Learning Static Object Segmentation from Motion Segmentation

Michael G. Ross and **Leslie Pack Kaelbling**
MIT Computer Science and Artificial Intelligence Laboratory
mgross@csail.mit.edu, lpk@csail.mit.edu

Abstract

Dividing an image into its constituent objects can be a useful first step in many visual processing tasks, such as object classification or determining the arrangement of obstacles in an environment. Motion segmentation is a rich source of training data for learning to segment objects by their static image properties. Background subtraction can distinguish between moving objects and their surroundings, and the techniques of statistical machine learning can capture information about objects' shape, size, color, brightness, and texture properties. Presented with a new, static image, the trained model can infer the proper segmentation of the objects present in a scene. The algorithm presented in this work uses the techniques of Markov random field modeling and belief propagation inference, outperforms a standard segmentation algorithm on an object segmentation task, and outperforms a learned boundary detector at determining object boundaries on the test data.

Introduction

Image segmentation is the discovery of salient regions in static images and has a history reaching back to the Gestalt psychologists (Palmer 1999). There are many computer vision approaches to this problem, but they are difficult to compare because there is no readily accessible ground truth. In recent years, this situation has improved as researchers such as Martin et al. (2004) and Konishi et al. (2003) have used human-segmented data to train and test boundary detection algorithms.

This paper further grounds the image segmentation problem by replacing it with the better-defined goal of object segmentation and by using an automatically segmented database as the training and testing set. Object segmentation is the task of grouping pixels in a static image into regions that correspond to the objects in the underlying scene (similar to figure-ground segmentation). In this formulation, objects are sets of elements that move coherently in the world. If objects can be distinguished by their motion, a motion segmentation algorithm, which divides moving objects from their static surroundings, can provide a partially labeled database of object segmentations. Such a database can be automatically gathered by a robot or other vision-processing agent by simply observing object motion in a domain. The static image and shape properties associated with the boundaries and interiors of moving objects can be used to train an object segmentation model. Then the model can determine the segmentation of individual static images without using motion information. Its performance can be measured by comparing the segmentation it produces on individual video frames to the motion segmentation.

This paper describes the Segmentation According to Natural Examples (SANE) object segmentation algorithm, which is trained and tested in this framework. Videos of moving objects are motion-segmented using background subtraction. This automatically labeled data provides a large, cheap training set that is used to learn the shape and image statistics of object segmentation. Then, presented with a new, static image, the learned model constructs a Markov random field (MRF) model which can infer the underlying object segmentation using the belief propagation algorithm (Pearl 1988). This algorithm outperforms a standard implementation of the general-purpose normalized cuts segmentation algorithm (Shi & Malik 1997) on the object segmentation task and outperforms the trained Martin boundary detectors (Martin, Fowlkes, & Malik 2004) on detecting the object boundaries.

Related work

Recent work in learning edge detection include Konishi et al. (2003) and Martin et al. (2004) who improved on standard edge detectors by learning detectors from human-labeled databases. These methods rely on manually segmented training data, which requires a time-consuming process that produces results dependent on the subjective judgment of the human labelers. Borenstein and Ullman have developed a model of class-specific segmentation that learns to perform figure-ground segmentations for a particular class of objects by building a database of fragments that can be assembled like puzzle pieces (2002).

The models employed in SANE are similar to, and were initially inspired by, the work by Freeman et al. on super-resolution and other problems (2000). The first use of MRFs in image segmentation was Geman and Geman's image restoration algorithm (1984).

Spelke et al. have discovered that human infants lack the

ability to segment objects according to their static monocular image properties and instead rely on motion and depth information (1994). This suggests that these built-in segmentation abilities may form the basis for learning static, single-image segmentation.

