Ground From Figure Discrimination

A. Amir
Computer Science Dept.
IBM Almaden Research Center
San Jose, CA 95120
arnon@almaden.ibm.com

M. Lindenbaum
Computer Science Dept.
Technion — I.I.T
Haifa 32000, Israel
mic@cs.technion.ac.il

Abstract

This paper proposes a new, efficient, figure from ground method. At every stage the data features are classified to either "background" or "unknown yet" classes, thus emphasizing the background detection task (and implying the name of the method). The sequential application of such classification stages creates a bootstrap mechanism which improves performance in very cluttered scenes. This method can be applied to many perceptual grouping cues, and an application to smoothness-based classification of edge points is given. A fast implementation using a kd-tree allows to work on large, realistic images.

1 introduction

Visual processes deal with analyzing images and extracting information from them. One reason that makes these processes hard is that only a few subsets of the data features contain the useful information, while all others, which are not relevant for the task, just make it harder, slower and less accurate. Figure ground discrimination processes split the data features into two sets: the *figure* set and the *background* set. The figure set includes all data features that are part of a *structure*, while the background set includes all the unstructured data. The definition of structure (organization) can take many forms, depending on the visual task. An illustration is shown in Figure 1: the data features are edgels (i.e. edge points and their associated gradient directions). The figure set includes all the edgels which lie on smooth curves (a common prescription of structure), and the background set includes the rest, irrelevant edge points, which are associated with clutter, texture and noise.

The human visual system has a powerful figure ground discrimination ability, first demonstrated by the Gestalt psychologists in the early twenties. Gestaltists have searched for the geometric properties and rules that are used by the human vision system for this task. Gordon summarizes some of their work in his book [6]. Using his words, "figure tends to be complete, coherent and in front of ground, which is seen as less distinct, is attended to less readily, and is often seen as floating behind the figure". The human visual system uses completeness, coherence, depth, motion and many other grouping cues to segment the figure out of the received image. Current computer vision systems try to mimic this ability but usually limit themselves to particular domains, like the one demonstrated in Figure 1. This filtering process is useful for many computer vision tasks. Following visual processes, like object recognition, can focus on the figure set and ignore the background (or save it for later use). This way, the number of data features to be processed is reduced, saving computational time and space. It also improves the signal to noise ratio, calling for better, more accurate, results (in contrary to random sub-sampling, that is sometimes used for similar saving reasons, e.g., with the Hough transform [15], but has no effect on the signal to noise ratio). Figure ground discrimination is, therefore, a useful pre-processing filtering stage for many computer vision applications [7, 13, 4].

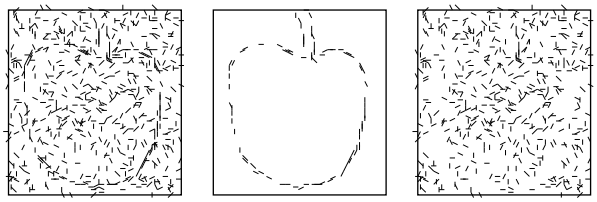


Figure 1: A figure ground discrimination process splits the data set (left) into a figure set (middle) and a background set (right).

Our starting point is the following observation. We argue that the figure ground discrimination problem is asymmetric. A data feature which belong to the figure set must be a part of some structure. In order to figure

this out, we need to find this structure, either explicitly or implicitly. On the other hand, a data feature that belong to the background set does not belong to any structure. We argue that it is easier to assert that a background data feature belongs to no structure at all than to find the structure to which a figure data feature is belong. Therefore, we propose to look only for background data features, and to move them into the background set, until no more background data features can be found. The rest of the data features will form the figure set.

1.1 Related Work

The detection of saliency is closely related to figure ground discrimination. A saliency map assigns a saliency measure to image points, and essentially represent the consistency of the point with some (unspecified) large set of such points. The figure points are expected to have higher saliency, and could be classified this way. Relevant work that fall into this category is of Shashua and Ullman [14], and of Guy and Medioni [9]. In some cases the saliency map is specified in a 3D domain, of pose and orientation (e.g., [16]). It has an advantage in ambiguous points, as junctions and corners, but it requires larger memory and computational effort. It seems that, Hough-like methods, like [9], are expected to suffer from the same high accumulated bias problems that appears with the Hough transform, and thus may not perform well for large images containing a lot of clutter.

