Pairwise Grouping Using Color

Marius Leordeanu and Martial Hebert

Technical Report TR-0846
December 6th, 2008

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Grouping was recognized in computer vision early on as having the potential of improving both matching and recognition. Most papers consider grouping as a segmentation problem and a hard decision is made about which pixels in the image belong to the same object. In this paper we instead focus on soft pairwise grouping, that is computing affinities between pairs of pixels that reflect how likely that pair is to belong to the same object. This fits perfectly with our recognition approach, where we consider pairwise relationships between features/pixels. Some other papers also considered soft pairwise grouping between features, but they focused more on geometry than appearance. In this paper we take a different approach and show how color could also be used for pairwise grouping. We present a simple but effective method to group pixels based on color statistics. By using only color information and no prior higher level knowledge about objects and scenes we develop an efficient classifier that can separate the pixels that belong to the same object from those that do not. In the context of segmentation where color is also used only nearby pixels are generally considered, and very simple color information is taken into account. We use global color information instead and develop an efficient algorithm that can successfully classify even pairs of pixels that are far apart.

1. Introduction

Grouping was recognized in computer vision early on as having the potential of improving both matching and recognition. We are interested in soft pairwise grouping mainly because of its potential of improving our recognition and matching approaches using pairwise relationships between features [7] and Section 5. Grouping is one way of constraining the matching/recognition search space by considering only features that are likely to come from the same object. This low level process is essential for higher level tasks such as improving recognition and matching because it uses general, category independent information to prune the search space and guide the recognition process on the right path. In Figure 1 we show two examples that intuitively explain this idea. The images in the left column contain edges extracted from a scene. We notice that without grouping the objects are not easily distinguished (e.g. the bus, or the horse). However, after using color information for perceptual grouping we are able to retain only the edges that are likely to belong to the same object as the edge pointed out by the red circle (right column). Grouping could also bring a second benefit to the recognition process, because, without it, matching could be very expensive especially when the image contains a lot of background features/clutter. As Grimson [5] has shown, the complexity of the matching process when the search is constrained is reduced from an exponential to a low order polynomial. Therefore, it is important to be able to establish *a priori* which pairs of features are likely to be part of the same object, and discard all the other pairs. To summarize, perceptual grouping does not only improve the recognition performance but it also reduces the computational complexity.

In our work [7] we are most interested in grouping pairs of features, that is, establishing which pairs are likely to belong to the same object. This could be beneficial to our matching approach (also used in recognition) that is mostly based on pairwise relationships between features.

2. Soft Grouping

Grouping is the task of establishing which features in the image are likely to belong to the same object, based on cues that do not include the knowledge about the specific object or object category. In his pioneering book [11] Marr argued, based on medical human studies, that a vision system should be able to recover the 3D shape of objects without knowledge about the objects' class or about the scene. In support of this idea come simple facts from everyday life. Humans are able to see a car parked in a room, even though that would be totally unexpected based on prior experience. Also, humans are able to perceive the shape of an abstract sculpture from a single image, even if they have never seen it before and have no clue about what it might represent. It seems clear that for humans both knowledge about the object class and the scene is not necessary for shape perception. But if one is able to perceive the 3D shape of the scene with respect to its own reference frame, then in most cases it would be relatively easy to figure out which features in the scene should belong together. In fact perceptual grouping most probably helps humans in the process of 3D shape recovery and not the other way around. We consider grouping as a process that happens entirely before the object recognition stage, and uses cues such as color, texture, shape, and perceptual principles that apply to most objects in general, regardless of their category or specific identity.

Unlike prior work in grouping [10], [15], [3], [6], [13], [12], [9], we do not make a hard decision about which features belong together. And that is for an important reason: it is sometimes impossible to divide features into their correct groups without the knowledge of the specific category (or the desired level of detail): for example, is the wheel of a car a separate

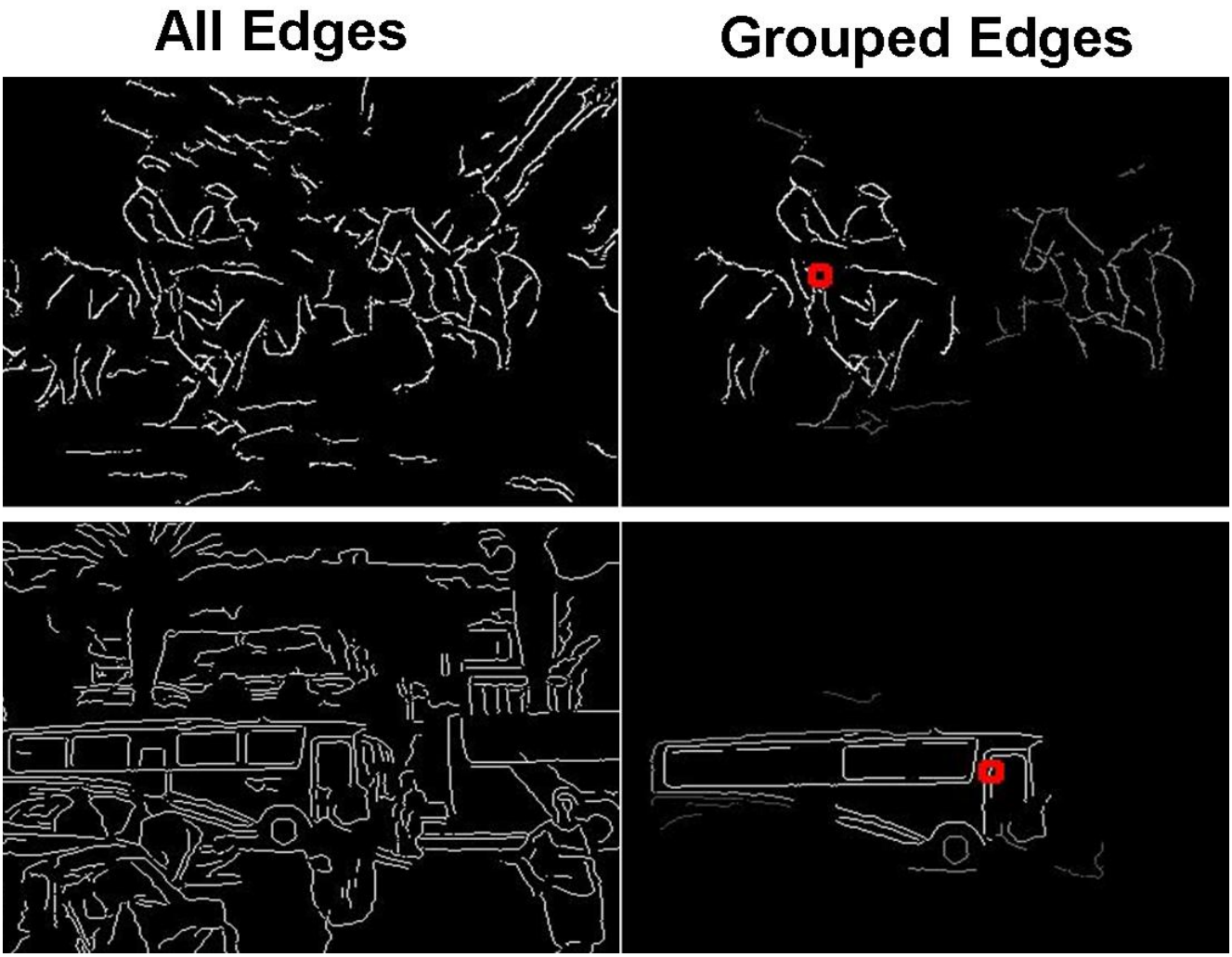

Figure 1. If grouping is not used it is very hard to distinguish the separate objects (left column). After grouping (right column) it is perceptually easier to distinguish them (the bus and the horse)

object or is it part of the whole car? We believe that both situations can be true at the same time, depending on what we are looking for. If we are looking for whole cars, than the wheel is definitely a part of it. If we are looking just for wheels then (at least conceptually) it is not. While perceptual grouping alone should most of the time separate correctly most objects (the ones that are clearly at different depths, such as a flying plane from a close car), it sometimes does not have access to enough information to make the correct hard decisions. We immediately see why it is important to keep most the grouping information around and transmit it to the higher recognition processes (without making hard decisions, except for pruning the cases when the pairwise grouping relationship is extremely weak). Instead of being interested on forming exact feature groups based on perceptual information alone, we rather focus on the quality of pair-wise grouping relationships and how to integrate them into our recognition step. Since we use pair-wise relationships at both the grouping and the recognition levels,

