Chameleon-pixels: Filling in context given a sparse visual percept

Deborah Hanus, Ryan Schoen, and Emily Zhao
Massachusetts Institute of Technology
77 Massachusetts Avenue
dhanus, rschoen, ezy **@mit.edu**

Abstract

Humans can recognize even a few pixels in the context of a 32x32 pixel image, although those same pixels are completely indiscernible when they are taken out of the image context. We model the problem of pixel identification as a distributed system. Using one-hop communication and constraint propagation, each chameleon pixel calculates a probability distribution on its location given observations of its neighbors. We use human fixation data to choose the starting points of the constraint propagation, and compare the performance of our Chameleon-pixels with the human visual system. We find that given human fixation data, the Chameleon-pixels make a similar pattern of errors to those we would expect from a human observer, lending credence to the integral role human fixations play in the role of object recognition. This insight can be used to improve the efficiency of future image reproduction and recognition systems.

1. Introduction

If we are to create a computer vision system that recognizes objects as accurately and efficiently as a human, we must first understand how humans use context in visual perception and recognition. Numerous studies have demonstrated that context plays an integral role in human object perception. Humans can easily recognize even a few pixels in the context of a 32x32 pixel image, even when those same pixels are completely indiscernible when they are taken out of the image context [15]. Showing an object with a semantically consistent background or even similar objects greatly facilitates human object recognition [4, 5]. When shown an extremely fuzzy scene, the human visual system even fills in semantic meaning where there is none [11]. The ability of the human visual system to "fill in" context often greatly facilitates object perception.

This human ability to take a limited amount of visual information and "fill in" the surroundings turns out to be a fundamental aspect of how each of us sees the world. The eye's most sensitive receptors are found in the fovea, a very small region of the eye that enables high-resolution vision. When viewing a scene, a human can only perceive 10-12 degrees of it, with high-precision, at any given time; however, most people report that they perceive objects within 200 degrees of visual angle clearly. How are they able to accomplish this?

Selective attention enables humans to direct our gaze rapidly towards objects of interest in our visual environment [6]. A series of saccades, a rapid series of fixations in a given area, enable humans to perceive several parts of a scene in high resolution. The brain somehow combines the high-resolution information from these saccades into an apparently high-resolution percept. Eye-tracking provides precise temporal and spatial information about the fixation points of saccades, giving us valuable insight into which small pieces of visual information are necessary for a human to recognize an object or scene.

Computer vision systems have often attempted to replicate the human capacity to incorporate contextual cues into a visual percept to facilitate recognition of individual objects; however, as of yet, no object recognition system has managed to incorporate these contextual cues quite as efficiently as the human visual system [16, 2, 1].

It is interesting to consider how the disparate high-resolution visual information from each fixation is combined into a complete visual percept, and at what point in visual processing the combination occurs. One might consider that this filling in occurs early in visual processing, because fixation information is stored a early in the superior colliculus [14, 9]. However, it is more generally accepted that this information is combined via top-down influence from the frontal cortex [3, 12, 10]. To simulate the necessary "filling in" step that occurs in human vision, we borrowed from a model often used in robotics: a *distributed system*. We consider each pixel to be a state machine with some "idea" of where it fits into the target image based on

probability distributions it has calculated by observing its neighbors via *single-hop communication*.

The aim of our project is to model this visual phenomena of "filling in" contextual information using two images; a target image and a chameleon reproduction of the target image. Given an 32x32 pixel target image, each "chameleon pixel" in our system determines its location in the image and takes on the appropriate hue.

2. Methods

Torralba et al. (2008) demonstrated that humans could easily recognize even a few pixels in the context of a 32x32 pixel image, even when those same pixels are completely indiscernible when they are taken out of the image context. Similarly, our aim for our the chameleon pixels is that each pixel, one element in a distributed system, determines its location within the target image. We compare our system's pattern of errors to those of humans by running our system on images for which we have human fixation data [8, 7].

2.1. Data

We used the dataset of Judd, et al. [8, 7]. These images were 193 random images collected from Flickr creative commons and LabelMe [13]. The eye tracking data was recorded from fifteen users who free viewed these images. The longest dimension of each image was 1024 pixels and the other dimension ranged from 405 to 1024 with the majority at 768 pixels.

2.2. Distributed system

In the Chameleon-pixel system, each pixel is modeled as a machine in a distributed system, capable of displaying a color and observing the color of each of its immediate neighbors. We assume each pixel can also portray their confidence in choice of color by varying the intensity with which they display it.