Model and algorithm

The SANE object segmentation model divides an image into a lattice of 5 pixel by 5 pixel, non-overlapping patches and assigns a variable to represent the segmentation at each patch. For each image patch i there is a hidden segmentation variable $S_i = (E_i, P_i)$ and visible image features I_i. I_i consists of real-valued image features (e.g., brightness, color, gradient) from the underlying patch. E_i specifies the shape of the object boundary present at location i and P_i specifies the boundary's parity. In our representation, each edge patch is parameterized by three variables: the locations of its entry and exit from the patch, and a possible inflection point inside the patch. On a 5 by 5 patch this produces approximately 3000 possible edge values. The no-boundary case, the "empty edge," is added as a special case. Assuming that the objects in a scene do not overlap in the image, dividing objects from their surroundings only requires two segments, so the parity of an edge is a binary value that determines which segment is on which side of the boundary (see Figure 1). A horizontal edge, for example, can have region 0 above it and region 1 below it, or vice versa.

The variables are linked into an MRF, an undirected graphical probabilistic model. Each node $N_{i,j} = (S_{i,j}, I_{i,j})$ is connected to its first-order lattice neighbors: $N_{i+1,j}$, $N_{i-1,j}$, $N_{i,j+1}$, and $N_{i,j-1}$ (see Figure 1). If N is the set of all nodes in the model, the Markov property ensures that $P(N_{i,j}|N \setminus N_{i,j}) = P(N_{i,j}|N_{i+1,j}, N_{i-1,j}, N_{i,j+1}, N_{i,j-1})$. Traditional MRF models keep hidden and visible variables as separate graph nodes. In this model, they are combined because in the segmentation problem the image properties at each node can exercise a strong influence on the joint probability of neighboring edge variables. Consider the situation in Figure 2. Each patch has local evidence indicating a horizontal edge, and a reasonable boundary model might assign high probability to two neighboring horizontal edges. But the assignment's probability might be negatively influenced by the fact

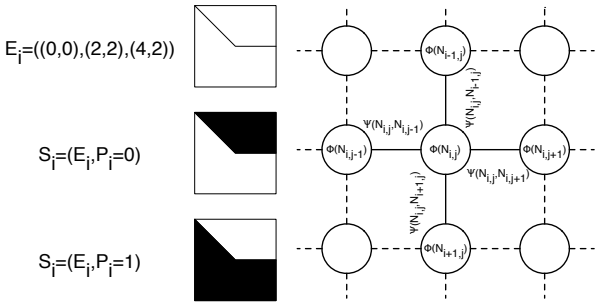

Figure 1: Left: An example edge assignment and the two possible segmentation assignments it can create depending on its parity. Right: A section of a Markov random field.

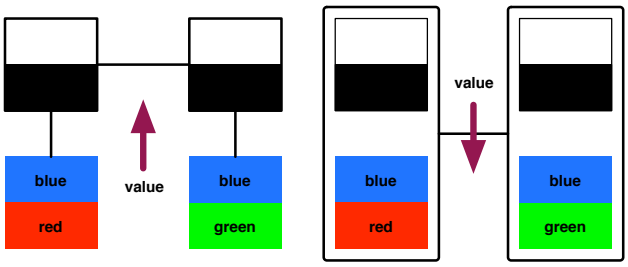

Figure 2: Left: A traditional MRF might give a high value to this pairing of horizontal edges, since each image patch gives strong local evidence and the edges continue each other. Right: An MRF that combines the edge and image variables into the same nodes can give a low value to this combination because it is creating a segment with differently colored neighboring pixels.

that the horizontal edge assignments would put green and red pixels in the same segment. The presence of the image information in each node allows the model to make precisely this type of decision.

In order to specify a segmentation MRF on a particular image, each node needs a positive $\Phi_i(N_i)$ function that provides local evidence for its hidden value S_i and every neighboring pair of nodes needs a positive $\Psi_{i,j}(N_i, N_j)$ function to represent the relationship between neighboring values (see Figure 1) (Besag 1974). In this context, all the Φ_i functions are identical due to translational invariance and there are only two types of Ψ functions, one for vertical neighbors and the other for horizontal neighbors (these functions are related by a $90°$ rotation of their arguments). The relationship between neighbors i and j will be denoted $r(i, j)$.

These functions are learned from the training data. Unfortunately, computing the marginal distribution on any node or pair of nodes requires normalizing the graph by its partition function, a computationally intractable operation on non-trivial models. However, there are useful approximations. Wainwright et al. (2003) note that setting $\Phi(N_i) = P(N_i)$ and $\Psi_{r(i,j)}(N_i, N_j) = P_{r(i,j)}(N_i, N_j)/(P(N_i)P(N_j))$ is an approximate maximum-likelihood estimate for the compatibility functions. This formulation corresponds to the estimates utilized in some of Freeman et al.'s work (2000).

