. We adopt a simple greedy method to refine the segmentation. For a given iteration we change each region in turn to each of the 7 classes and record the resulting energy. We then choose the region/class change with the lowest energy and repeat the process. We have also experimented with MCMC and a tree based search, choosing some n of the top changes at each iteration, however the greedy approach has performed the best.

Error Amplification Initial experiments had shown an unexpected side effect of iteratively improving a segmentation region by region. We have observed that the amount of the improvement obtained appeared much smaller for segmentations that were mostly correct. In fact it can be shown that it becomes harder and harder to iteratively choose regions to change and improve a segmentation as more of the regions are labeled correctly. Due to length constraints the proof is omitted. However, consider the following key observation. Given an already reasonably good segmentation there are many more potential degradations than improvements. By updating a segmentation region by region we must consider changing each region to every possible class at each iteration. Within the EBM framework this is done by choosing the change that minimizes the learned energy function. As the segmentation improves, however, the number of changes that lead to worse segmentations rapidly outnumbers the number of good changes. The trained EBM is not perfect, if used to label transitions as improvements it will have some number of false positives. The number of false positives are in fact amplified as the number of potential mistakes gets larger than the potential improvements. We refer to this as error amplification. Note that if the EBM were perfect this would not be an issue.

Quality Prediction One way to deal with the above situation is to focus on the worst segmentations. Using the same global features we train a boosted tree classifier. The training data again consists of randomly damaged ground truth segmentations. However, this time we make a positive dataset with those segmentations that have greater than or equal to some percentage of the image labeled correctly, in our experiments 80%. The negative dataset will consist of all other segmentations. The local segmentations from test images are first passed through this classifier. If classified as having a classification rate less than the threshold value we attempt to improve the segmentation using the EBM described in the previous section.

4 Experiments

Our data consists of 105 annotated indoor images and 7 classes: ceiling, chair, desk, door, floor, wall and other (kindly provided by Toyota Motor Corporation). The images were taken in locations which varied in appearance (styles

	High	Medium	Low
Initial	62.11%	76.76%	89.38%
Final	75.46% (13.35%)	83.73% (6.97%)	89.10% (-0.28%)

Table 3. *Initial and post global refinement classification rates for test images with damaged initial segmentations. High, medium and low indicate the amount of damage imposed to the ground truth segmentations. The values in the "initial" row indicates the resulting average percentage of correctly labeled pixels of these damaged segmentations. The values in the "final" row indicate the average percentage of correctly labeled pixels after the global refinement. The value in parenthesis indicates the improvement from the initial value.*

and furniture), under various lighting conditions and arbitrary viewpoints. In our experiments, we randomly generate 10 splits of this data, each with 75% used as training data and the remaining 25% as test data.

For the local classifiers, we run the logistic version of Adaboost for 200 iterations, using three-level decision trees as weak classifiers. A validation set is used to verify that the model is not over-fitting. We biased the positive data so that classifying them correctly was 3 times as important as the negative data. In addition, the importance of classifying each region correctly is biased by the region's size. For the global energy model we train with 20 positive examples per training image, one the ground truth and 19 noisy versions of the ground truth. Each of these in turn has 200 contrastive examples. This training data is generated once at the beginning of training.

We begin by testing the effect of the error amplification. Rather than starting with the segmentation provided by the local classifiers and the CGARs we randomly damage the ground truth segmentations of the test images in the manner specified for the EBM training data. The amount of damage is varied to create three categories: one with low, one with medium, and one with a high amount of damage. The low category will have relatively fewer incorrectly labeled regions as compared to medium and high and so on. We then apply the greedy global refinement discussed in Section 3.3 to these initial segmentations. We do not consider the local term in these experiments, so none of the observed improvements are a result of simply trying to maximize the local classification results. Also, we have observed that the greedy refinement is only reliable when the change in energy is large, thus we stop making changes when the change in energy becomes small. The results are shown in Table 3. What should be noted from this synthetic test is the amount of improvement seen in each case. In the low category, with little initially incorrect, the segmentation actually gets worse. On the other hand the other two categories show considerable improvement.