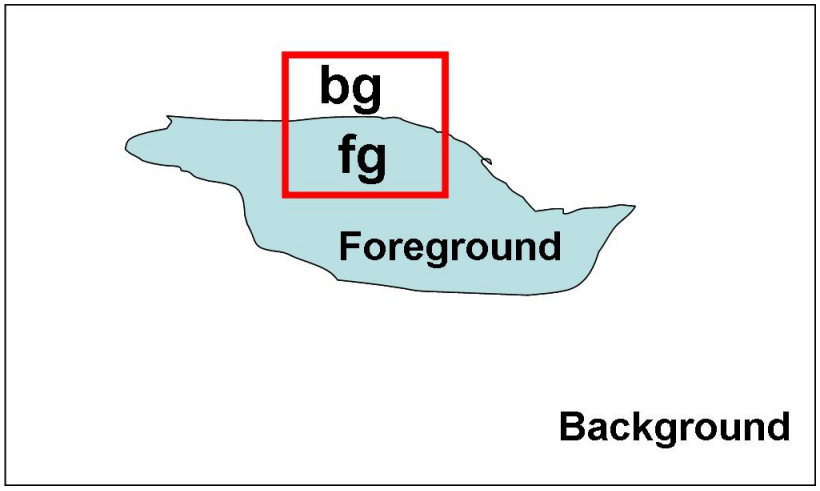

Figure 2. Automatically discovering the foreground mask

the two could be naturally integrated.

Our pair-wise grouping relationships are soft weights that should reflect the likelihood that the two features belong together (prior to recognition, that is before using any category specific knowledge). In this paper we focus only on pairwise grouping using color information: objects tend to have unique and relatively homogenous color distributions. We propose a novel and effective way of using color histograms for inferring pair-wise grouping relationships.

Color histograms are simple but powerful global statistics about objects' appearances that have been successfully used in some recognition applications. We present a novel algorithm for automatically discovering soft object masks (based on their color distributions), without knowledge about their locations or shapes. This method becomes very useful for pairwise grouping of features, because two features that belong (in a soft way) to the same mask are more likely to belong to the same object then pairs of features that belong to different masks.

2.1. Pairwise Grouping using the Geometry of Line Segments

Table 1. Perceptual cues used to describe the relationship between pairs of lines. Based on these cues we estimate the likelihood that pairs of lines belong to the same object or not

Cue	Description
Proximity	$\frac{d_p}{l_i+l_j}$
Distance	$\frac{d_i+d_j}{l_i+l_j}$
Overlap	$\frac{d_{oi}+d_{oj}}{l_i+l_j}$
Continuity	c
Parallelism	α
Perpendicularity	β
$Color_1$	difference in mean colors
$Color_2$	difference in color histograms
$Color_3$	difference in color entropies

Before we discuss our approach to color grouping we first present a method for grouping using geometry. Geometric perceptual cues are particularly important because of their connection to important studies in human vision (mainly from the Gestalt school). In our experiments we found that geometric grouping is more local than grouping based on color, because

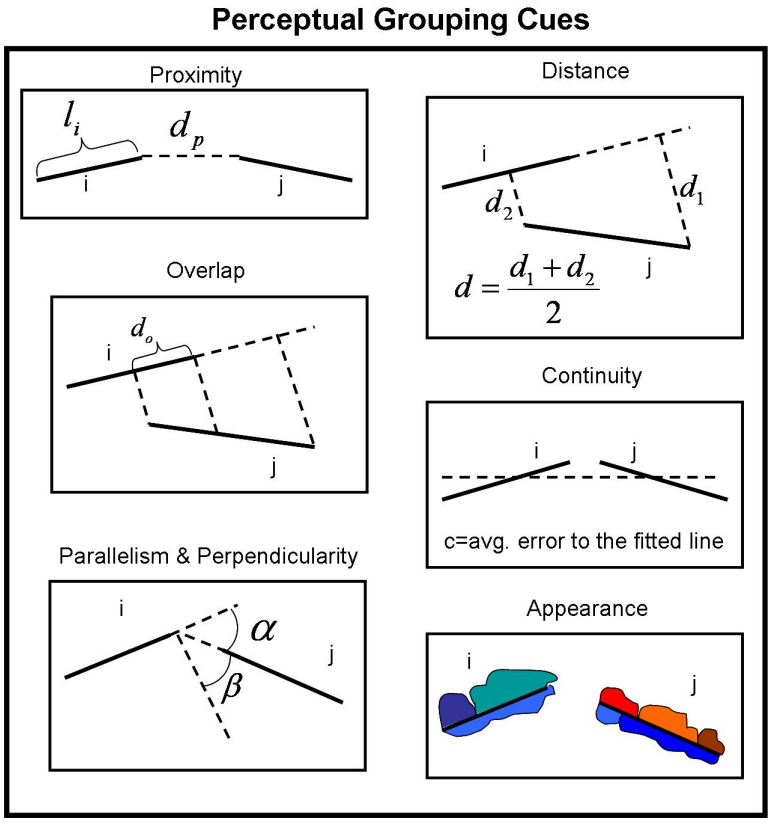

Figure 3. Geometric perceptual cues used for grouping pairs of line features

faraway pixels are harder to group using only geometry, with no color or texture information. We believe that in a complete grouping system one should use as many cues as possible including both geometry and color.

Our main features for object recognition are pieces of contours extracted from the image. In the grouping stage we approximate these contours by fitted line segments. The geometric grouping cues we propose to use consist of specific relationships between pairs of such line segments (i, j): proximity, distance, overlap, continuity, parallelism and perpendicularity as shown in Figure 3. We also use some local appearance cues (which are different than the global color histograms used in the next section), which are computed over the super-pixels adjacent to the pair of lines, such as: difference between the mean colors of the super-pixels belonging to each line, as well as the differences in color histogram and color entropy. All these cues, both geometric and appearance based, form a *relation* vector $\mathbf{r}(\mathbf{i}, \mathbf{j})$ for any pair of lines (i, j) whose elements are described in Table 1 (each row of the table corresponds to an element of $\mathbf{r}(\mathbf{i}, \mathbf{j})$). We have already performed some preliminary experiments with a basic implementation of this idea. We have manually collected about 300 positive pairs of lines (lines that belong to the same object) and 1000 negative ones (pairs of lines which do not belong to the same object), and learned a binary classifier on the corresponding relation vectors $\mathbf{r}$, using the logistic regression version of Adaboost [1] with weak learners based on decision trees [4].

In Figure 4 we present some results. The contours shown in red belong to line segments that were classified as being part of the same object as the line segment pointed by the white circle. We notice that in general only lines that are relatively close to the white circle are positively classified. This is due to the fact that in general geometric perceptual grouping is a local process and is not able to link directly pairs of faraway lines. Such pairs could be ultimately connected indirectly thorough intermediate lines. Of course, these are only preliminary results, and more thorough experimentation is needed.

