A Visual Attention Algorithm Designed for Coupled Oscillator Acceleration

Christopher Thomas Adriana Kovashka Donald Chiarulli Steven Levitan
Department of Computer Science
University of Pittsburgh
{chris, kovashka, don}@cs.pitt.edu

Abstract

We present a new top-down and bottom-up saliency algorithm designed to exploit the capabilities of coupled oscillators: an ultra-low-power, high performance, non-boolean computer architecture designed to serve as a special purpose embedded accelerator for vision applications. To do this, we extend a widely used neuromorphic bottom-up saliency pipeline by introducing a top-down channel which looks for objects of a particular type. The proposed channel relies on a segmentation of the input image to identify exemplar object segments resembling those encountered in training. The channel leverages pre-computed bottom-up feature maps to produce a novel scale-invariant descriptor for each segment with little computational overhead. We also introduce a new technique to automatically determine exemplar segments during training, without the need for annotations per segment. We evaluate our method on both NeoVision2 DARPA challenge datasets, illustrating significant gains in performance compared to all baseline approaches.

1. Introduction

Advances in computer vision have led to a proliferation of vision applications increasingly used by the general public. These advances have coincided with a changing computational landscape, with the majority of the public's computer usage taking place on mobile devices such as smartphones or tablets rather than personal computers [27]. Cameras usually come standard on these platforms, resulting in a greater than ever demand for real-time vision applications. However, the mobile platform poses its own unique challenges for vision developers: significantly less memory is available, central processing units (CPUs) are slower, network activity may be costly and slower, and power is limited. Furthermore, because increases in CPU processing speed require increased power usage, researchers have noted that processing speed is unlikely to significantly increase in mobile devices (due to battery life) and that custom special-purpose units will likely be devised for certain tasks such as vision [40]. A recent example of this phenomenon are the "Myriad" mobile computer vision processors which use architectural design features such as VLIW (very-long instruction word) and numerous cores operating at low frequencies to perform certain vector computations with very low power consumption [3]. Specialized chips may very well be the future of computer vision on mobile devices.

Other researchers pursuing faster and more energy efficient computational mediums turn to the brain for inspiration. The human brain performs the equivalent of 20 petaFLOPS while consuming 20 watts of power. In contrast, the IBM Sequoia supercomputer is capable of computing at 16.3 petaFLOPS but consumes 8 *mega*watts [34]. Just as simulating the brain's computational processes has brought about performance improvements in a number of machine learning tasks through artificial neural networks [22, 38, 25, 37], researchers are attempting to replicate the physical computational structure of the brain (rather than merely simulating it) in the hopes of achieving both energy efficiency and high performance [23, 43]. These *neuromorphic* or *bio-mimetic* computer architectures attempt to replicate how the brain computes, relying on non-boolean, ultra-low-power, massively distributed devices.

One such architecture that can accelerate computer vision tasks involves coupled arrays of nano-oscillators which oscillate at certain frequencies dependent on the supplied voltage [32]. Though a detailed discussion of their function is beyond the scope of this paper, the basic idea is that intensity values of an image can be translated to voltages, each of which is input to an oscillator in a 2D array whose outputs are connected to other nearby oscillators. These connections create a feedback loop which causes neighboring oscillators to synchronize. This process is analogous to patterns of neural firings in the brain and visual cortex [36]. This notion can be extended to arbitrary input vectors (not just images) and the amount of convergence in the oscillations can function as a Degree-of-Match function between the vectors or as a simple classifier [19]. In other words, this means that *we can use the coupling behavior of oscilla-*

tor networks to compute vector distances, segment images, and perform convolutions faster and with much less power than traditional computer hardware. However, new computer vision methods need to be developed that can exploit these advantageous hardware features.

One fundamental computer vision task is *saliency prediction*. Saliency prediction algorithms attempt to predict where in an image a person might look based on what "stands out" (bottom-up influences) and on prior beliefs, knowledge, and goals (top-down effects). Recently, object recognition algorithms have been modified for acceleration on coupled oscillators [10, 19], but these require the identification of regions of the visual field by a saliency algorithm. However, no saliency algorithm designed for oscillatory acceleration currently exists which takes into account top-down information, which has been shown to dominate real-world visual search [11]. We are the first to propose a *visual attention algorithm designed for acceleration on coupled oscillators which considers both bottom-up and top-down effects.*

We make several novel contributions: I.) we address the lack of an oscillatory acceleratable top-down saliency algorithm by developing the first such algorithm which can be entirely accelerated by oscillators; II.) we leverage the information in the bottom-up saliency maps to create a novel, scale-invariant segment descriptor with minimal computational overhead; III.) we are the first to use acceleratable one-class support-vector machines (SVMs) for saliency estimation; and IV.) we develop a new technique for determining representative (exemplar) segments of objects during training using cosegmentation. The novelty of our algorithm is directly related to its target domain as currently no top-down method can be hardware-accelerated, yet research [11] says object search is driven by top-down effects. Our paper fills that need.

