CINNIC, a new computational algorithm for the modeling of early visual contour integration in humans

T. Nathan Mundhenk[1]
Computer Science Department
University of Southern California
Los Angles, California, 90089
mundhenk@usc.edu

Laurent Itti[1]
Computer Science Department
University of Southern California
Los Angles, California, 90089
itti@pollux.usc.edu

Abstract

In order to gain a better understanding of visual saliency, we have developed and algorithm which simulates the phenomenon of contour integration for the purpose of visual saliency. The model developed consists of the classical butterfly pattern of connection between orientation selective neurons in the primary visual cortex. In addition, we also add a local group suppression gain control to eliminate extraneous noise and a fast plasticity term which helps to account for closure effect often observed in humans exposed to closed contour maps. Results from real world images suggest that our algorithm is effective at picking out reasonable contours from a scene. The results improved with the introduction of both the fast plasticity and group suppression. An addition of multi scale analysis has also increased the effectiveness as well.

Keywords: contour; integration; visual; saliency; model

1 Introduction

We have created an algorithm, which will integrate contours in real world images. The idea is to emulate the way the human brain integrates contours for visual salience in early visual pre-processing. To this end, our goal is to simulate saliency for a given contour. The important components of this model are not only its abilities to find contours in an image, but for it to find contours that a human finds salient as well. Such things include contour continuity, length, closure and the uniqueness of the contour when compared to its background.

Our approach has been to start with a simple model of neural connections. We use here a standard butterfly shape for connections, which has been tried with success in the past (for instance Li[2]). The elements of the butterfly pattern are connections that branch out like wings to co-linear neurons, which creates excitation on the neurons it connects to. However, suppression is also added with a set of orthogonal wings that are used to suppress parallel contours. The end result should be that the butterfly

[1] More information at http://iLab.usc.edu

model picks out salient contours in an image using co-linearity. However, it seems that the original butterfly shape of connections is difficult to manually tune for multiple images suggesting that the brain has adaptive centers to do this. In addition to this, the butterfly shape also does not necessarily account for contour closure or the observation that contour effects seem to extend beyond their receptive field.

To address these factors, we propose to use several devices that should aid in explaining several observed effects. We have added an adaptive layer using group suppression for when a group of neurons seems to get to excited. We have also added fast plasticity described by Braun [1], for neural weight to allow connections to dynamically adapt to local neural activity. In addition we are using the model at multi scales, which should help in preventing otherwise salient contours from being excluded.

2 MODEL

The current model which we have named CINNIC (Carefully Implemented Neural Network for Integrating Contours) starts with the basic butterfly local connections, but in addition to this, we have added the use of multiscale analysis, a local group suppression gain control and fast plasticity for long range effects. Figure 1 shows a schematic description of our model, which is also described by equations (1-5). The first step is analyzing the input image using Gabor filters turned for 12 different angles. This produces 12 images that represent elements from the original image at increments of 15 degrees. A noise factor of approx. 2% is introduced at this stage. These 12 images are then reduced into three different scales 64x64, 32x32 and 16x16 pixels in resolution, which are run separately and do not interact. A 4D convolution is run to simulate the interaction between the different orientation images. The convolution is done using a set of 144 kernels that represent all possible interactions between pairs of the 12 orientation images. These kernels specify the excitation and suppression that should occur between two elements in the images. The kernels take into account the colinearity of two elements as well as their separating distance. For simplicity, interaction strength decreases as a ramp function as two elements are further separated. The kernels are statically specified at the beginning of a program run by input parameters and do not interact.

Each scale is run separately from the other. Each element is convolved against every other element within its range in such a way that collinear elements tend to excite, while parallel elements tend to suppress each other (eq. 1). This is expressed as $x_{ij\alpha}$ being the source image pixel at location (i, j) and orientation α, and $x_{kl\beta}$ being the other image pixel at location (k,l) and orientation β then taking the product of these two by the kernel $k_{\alpha\beta(k-I)(l-j)}$. It should be noted that m and n equal the image scale for instance 64,32 or 16. Further, $(S_{ij})^t$ is a group suppression term for the current group (detailed below) with *t* being the current iteration. $(P_{ij\alpha})^t$ is a plasticity term (also detailed below). The resulting potential from a single iteration is sent to a saliency map $(V_{ij})^{t+1}$. Each pixel in the saliency map represents a column of pixels from each of the twelve orientation images summed. The saliency map itself is made up of leaky integrator neurons, which lose