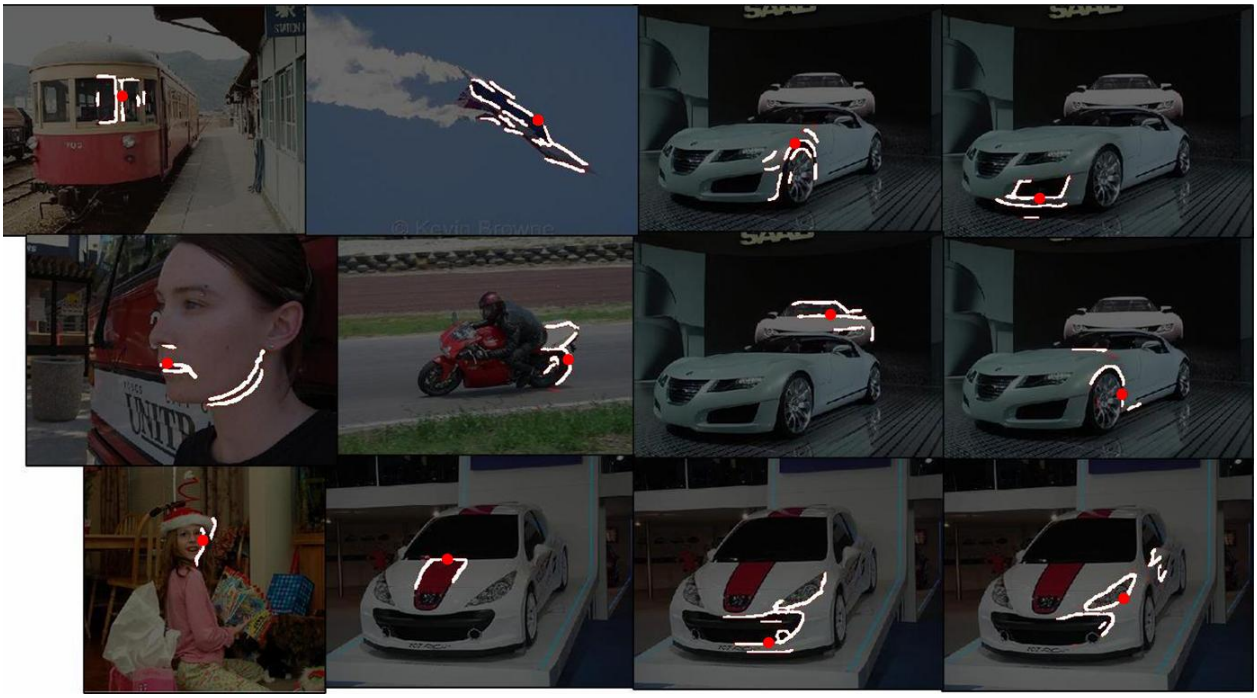

Figure 4. Pairwise grouping constraints based on geometry. The contours in white are the ones that establish a pairwise grouping relationship with the contour pointed out by the red circle

3. Unsupervised discovery of the foreground mask

The main focus of this paper is grouping based on color. The algorithm we will present for color grouping is more efficient and simpler than the one used for geometric grouping, and , yet, it gives more global results, being able to group pixels (or line features) that are far away in the image. We show that color histograms alone are often powerful enough to segment the foreground vs. the background, without using any local intensity information or edges. Other work, not closely related to ours, that used color histograms to separate the foreground from the background given minimal user input includes GrabCut [14] and Lazy Snapping [8]. In our case we want to discover, in a soft way, the object mask relative to a given point on the foreground, without any other information given. We want to be able to use the power of color histograms without knowing the true mask (or bounding box) of the foreground. But this seems almost impossible. How can we compute the color histogram of the foreground if we have no clue about the foreground size and rough shape?

For now, let us assume that we have the bounding box of an object in an image. Color grouping should separate the foreground vs the background in terms of certain global statistics using the bounding box given. In this case we use color likelihoods derived from color histograms: the histogram for the foreground object is computed from the bounding box of the object and the histogram of the background is computed from the rest of the image, similar to the idea used in object tracking [2]). This is reasonable if we had a correct bounding box. Here we show that even a completely wrong bounding box, that meets certain assumptions will give a reasonably good result. Then, at any given pixel, if we choose a bounding box that that meets those assumptions, we could potentially find which other points in the image are likely to be on the same object as that particular point. As we will show later a fairly good foreground mask can be obtained based on color distributions from a bounding box centered on the object, but of completely wrong shape, size and location. This is explained theoretically if we make certain assumptions. In Figure 2 the foreground is shown in blue and the bounding box in red. The bounding box's center is on the foreground, but its size and location are obviously wrong. We make the following assumptions, that are easily met in practice, even without any prior knowledge about the shape and size of the foreground:

1. The area of the foreground is smaller than that of the background: this is true for most objects in images

2. The majority of pixels inside the bounding box belong to the true foreground: this is also easily met in practice since

the center of the bounding box is considered to be on the foreground object by (our own) definition.

3. The color distributions of the true background and foreground are independent of position: this assumption is reasonable in practice, but harder to satisfy than the first two, since color is sometimes dependent on location (e.g. the head of a person has a different distribution than that person s clothes).

Even though the three assumptions above are not necessarily true all the time in practice, most of the time they do not need to be *perfectly true* for the following result to hold (they represent only loose sufficient conditions): let $p(c|obj)$ and $p(c|bg)$ be the true foreground and background color probabilities for a given color c, and $p(c|box)$ and $p(c|\neg box)$ the ones computed using the (wrong) bounding box satisfying the assumptions above. We want to prove that for any color c such that $p(c|obj) > p(c|bg)$ we must also have $p(c|box) > p(c|\neg box)$ and vice-versa. This result basically shows that whenever a color c is more often found on the true object than in the background, it is also true that c will be more likely to be found inside the bounding box than outside of it, so a likelihood ratio test (> 1) would give the same result if using the bounding box instead of the true object mask. This result enables us to use color histograms as if we knew the true object mask, by using any bounding box satisfying the assumptions above. The proof is straight forward and it is based on those assumptions:

$$p(c|box) = p(c|obj, box)p(obj|box) + p(c|bg, box)p(bg|box), \tag{1}$$

and

$$p(c|\neg box) = p(c|obj, \neg box)p(obj|\neg box) + p(c|bg, \neg box)p(bg|\neg box). \tag{2}$$

Assuming that the color distribution is independent of location (third assumption) for both the object and the background, we have $p(c|obj, box) = p(c|obj)$ and $p(c|bg, box) = p(c|bg)$. Then we have: $p(c|box) = p(c|obj)p(obj|box) + p(c|bg)p(bg|box)$ and similarly $p(c|\neg box) = p(c|obj)p(obj|\neg box) + p(c|bg)p(bg|\neg box)$.

Since the object is smaller than the background (first assumption) but the main part of the bounding box is covered by the object (second assumption) we have $p(obj|box) > 0.5 > p(bg|box)$ and $p(obj|\neg box) < 0.5 < p(bg|\neg box)$. By also using $p(c|obj) > p(c|bg)$, $p(bg|box) = 1 - p(obj|box)$ and $p(bg|\neg box) = 1 - p(obj|\neg box)$, we finally get our result $p(c|box) = p(obj|box)(p(c|obj) - p(c|bg)) + p(c|bg) > p(obj|\neg box)(p(c|obj) - p(c|bg)) + p(c|bg) = p(c|\neg box)$ (since $p(obj|box) > p(obj|\neg box)$). The reciprocal result is obtained in the same fashion, by noticing that in order for the previous result to hold we must have $p(c|obj) - p(c|bg) > 0$, since $p(obj|box) > p(obj|\neg box)$.