Herault and Horaud have used simulated annealing for figure ground discrimination [10]. Their interpretation of the binary relations between data features, which are affected by the presence of a structure, is inspired by the spin model in particle physics. They use an "interacting spin" system, described by the $n \times n$ adjacency matrix (n is the number of data features), and consider a combinatorial optimization problem, where the cost function is analogue to a global energy. They compare between a few different annealing methods, as simulated annealing, and mean field annealing. They get nice results, but the algorithms are computationally expensive.

Gutfinger and Sklansky have developed a classification method which combines supervised and unsupervised learning [8]. They demonstrate its applicability to separate the figure points from synthetic sets of points, which include structured points and scattered noise points. Each point is associated with a 12-dimensional feature vector, which includes various statistics of the rest of the points. The classification takes place in this high-dimensional feature space, using an iterative algorithm. The evaluation of these feature vectors for all points requires high computational efforts.

Our method can be classified as an asymmetric relaxation labeling process. Parent and Zucker [12], and also Deng and Iyengar [4], have used relaxation labeling to detect smooth edges. Each pixel can take one of two labels - figure or background. After an initial labeling assignment is made (for example, by using an edge detector), the labels are updated iteratively, until convergence. The updating rule, applied separately to each pixel, is a function of the neighbors configuration - they can either support its labeling or call for a change. Our algorithm is a limited relaxation labeling technique - it allows only one type of changes: a figure label can be changed to a background label, but not vice versa. This ensures the fast convergence of our algorithm, within a linear number of iterations, in the worst case. Moreover, the total number of label changes in less then n. It requires, however, that the initial labeling algorithm, that is, the edge detector, should preferably be tuned to miss less true edge points with the expense of producing more false alarms (false edges).

Our approach is different from other relaxation labeling techniques in another crucial issue. Usually, the neighboring system is selected according to the image grid - for example, a 3×3 or 5×5 neighborhood. It is assumed that the global structure should be inferred by the propagation of information inside the graph. However, in the absent of an appropriate propagating quality measure, that accumulates the length of the path (e.g., the one proposed in [14]), there is no way to distinguish between a pixel that lies on a long boundary curve and a pixel that lies on a very short edge, only few pixels longer then the neighborhood size. This is apparent in the results reported in [4]. We use a different, non regular neighboring system, which yields, for the same number of neighbors, much larger effective neighborhood.

2 Grouping Cues and Their Graph Representation.

The presence of a local structure in the image can be characterized by local relationships between pairs (or larger subsets) of data features, even without finding and explicitly isolating the structure from its background. The mutual dependencies between data features are measured by perceptual grouping cues.

We apply here the framework developed in our work on generic grouping algorithms and their analysis [1]. The grouping information available to the process is represented by the *grouping likelihood graph*. In this framework, *grouping cues* are considered to be random

binary functions of data features pairs (e.g. pairs of edge points). For a pair of edgels, $e = (u, v)$, the grouping cue $c(e)$ provides a decision which may be either that the two edgels belong to the same group ($c(e) = 1$) or that they do not ($c(e) = 0$). Let $t(e)$ denote the true association between the data features u and v. That is, if both features indeed belong to the same group then $t(e) = 1$, otherwise $t(e) = 0$. Using this notation, the cues' reliability may be quantified by the two probabilities that the cues provide wrong decisions,

$$\begin{aligned} \epsilon_{fa} &= Prob\{c(e) = 1 \mid t(e) = 0\} \\ \epsilon_{miss} &= Prob\{c(e) = 0 \mid t(e) = 1\}. \end{aligned} \quad (1)$$

This model is easily extendible to more general cue functions which provide a non-binary output, such as the likelihood of the feature pair to be in the same group. Cues which operate on three or more data features are also a direct extension (see [2]). In our experimentation the grouping cue is a smoothness criterion: a function which decide whether or not two edgels can belong to the same smooth curve. The reliability of grouping cues, quantified by $\epsilon_{miss}, \epsilon_{fa}$, may be estimated analytically or experimentally. We used a Monte-Carlo process with marked synthetic data to estimate these distributions. Typical values are $\epsilon_{miss} = 0.1$, $\epsilon_{fa} = 0.2$.