In order to improve the segmentation results, the SANE MRFs modify the Wainwright neighborhood compatibilities to enforce boundary continuity and segmentation consistency. Boundary continuity requires that an edge assignment whose endpoint is adjacent to a neighboring patch be continued by the edge selected for the neighboring patch. Segmentation compatibility requires that segmentation assignments implied by neighboring edge parities are consistent, except where the pixels are separated by an edge. The function cont(S_i, S_j) equals 0 if S_i and S_j violate either of these properties, and 1 otherwise.

Therefore, the compatibilities are

$$\Phi(N_i) = P(E_i, I_i)$$

and

$$\Psi_{r(i,j)}(N_i, N_j) = \text{cont}(S_i, S_j)\frac{P_{r(i,j)}(E_i, I_i, E_j, I_j)}{P(E_i, I_i)P(E_j, I_j)} + (1 - \text{cont}(S_i, S_j))\epsilon.$$

The relationships between pixels and boundaries are learned from the data. The parity information is only needed to keep outputs in the space of valid segmentations that obey the boundaries inferred in the image. In addition to enforcing continuity and spreading boundary information, the parity bits also allow any assignment to be interpreted as a valid segmentation—the parity of a patch specifies which of its pixels are in region 0 and which are in region 1. The ϵ compatibility is used because zero compatibility values are disallowed by the Hammersley-Clifford theorem (Besag 1974).

Because the parities do not have semantic meaning apart from the requirements of continuity, they can produce local assignment ties. Without some knowledge of the parity at neighboring nodes, there is no information to favor choosing one parity over another. Belief propagation cannot break ties without assistance, so the situation is avoided by fixing the upper-left node to only have parity 0. The need to match parities with that assignment allows belief propagation to set the parities across the entire image correctly.

Inference on a new image is performed by constructing an MRF, as described above, using the density functions estimated from the training data, and applying the belief propagation algorithm. Belief propagation only provides exact inference on loopless graphical models, but it produces useful approximate inference on loopy models such as ours (Weiss 1997).

Belief propagation can fail to converge on a loopy graph and can give neighboring nodes incompatible assignments. For these reasons, the belief propagation estimate is post-processed with the iterative conditional modes (ICM) algorithm (Besag 1986). A restricted form of ICM is allowed to flip the parities of the marginal MAP estimate of the nodes, but not change the edge assignments. The parity-flips can repair mismatches between neighboring segmentation labels caused by non-convergence of belief propagation.

To make inference more tractable, the algorithm initializes the set of possible assignments at each node to the 20 edge assignments that are most likely given the local image data. Then, edges are added (in order of decreasing local probability) as is necessary to continue the possible assignments of neighboring patches. This ensures that complete, closed contours are always possible. Finally, the number of possible assignments at each patch (except the upper-left patch, as discussed above) is doubled by pairing each possible edge with both potential parities to create the set of possible S_i values at each patch.

In some cases, better results can be achieved with a multiresolution model. Two MRFs are constructed, one on a full-size image and one on that image at half-resolution. Note that this requires training on half-scale training data in order to construct the half-scale model. Then the models are linked such that each node in the half-scale model is the neighbor of four full-scale nodes. The interlevel node compatibilities are set so they are 1 if the segment labels of the full-scale node matches those of the relevant quadrant of its half-scale parent and ϵ otherwise. This encourages the full-scale model to find an assignment that is compatible with larger-scale shape and image information.

Training

Motion segmentation, provided by background subtraction, gives a partially and imperfectly labeled segmentation database. Background subtraction can only label moving objects, so, in training, only the moving objects and their immediate surroundings are used to learn all the necessary probability distributions. If multiple moving objects are present, the image is not subdivided into subimages, but a region that contains all the objects is used.