In Figures 5, 10, 11, 12 we present some results using this idea. The soft masks are computed as follows: each pixel of color c in the image is given the posterior value $p(c|box)/(p(c|box) + p(c|\neg box))$ in the mask. By the result obtained previously we know that this posterior is greater than 0.5 whenever the true posterior is also greater than 0.5. Therefore we expect that the soft mask we obtain to be similar to the one we would have obtained if we had used the true mask instead of the bounding box for computing the color distributions. We compute such masks at a given location over four different bounding boxes of increasing sizes (the sizes are fixed, the same for all images in all our experiments). At each scale we zero out all pixels of value less than 0.5 and keep only the largest connected component that touches the inside of the bounding box. That is the soft mask for a given scale. Then, as a final result, we average the soft masks over all four scales. In Figure 9 we show that the mask obtained is robust to the location of the bounding boxes, so long as the bounding boxes' centers are on the object of interest.

4. Pairwise Grouping Using Color

As we mentioned already, the idea of automatically discovering foreground masks at given locations can be easily used for pairwise grouping of features. To prove our point, we present a simple approach for using such masks for color grouping. Given two features (i, j) one can compute the associated soft masks m_i and m_j by centering the bounding boxes at the features locations (x_i, y_i) and (x_j, y_j) and follow the procedure explained previously, using bounding boxes of those four fixed different sizes. The values $m_i(y_j, x_j)$ and $m_j(y_i, x_i)$ can then be used by a classifier to establish the likelihood that the two features belong to the same object. In Figure 13 we present some preliminary results of this idea. The red circles represent the location of some contour (feature) i. In white we show all those contours (features) j that were automatically classified as likely to belong to the object of i. Here the classifier was simply thresholding the average $(m_i(y_j, x_j) + m_j(y_i, x_i))/2$ at 0.5 (everything above 0.5 was considered positive). The images show weighted contours for the positive examples, and no contours for the negative. It is important to note that the shape and extent of the foreground is not known, and that all internal parameters are fixed (such as the four fixed bounding box sizes).

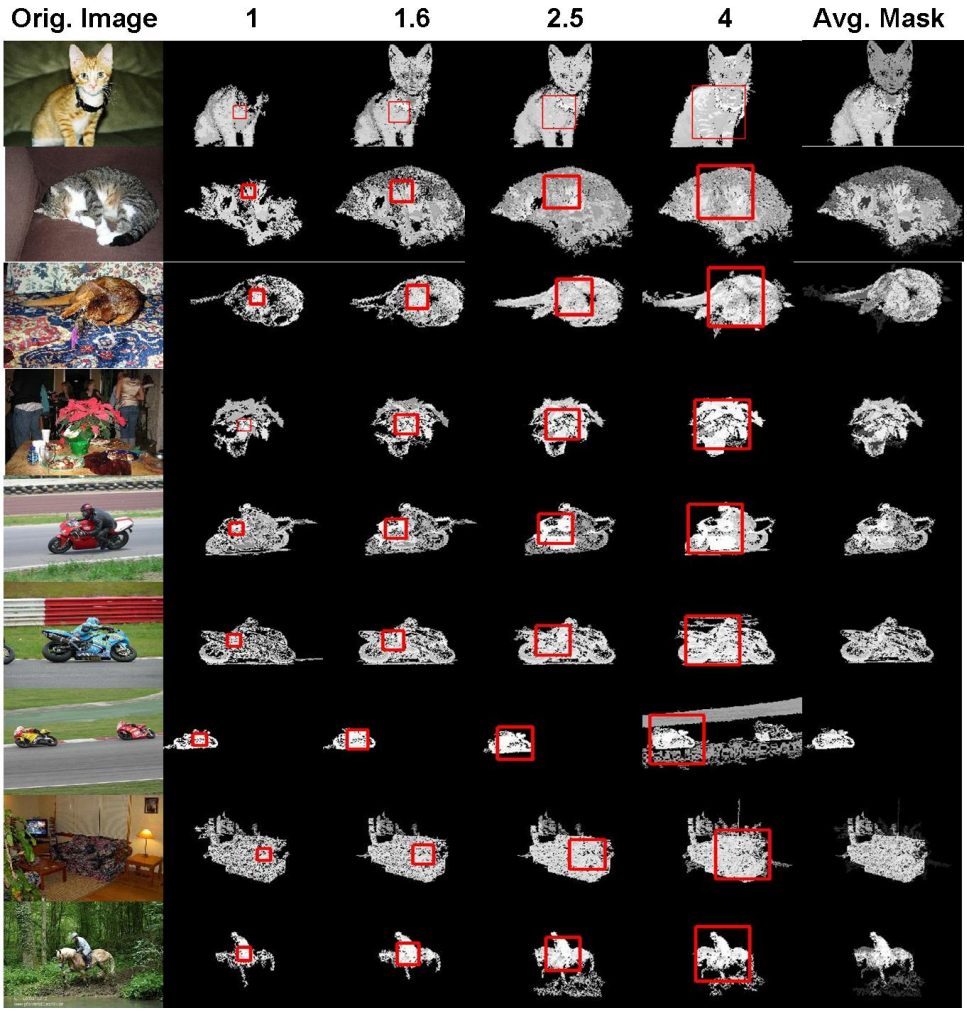

Figure 5. Object masks obtained from wrong bounding boxes, centered on the object of interest. Averaging over several fixed scales improves the result. The masks shown are computed from color likelihood histograms based on the bounding boxes (completely wrong) shown, which are centered on the object of interest. The interior of the boxes is considered to be the foreground, while their exterior is the background. The posterior color likelihood image is obtained, threshold-ed at 0.5 and the largest connected component touching the interior of the bounding box is retained. We notice the even when the bounding boxes have wrong locations and sizes the masks obtained are close to the ground truth. Different bounding boxes sizes (which are fixed for every image, starting at 8 pixels and growing at a rate of 1.6) are tried and the average mask is shown in the rightmost column.

In Figure 14 we present some *failure* examples of the color pairwise constraints. The algorithm does fail in the sense that it connects contours from the house to contours from the car, but it also connects car contours among themselves, so it should still improve the recognition performance (because most clutter is removed). The main reason for these lower quality results

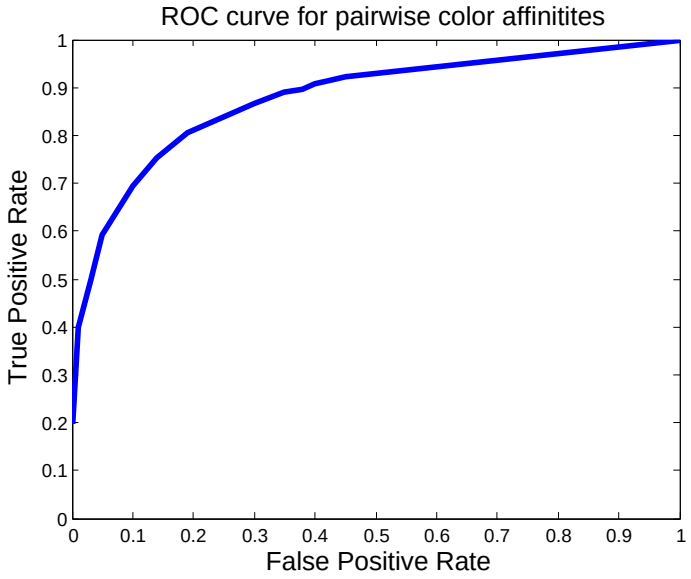

Figure 6. The ROC curve for the pairwise classifier on the MSR database, on about 5000 positive and 5000 negative pairs. Notice that we could eliminate almost all negative pairs ($FP = 0$) while keeping a significant part of the positive ones ($TP = 0.4$). For the purpose of recognition and matching we do not need to classify correctly all positive pairs. It is more important to remove most of the negative ones (very small false positive rate).

is that the house and the car have similar colors (relative to the histogram bin-ing). This issue could probably be solved to a certain extent by improving the color histogram-ing. We also want to point out that these *failures* are the exception and not the rule. In fact the vast majority of our results are of similar quality with those in Figure 13.