$$(v_{ij\alpha})^{t+1} = (S_{ij})^t (P_{ij\alpha})^t (x_{ij\alpha}) \sum_{\substack{k\in[[0,m]] \\ l\in[[0,n]] \\ \beta\in[[0,11]]}} (x_{kl\beta})(k_{\alpha\beta(k-i)(l-j)}) \tag{1}$$

$$(V_{ij})^{t+1} = \sum_{\substack{k\in[[0,m]] \\ l\in[[0,n]] \\ \beta\in[[0,11]]}} (v_{kl\alpha})^{t+1} - L \tag{2}$$

$$(S_{ij})^t = \upsilon\Big[\sum_{(k,l)\in N_i \times N_j} ((V_{kl})^t - (V_{kl})^{t-1}\Big] - T \tag{3}$$

with

$$N_i = [[i-(m/8);i+(m/8)]]$$
$$N_j = [[j-(m/8);j+(m/8)]]$$

$$(P_{ij\alpha})^t = (v_{ij\alpha})^t(C) \tag{4}$$

$$I_{ij} = \mathrm{sig}((V_{ij})^t) \tag{5}$$

some constant potential *L* from one iteration to the next (eq. 2). To form a final saliency map for one of the three image resolutions, the potential from the leaky integrator neurons are fed through a sigmoidal function that simulates neural firing patterns (eq. 5) with I_{ij} being the final saliency map pixel for this scale.

Non-linearities are introduced in the form of the group suppression gain control (eq. 3) where T is the threshold constant and $(V_{kl})^t$ is the potential for a neuron in this group which are all summed for that group with υ as a constant multiplied by that sum. m and n represent the image size at that scale The suppression is based upon the rate of the change of excitation. Fast plasticity (eq. 4) is introduced as $v_{ij\alpha}$ being the potential this neuron had multiplied by a constant C. The fast plasticity works by increasing all weights for a single simulated neuron, proportionally to the excitation it received in the previous iteration. That is, neurons that are stimulated more tend to stimulate collinear neighbors more as well as to suppress parallel neighbors more. This function is introduced to re-create non-local interactions that are observed in human subjects in an attempt to account for observed contour closure effect. The fast plasticity used here is bounded to 5 times the original connection strength for any given neuron.

In our model the usage of fast plasticity was chosen for several reasons. The first was the suggestion by Braun [1] that other methods that attempted to explain contour

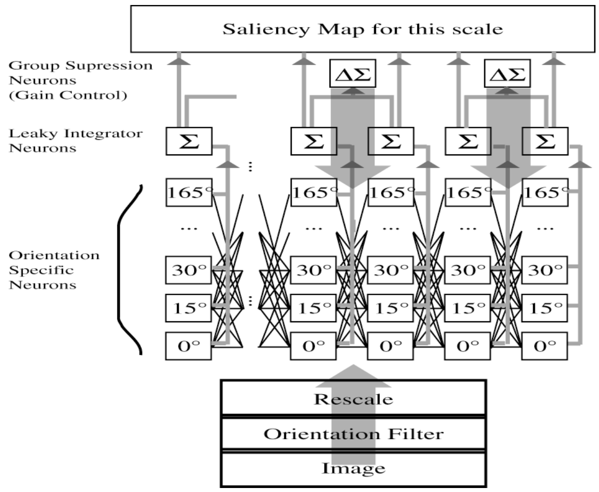

Fig. 1. This is a basic representation of the CINNIC algorithm. An input image is filtered, rescaled then interacted with images at other orientations including itself. The output goes to a saliency map of leaky integrator neurons. Group suppression is fed back from the change in group potential.

closure either occurred too fast such as cumulative propagation [2], or were too slow such as temporal synchronization [3] as to explain the time it takes for closure effect to happen which Braun measured at about 250 ms. Second we wished to test the idea of fast plasticity and find if it was a viable mechanism for explaining closure effect.

Another non-linearity introduced is a simple local gain control using group suppression. Neurons are grouped into local neighborhoods of size 1/8x1/8 pixels of the image size at the current scale (e.g. 8x8 pixels for a 64x64 pixel scale). If the total change in potential from a group surpasses a threshold then the neurons increase their suppression of parallel neighbors proportionally to the increase past threshold. The group includes all neurons in all orientation maps for a given visual location, which report to the same image location. There is no current cap on how much additional suppression can be added using this method.

The algorithm runs for eight iterations, which was a number chosen based upon its observed optimality. After the final iteration, the three scales are brought back together and combined using a fixed weighted average. This average is the total saliency map of contours for the input image. The entire process takes approximately 2 minutes using an Athlon 1400 MHz based PC running Linux. The time is mostly due to the enormous amount of computation needed to compute interactions between neurons from all possible pairs of the 12 images using 144 2D kernels.