For the rest of our experiments we start with an initial segmentation produced by the local classifiers and 1000 iterations of the CGARs. These segmentations are greedily refined using our global energy model. We experiment with and without the local confidence term in the EBM. The weight assigned to the local term regulates the relative importance of the local and global features (shown in Figure 1 for one of the dataset splits). If the magnitude of this weight is high then

Method	Classification Rate	Classification Rate after Quality Prediction
Initial	76.55%	68.83%
Final	78.12% (1.66%)	72.08% (3.25%)
Final (no local term)	78.21% (1.31%)	72.28% (3.45%)

Table 4. *Classification rates for the various methods: local features only, global features + local term, global features only (initialized by local segmentation). Middle column gives results averaged over all test images. Right column gives results on test images classified as being initially worse than* 80% *correct via the quality prediction discussed in Section 3.3. Values in parentheses are the average improvements over the local segmentations. Note, the quality prediction step labeled* 30.3% *of the test images as being less than* 80% *correct (an accuracy of* 69.1%*).*

only local information is considered and the improvement will be 0 since nothing will change from the initial segmentation. If the magnitude is too low then the global information completely overrides the local information which sometimes results in worse segmentations. When set to a more appropriate value, chosen by the training of the EBM, the weight of the local term allows improvements from the global information without completely ignoring the initial local information.

Improvement results obtained by starting from these local segmentations are shown in Table 4. Overall, the CGARs labeled 76.55% of the pixels correctly. With the use of the global refinement, we were able to increase the percent of correctly labeled pixels by an average of 1.66%. Like Shotton et al. [9] who use context to improve segmentation results, we see a seemingly small numerical improvement overall. While in their case small improvements appear to be distributed over all their test images, our case involves large improvements that occur on the worst initial segmentations (as much as 20%, see Figure 2). We mention again that image context alone, a strong cue within the MSRC dataset used in [9], would likely not lead to improvements on the Toyota dataset as every image contains nearly every class. If we use the quality prediction described in Section 3.3 and only attempt to improve those segmentations that were classified as being below 80% correct we achieve an average improvement of 3.25%. As an interesting side note, the last row of Table 4 uses global information alone without the local term (i.e. no image information) and achieves and average improvement of 1.31%.

Qualitative results are shown in Figure 2. In particular, consider the images in the first two rows. In Figures 3(a) and 3(b), we show these two images, their over-segmentations, and the local probabilities assigned for each class. The results from the local segmentations can be seen in the third column of Figure 2. In Figure 3(a), the local probabilities are not too bad, and after the global refinement the overall recall only improves slightly, from 78.8% to 79.44% (fourth column of Figure 2). On the other hand, in Figure 3(b) large portions of the desk are assigned small probabilities of being desk and high probabilities as to being other. In this case the improvement from the global refinement is considerable, with the recall going from 63.2% to 83.13%. The global refinement, concerned with the quality of the overall labeling of the image, is able to recover most of

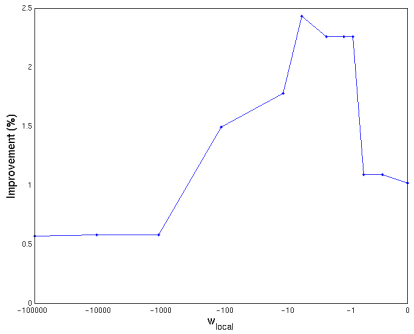


Fig. 1. *The effect of the local term on the improvements obtained. The weight,* w_{local}*, assigned to the local term within the EBM is a negative value whose magnitude indicates how much trust should be given to the local classifier results. As the magnitude of the weight gets larger, left on the x-axis, the improvement becomes smaller as this tends to ignore the global information and effectively returns the original local segmentation. As the magnitude of the weight gets smaller, right on the x-axis towards 0, the improvement also gets smaller. In this case the global information completely overrides the local information which can result in worse segmentations. The trained EBM sets* w_{local} *to* -1.051 *which is near the peak on the plot.*

the desk even though the local probability of being desk is low. Similarly in rows 3 through 8 of Figure 2, pieces of walls, floors, and chairs are recovered despite being misclassified by the local classifiers.

5 Conclusion

We have proposed a means of estimating the quality of a segmentation and using this estimate to attempt to improve it. This is performed via an energy-based model trained with global features so as to prefer segmentations that are more correct. In addition to the Toyota dataset discussed in this paper we have conducted experiments on our own larger indoor dataset, containing 232 images and the same 7 classes. Results are similar with an average improvement of 1.56% overall and 2.15% with quality prediction (where the quality prediction labeled the worst segmentations with an accuracy of 65.0%).