We measured the performance of this simple color grouping algorithm on the MSR database, for which ground truth masks are provided. For each image we randomly picked inside the true foreground object mask 100 positive pairs of points, and similarly 100 negative pairs for which one point is randomly picked inside the mask and the other outside of it. This database may not be the best choice for our algorithm because the foreground objects are sometimes very large and thus tend to violate our first assumption, but the results are encouraging. In Figures 10, 11, 12 we show some qualitative results on about half of the images from the database. We also measured the performance of the pairwise classifier quantitatively, Figures 6, 7. We plan on using this pairwise classifier for matching and recognition. Grouping has the potential of considerably pruning the search space for our matching algorithm that uses pairwise constraints. Our results indicate that indeed the pairwise grouping can group most of the positive pairs, while correctly separating almost all negative ones. In cluttered scenes with large backgrounds this can be very useful.

5. Combining Matching with Grouping

Our matching algorithm is based on the graph matching problem, which consists of finding the indicator vector $\mathbf{x}^*$ that maximizes a certain quadratic score function:

$$\mathbf{x}^* = argmax(\mathbf{x^T M x}). \tag{3}$$

Here $\mathbf{x}$ is an indicator vector such that $\mathbf{x}_{ia} = 1$ if feature i from one image (or object model) is matched to feature a from

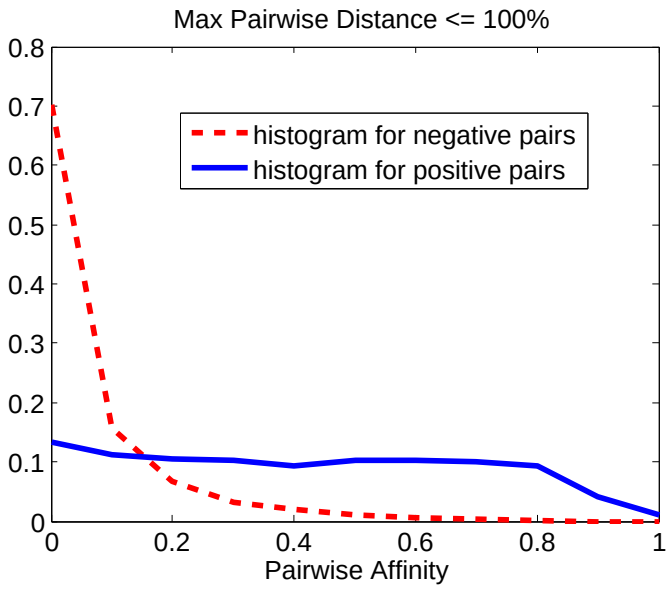

Figure 7. Histograms of pairwise affinities based on color. Notice that the negative pairs have very low affinities, close to zero most of the time. Thresholding at a value of 0.2 or higher would practically remove all negative pairs.

the other image (or object model) and zero otherwise. Usually, one-to-one constraints are imposed on $\mathbf{x}$ such that one feature from one image can be matched to at most one other feature from the other image. In spectral matching $\mathbf{M}$ is a matrix with positive elements containing the pairwise score functions, such that $\mathbf{M}_{ia;jb}$ measures how well the pair of features (i, j) from one image agrees in terms of geometry and appearance (e.g. difference in local appearance descriptors, pairwise distances, angles, etc) with a pair of candidate matches (a, b) from the other. The local appearance terms of candidate correspondences can be stored on the diagonal of $\mathbf{M}$; in practice we noticed that including them in the pairwise scores $\mathbf{M}_{ia;jb}$, and leaving zeros on the diagonal gives better results; We plan to combine the grouping constraints with the object/category specific constraints, by integrating the pairwise affinities based on color into the pairwise scores $M(ia; jb)$. Also, we will use the pairwise scores in order to prune the negative pairs. Instead of considering all pairs of features in an image, we could simply ignore the ones with a color affinity that is less than a certain threshold (e.g. 0.2). Thus most of the negative pairs will be ignored, which will improve both the computation time as well as the quality of matching.

In Figures 1 13we present the potential advantage of using grouping. On the left we show all the pieces of contours (in white). Since no grouping information was used, all contours are considered. On the right, we show the pieces of contours that are likely to be on the same object with the red circle, if we are using the color histogram based grouping. As already discussed, it is clear that grouping could significantly improve the performance of recognition, since most pairs that should not be considered can be automatically discarded. At this point we do not know precisely what is the way of optimally combining pairwise grouping with feature matching, but it should be clear that even a straight forward, natural solution, such as the one we presented in this section, should improve the recognition performance.

6. Conclusions

We presented an efficient algorithm for pairwise grouping of features using color. We showed that the color grouping has the potential of being more global than grouping based on geometry, as it is able to reliably link faraway features that belong

to the same object. The method we presented is intuitive, simple and fast, making a compelling choice for real applications in recognition or matching, for which pairwise grouping is useful. Even though we focused only on grouping pairs of features, this method can be used as the starting point for forming larger groups of features. As future work we need to experiment with this idea on other datasets and measure its impact on recognition and matching.

7. Acknowledgements

This research was supported in part by the National Science Foundation Grant IIS0713406, and by the Intel Graduate Fellowship program.

References

[1] M. Collins, R. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. In *Machine Learning*, 2002. 4

[2] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. In *PAMI*, 2005. 5

[3] A. Etemadi, J.-P. Schmidt, G. Matas, J. Illingworth, , and J. Kittler. Low-level grouping of straight line segments. In *BMVC*, 1991. 1

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. In *Annals of Statistics*, 2000. 4

[5] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990. 1

[6] D. Jacobs. Robust and efficient detection of salient convex groups. In *PAMI*, 1996. 1

[7] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. Technical Report TR-05-16, The Robotics Institute, Carnegie Mellon University, 2005. 1

[8] Y. Li, J. Sun, and C. T. H. Shum. Lazy snapping. 2004. 5

[9] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Boston, 1985. 1

[10] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. In *PAMI*, 2003. 1

[11] D. Marr. *Vision*. Freeman, San Francisco, 1982. 1

[12] R. Mohan and R. Nevatia. Using perceptual organization to extract 3-d structures. In *PAMI*, 1994. 1

[13] G. Roth and M. Levine. Geometric primitive extraction using a genetic algorithm. In *PAMI*, 1994. 1

[14] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. 5

[15] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. In *PAMI*, May 2000. 1