If an oscillatory accelerator is available on a mobile phone, algorithms capable of utilizing oscillatory operations as opposed to CPU operations will use less power, run faster, and be able to do more computations with less overhead. Because even CMOS-based accelerators can only accelerate a subset of all CPU operations, specialized algorithms are necessary for maximum performance gain.

The remainder of this paper is structured as follows. In Sec. 2, we survey relevant research. In Sec. 3, we present our proposed oscillatory-correlation-based visual attention system and describe how it is trained. Sec. 4 illustrates how our algorithm, SegSaliency, improves the performance of other commonly used models. Sec. 5 concludes the paper.

2. Related Work

Primates' innate ability to discard uninteresting parts of an image and focus on those of importance is of great interest to the computer vision community because of its widespread applicability to a variety of tasks [6] including object detection [5], tracking [7], recognition [41], video compression [17], and many others. Visual attention research can be broadly categorized into 3 categories: bottom-up algorithms, which find regions of the visual field that "stand out" from the rest of the image; top-down algorithms, which modulate the bottom-up effects based on prior knowledge and current goals; and algorithms which combine both top-down and bottom-up influences [6]. Many neuromorphic models of bottom-up saliency have been proposed, with perhaps the most well known being the Itti-Koch model [18]. This model has been shown to correlate well with human eye-tracking data and has become a standard baseline for evaluating saliency methods [44]. The algorithm pools image intensity values and edge and color detector responses to produce a "saliency map," where the intensity value of each pixel indicates its saliency. Many bottom-up algorithms have extended [18], by pooling additional channels, *e.g*. skin color, horizontal line, gist, flicker, texture, and depth [6].

Top-down attention drives visual search in a goal-oriented manner. While bottom-up saliency is a property of the visual stimulus, top-down saliency depends on the observer's current goals, beliefs, and prior knowledge [6]. Many top-down attention algorithms have been proposed which attempt to model a variety of phenomena (*e.g*., task and context), but the closest to ours are those that focus on object search (salient object detection) [14]. Various attempts have been made to extend the Itti-Koch model in [18] with top-down feedback by learning a channel weighting scheme for each object class [29, 28, 30, 31]. For example, if the system is searching for an apple, weights may be increased on the red color channel causing red objects to appear more salient.

A primary differentiator of our work from these is that we do not represent top-down influences by weighting bottom-up maps. Instead, we leverage the information computed during the bottom-up saliency phase to produce segment descriptors used in classifiers trained to recognize exemplar object segments. The top-down method described in [20] also samples from bottom-up saliency maps (among many other maps) to form a pixel-level descriptor. However, this descriptor also contains dimensions obtained from multiple computationally expensive sliding window object detectors, exactly what our work seeks to avoid. Further, one descriptor is required for every pixel of the image, whereas our system only produces one descriptor *per segment*, significantly reducing the number of classifications and distance computations necessary. Additionally, the purpose of [20] is to predict human eye saccades whereas we perform coarse object search.

Recent work [10, 19] shows how the neuromorphic "Hierarchical Model and X" (HMAX) object recognition al-

Operation	Supported By Oscillators
Low-Dimensional Convolutions	Y [2]
High-Dimensional Convolutions	N
Image Segmentation	Y [42]
Nearest Neighbor Classifier	Y [19]
SVMs	N
Vector Degree of Match	Y [10]

Table 1: Common computational operations in saliency algorithms. Our algorithm uses the operations shown in bold.

gorithm can be modified for oscillatory acceleration by using the previously discussed Degree-of-Match behavior to calculate distance norms and function as a classifier. Both studies note, however, that their algorithms comprise the "back-end" of an image processing pipeline which requires subregions of the image to be selected by an object search saliency algorithm like SegSaliency. As our experiments demonstrate, the inclusion of top-down information substantially improves performance for object search over purely bottom-up approaches.

