Nathan Merritt
December 12, 2010

Who Needs Pixels? A Cognitive Approach to Machine Vision

In the following paper I will discuss an architecture for abstract object recognition. At the system's core is the difference of Gaussian (DoG) function. The DoG function is used to build "receptive fields" of varying sizes in the image. These receptive fields are further processed to both segment the image into 7 +/- 2 regions of space and to locate keypoints in the image. A DoG-based approach has its roots in the neurophysiological basis of the brain; each receptive field can be thought of as a single neuron in primary visual cortex. In fact, neurons with response patterns similar to various levels of the DoG function have been located in real brains (citation).

The image segmentation algorithm can be thought to be a primitive analogue of the human dorsal system, while the keypoint recognition algorithm is a candidate to provide the first layer of input to the ventral system. Separating the two systems increases the efficiency of each and allows for more natural abstraction. Furthermore, interaction between the two systems will allow for a simple correlate of focus. Segments within the image that contain many keypoints will be considered highly relevant. Additional keypoints located in a relevant section of the image will have a lower threshold of activation. Thus our system will have the ability to regulate its function from the top-down in a neurally inspired manner.

Building neurologically plausible receptive fields

Both our image segmentation and object recognition algorithms rely on the ability to differentiate areas of space. To build this functionality from the ground up, we must first construct a measure of the difference in color at any given point in the image. To do this, we will run a guassian over the image, with various sigma parameters. This will produce different levels of smoothing on the image. We obtain the difference of guassian function by subtracting two guassians with different sigmas. This produces a high magnitude “cell” when there is high contrast between the center of the field and its surroundings. We can then use the absolute value, so that our cells respond to any point in the image with rapid color shift, no matter whether it goes from light to dark or vice versa. The radius of our receptive field is dependent on the sigma values used in each of the two gaussian functions.

Now that we've constructed small, highly overlapping receptive fields across the image we can focus on building an abstract representation from these fields. The first step is to use the concept of lateral inhibition to greatly reduce the number of cells we listen to. We'll approximate biological inhibition by doing a simple hill-climb across all the receptive field cells with the same radius (sensitivity), discarding all but the maximally responding cells. This will reduce our image from having a cell centered at every pixel to one with much less overlap. To use Lesperance's naming convention, we are now dealing with the @peak cells at each resolution. These @peak cells can now be used to segment the image and identify key features to use in object recognition.

Image segmentation

Our image segmentation step has two major goals: to break the image into 7 +/- 2 sections, and for those partitions to correspond to groupings that a human might suggest. Lesperance achieves these goals by using several layers of receptive fields with different sizes. In his scheme, the smaller @peak cells detect edges of objects, while the larger fields are maximally active when the object is centered and the same size as the receptive field (maximum contrast). He then groups highly active cells together, so that a region consists of all the edges (small receptive fields) that are encompassed by a highly active larger receptive field. Lesperance suggests several improvements related to using slightly more complicated receptive fields. His improvements would help to eliminate certain light/dark patterns not being detected as an edge, but do not address his system's reliance on extremely large receptive fields.

One potential solution to this shortcoming is to use a cell-assembly network for all associative post processing on the @peak cells. Ultimately, we want to associate similarly sized highly active @peak cells that are close together with each other, and group them with a larger @peak cell. The small cells will indicate the region's boundaries, while the larger cells provide a better general representation of the area. A cell-assembly network with variously sized, spatially consistent input layers is a possible way to achieve this. The network should have a 1-1 mapping of receptive field cells to input neurons. It should be arranged so that spatial relationships among the cells are maintained in the network. If done right, within-layer lateral inhibition in this network could eliminate the need for Lesperance's hill climbing step, @peak cells would arise naturally as the only cells to still be firing after lateral competiton. Further associative grouping could be handled by interactions between the different input layers, so that activity from an extremely large peak cell could influence lower layers in a top-down manner.

Feature Recognition

Feature recognition also has its roots in the @peak receptive fields. Lowe (2004) describes a method by which scale and rotation invariant features can be extracted from an image. I'll briefly discuss his method, and then describe some modifications I think can be applied to both optimize the processing and make the resulting features (“keypoints” in Lowe's terminology) more useful for general object recognition with a cell-assembly network.

Interestingly, much of Lowe's architecture is common with Lesperance's image segmentation algorithm. Both approaches use a series of Guassian convolutions over an image, and then calculate the difference of guassian (DoG). Lowe adds an additional step to achieve scale invariance. Once the DoGs have been calculated for the initial image, the image is down-sampled by a factor of two and the DoGs are calculated again. This ensures that some features of an image will be recognized no matter its size in the visual field. The next generalization Lowe applies is a rotation. Each @peak cell represents a point of high dark/light contrast, and so by finding the gradient at that point we can rotate and thereby normalize the pixel. This step will allow our recognition algorithm to identify a key point even if the object that contains it has been rotated. This step could definitely be incorporated into Lesperance's image segmentation algorithm, since both are dealing with fundamentally similar problems.