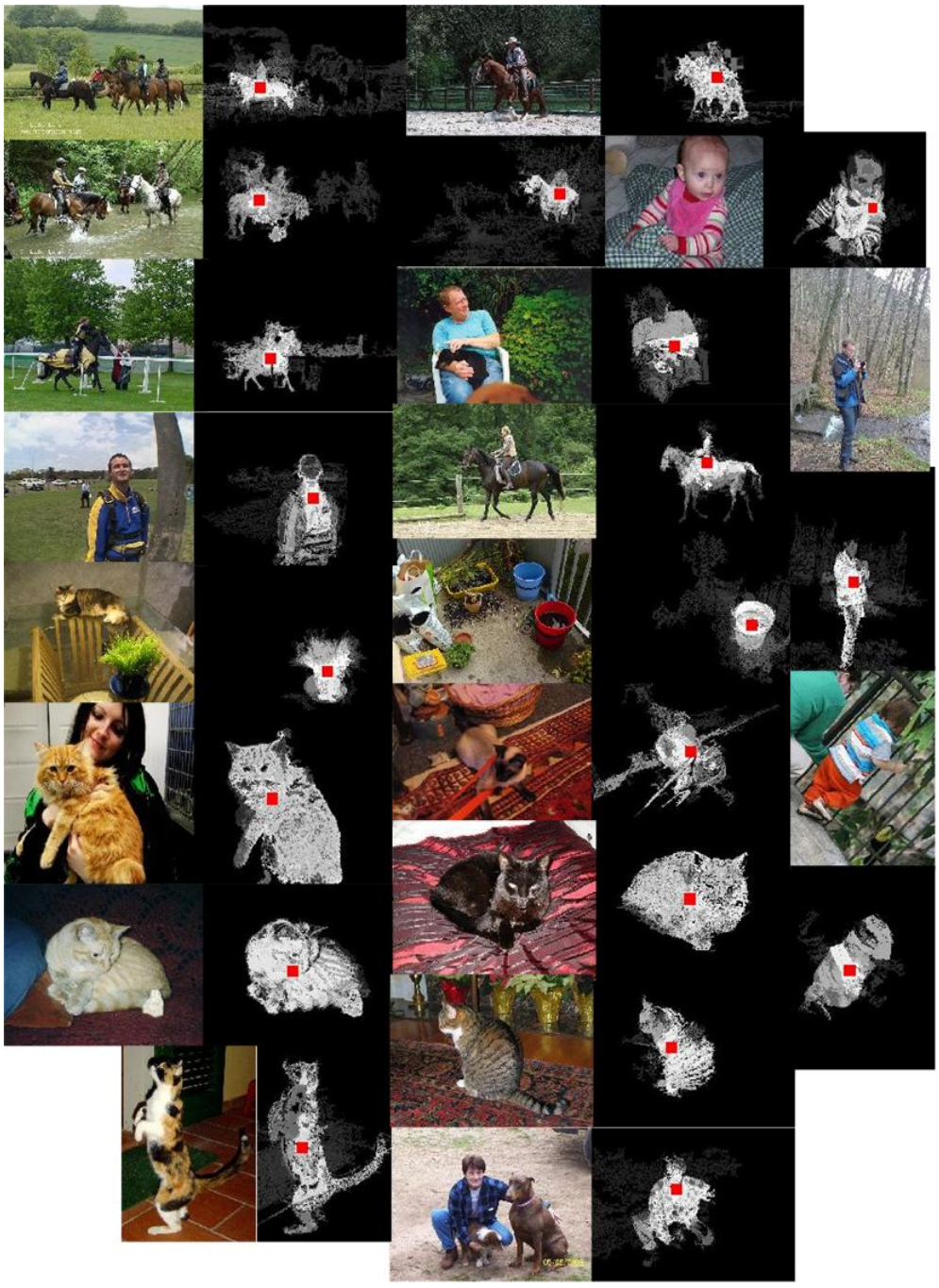

Figure 8. Results on images from the Pascal 2007 challenge database. The red mark indicates the location of the point relative to which the mask is computed. The white pixels are the ones more likely to be on the same object with the red point. The intensity indicates the strength of this likelyhood

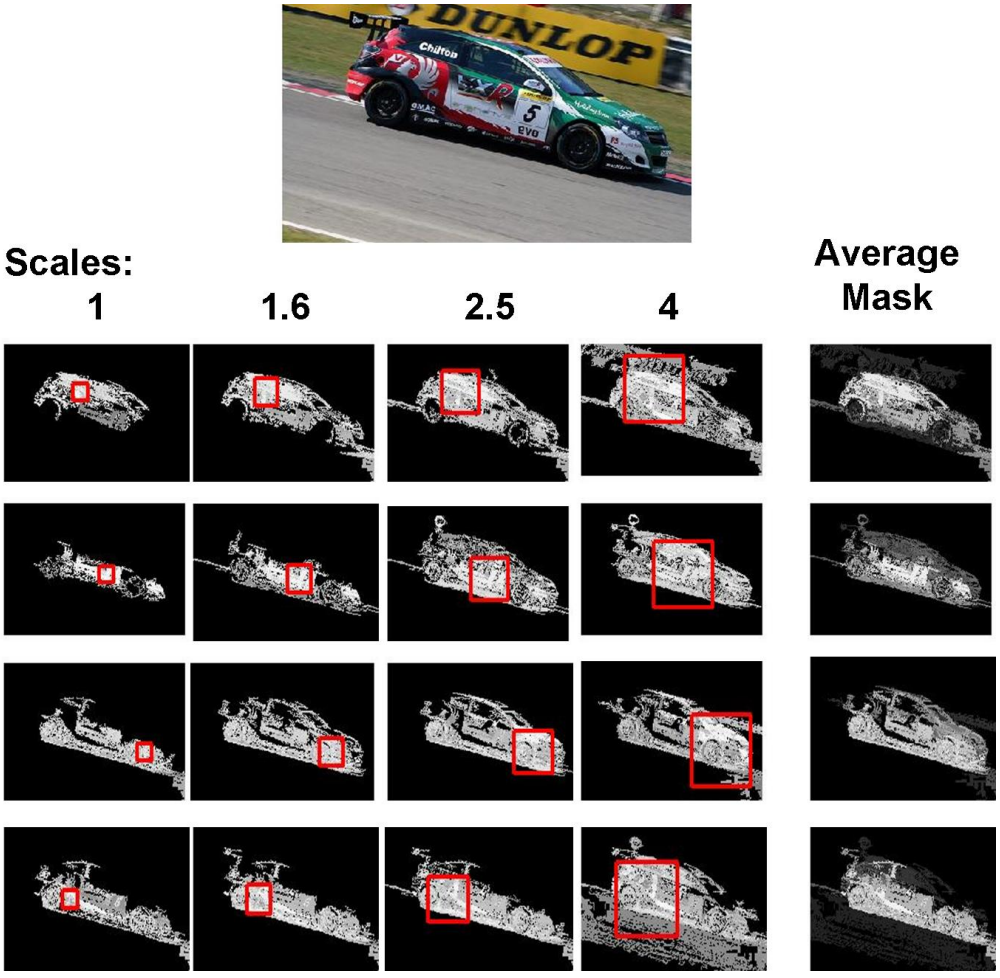

Figure 9. As in the previous Figure, finding reasonable object masks is robust to the location of the bounding box, even for objects with a complex color distribution.