We use graphs to represent the set of tested edgel pairs in the image. The *underlying graph*, $G_u = (V, E_u)$, represents the set of edgel pairs to be tested by the cue. Each node represents an edgel, and each arc represents a tested pair. The *measured graph*, $G_m = (V, E_m)$, represents the cue results. It has the same nodes set as G_u, but only the subset of its arcs for which the cue result is $c(e) = 1$. For the included examples, G_u is build up by connecting each edgel to its k nearest edgels (that is, it ignores the non-edge pixels). Intuitively, nodes that correspond to edgels which are lying on a smooth curve are expected to have many neighbors in G_m, while nodes which correspond to clutter, non smooth edges, are expected to have very few neighbors in G_m. Using this framework, the ground from figure discrimination is a node pruning process. This graph representation of perceptual grouping information, denoted as the Grouping Likelihood Graph, is effective also for describing other cues and implementing other visual tasks which rely on perceptual grouping information [2, 3].

3 The Algorithm

3.1 A Fast Graph Construction

The underlying graph is build by connecting every edge point to its k nearest neighbors (typical values are between $k = 10$ to $k = 30$). We use the algorithm of Fridman, Bentley and Finkel, that uses a kd-tree to efficiently find the k nearest neighbors to a query point [5]. The tree building and the graph construction is done in time $O(|V|(k + \log|V|))$. Note that the minimal degree in the graph is k, but some nodes are likely to have higher degree, as all arcs are duplicated to create an undirected graph.

3.2 The Smoothness Grouping Cue

After building the underlying graph, the grouping cue is applied to all the edgels pairs which are connected by an arc in the graph. The grouping cue that was used in the following experimentation is a binary function that receives two edgels and returns 1 if they could lie on a smooth curve and 0 otherwise. There are many different ways to specify a smoothness criterion (e.g., [10, 14, 12, 9]). The one we use is a combination of proximity, cocircularity and low curvature criteria. Let v_1, v_2 denote the two edgels. Let d denote the distance between v_1 and v_2. Let α_1, α_2 denote the angles between the edgel direction (orthogonal to the gray level gradient) and the line connecting the two edgels (see Figure 2), then:

- *Proximity* - the distance d must not exceed some maximal distance. This maximal distance should depend on k, because larger k means that the effective neighborhood is larger, and thus includes a longer part of the possibly existing smooth curve. Here we use the criterion $d < 0.6k$, determined so that a point at the end of a smooth curve will have most of its k nearest neighbors inside a circle of that radius.

- *Co-circularity* - a smooth curve should not have fast curvature changes. Therefore, it should locally match with a circle, and the angles α_1 and α_2 are expected to be similar. Due to the high inaccuracy involved in measuring the gradient direction, we used the criterion $|\alpha_1 - \alpha_2| < \pi/6$.

- *Similar orientation* - Two cocircular edgels might not fit to any smooth curve if their orientation is very different (see Figure 2(b)). Therefore, the difference between the two angles was also limited: $\beta < \pi/6$.

- *Minimal radius* - two very close edgels can meet with all the above constraints and still form a high-curvature curve (see Figure 2(c)). This constraint limits the local curvature: $d/\cos(\frac{\alpha_1+\pi-\alpha_2}{2}) > r_{min}$. The value of r_{min} de-

pends on the subjective definition of what is the maximum curvature of a smooth curve.

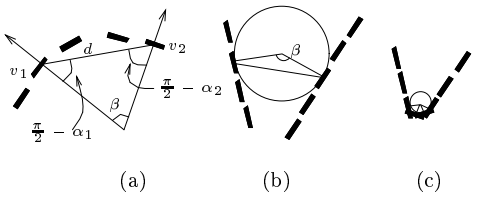


Figure 2: The smoothness grouping cue (see text).

The smoothness criterion returns 1 if and only if all four constraints hold. Note that this criterion is invariant to similarity transformation (assuming that the pixel size follows the same scale change). This meets Lowe's first requirement for good cues [11]: a good grouping cue must a. be viewpoint invariant, and b. have a good discrimination power (the detection condition). Under our framework, this second demand means that the cue should be reliable. That is, it should have small error probabilities, $\epsilon_{miss}, \epsilon_{fa}$. These error probabilities are used with the figure ground algorithm in the next section.