Previous approaches have used oscillators to accelerate object search [30, 33], but all of these works consider only bottom-up saliency and ignore top-down effects. To the best of our knowledge, our algorithm is the first that considers top-down saliency designed for oscillatory acceleration. The method presented in [33] is perhaps the closest to ours. Their technique extends the standard Itti-Koch pipeline by producing a segmentation of the input image in order to reduce cases where the output bounding box from the saliency map does not fully cover the salient object, which could hamper an object recognizer. Unlike our algorithm, this method provides no mechanism to tune the visual search towards objects of interest and instead relies solely on bottom-up saliency.

We provide a look at the computational capabilities of coupled oscillators in the context of saliency algorithms in Tab. 1. We show the operations used by our approach in bold. Arrays of coupled oscillators can perform low dimensional convolutions, such as the Gabor filters in the bottom-up saliency phase of our algorithm, can perform image segmentations, compute vector degrees of match, and serve as nearest neighbor classifiers. These are the only computational operations which oscillator arrays have been shown capable of.

Many top-performing algorithms on the MIT 300 saliency benchmark [8] are deep convolutional neural networks, but these models require high dimensional convolutions and mathematical functions such as hyperbolic tangent, sigmoid, and logit which are not acceleratable using oscillators. Additionally, due to their high memory demands, they are not good candidates for real-time acceleration on mobile platforms. [20], another top saliency method, relies on SVMs, which cannot be accelerated by oscillators. In contrast, our simple one-class SVM outputs a degree of match between an input vector and a vector

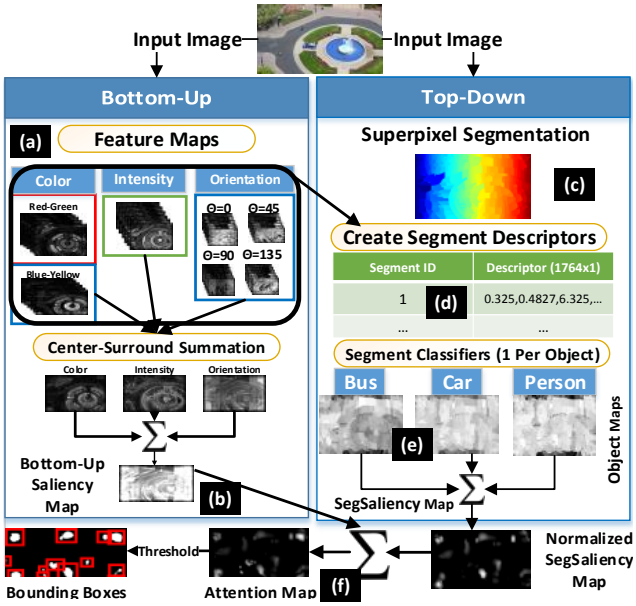

Figure 1: Itti-Koch saliency pipeline enhanced with an additional top-down channel. Bottom-up saliency features are reused to create descriptors for each segment to create top-down object maps. The top-down and bottom-up maps are combined to form an attention map.

learned to be a representative of the target class. The degree of match calculation can be accelerated using oscillators [10].

3. Approach

The purpose of our algorithm is to serve as an object detector, producing bounding boxes that likely contain objects of interest for further processing by an object recognizer. We provide an illustration of our algorithm in Fig. 1. The Itti-Koch bottom-up pipeline [18] serves as the base of our approach. In order to create the bottom-up saliency map, the Itti-Koch model creates multiple intermediate "feature maps" to discover regions which have high intensities or contrasting orientations and colors (Fig. 1(a)). Our algorithm begins by computing the bottom-up saliency map of [18] (Fig. 1(b)) and preserves the intermediate feature maps for later re-use. Next, our algorithm performs an oversegmentation of the original input image (Fig. 1(c)). Using the previously computed feature maps, a segment descriptor is produced for each segment (Fig. 1(d)). Each segment is then classified by the classifier for each object of interest and assigned a saliency value for that object (Fig. 1(e)). These bottom-up and top-down outputs are combined to form the final attention map (Fig. 1(f)). Because segmentation is such a core feature of our algorithm, we call it *SegSaliency*.

3.1. Bottom-up Saliency Architecture

The Itti-Koch bottom-up phase consists of mostly low-dimensional convolutions which can be accelerated with os-

cillators [2], so it is an appropriate basis for our method. Recent research has shown that it continues to achieve state-of-the-art performance for salient object segmentation, even outperforming some top-down methods [13].