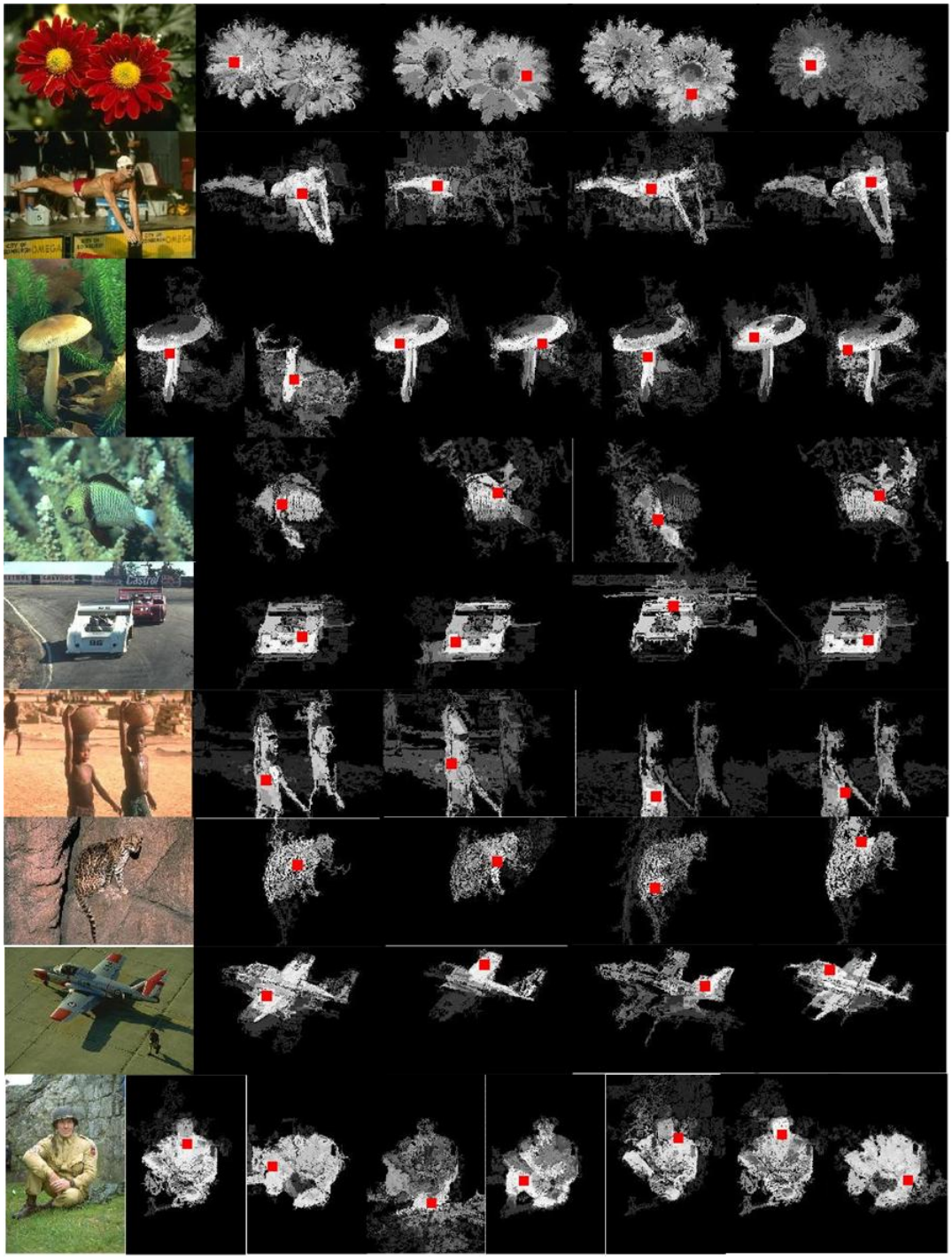

Figure 10. Results obtained from the MSR database. 30 points are chosen randomly on the object for each image and for each point a soft mask is computed. Here we show only a few representative results for each image.

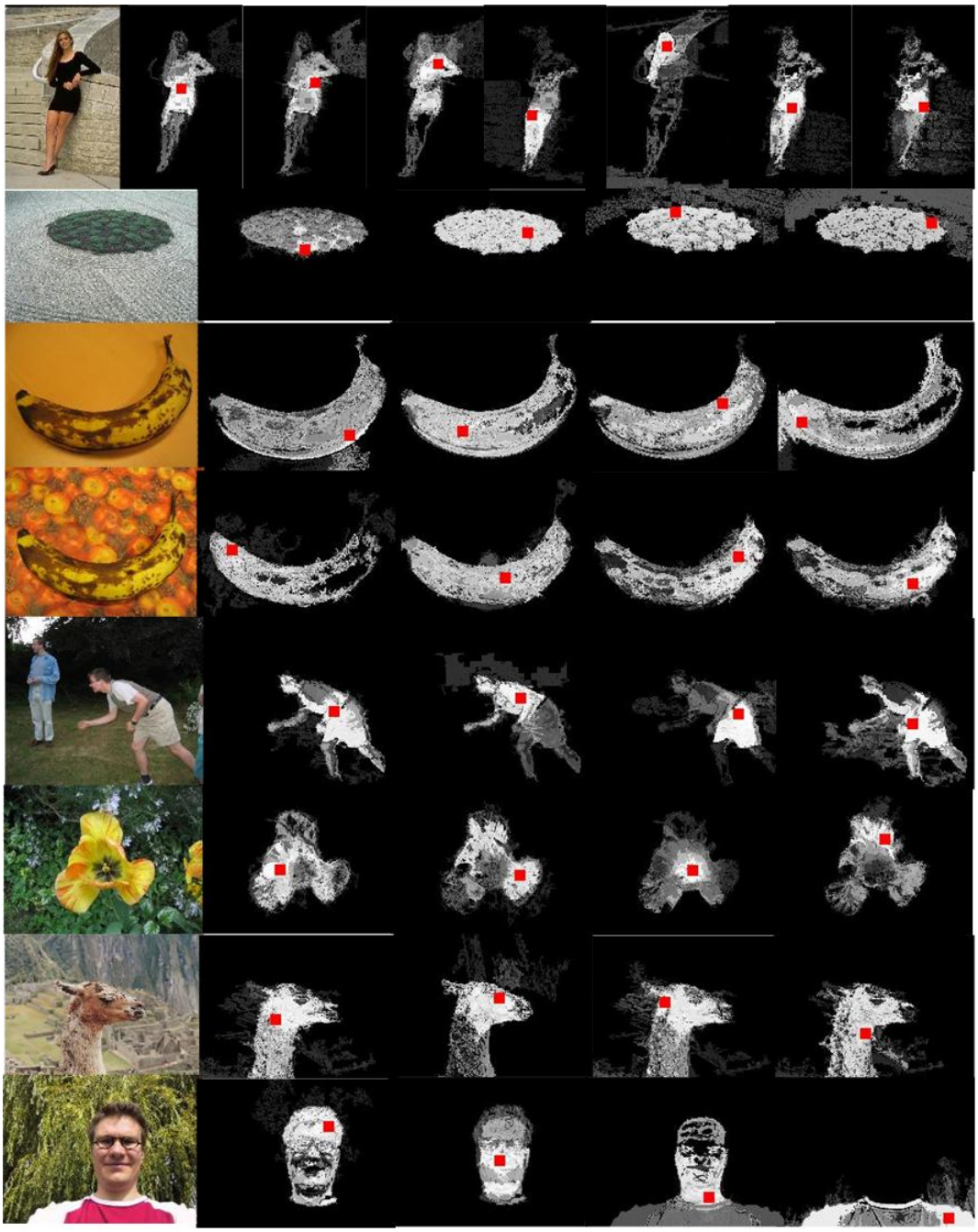

Figure 11. Results obtained from the MSR database. 30 points are chosen randomly on the object for each image and for each point a soft mask is computed. Here we show only a few representative results for each image.