3.3 The Iterative Pruning Process and the Bootstrap Mechanism

Given the underlying graph, G_u, and the measured graph, G_m, the pruning process iteratively removes "background" nodes, classifying the nodes into figure and background sets. Each iteration includes two phases: a *node-pruning* phase and an *arc-filling* phase. Let $d_u(v)$ and $d_m(v)$ denote the degree of a node v in G_u and in G_m, respectively. Let $r(v) = d_m(v)/d_u(v)$ denote the degree ratio of v. This random variable is the sum of $d_u(v)$ independent binary random variables (grouping cues). Therefore it is a Bernoulli trial, with a binomial distribution.

A node-pruning phase takes the underlying graph and removes nodes that are classified as "background". A node is classified as a "background" if $r(v) < \gamma$, where γ is the *degree ratio threshold*, specified from the error probabilities (see below). Otherwise it is classified as "unknown yet". After this stage, it is likely that some (or even many) of the remaining, "unknown yet" nodes in G_u, are left with less than k neighbors. If the process continues with a low number of neighbors, then the next classification will be unreliable. Therefore, the node-pruning phase is followed by an arc-filling phase.

In the arc-filling phase, new arcs are inserted into G_u, such that at the end each node is connected to (at least) k neighbors. The measured graph is similarly updated: marked background nodes are deleted, and the new grouping cues are evaluated. The algorithm keep removing nodes and adding arcs (and cues), until it converges. The algorithm is summarized in Figure 3.

```
Build the underlying graph, G_u.
Evaluate the measured graph, G_m.
repeat
   for v ∈ V  // node-pruning phase
      if d_m(v) ≤ γ d_u(v) then
         delete(v)
   for v ∈ V  // arc-filling phase
      if d_u(v) < k then
         add_arcs(v,G_u)
   update(G_m)
Until no changes occur.
```

Figure 3: The ground from figure algorithm.

For the following analysis, let v_f,v_b denote a (ground truth) figure node and background node, respectively. Before the algorithm starts, the expected degree ratio of a background node is $E(r(v_b)) = \epsilon_{fa}$, ((1)). However, for a figure edgel, v_f, there is an effective miss ratio which is much higher than ϵ_{miss}, as many of its neighbors are background edgels. Let k' denote the number of its figure neighbors, then $E(r(v_f)) = (k' * (1 - \epsilon_{miss}) + (k - k') * \epsilon_{fa})/k$. Hence we cannot expect $r(v_f)$ to be as high as $(1 - \epsilon_{miss})$, and therefore the classification of v_f has a high uncertainty. At this early stage, the algorithm does not make any decision about figure points. As the iterations proceed, $E(r(v_b))$ remains unchanged. Each figure node, however, "loses" background neighbors, and is being connected (by the arc-filling process) to more figure nodes in its larger neighborhood. Therefore, k' increases, and $E(r(v_f))$ increases too. The figure node, v_f, becomes more strongly connected in G_m, decreasing its chance to be deleted later on. This is a bootstrap effect which significantly improves the performance.

The degree ratio threshold, γ, is determined by the error probabilities and feature densities. It should always be in the range $\epsilon_{fa} < \gamma < (k' * (1 - \epsilon_{miss}) + (k - k') * \epsilon_{fa})/k$. As the algorithm starts, γ is set to be slightly above ϵ_{fa}, leading to the deletion of many background elements. As the iterations proceeds, k' increases, the upper bound gets closer to $1 - \epsilon_{miss}$, allowing γ to be increased, and leading to a more accurate classification. In our experimentation we used a

fixed value of γ (between $\gamma = 0.2$ to $\gamma = 0.3$), depending on the noise level for that image domain, which reflects in the initial value of k'. If the noise level is higher, then k' is lower, and γ should be set closer to ϵ_{fa}. We currently study strategies for an adaptive control of γ. Note that γ indirectly controls the tradeoff between the number of *misses* (figure edgels which are deleted from the resulting figure set) to the number of *false alarms* (background edgels that ate falsely added to it). If γ is very small, very few misses are expected, but the resulted figure set might include many false alarms. Similarly, if γ is very high (close to 1), then very few false alarms are expected, on the expense of having more misses.