3 RESULTS and CONCLUSION

Initially, the model had been tested on a number of both real world images as well as contour simulation images. The results at face value seemed to yield positive results.

Large amounts of noise are cleaned out with the adaptive group suppression. The fast plasticity enhances the contours of the test images and the multi scale model insures that fewer contours are missed.

Quantitative analysis has also been conducted. 24 real world images representing a wide variety of contexts were used. A priori we used Photo Shop or GIMP to trace the outline of what seemed to us to be reasonable salient contours. The outline images where taken and then compared with the output images from CINNIC. It should be noted that, all images from CINNIC were run with the same values and no manual tuning was done between the time images were run.

Analysis was done between the test image and CINNIC output image for all 24 images using Euclidian distance as well as linear regression correlation under several different conditions. Without group suppression the mean Euclidian distance was poorer. Also, the variance was twice as much without group suppression for both correlation and Euclidian Distance suggesting that group suppression increases the consistency of outcomes. Fast plasticity also seemed to help. Without fast plasticity the minimum correlation was negative among the 24 images. However, with fast plasticity, all correlational values were positive.

The data also supports the notion of improvement from the use of multi scales. Each of the scale images by themselves was compared with the outline images. In neither case did the mean correlational value or Euclidian distance surpass the values of the final max selected composite image.

Finally, figure 2 shows the output from the algorithm on three real world images used in the sample of 24 images. The five most salient points are circled by the algorithm and shown against the original image in the top row. The bottom row represents the

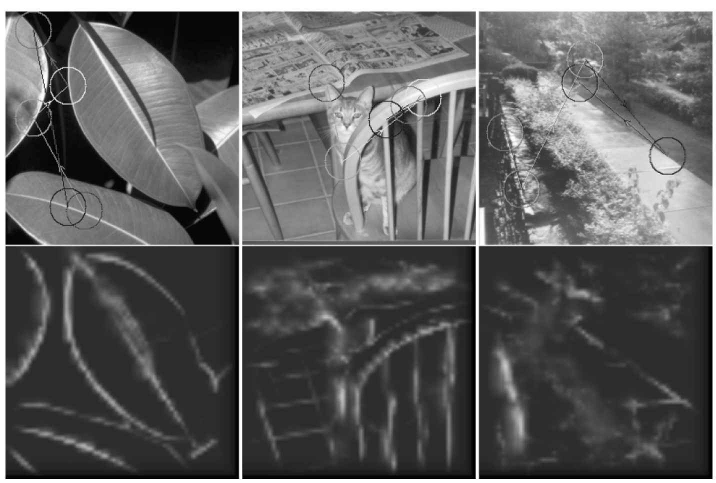

Fig. 2. The top row are real world images with the top 5 most salient points circled by the algorithm. The bottom row represents the raw saliency map for each image.

raw saliency map output from which the five most salient points are selected using the five brightest regions.

Qualitative analysis suggests that CINNIC has an affinity for longer straighter contour segments with greater continuity, which seems to agree with literature on contour integration. CINNIC is also more effective at finding contours that are more unique given its region, as is also suggested by literature on contour integration. CINNIC's abilities with closure effect are currently being assessed, but the data regarding it as of yet is not conclusive.

Acknowledgements

We would like to acknowledge Jochen Braun, Christof Koch and Mike Olson for their help and suggestions. This research is supported by the National Imagery and Mapping Agency, the National Science Foundation, the National Eye Institute, the Zumberge Faculty Innovation Research Fund and the Charles Lee Powell Foundation.

References

[1] Braun J , 1999, On the detection of salient contours, *Spatial Vision*, **12**(2):211-225
[2]Yen S, Finkel L H, 1998, Extraction of perceptually salient contours by striate cortical networks, *Vision Research*, **38**(5):719-741
[3] Li Z, 1998, A neural model of contour integration, *Neural Computation*, **10**:903-940

T. Nathan Mundhenk received his BA in psychology from the University of Cincinnati in 2000 and was team leader of the University of Cincinnati Robotics Team. He is a current member of iLab at the University of Southern California (Los Angeles, California) where he is a graduate student in computer science. He is currently involved in research into contour integration, vision in ego motion and is the team leader for the Beobot robotics project at iLab.

Laurent Itti received his MS in image processing from the Ecole Nationale Super-euredes Telecommunications (Paris) in 1994, and PhD in computation and neural systems from Caltech (Pasadena, California) in 2000. He is currently an assistant professor of computer science at the University of Southern California in Los Angeles. He is the Director of iLab, which pursues research in biologically motivated computer vision and neuromorphic robotics.