The first step in the computation of bottom-up saliency is the creation of multiple Gaussian pyramids of feature detector responses. The three features covered by the Itti-Koch model are color, intensity, and orientation and were chosen because they are known to have direct correlates in the early primate vision system [18]. The color channel is further subdivided into two sub-channels, designed to mimic biological color opponencies. We denote these by F_{RG} for Red-Green, and F_{BY} for Blue-Yellow. The F_{RG} and F_{BY} feature pyramids are constructed by creating a Gaussian pyramid of the original image and applying the following transformations on a pixel by pixel basis:

$$F_{RG} = \frac{r-g}{\max(r,g,b)}, \qquad F_{BY} = \frac{b-\min(r,g)}{\max(r,g,b)} \tag{1}$$

Similarly, the intensity pyramid is constructed using the formula $F_I = (r+g+b)/3$, where r, g, b are pixel values from the red, green, and blue color channels, respectively. Finally, oriented Gabor filters at $\theta \in \{0°, 45°, 90°, 135°\}$ are passed over the image pyramid to produce four orientation pyramids used to discover contrasting orientations.

The next phase of the process simulates the center-surround mechanism of retinal ganglial cells. The purpose of this process is to amplify the salience of regions (the center) which contrast with their surroundings (the surround) while muffling those which do not. This is done by subtracting coarser-grained levels in the Gaussian pyramid (whose pixels represent a larger area of the original image) from finer-grained levels. The center takes on levels $c \in \{2, 3, 4\}$ and the surround is at levels $s = c + \delta$, where $\delta \in \{3, 4\}$. The center-surround differences operation is denoted by the symbol $\ominus$. This procedure produces six *feature maps* for each type, denoted as M_{type} for each feature type. These feature maps are later used again by our top-down channel, so they are preserved after computation. The feature maps are produced by the following formula, where $N(.)$ represents an iterative normalization routine designed to suppress uninteresting noise within each map:

$$M_{\text{type}}(c,s) = N(|F_{\text{type}}(c) \ominus F_{\text{type}}(s)|) \tag{2}$$

The above process results in forty-two maps (twelve for color, six for intensity, and 24 for orientation). The feature pyramids are then combined into three "conspicuity maps" which represent the feature's response across all scales of the pyramid. The orientation conspicuity maps are created by performing across scale addition for each orientation and then summing all orientations. The overall bottom-up saliency map S is created by taking a normalized average of the three conspicuity maps. See [18] for more details.

3.2. Top-Down Attention Modulation

The bottom-up algorithm just described provides no mechanism for biasing the search towards objects of interest. In order to achieve this, we introduce a learning phase to learn a profile of how target objects appear in the bottom-up feature maps. As with [33], we use segments as the base unit of attention. At a high level, our top-down channel works by assigning each segment a saliency score based on how similar the segment is to each learned object model. Using these segment scores, we are able to produce top-down object maps showing how strongly each segment responds to each object class. Much as the bottom-up saliency algorithm combined multiple feature maps, our algorithm combines all object maps to form a single top-down map. This top-down map is then combined with the bottom-up saliency map to produce the overall attention map. See Fig. 1 for an illustration of this process.

The channel begins by performing a segmentation of the input image using a simplified version of the approach described in [1]. The granularity of the segmentation is a tunable parameter of the algorithm and will depend on the desired size of the target objects in the visual field, the distance of the camera to the objects, etc. Automatically tuning this parameter based on scene analysis is conceivable, but for this study we set it to a constant. Any segmentation algorithm capable of segmenting an image using coupled oscillators, such as [33, 12], could be used for this process.

The next phase of the process is the computation of a descriptor for each of the segments produced during the first phase. The segment descriptor is computed according to the following formula, where SD_i^j is the segment descriptor for segment i in image j, $\widehat{FM}_{type}$ are the feature maps produced during bottom-up saliency (rescaled to the size of the original image using a Laplacian interpolation), x, y are the pixel coordinates of each pixel in the segment, c, s are defined as in Eq. 2, and $\odot$ is the vector concatenation operator (note that the parameter θ, the orientation of the Gabor filters, does not apply to the color and intensity maps):

$$SD_i^j = \bigodot_{c=2}^{4} \bigodot_{s=c+3}^{c+4} \bigodot_{t\in\{C,I,O\}} \bigodot_{\theta=0°}^{130°} \widehat{FM}_t(c,s,\theta)(x,y) \tag{3}$$

Because there are 42 feature maps after center-surround differences, each segment descriptor is of size $42n$, where n is the number of pixels in the segment. Since segments are of unequal size, we reduce the dimensionality of the segment descriptor by performing principal component analysis (PCA) to obtain the 1764 (42^2) dimensional segment descriptor used in our algorithm.