The pixels are arranged in a grid corresponding to the length and width of a target image, and each pixel is given a complete copy of that image. However, each pixel does not know where in the image it is located. Based on the presence or absence of neighbors in each direction, each pixel can tell if it is on an edge or corner (and accordingly, which edge or corner), but internal pixels cannot determine anything besides the fact that they are internal.

At each time step, every pixels observes the color and intensity output by its four neighbors. Given these observations, the pixel calculates its likelihood of being at each possible position in the image. The pixel then displays the color of the most likely pixel, with intensity (confidence) proportional to its likelihood of being correct.

Given this system, we initialize certainty using two methods: one based on machine vision, and another based on human vision. In the first method, certainty begins at the corners: each corner has full knowledge about where it is, but can only communicate color and confidence to its neighbors. Information therefore flows from the corners to the center of the image, ideally converging to the target image. In the second method, certainty is initialized at the points of the image that humans fixated on most when they were asked to view the image naturally. In this case, the pixels that the humans looked at most begin with full certainty of their location, and then can only communicate color and confidence to its neighbors.

To explain the system, we first must define a similarity metric between two colors. In our system, the similarity metric is the normalized closeness in RGB space:

$$S(x,y) = 1 - \frac{\sqrt{(x_R - y_R)^2 + (x_B - y_B)^2 + (x_G - y_G)^2}}{\sqrt{3 * 255^2}} \tag{1}$$

The algorithm is then as follows:

Algorithm 1 Pixel identification algorithm.

the target image T is given
$N_{x,y}(\delta)$ is the true neighbor of (x, y) in the δ direction
$\delta_{neigh} \leftarrow [(0,-1),(0,1),(1,0),(-1,0)]$
for $(x, y) \in T$ **do**
  if pixel(x,y).conf == 1 **then**
    continue
  end if
  $\hat{p} \leftarrow 0$
  for $(i, j) \in T$ **do**
    $p \leftarrow 1$
    for $\delta \in \delta_{neigh}$ **do**
      $N' \leftarrow T(i + \delta(0), j + \delta(1))$
      $p \leftarrow p * S(N_{x,y}(\delta), N') * N_{x,y}(\delta).conf$
    end for
    if $p > \hat{p}$ **then**
      $\hat{p} \leftarrow p$
      $(\hat{x}, \hat{y}) \leftarrow (i, j)$
    end if
  end for
  pixel(x,y) $\leftarrow T(\hat{x}, \hat{y})$
  pixel(x,y).conf $\leftarrow \hat{p}$
end for

The algorithm can also be expressed mathematically. Given the target image T, a pixel (x, y), and one of its neighbors $N_{x,y}(\delta)$, we can define the likelihood due to the neighbor of the pixel being located at (i, j):

$$p_{x,y}(i, j, \delta) = S(N_{x,y}(\delta), T(i + \delta, j + \delta)) * N_{x,y}(\delta).conf \tag{2}$$

The overall probability of a pixel (x, y) being located at a position (i, j) is then given by the product of the likelihood of its being there due to all its neighbors:

$$P_{x,y}(i,j) = \prod_{\delta \in \delta_{neigh}} p_{x,y}(i,j,\delta) \quad (3)$$

where δ_{neigh} contains the four offsets defining the relative position of each neighbor. The maximum likelihood location and the confidence in that location can then be found as follows:

$$(\hat{x}, \hat{y}) = \arg\max_{i,j} P_{x,y}(i,j) \quad (4)$$

$$conf_{x,y} = \max_{i,j} P_{x,y}(i,j) \quad (5)$$

In each iteration, this update happens once for each pixel that has confidence lower than 1.

3. Results

We ran our Chameleon-pixel system on the dataset in two conditions. First, we ran the system on the dataset, initializing the points of certainty at the corners of image. We chose these points because the corners, with only two neighbor-pixels rather than four, are the most constrained points of the image. Second, we ran the system on the same dataset, initializing our points of certainty based on the points of the image that human subjects fixated on most. We found that, when we initialized the points of uncertainty based on human fixations, the Chameleon system made a pattern of errors that is suggestively similar to the pattern of errors that might be generated by a human participant.

3.1. Convergence

When we run the Chameleon-pixel algorithm on a 32x32 pixel image, we find that the system produces a good reproduction of the target image. Most of the pixels are able to correctly identify their appropriate color in 30-40 iterations of constraint propagation (Figure 1).

Initially we tried to get our system to run on larger images as well, however, we found that the algorithm took a prohibitively long time to run, and produced several several errors. In order to make a scientific contribution within a semester's time constraints, we elected to analyze only images with 32 pixel dimensions.