3.4 Complexity and Run Time.

The algorithm is guaranteed to stop after at most $|V|$ iterations, because at least one node is being deleted each time. In practice, it converges much faster. Moreover, in our implementation we have used an adaptive stopping rule, that stops the process if the number of changes in the last iteration was less than 0.2% of the remaining nodes. Using this stopping rule, the algorithm have stopped after $5 - 15$ iterations in all our experiments. From our experience, if one does proceeds with the processing until no changes occur, it makes an indistinguishable result after a few more iterations.

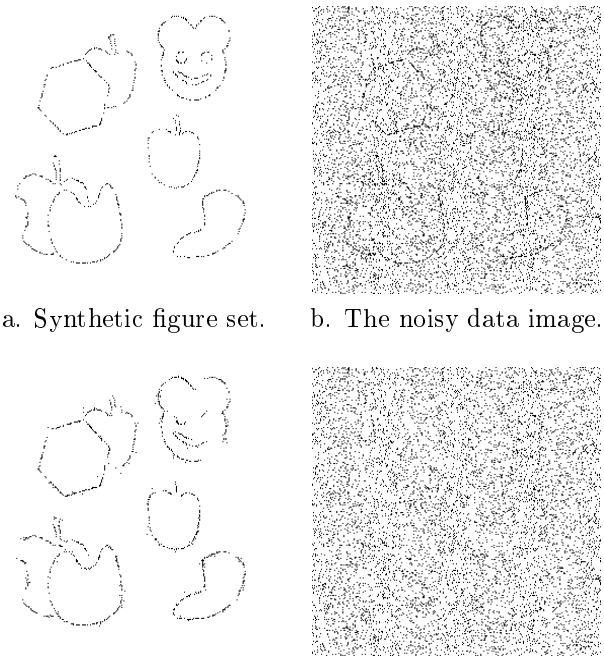
Every iteration takes $O(|E|)$ for the node-pruning, and $O(|V|(\log|V| + k))$ for the arc-filling (an average time), assuming that a new underlying graph is build at each iteration Evaluating the grouping cues takes $O(|E|) = O(k|V|)$. Note that the overall complexity grows linearly with the neighborhood size, k (unlike, for example, [4]).

In the following examples, the processing cpu time (on a Sparc 10) varied between 16sec for the synthetic example ($8,871$ nodes, 6 iterations), up to 109sec for the Lizard images ($18,145$ nodes, 10 iterations). Note that comparing to other methods, operating on data sets of similar size, this is a relatively fast algorithm. Furthermore, we believe that the run time can be reduced by almost an order of magnitude by avoiding the re-creation of the entire kd-tree, the underlying graph and the measured graph at each iteration.

4 Experimentation

The algorithm was tested on synthetic images, on computerised tomography (CT) images, and on natural images. Here we provide three detailed examples, reflecting the wide spectrum of tested images. A locally connected underlying graph ($k = 15$) was used for all examples. Edges were extracted by the Khoros standard edge detector, and gradient direction was evaluated by convolving the image with G_x and G_y — the partial derivatives of a Gaussian ($\sigma = 1[pixel]$).

The first example is a synthetic dot image, shown in Figure 4. This example is similar to those used in past work (e.g., [10, 14, 8]), but it is much larger (including 8,871 edgels), and is very noisy (the figure set is only 10% of the edgels). The edgels data image (Figure 4(b)) includes 50% of the edge detector result on the synthetic image (randomly selected, Figure 4(a)), and nine times more additional clutter pixels with random orientations. The objects indeed seem to be lost in the noise, although they are more visible with the gradient direction (like the zoomed region shown in Figure 1). The resulted figure and background sets are shown in Figure 4(c), and Figure 4(d), respectively. The low number of missed points, and the low number of false alarms, is an evidence to the high quality of the result.

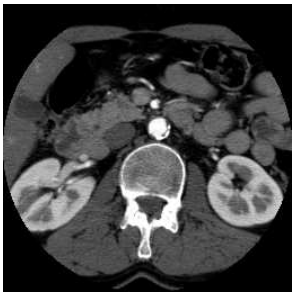


a. Synthetic figure set. b. The noisy data image.