So far, we have shown how the bottom-up feature maps computed by the Itti-Koch algorithm can be paired with a segmentation of the image to form a descriptor for each segment. The next part of our algorithm uses this descriptor

to determine how close each segment is to each object of interest.

In the next phase, each segment descriptor is classified by one-class SVM [35] classifiers to produce a saliency score for each segment. These classifiers can be accelerated similar to the oscillator accelerated nearest-neighbor classifiers of [10, 19]. More details on how these are trained is given in Sec. 3.3. The classifiers attempt to identify segment descriptors matching certain *exemplar segments*, or segments common to many instances of the same class. Exemplar segments can be, for instance, the wheels of a car, the spokes of a bicycle, *etc*. The more positive the response of a classifier is, the better the segment fits with the known exemplar segments for that object class. Each pixel position covered by the segment is then filled with the score from the classifier. These filled segments serve as the object-based top-down bias of the bottom-up saliency map.

We have discussed how a top-down object detection map can be created and filled with scores from the classifier. Our training procedure (Sec. 3.3) creates one classifier per object of interest, which allows each classifier to focus on learning exemplar segments individually for each object. Because there are multiple classifiers, we create one object detection map for each object of interest by classifying every segment by every classifier. For example, a segment which is the wheel of a car should have a high saliency value in the "car" object map but a lower value in the "person" map. The value at each pixel position is a measure of the strength of how well the segment to which the pixel belongs matches the learned object model. Because some object classes are much more frequent than others, the exemplar learning procedure in Sec. 3.3 will produce higher values for some classes than others, which in turn will cause the distribution of the values produced by each classifier to be different. In other words, the "car" classifier's output value for a strong car detection may be much higher than the "truck" classifier's score for a strong truck detection. To prevent objects with high value distributions from dominating those with lower distributions, each object feature map is normalized using the iterative normalization technique proposed by [18], resulting in crisp, focused maps to mitigate the effects of uncertain predictions. The object maps are then summed and normalized to produce the SegSaliency map. Just as the bottom-up saliency algorithm combined multiple feature maps to produce a single saliency map, our SegSaliency map combines all object maps to produce a single top-down map.

At this point, the system has two maps: the bottom-up saliency map and the top-down map. To combine both bottom-up and top-down effects in a single map, the SegSaliency map is added to the bottom-up map and an additional normalization is applied. We call this map combining top-down and bottom-up methods the *attention map*.

All that remains for the algorithm to do is to extract object bounding boxes from the attention map. A thresholding procedure is applied to convert the 2D intensity image that represents the attention map into multiple rectangular bounding boxes believed to contain objects of interest. We replace the winner-take-all neural network provided by the Itti-Koch algorithm with a simpler, more efficient mechanism. The original fixation prediction routine outputs fixed-size circular regions from the image. However, for larger objects or rectangular objects, the entire object may not be captured in one fixation, and it is unlikely that only the object of interest will be in the fixation region because the circular region size is fixed.

To address both of these problems, we again use a segmentation which can be performed by oscillators. A *coarse* segmentation of the attention map is first performed. Each segment is then enclosed by the smallest possible rectangular bounding box. Bounding boxes which are deemed too large by a tunable parameter are discarded and recursively segmented at finer granularities. The average salience of each box is determined and boxes above a threshold are output by the algorithm. This pipeline attempts to ensure that the regions output by the algorithm capture the entire object with minimal background noise.

3.3. Exemplar Segment Learning

To perform the top-down biasing of the visual search (Sec. 3.2), we need classifiers trained to recognize segments from target objects. Training the classifiers used by the top-down saliency algorithm to recognize exemplar segments requires first finding exemplar segments for each object of interest. Our training dataset consists of images with bounding boxes drawn around each object of interest. Because the object annotations in our dataset are not at the segment level, segments containing background and even other objects of interest inevitably leak into the annotations. For example, in the NeoVision2 DARPA dataset used for this study we observed that many segments of trees, people, and the road also appear in the boxes humans drew and labeled "car."

Because these background segments cause confusion by our classifiers, we need to remove segments contained within the human annotations which are not part of the actual object. To solve this problem, we use a novel supervised cosegmentation procedure to find discriminative "exemplar" segments shared over many instances of each object class. Since training is done offline, power usage and performance are not critical issues. Thus, it is not necessary to constrain our approach to those amenable to oscillatory acceleration. While a number of techniques exist to perform unsupervised cosegmentation [39], the dataset used in this study is highly cluttered and contains extraneous objects which appear in every image. In such a situation, it

is impossible to distinguish objects of interest from other objects without some degree of supervision.