3.2. Chameleon vs. human errors

We chose how to initialize the Chameleon-pixels' the points of uncertainty using two methods. In one method, we initialized the points of certainty to be the most constrained points – the corners (a), and in the other, we initialized the points of certainty to be the points humans fixated on most when viewing the image naturally (b). In figures 2–4, we show the Chameleon-pixel algorithm's reproduction of the target image after 10 iterations of the algorithm (c, d), completion (e, f), and a heat map of errors when the algorithm is completed (black: correct, white: incorrect).

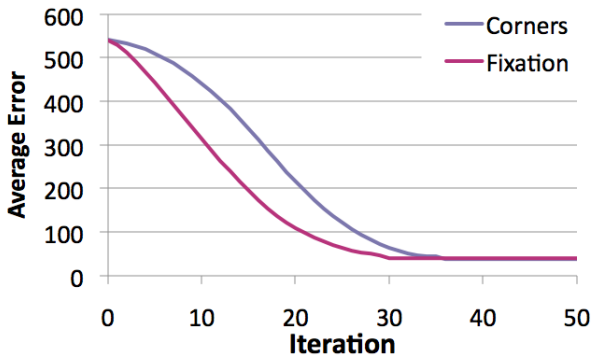

Figure 1. Chameleon-pixel algorithm converges. When we run the algorithm on a 32x32 pixel image, we find that the algorithm converges to low error rates, such that most of the pixels are able to correctly identify their appropriate color. When the points of uncertainty are chosen to be the same as human fixation points (red), on average, the error decreases more quickly than when the points of uncertainty are chosen at the corners (blue).

Here, we discuss the results of reproducing three representative target images that produced a particularly interesting patterns of errors. We also ran the Chameleon-pixel algorithm on the entire dataset of [8], but most of the patterns of error we found in the other images confirmed the patterns that we noted in our representative image set (see Figure 2). We find the Chameleon-pixel system makes a pattern of errors similar to those that we might expect a human to make when asked to report information about the image.

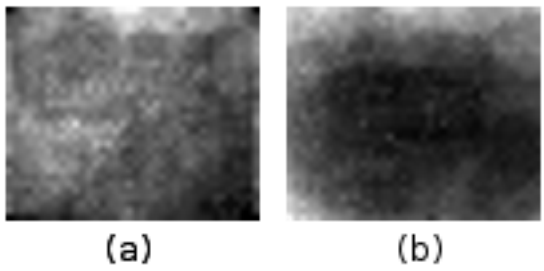

Figure 2. Comparison of accuracy across all images. When certainty began at the corners (a), we found the Chameleon-pixel system made fewer errors in the corners, but otherwise the locations of the systems errors was unpredictable. When certainty began at points of human fixation (b), we found the Chameleon-pixel system made fewer errors in the center of the image, where people are more likely to fixate, and fewer errors along the periphery. These errors are much more like those we would expect from humans.

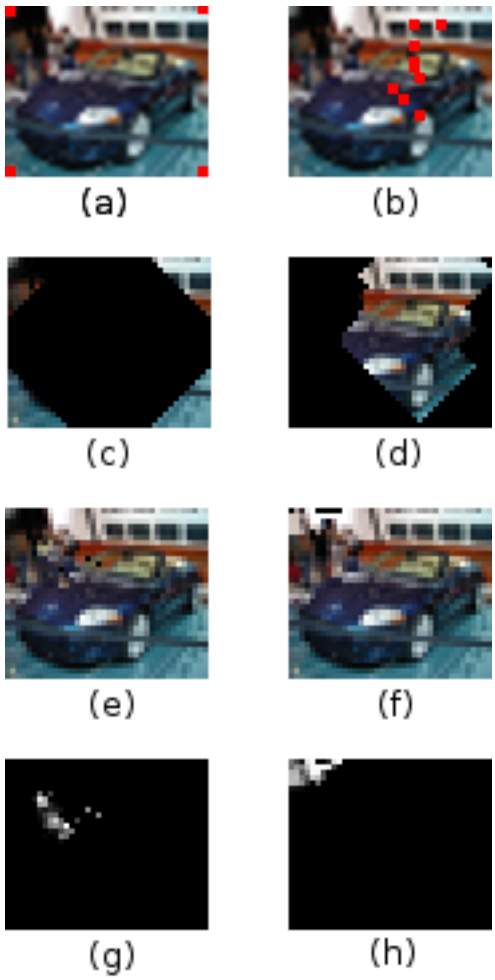

Figure 3. Comparison of errors on a large land vehicle. We initialized the points of uncertainty to be at the corners of the image (a) or the points humans fixated on most when viewing the image naturally. After several iterations, the image is still not recognizable using the first method (c), but is already recognizable as a car using the human fixation method (d). Chameleon-pixel system's reproduction of each image is shown in (e) and (f), respectively. Heat maps showing the systems certainty are shown for the two conditions (g) and (h) reveal interesting results (black: correct; white: incorrect). When the system begins at the corners, it makes errors on the people standing behind the car, but when it begins at the fixation points, it only produces a few errors in the periphery of the image.