Background subtraction and cropping provides a set of images paired with binary masks indicating which pixels belong to the foreground and which belong to the background. Scanning the rows and columns of the binary image and marking the transitions between foreground and background produces an edge image. Every image and edge image is divided into 5 pixel x 5 pixel tiles (in order to maximize training data, all valid offsets of the tiling are also used, as are local rotations and reflections of the training data). Each edge patch is matched to the most similar parameterized edge. Factoring $P(E_i, I_i)$ into $P(I_i|E_i)P(E_i)$ and $P(E_i, I_i, E_j, I_j)$ into $P(I_i, I_j|E_i, E_j)P(E_i, E_j)$ produces both continuous and discrete probability distributions. The discrete distributions are estimated by counting and the continuous distributions are estimated by Gaussian kernel density estimates.

The value of I_i could be a full color image of the relevant patch, but estimating distributions of such high dimensionality would require an unreasonable amount of training data and using them would be too computationally expensive. Instead, the patches are represented by relatively few features: the average brightness of the pixels in in the top, left, right, and bottom patch areas and, when color is used, the average red, green, and blue values of all the patch pixels.

The required features of the associated patches are extracted and used to fit kernel density estimates for the $P(I_i, I_j|E_i, E_j)$ distributions. The features are preprocessed to have zero mean and unit variance. The training points available for each distribution are split into kernel points and test points. The kernel variances are fit by searching for a maximum of the likelihood of the test points.

The single node $P(I_i|E_i)$ distributions are not estimated independently since they are equal to

$$\sum_{E_j} \int_{I_j} P(I_i, I_j|E_i, E_j)P(E_j|E_i).$$

Independent estimates would provide substantial computational savings in inference, but in our experience the Wainwright compatibility formulas are very sensitive to any mismatch between $P(I_i|E_i)$ and the exact marginalization given above, so the explicit marginalization is necessary.

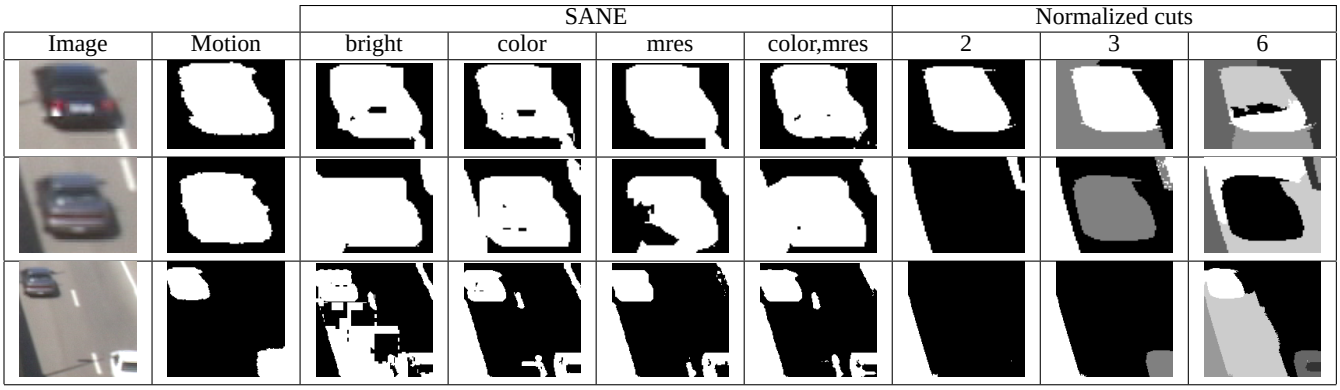

Figure 3: The images from the car sequence illustrate that SANE can find a reasonable object segmentation in these examples, but there is no number of regions that allows normalized cuts to correctly segment the objects in all three images.