Our procedure starts by picking a human-drawn bounding box around an object from the training set. Segments within this bounding box are matched with segments within bounding boxes of the same object class from other images. We do this to determine which segments lying within the original bounding box are most representative of the object. For example, for the class "car", one would expect the segments containing tires to match other annotations of cars more frequently than do background segments. To perform the matching between segments, we compute a speeded-up robust features (SURF) [4] descriptor for interest points found in each bounding box. SURF is a classic scale-invariant, local neighborhood feature descriptor. We associate each segment with zero or more SURF descriptors that lie within it.

We then scan the dataset and locate other images containing bounding boxes of the same class. Note that the latter do not need to be segmented. We then match the segmented image's descriptors to all other annotated patches using the technique described in [26]. Every time one of the SURF descriptors associated with a segment matches a SURF descriptor in another bounding box of the same class, a counter associated with the segment is increased by 1. These counts enable us to determine which segments have matched most with other annotations of the object, hence which segments are most representative of the class of interest. Segments with lower counts are either background or are not representative of the object class.

At this point, the algorithm knows how often each segment in the bounding box annotation matched with other object instances. We compute a segment descriptor for each segment in the annotation using the approach described in Section 3.2. With each segment descriptor, we also store the number of matches that segment had. These counts will be used later to remove the descriptors associated with low-performing segments. This process of segmenting an annotation and matching it against other annotations is performed repeatedly over thousands of object instances. When this process finishes, a large list of segment descriptors and counts is obtained.

To weed out segments which are not representative of the object, we perform a k-means clustering on the stored counts. We chose $k = 2$ based on our observation that this procedure tends to produce a large number of segments with low counts (non-representative or background segments) and a few segments with very high counts (the exemplars). In other words, the clustering procedure splits the highest-performing segments and the poorly-performing segments. The cluster with the higher-valued count centroid is the cluster of the exemplar counts, and segments in the other cluster are discarded. While more advanced techniques (such as per-cluster weighting schemes) could be used rather than k-means for discarding background segments, we found k-means performed well in practice. Using these segment descriptors, we train a one-class SVM classifier [35] with a radial basis function (RBF) kernel and weight each segment descriptor with its count. Thus, segment descriptors whose segments were matched the most (better exemplars) have the most weight in training the classifier.

The entire training procedure is repeated for each object class in the training set, creating one classifier for each known object type. The classifiers produced by this process are used to assign segments a top-down saliency value at runtime. These values are used to create the object maps which provide the top-down bias towards objects of interest in our algorithm. We qualitatively observed a large difference in the quality of the segments in our training data when we implemented the cosegmentation-based training.

4. Evaluation

We used the entire DARPA Neovision2[1] dataset to evaluate our model. The Neovision2 dataset is a standard benchmark dataset for evaluating neuromorphic object detection, recognition, and tracking algorithms [21, 9, 24] and is comprised of two separate collections: Tower and Heli. Ground truth annotations are provided with the dataset for ten object types: cars, trucks, tractor-trailers, buses, containers, boats, planes, helicopters, people, and cyclists. Each object is annotated with a tight polygonal bounding box.

The Tower dataset was acquired from a fixed camera on top of the Stanford Hoover tower and consists of variable lighting conditions (sunny and overcast). Additionally, the camera is rotated to the side in some videos, introducing rotational variance into the training and test sets. Because the camera is high on the building, many objects below are quite small. Further, the images are extremely cluttered, with light poles, park benches, a water fountain, and other distractions present. Only cars, trucks, buses, people, and cyclists appear in the Tower dataset.

The Heli dataset was acquired from a helicopter flying over the Los Angeles metro area. The data represents a variety of different settings, including the ocean, beaches, freeways, airports, train stations, etc. The data is highly variable from image to image and contains all ten object classes in a variety of settings and orientations. As such, it is a more realistic and challenging.

To simulate realistic operating conditions, we merged the Heli and Tower datasets together into one large training and test set (rather than developing two distinct top-down models for each), which greatly increased the experiment's difficulty. The training and test sets are disjoint and do not

[1] Available online at: `http://ilab.usc.edu/neo2/`, accessed 2014.