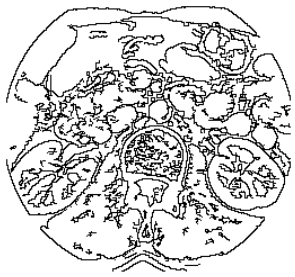
c. Resulted figure set. d. Resulted background.

Figure 4: Results for a 284×264 synthetic data, containing 8,871 edge points (graph nodes). The algorithm converges after 6 iterations in 16 sec.

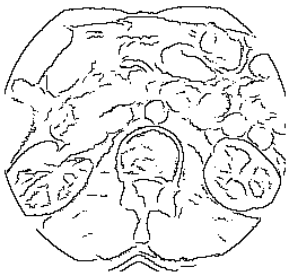
The next example is a CT image (Figure 5). This image contains sharp and smooth edges, associated with different gradient magnitudes. It is therefore unlikely that a single edge operator will find all salient edges and ignore the structurally unimportant ones. The algorithm removes much of the edges which do not lie on smooth curves, and leaves, in the figure set,



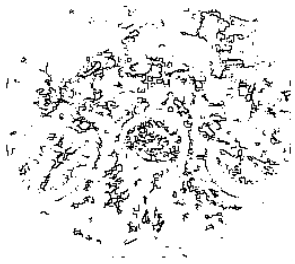
a. A CT image.



b. Edgels image.



c. Resulted figure set.

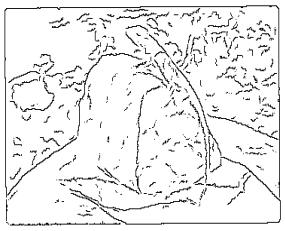


d. Resulted background.

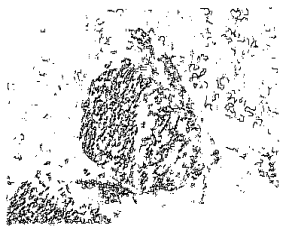
Figure 5: Results on a 512×512 CT image.

a. The Lizard image.

b. Edgels image.



c. Resulted figure set.



d. Resulted background.

Figure 6: The Lizard, 383×255 image, contains a rich texture, which cause the detection of a lot of edgels (18,145 nodes). The algorithm successfully removes most of the edgels corresponding with the texture, while keeping most of the edgels corresponding with the smooth boundaries (10 iterations, 109 sec).

most of the boundaries of the main regions in the original image. Therefore, higher tasks, such as search for a particular organ, is easier and as reliable when one starts from the figure set and not from the original edge map.

The Lizard image (Figure 6) is typical to natural scenes. It contains a rich texture and high contrast, which cause the edge detector to detect many edgels. The algorithm successfully removes much of the texture-induced edgels (Figure 6(d)), while keeping most of the edgels corresponding to real object boundaries (Figure 6(c)).

Other perceptual grouping processes may also benefit from a ground from figure pre-processing. The grouping results, shown in Figure 7, were obtained by the algorithm proposed in [1], applied on the resulted figure set. Note that the complexity of grouping processes is usually much higher than of this figure ground discrimination algorithm. Therefore, the grouping of the figure set is much easier and faster.

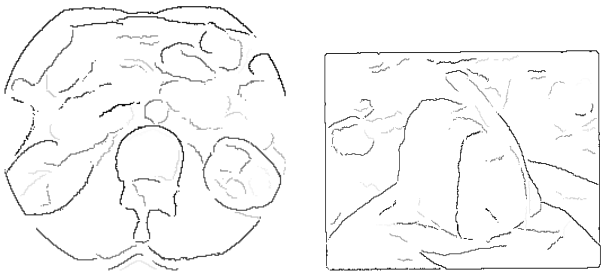
As usual in this type of tasks, neither the figure from ground nor the grouping are perfect. Note, for example, the wrong grouping of the lizard's back with the right-bottom corner of the rock, which meet quite smoothly at the occlusion point. We did not try to make the figure sets prettier by closing the gaps using some edge completion technique. The results are the exact two disjoint subsets of the original edge image. We believe that some of the ground from figure misses may be corrected by edge completion methods.