Now that we've identified preliminary keypoints, and abstracted their size and orientation, it's time to describe them. Lowe builds a keypoint descriptor by inspecting the gradient of nearby points. First, each point is weighted by a Guassian function with a sigma of one half the descriptor window. In effect, this makes the magnitude of the gradients in the center more important than the gradients that are further out. The space around the keypoint is divided into n*n regions (Lesperance uses 4x4 in his experiments, but I think a 2x2 descriptor array will work better for reasons I'll discuss below), and in each region the magnitudes of the gradients are used to build a histogram. These histograms are then used to build an n-dimensional feature vector. The feature vector's dimensionality is determined by the number of histograms, multiplied by the number of bins in each histogram. By scaling this vector to unit length, we can achieve invariance over illumination. This works mainly because we are already dealing with gradients, and also because a linear change in illumination will cause a similar increase in the magnitude of the gradients (so we end up being more concerned with gradient distribution).

Lesperance's goal was to build keypoints that could then be used in a generalized Hough transform. As such, he wanted few false-positives but wasn't that worried about false negatives. Identifying even a few keypoints associated with an image would be enough, as the Hough transform deals with obstruction extremely well already. This is why his experiments worked the best with a massive descriptive vector: 4x4 histograms with 8 bins in each (128 dimensions)! Our cell assemblies function similarly to a Hough transform in that they should be able to handle (and even thrive) under non-optimal conditions. However, to build a coherent network we will need lots of input. I think that we will find much better results with a significantly smaller descriptive vector, 2x2x8 should be more than enough information about each point to allow an associative network to function (this is entirely a gut feeling...). Instinctively, this makes sense to me because reducing the amount of information in any keypoint will further generalize it and we want to provide the lower layers of the network with as much general information as possible. Also, if we later decide we need a larger input layer, it would be straightforward to incorporate location within the image. Since Lowe uses a Hough transform, his algorithm doesn't care where in the image a given key point is located until the post-processing phase (to draw a box around the already recognized object).

Plugging our data into a cell-assembly network

At this point, our visual system consists of two separate mechanisms. Both rely on the DoG to identify points of contrast within an image. Our segmentation algorithm will split an image into several regions, and our object recognition system can pick out highly localized patterns or so called “key points.” In human vision, image segmentation and feature recognition are likely part of the same system (the ventral stream). The dorsal system's task is to abstract motion and location, something I'll discuss briefly but not focus on. Our goal is to combine the information from the two algorithms to produce coherent and yet abstract representations of things in the visual field. This representations will necessarily be semantically transparent – they are built up entirely from perception and as such do not suffer from the symbol grounding problem.

The first step is to build a cell-assembly network with an input cell for every possible keypoint descriptive vector. This will allow an easy 1-1 mapping from observed keypoints to next stage in processing. It is important that the cell-assembly be wired hierarchically, but very densely to start. As the network ages, connections that haven't been heavily used will be pruned. A dense wiring ensures that any subset of keypoints can fire together and trigger a cell in the next level of the hierarchy. Pruning the network will help to control it; we don't want a new visual stimulus to be able to trigger an explosion of activity.

Another portion of the network will code for an image's location in the visual field, and allow for a computational correlate to focus. When a keypoint is found, its location (X/Y) will also activate a cell in a neighboring network. This network will also receive top-down input from the segmentation algorithm. Each segment will have a high-level node whose activity will indicate how many keypoints have been found within the segment. As more relevant keypoints are found, the segment will stimulate all pixels within its boundaries (conversely, it will inhibit its pixels when few keypoints have been found). This will promote the activation of keypoints that may not have otherwise been judged important enough to process. In theory, this feedback loop will allow the system to focus in on a specific segment if a few highly relevant (judged by their activation on higher levels of the object cell assembly, perhaps?) keypoints are found. So in a blurry image, the system can be more tolerant of input in specific portions, while still ignoring the majority of the noise. Also, the activity level of a segment will serve as a rough indicator of how much is "going on" in that segment. This will allow us to differentiate the foreground and background of an image.

If we're building a system to search for a specific object which we may or may not be seeing, then this mechanism could be used to scan the entire visual field. If the object hasn't been found after a first pass, we could then inhibit the highly active segments and activate others, bringing out more keypoints in them (hopefully enough to allow our network to recognize the image).

Conclusions

I thought it was extremely interesting that both Lesperance and Lowe used the difference of guassian function as the basis for very different tasks in machine vision. I think that a cleverly wired series of associative networks could likely be trained to perform more accurately and faster than Lowe's database and Hough transform or Lesperance's complicated post processing. In my view, the most difficult part of the project will be figuring out how to keep the network stable after an initial period of training. Neural networks are well known to suffer from catastrophic forgetting, but human brains do not. I think that we may be able to prevent this by pruning the network after training, but the pruning may stunt the system's flexibility.

Additionally, it may be possible to perform the majority of the computation before the two systems split. The DoG and lateral inhibition (@peak cell determination) steps could be run in a highly optimized hardware loop which dumps its output into a common cell-assembly input layer. Both systems could then begin processing on this layer, and modulate it as top-down conditions dictate.

Bibliography

Ideas from many sources discussed in Computer Science 355, uncited. For details see the research of Kaplan, Chown and others from the Michigan Dept. of Computer Science. CS355 was taught at Bowdoin College in the Fall of 2010 by professor Eric Chown (echown@bowdoin.edu).

Grossberg S, Mingolla E, Ross WD. A neural theory of attentive visual search: Interactions of boundary, surface, spatial, and object representations. *CAS/CNS Technical Report Series*. 2010;(038).

Lesperance RM-K. The location system: using approximate location and size information for scene segmentation. 1990. Available at: http://portal.acm.org/citation.cfm?id=917131.

Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004;60(2):91-110.