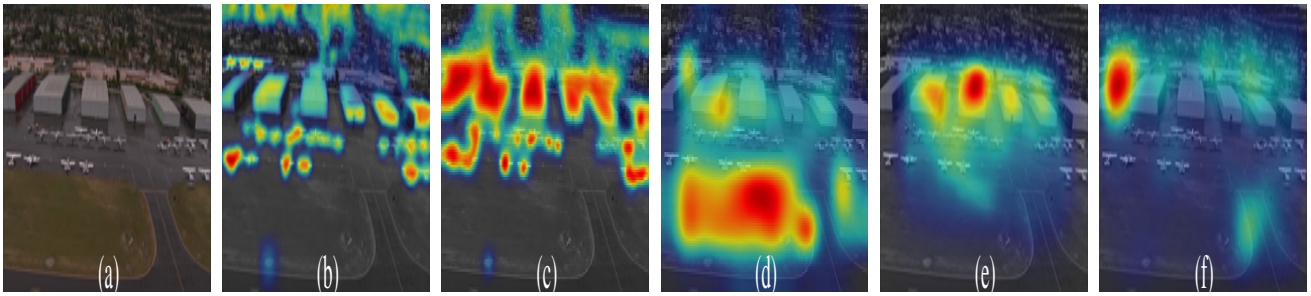

Figure 2: A variety of saliency maps computed on an image from our dataset. Airplanes are a target object class in our dataset. The ground truth of this image would have the airplanes covered with a high saliency value (in red) and little else covered (most similar to (b)). (a) Original Image (b) SegSaliency Object Map (looks for *all* target objects) (c) SegSaliency Airplane Object Map (looks for airplanes) (d) Itti-Koch [18] (e) Graph Based Visual Saliency [15] (f) Signature Saliency [16]

share even the same capture locations (except for Tower, in which the camera is at a fixed location and rotated). Because many target objects appear very small in some images, we set the single segmentation threshold of [1] to 0.01 in order to produce a very fine-grained segmentation so that even small objects are composed of multiple segments. Our training and evaluation sets are comprised of approximately 40,000 images each. Due to the large size of the Neovision2 dataset, we sampled one image from every 20 images from the training and evaluation sets (provided by the dataset) to form our train and test sets. In practice, this amounts to sampling approximately one image per second of video. Note that because the camera position frequently changes in both data sets (it is explicitly flipped in the Tower dataset), rotation and scale variance are implicit in the data.

We performed experiments to test how well our algorithm, SegSaliency (SS), performs object detection, and compared it to several unguided bottom-up search methods: Graph-Based Visual Saliency (GBVS) [15], Itti-Koch [18], and Signature Saliency (SigSal) [16]. Examples of saliency maps produced by these algorithms are shown in Fig. 2. We performed two sets of experiments, one designed to test the algorithms on unmodified test data, and a second designed to test the scale-invariance of our method by randomly resizing the same input data while keeping the training data constant. We combined SegSaliency with each bottom-up method by normalizing and adding SS's top-down attention map to the bottom-up saliency map produced by each method. Each method's final attention map was then used to output a number of bounding boxes around regions calculated to be salient using the method described in Sec. 3.2.

Using the bounding boxes produced by each algorithm, we then determined whether each box overlaps with the human-provided bounding boxes. We computed the standard intersection over union metric between the output bounding box and each manually annotated region in the image. If the overlap exceeded a fixed region-of-interest (ROI) threshold noted below, the output was considered a correct detection of that object. Bounding boxes which did not contain any ground truth annotations above the ROI threshold parameter were considered false positives. Using the boxes output by each algorithm, we also determined which manually annotated objects in each image were found (true positives) and which were missed (false negatives). We swept the ROI threshold at 20 positions between 0.4 (40% overlap) and 1 (100% overlap) on the X-axis in Figs. 3, 4, and 5. Our cutoff choice of 0.4 for Figs. 3 and 4 was based on our observation that bounding boxes which had less than 40% overlap contained so little of the target object that they were essentially useless. Due to space limitations, we only present results at the four closest to equally spaced ROI threshold positions we tested in Figs. 3 and 4. The omitted values are intermediates between these positions and follow the same general pattern (as can be seen in Fig. 5).

Figs. 3 and 4 show each algorithm's F1 score as a function of the ROI-threshold parameter on the test set. The F1 score is a weighted average of precision and recall. Fig. 3 shows the results of the first set of experiments on the unmodified data set. Fig. 4 shows the results of the second experiment, where each test image is randomly resized to either half or double its original size. In each chart, SS refers to the top-down map produced by our algorithm and used to complement the bottom-up maps. We first show the performance of each algorithm without modification. Next, we use SegSaliency's top-down modulation to bias the search of the bottom-up map and show the results next to the original. The introduction of our top-down channel always improves performance over the original bottom-up algorithms (compare the F1-score of each algorithm to the algorithm's name+SS). Despite the random resizing of the test images, Fig. 4 shows that our technique still significantly improves the performance of the bottom-up algorithm.