Our experiments have revealed an error amplification that is inherent to any iterative refinement method such as the greedy refinement described in Section 3.3. We deal with this by proposing a quality prediction step that allows us to focus our efforts on the lower quality segmentations which are in most need of improvement. An interesting question is whether or not other, common, methods for combining information at various levels suffer from an error amplification as well. If so incorporating non-local information to a scheme that produced poor segmentations with local information alone would show large improvements where it would otherwise not if the local segmentation was better. This is a topic we wish to investigate further.

References

1. Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. (2003)
2. Paragios, N., Deriche, R.: Coupled geodesic active regions for image segmentation: A level set approach. (2000)
3. Roussson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. (2003)
4. Ren, X., Malik, J.: Learning a classification model for segmentation. (2003)
5. Hoiem, D., Efros, A., Hebert, M.: Geometric context from a single image. (2005)
6. He, X., Zemel, R., Carreira-Perpinan, M.: Multiscale conditional random fields for image labeling. (2004)
7. Kumar, S., Hebert, M.: Discriminative fields for modeling spatial dependencies in natural images. (2003)
8. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. (2007)
9. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearence, shape and context modeling for multi-class object recognition and segmentation. (2006)
10. Riklin-Raviv, T., Kiryati, N., Sochen, N.: Segmentation by level sets and symmetry. (2006)
11. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. (2002)
12. Cremers, D., Soatto, S.: A pseudo-distance for shape priors in level set segmentation. 2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision (2003)
13. Cremers, D., Sochen, N., Schnorr, C.: Towards recognition-based variational segmentation using shape priors and dynamic labeling. International Conference on Scale Space Theories in Computer Vision (2003)
14. Rousson, M., Paragious, N.: Shape priors for level set representations. (2002)
15. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. (2006)
16. LeCun, Y., Chopra, S., Huang, F., Ranzato, M.: A Tutorial on Energy Based Learning. In: Predicting Structured Outputs. MIT Press (2006)
17. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. (2004) 167–181
18. Malisiewicz, T., Efros, A.: Improving spatial support for objects via multiple segmentations. (2007)
19. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. (2002) 255–271
20. Collins, M., Schapire, R., Singer, Y.: Logistic regression, adaboost and bregman distances. Machine Learning (2002)
21. Schapire, R., Rochery, M., Rahim, M., Gupta, N.: Incorporating prior knowledge into boosting. 19th International Conference on Machine Learning (2002)
22. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Computation (2002)