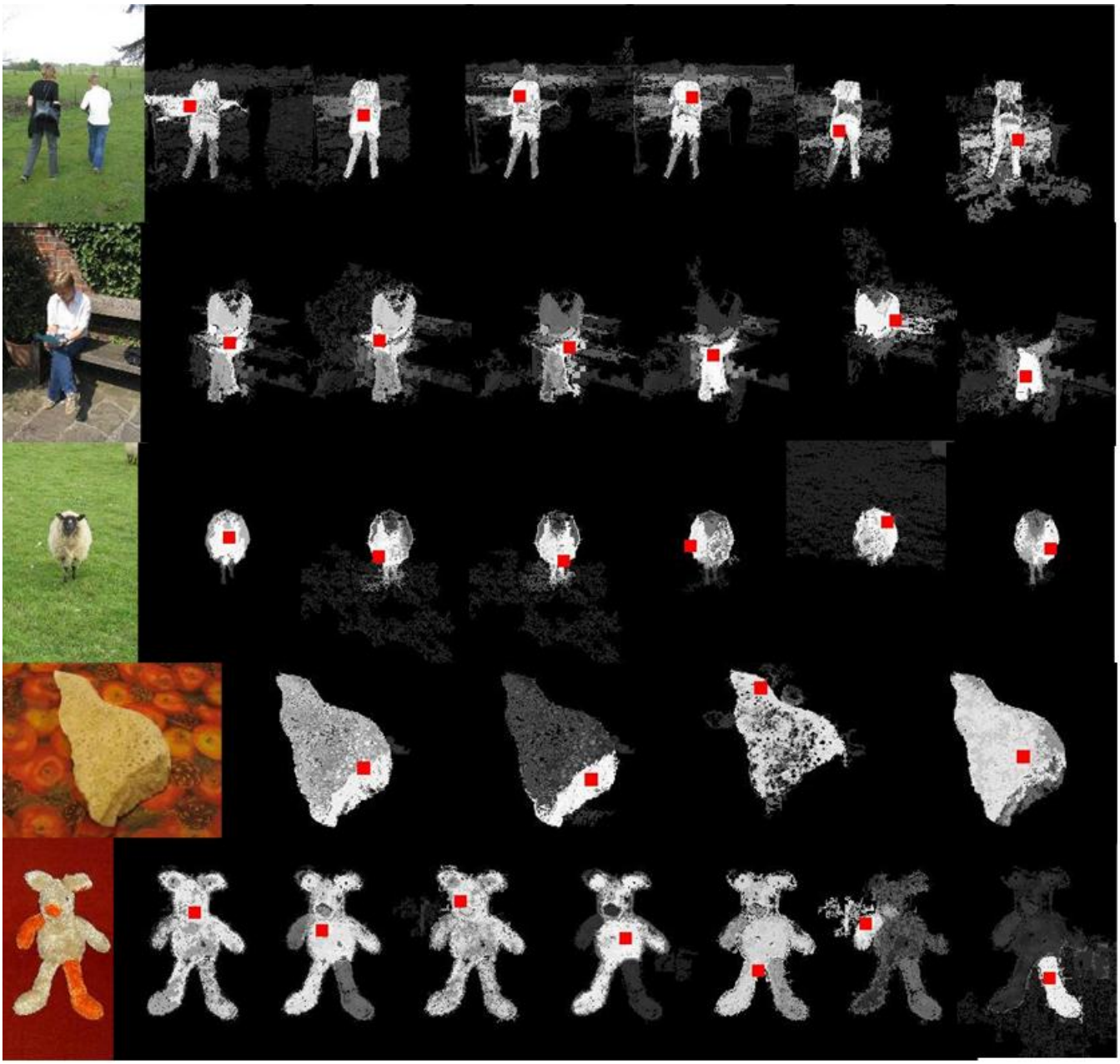

Figure 12. Results obtained from the MSR database. 30 points are chosen randomly on the object for each image and for each point a soft mask is computed. Here we show only a few representative results for each image.

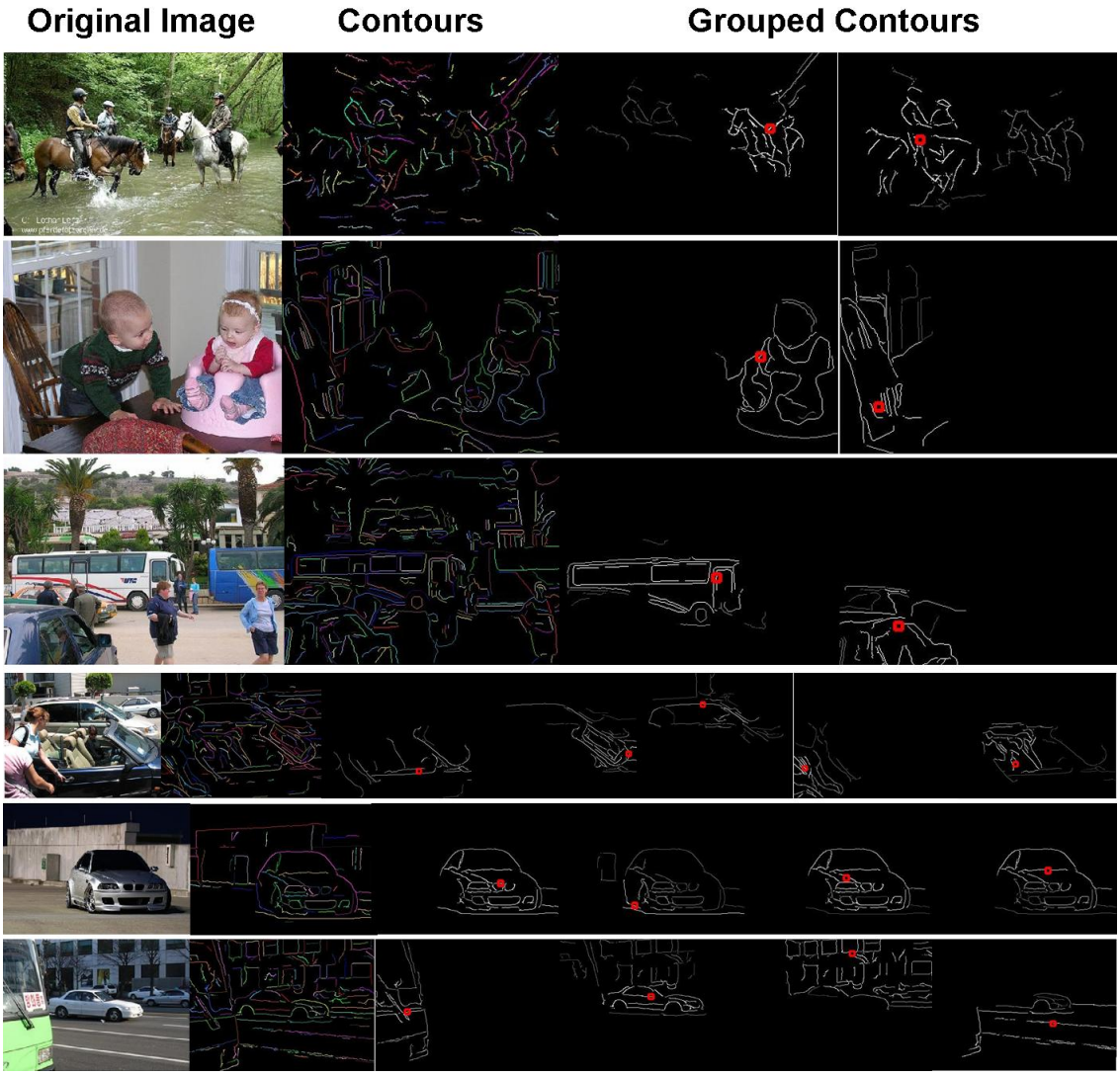

Figure 13. Pairwise grouping relationships (constraints) based on color distribution. The contours shown in white are the ones establishing a pairwise grouping relationship based on color with the contour pointed out by the red circle. Notice some difficult cases from very cluttered scenes. The second column (next to the original image) shows all the contours extracted, while the next images show the contours that form a positive grouping relationship with the contour shown by the red circles.

Figure 14. Examples when color grouping does not work so well. Upper left corner: original image. Upper middle: all the contours extracted. The rest: the results are shown in the same style as in Figure 13. The results are of worse quality than the ones from Figure 13. We can see that parts of the house are weakly connected to contours from the car. This happened mainly because the house and the car have similar colors, and the differences were lost during histogram bin-ing.