5 Discussion

A new figure from background discrimination algorithm was presented and its performance was demonstrated experimentally over a set of images, showing good results. The tested images are taken from three different domains: a synthetic edge image, a CT image, and a natural out-door image.

The algorithm is based on the grouping likelihood graph - a data structure that represents the atoms of perceptual evidence: the results of the grouping cues. The proposed algorithm may be considered as a simplified and asymmetric relaxation labeling technique. It is asymmetric because it iteratively prunes background data features, until it stops and hypothesize that the remaining feature is the figure set.

The algorithm implementation and its performance was demonstrated in the domain of an edge image. However, we argue that it should work also in other domains. Another domain could be, for example, a motion based ground from figure discrimination. Pixels, or small regions, that have unique motion parameters, are not connected to any other region, and



a. Grouping of Fig.5(c). b. Grouping of Fig.6(c).

Figure 7: The figure set is fed into a grouping process, which separates it into smooth curve segments. All segments having 20 points or more are displayed. The gray level indicates the group number, sorted by group size (darker groups are larger). Note that the left and right kidneys, the bone, and few other main parts, correspond to very large single groups.

therefor will be classified as background. On the other hand, small regions which have the same motion parameters will support each other, and will remain in the figure set (see [2] for a multi-domain (generic) grouping algorithm which uses the same graph representation but changes the cue depending on the domains.)

Finally, we believe that various perceptual grouping processes should be combined in order to make a complete system. Figure from ground would be the first filtering stage, followed by edge completion and perceptual grouping. All these stages can use the same data structure and the same grouping cues as their source of information, but may also use different cues in different stages. This is not a new idea, but we believe that the framework proposed in [2] and used here is a convenient tool to combine all these tasks.

Acknowledgments

Thanks to R. Bar-Haim, E. Dvi-Aharon and Y. Lerner for their great help with the implementation. Thanks to Elscint Ltd for providing us with the CT data.

References

[1] A. Amir and M. Lindenbaum. Quantitative analysis of grouping processes. In *ECCV-96, Cambridge* (1996), vol. 1, pp. 371–384.

[2] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence 20*, 2 (Feb. 1998), 168–185.

[3] A. Amir and M. Lindenbaum. Grouping-based nonadditive verification. *IEEE Trans. Pattern Analysis and Machine Intelligence 20*, 2 (Feb. 1998), 186–192.

[4] W. Deng and S. S. Iyenger. A new probabilistic relaxation scheme and its application to edge detection. *PAMI 18*, 4 (Apr. 1996), 432–437.

[5] J. H. Fridman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. on Mathematical Software 3*, 3 (Sept. 1977), 209–226.

[6] I. E. Gordon. *Theories of Visual Perception*, first ed. John Wiley and sons, 1989.

[7] W. E. L. Grimson. *Object Recognition By Computer: The Role of Geometric Constraints*, first ed. The MIT press, 1990.

[8] D. Gutfinger and J. Sklansky. Robust classifiers by mixed adaptation. *PAMI 13*, 6 (June 1991), 552–566.

[9] G. Guy and G. Medioni. Perceptual grouping using global saliency-enhancing operators. In *ICPR-92* (1992), vol. I, pp. 99–103.

[10] L. Herault and R. Horaud. Figure-ground discrimination: a combinatorial optimization approach. *IEEE Trans. Pattern Analysis and Machine Intelligence 15*, 9 (Sep 1993), 899–914.

[11] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Pub., 1985.

[12] P. Parent and S. W. Zucker. Trace interface, curvature consistency, and curve detection. *IEEE Trans. Pattern Analysis and Machine Intelligence 11*, 8 (August 1989), 823–839.

[13] S. Sarkar and K. L. Boyer. Perceptual organization in computer vision: a review and proposal for a classifactory structure. *IEEE Transactions on System, Man and Cybernetics 23*, 2 (March/April 1993), 382–399.

[14] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using locally connected network. In *ICCV-88* (1988), pp. 321–327.

[15] D. Shaked, O. Yaron, and N. Kiryati. Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Computer Vision and Image Understanding 63*, 3 (May 1996), 512–526.

[16] L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. In *ICCV-95, MIT* (1995), pp. 408–415.