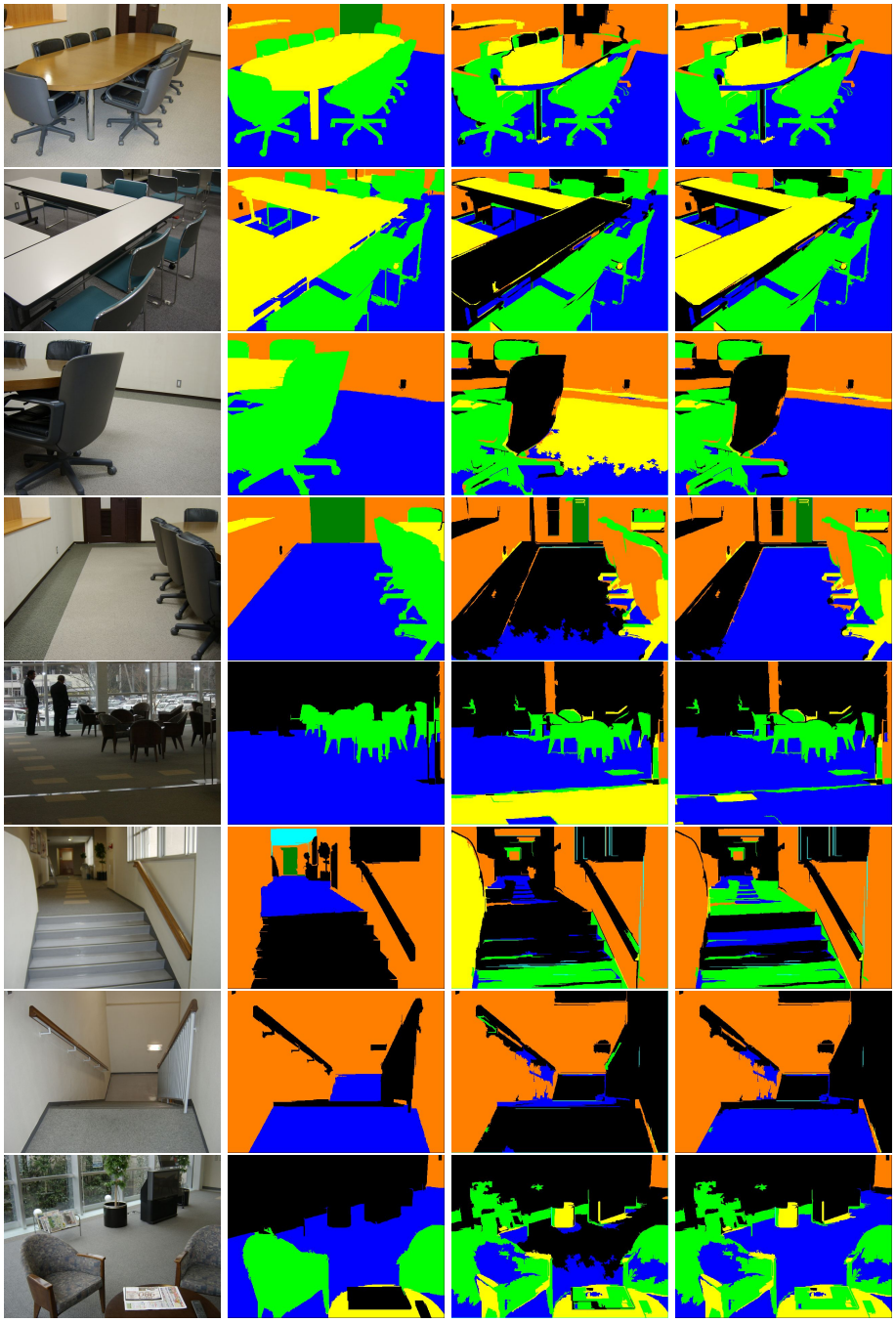
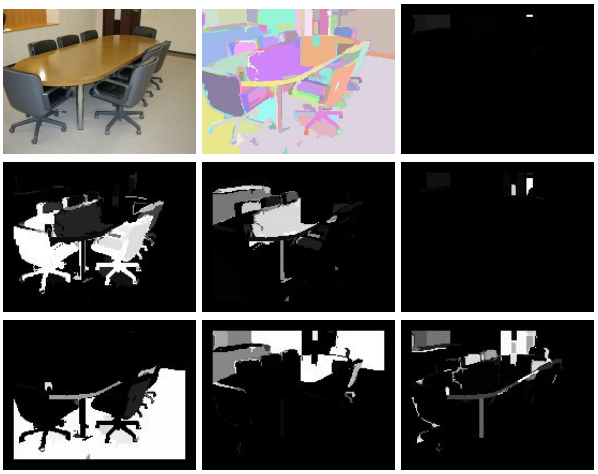
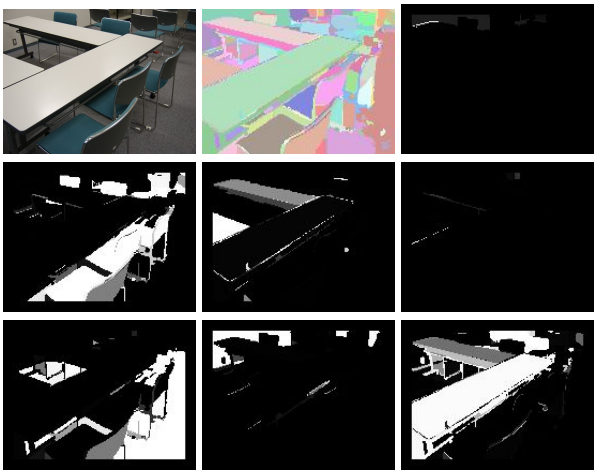


Fig. 2. *From* **left** *to* **right:** *test image, ground truth, output from CGAR, output from CGAR + Global Refinement. Initial vs. final error rates (from* **top** *to* **bottom***): 78.8% to 79.44%, 63.2% to 83.13%, 52.4% to 74.02%, 40.1% to 62.3%, 69.6% to 88.32%, 60.5% to 72.34%, 55.3% to 75.3%, 58.5% to 73.62%.*



(a) Test image 1.



(b) Test image 2.

Fig. 3. Top left: *a test image,* **top-middle:** *regions,* **top-right:** *probability of each region being of class ceiling,* **middle-left:** *probability of being of class chair,* **middle-middle:** *probability of being of class desk,* **middle-right:** *probability of being of class door.* **bottom-left:** *probability of being of class floor,* **bottom-middle:** *probability of being of class wall,* **bottom-right:** *probability of being of class other. Brighter values indicate higher probabilities.*