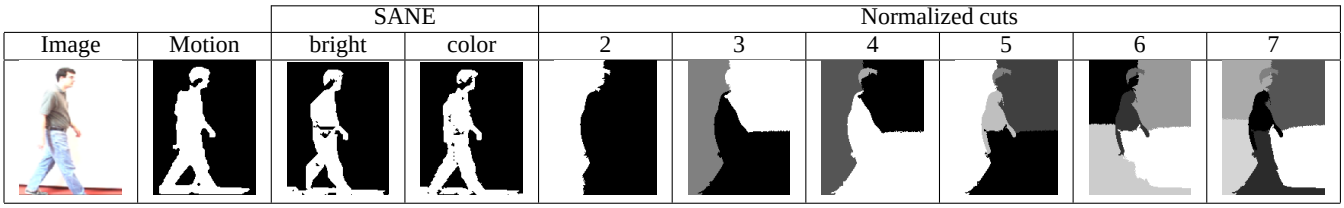

Figure 4: SANE separates the person from the whiteboard, but normalized cuts, lacking an object shape model and the ability to distinguish object boundaries from other differences, fails.

Results

SANE was tested on two data sets, a video of cars traveling on a highway and a video of a person walking back and forth in front of a whiteboard several times. In the traffic data set, the original video contained images of two highways: the left highway was used as the training data and the right highway as the testing data. In the walking-person data set, the first half of the frames formed the training set and the second half the test set. For the traffic results, 286 frames, uniformly sampled from the set of frames that contained moving cars, were used to train the model, and a set of 100 frames uniformly selected from the test set were used in testing. For the walking-person results, the first 200 frames that contained the moving person were used to train the model, and 40 frames from the remainder were used in testing. The car background subtractions were produced by the Stauffer and Grimson algorithm (1999) and the walking person background subtractions were produced by the Migdal and Grimson (2005) extension of that algorithm.

For each test image, each MRF node was given an initial set of the 19 most locally likely edge values and the empty edge. Belief propagation was run for 200 iterations—the MAP assignments typically converge by that point. Convergence is not guaranteed and waiting for full convergence is impractical for models of this complexity. The ϵ value used for incompatible neighboring assignments was 10^{-10}.

Four versions of the SANE algorithm were tested on the traffic data set. There were two choices of image features: the average brightness at the top, left, right and bottom patch areas and these values added to the average red, green, and blue values over the whole patch. We also used both single and multiresolution models, making four possible combinations of features and model type. Only single resolution brightness and color models were trained on the walking-person data set. Motion is not used in segmentation.

The "ground-truth" segmentations that the outputs are compared to are produced by the same background subtraction methods that are used to label the training data. Just as in the training process, the test frames are cropped so they only contain the moving objects and their immediate surroundings because there is no knowledge about the correct segmentation in static parts of the image.

SANE does not distinguish which of the two segment labels belongs to the objects and which to the background. When comparing the results to the motion segmentations, the inferred segment that has the highest overlap with the moving objects is considered the object segment label.

Table 1: SANE and normalized cuts on traffic data.

Algorithm	**F-measure**	**Recall**	**Precision**
SANE (color, mres)	0.803	0.816	0.790
SANE (color)	0.767	0.802	0.735
SANE (bright)	0.717	0.805	0.647
Ncuts (5)	0.677	0.629	0.734
Ncuts (6)	0.676	0.591	0.789
Ncuts (4)	0.656	0.641	0.672
Ncuts (7)	0.651	0.561	0.775
SANE (bright, mres)	0.646	0.801	0.541
Ncuts (8)	0.633	0.534	0.779
Ncuts (3)	0.609	0.705	0.536
Ncuts (9)	0.600	0.496	0.760
Ncuts (2)	0.580	0.793	0.457
Ncuts (10)	0.578	0.469	0.754

We measured precision (p) and recall (r) on the segmentation output, using the motion labels as ground truth. Preci-

sion is the percentage of inferred object pixels that were true object pixels and recall is the percentage true object pixels that were correctly labeled. Precision and recall are used to calculate the F-measure ($2pr/(p+r)$), which can be used as a measure of overall quality.

The motion data only labels moving objects in the frame, so in some image sequences it may be impossible or undesirable to achieve 100% precision and recall. If the traffic images included cars parked alongside the road, a good object segmentation algorithm might correctly discover their boundaries, but there would be no corresponding motion label. The test data used here largely eliminates that concern by only including image regions that contain moving objects, but there are some potential objects, such as the lane lines on the highway, that are never labeled as moving. Furthermore, the motion labels themselves have some noise.