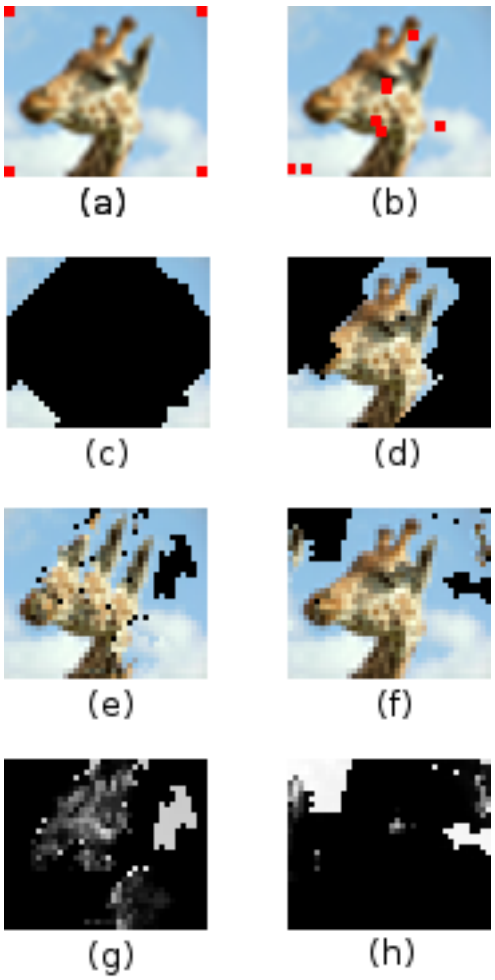

Figure 4. Comparison of errors on an animal. We initialized the points of uncertainty to be at the corners of the image (a) or the points humans fixated on most when viewing the image naturally. After several iterations, the image is still not recognizable using the first method (c), but is already recognizable as a face using the human fixation method (d). Chameleon-pixel system's reproduction of each image is shown in (e) and (f), respectively. Heat maps showing the systems certainty are shown for the two conditions (g) and (h) reveal interesting results (black: correct; white: incorrect). When the Chameleon-pixel system begins at the corners, the face of the giraffe is almost indistinguishable, but when it begins with certainty information about the eyes and the face of the giraffe as a human does, it reproduces a clear picture of the giraffe's face, only making errors in the periphery.

First, when the target image is a car with people in the background, we find that when the system begins at the corners, it makes errors on the people standing behind the car, but when it begins at the fixation points, it only produces a few errors in the periphery of the image (Figure 3).

Second, when the target image is an image with lots of similar patterns, such as the face of a giraffe, the human fixation points go straight to the giraffe's face and eyes. These points of certainty propagate through the rest of the image, so the Chameleon-pixels reproduce a recognizable giraffe face (Figure 4(f)); although, when the same system starts at the corners, it produces a nearly unrecognizable texture (Figure 4(g)).

Finally, when the target image contains human faces, the Chameleon-pixels combined with human fixation in-