The results of SANE using brightness and color features on the traffic and walker data can be found in Tables 1 and 2, respectively. Multiresolution SANE with both brightness and color image features were also tested on the traffic data. For comparison, the tables also present results produced by the well-known normalized cuts segmentation algorithm (Shi & Malik 1997) on the same data. The normalized cut implementation (Cour, Yu, & Shi 2004) only used brightness-based features, so it should not be directly compared to the SANE output that utilized color.

Table 2: SANE and normalized cuts on walker data.

Algorithm	F-measure	Recall	Precision
SANE (color)	0.742	0.811	0.683
SANE (bright)	0.613	0.709	0.540
Ncuts (2)	0.529	0.730	0.415
Ncuts (3)	0.516	0.627	0.438
Ncuts (4)	0.527	0.563	0.495
Ncuts (5)	0.512	0.539	0.487
Ncuts (6)	0.490	0.515	0.468
Ncuts (7)	0.480	0.501	0.462
Ncuts (8)	0.488	0.481	0.495
Ncuts (9)	0.466	0.456	0.477
Ncuts (10)	0.453	0.436	0.471

Normalized cuts is a general purpose segmentation algorithm that finds groups of similar pixels, with a bias against finding small regions. The implementation used here measures the similarity between two pixels by searching the space between them for edges that might indicate region boundaries. In the experiments, the non-touching normalized cut regions that maximally overlapped the object pixels in the motion segmentations were considered the object segments and all the others were considered the background. Thus dividing a single object into multiple segments is penalized. The set of object segments can be no larger than 3, since none of the test images contain more than 3 objects. The total number of segments found by normalized cuts was varied from 2 to 10 to discover if any setting was optimal for finding object segments across the entire data set. The intent is to demonstrate that object segmentation cannot be easily solved with a generic image segmentation algorithm.

On both data sets, the F-measure of brightness-only, single resolution SANE is higher than that for any setting of normalized cuts.[1] Because the object segmentation problem requires that an algorithm distinguish between object and non-object boundaries, normalized cuts labors under two disadvantages. First, it only has generic knowledge of image regions, while SANE is trained to recognize the region properties appropriate to particular objects and environments. In Figure 3, there is no number of normalized cut segments that works well across all three example images because it does not have a model that favors car boundaries over other types. Secondly, normalized cuts does not have a strong model of object shape. The boundaries of the walking person in Figure 4 are not very well defined due to saturation caused by the extremely bright background. SANE benefits enormously because it has learned a strong shape model that allows it to do a better job of pulling out the human boundary and ignoring other more visually striking regional divisions. Normalized cuts, on the other hand, prefers dividing the person and the background along the most striking brightness boundaries, which often results in subdividing the object of interest rather than separating it from the surroundings. The data and these examples speak to the advantages of using machine learned algorithms with strong shape models to perform object segmentation.

It's interesting to note the odd performance of the brightness-only, multiresolution SANE algorithm on the traffic data. It appears that on this data, multiresolution without color information degrades performance. Adding color, however, produces a multiresolution SANE that is superior to all single resolution versions. This phenomenon bears further investigation, but it may point to the utility of color information in modeling low-resolution image structures.

Figures 3 and 4 have examples of the performance of different SANE variants. In the traffic images, one of the brightness, single resolution examples is particularly bad because the belief propagation algorithm failed to converge. However, the addition of color and multiresolution structure in the other SANE models overcomes that problem.

To further examine the benefits of the extra shape information provided by the shape model and the closed boundary requirement imposed by segmentation, we also compared the output of SANE to the learned boundary detector created by Martin et al. (2004). The Martin code (Martin & Fowlkes 2004) was adapted to our data by using constant-sized features, since the image sizes of our data don't indicate the object scales. We trained the detectors on our traffic data using multiple feature scales and selected those with maximal F-measures on the task of detecting the boundaries in our test set. For comparison, we also measured how well the color, multiresolution SANE detector performed on the same data. A detected boundary pixel had to be within 3 pixels of a motion boundary to count as a correct detection.