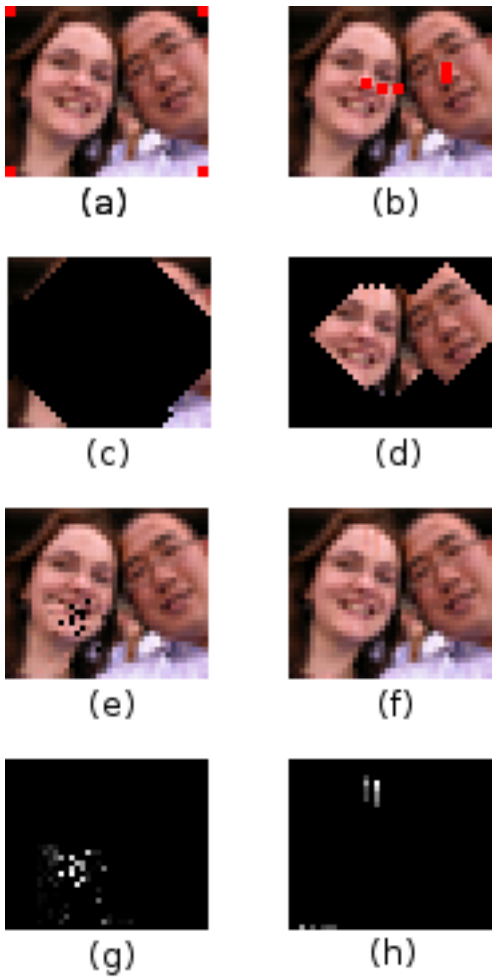

Figure 5. Comparison of errors on an human faces. We initialized the points of uncertainty to be at the corners of the image (a) or the points humans fixated on most when viewing the image naturally. After several iterations, the image is still not recognizable using the first method (c), but is already recognizable as human faces using the human fixation method (d). Chameleon-pixel system's reproduction of each image is shown in (e) and (f), respectively. Heat maps showing the systems certainty are shown for the two conditions (g) and (h) reveal interesting results (black: correct; white: incorrect). When the Chameleon-pixel system begins at the corners, the system makes errors around the girl's mouth and other facial features, where a human would not make mistakes. When certainty begins at the points of fixation, the system only produces minor uncertainty around the girl's hairline, a minor error that a human could plausibly make.

formation reproduces the image almost perfectly with only minor error around the woman's hairline (a place where a human observer may also be uncertain). While if the constraint propagation began at the corners of the image, the Chameleon-pixels made substantial errors around the woman's mouth, nose, and other essential features.

4. Conclusion

We conclude that the Chameleon system performs quite well, using constraint-propagation to reproduce a target image, and when given human fixation data, can perform even better. The errors that the system makes when given fixation data are similar to the errors that we might expect a human to make in similar circumstances. This suggests that understanding how a human visual system chooses the points on which it fixates may lend great insight into how an image reproduction or image recognition system might reproduce and recognize objects more efficiently.

5. Contributions

- Integrated methods from distributed robotics, cognitive science, and machine vision to create Chameleon-pixels
- Developed and implemented Chameleon-pixels, a distributed system that allows pixels to determine which hue they should be given a target image and a combination of probability distributions from its neighbors.
- Applied our system to human fixation data, enabling a comparison of the pattern of errors that arise from humans to those generated by the Chameleon-pixel system

6. Deborah's Contributions

- Applied knowledge of eye-tracking and human vision to the distributed system that Ryan and Emily originally proposed.
- Designed the experimental method with equal contribution from other group members.
- Developed the theoretical backing for how we could relate this system to an interesting problem in human and computer vision.
- Interpreted the resulting images in the context of human vision.
- Wrote the text of the final paper. Ryan and Emily's input was essential in determining which figures to use to describe the context of our experiments appropriately.

7. Acknowledgements

We extend our thanks to Prof. Antonio Torralba for his valuable feedback and advice as we developed our project into its current state.

References

[1] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[2] M. Brown and D. G. Lowe. Recognising panoramas. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1218–1225, 2007.

[3] S. M. Courtney, L. Petit, J. M. Maisog, L. G. Ungerleider, and J. V. Haxby. An area specialized for spatial working memory in human frontal cortex. *Science*, 279(5355):1347–1351, 1998.

[4] J. Davenport. Consistency effects between objects in scene processing. *Memory & Cognition*, 35:393–401, 2007.

[5] J. Davenport and M. Potter. Scene consistency in object and background perception. *Psychological Science*, 15:559–564, 2004.

[6] L. Itti and C. Koch. Computational modelling of visual attention. *Nature*, 2(3):194–203, 2001.

[7] T. Judd, F. Durand, and A. Torralba. Fixations on low-resolution images. *Journal of Vision*, 11(4):1–20, 2011.

[8] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[9] B. Kim and M. A. Basso. Saccade target selection in the superior colliculus: A signal detection theory approach. *The Journal of Neuroscience*, 28(12):2991–3007, 2008.

[10] T. Moore and K. M. Armstrong. Aselective gating of visual signals by microstimulation of frontal cortex. *Nature*, 421:370–373, 2003.

[11] A. Oliva and A. Torralba. The role of context in object recognition. *TRENDS in Cognitive Sciences*, 11(12):520–527, 2007.

[12] B. Postle, J. Berger, A. Taich, and M. DÉsposito. Activity in human frontal cortex associated with spatial working memory and saccadic behavior. *Journal of Cognitive Neuroscience*, 12(2):2–14, 2000.

[13] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. In *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[14] D. Sparks, W. Rohrer, and Y. Zhang. The role of the superior colliculus in saccade initiation: a study of express saccades and the gap effect. *Vision Research*, 40:2763–2777, 2000.

[15] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1218–1225, 2001.