As seen in Table 3, SANE outperformed all the different Martin detectors (brightness gradient (BG), color gradient (CG), texture gradient (TG), and the BGTG and CGTG

[1] These statistics are computed across all the pixels in all the test images. The best average per-image F-measures for SANE and normalized cuts on the traffic data are more similar, probably because smaller images are easier to segment correctly.

combinations). Since the Martin algorithm had access to the same training data and used a much more sophisticated set of local features, SANE's advantages must be due to its shape model and the sharing of information produced by the MRF. The Martin detectors are calculated independently at each location and do not allow for the pooling of information at many locations to make optimal joint decisions. Additionally, the SANE requirement that outputs must form valid segmentations, that they have closed boundaries, helps the model ignore boundaries that are internal to the object and continue boundaries in regions with ambiguous local data. Because the Martin detectors lack this requirement, they can detect edges that never form part of a complete boundary and cannot infer boundaries in data-poor regions.

Table 3: SANE and Martin et al. boundaries.

Algorithm	F-measure	Recall	Precision
SANE	0.642	0.759	0.556
Martin BGTG	0.605	0.813	0.482
Martin BG	0.599	0.704	0.521
Martin CGTG	0.597	0.779	0.484
Martin TG	0.579	0.810	0.450
Martin CG	0.376	0.477	0.311

Future work

The motion segmentations produced by background subtraction lack T-junctions, areas where three different regions meet, because there are only two region labels—foreground and background. Similarly, the SANE algorithm assumes only two region labels and therefore only distinguishes non-overlapping objects. The next extension of this work will be the ability to handle overlapping objects with T-junctions.

Conclusion

The SANE algorithm demonstrates the value of self-supervised learning and the combination of local image information, shape models, and global output constraints in the object segmentation task. Self-supervised learning allows the algorithm to adapt to new environments and outperform generic methods, making it ideal for integration with real-world systems. The combination of local boundary and region models with the shape relationships encoded in the MRF and the global requirement of producing a valid segmentation allow the model to better find the class of segmentations defined by the training data and to ignore distracting regional differences. Further advances could make this type of task and environment-specific segmentation a valuable part of actual, deployed computer vision systems.

Acknowledgments

This work was funded in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010, and in part by the Singapore-MIT Alliance agreement dated 11/6/98.

The authors thank Joshua Migdal, Chris Stauffer, David Martin, Jianbo Shi and his students, Sarah Finney, Luke Zettlemoyer, and the anonymous reviewers.

References

Besag, J. 1974. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B(Methodological)* 36(2).

Besag, J. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B(Methodoloical)* 48(3).

Borenstein, E., and Ullman, S. 2002. Class-specific, top-down segmentation. In *European Conference on Computer Vision*.

Cour, T.; Yu, S.; and Shi, J. 2004. Matlab normalized cuts segmentation code. http://www.cis.upenn.edu/∼jshi-/software/.

Freeman, W.; Pasztor, E.; and Carmichael, O. 2000. Learning low-level vision. *International Journal of Computer Vision* 40(1).

Geman, S., and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6).

Konishi, S.; Yuille, A.; Coughlan, J.; and Zhu, S. C. 2003. Statistical edge detection: Learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(1).

Martin, D., and Fowlkes, C. 2004. Matlab boundary detection code. http://www.cs.berkeley.edu/projects/vision-/grouping/segbench/.

Martin, D.; Fowlkes, C.; and Malik, J. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(5).

Migdal, J., and Grimson, W. 2005. Background subtraction using markov thresholds. In *IEEE Workshop on Motion and Video Computing*.

Palmer, S. 1999. *Vision Science: Photons to Phenomenology*. The MIT Press.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Shi, J., and Malik, J. 1997. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition*.

Spelke, E.; Vishton, P.; and von Hofsten, C. 1994. Object perception, object-directed action, and physical knowledge in infancy. In *The Cognitive Neurosciences*. The MIT Press. 165–179.

Stauffer, C., and Grimson, W. 1999. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*.

Wainwright, M.; Jaakkola, T.; and Willsky, A. 2003. Tree-reweighted belief propagation and approximate ML estimation by pseudo-moment matching. In *Workshop on Artificial Intelligence and Statistics*.

Weiss, Y. 1997. Belief propagation and revision in networks with loops. Technical Report 1616, MIT Artificial Intelligence Laboratory.