One possible explanation for the decrease in performance observed on the resized dataset (which affected all algorithms) compared to the non-resized dataset is that the parameters of the bounding box extraction process described in Sec. 3.2 were kept constant rather than adjusted to account for possible changes in scale of the objects. Another possibility is that the bottom-up saliency algorithms had difficulty finding salient regions in the resized images and needed to have their parameters tuned (*i.e.* wider or

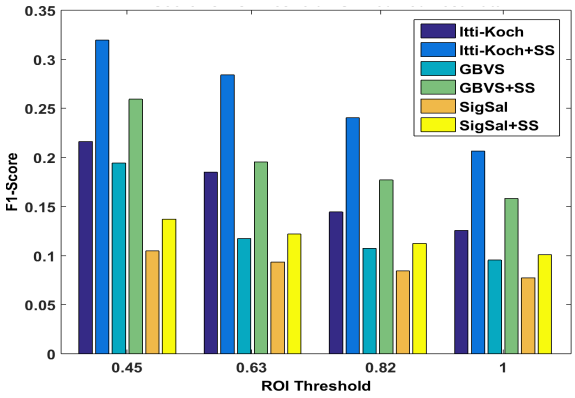

Figure 3: F1-Scores for unmodified dataset

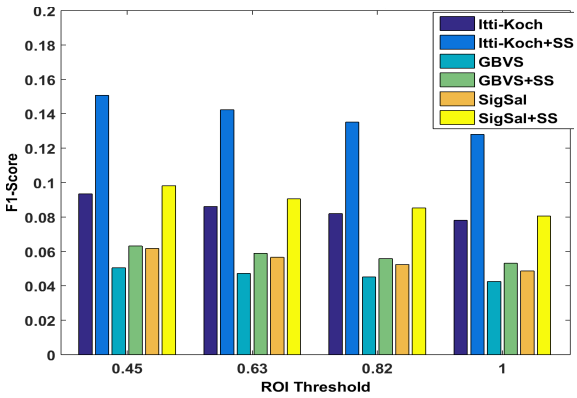

Figure 4: F1-Scores for randomly resized dataset

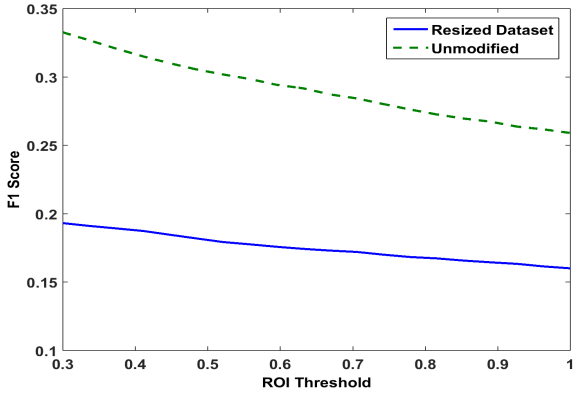

Figure 5: SegSaliency without bottom-up information

smaller filters) to produce consistent results to reflect scale differences. In all cases, SegSaliency still results in significant performance gains on both tests and never decreases the performance of the bottom-up algorithms.

In Fig. 5, we illustrate the performance of the top-down attention map when used *without* a bottom-up saliency map, on the two test sets. We first observe that the performance on the resized set is somewhat worse than on the unmodified test set. However, when SegSaliency's F1 scores on the resized test set (the blue line) are compared to the scores in Fig. 4, one notices that they are still higher than the best results obtained by combining the bottom-up and top-down map. In other words, top-down influences are sufficient to capture attention. We also observe that SegSaliency's F1 score (the green line) slightly outperforms the best combined approach in Fig. 3. One may ask, why bother combining top-down and bottom-up at all? The answer to this question is probably largely dataset dependent. Top-down saliency will not always outperform bottom-up saliency because not all datasets have the same properties, see [13]. Because the NeoVision2 dataset is an extremely cluttered dataset, bottom-up saliency returns numerous visually interesting regions which are not objects of interest. SegSaliency on the other hand, has the benefit of having been trained to look for those objects and can bias the search. This eliminates some unnecessary visual search while also finding objects which may not have originally appeared salient. For instance, objects with highly variable appearance could be missed by the top-down channel, only to be found by the bottom-up channel. In such situations, combining both top-down and bottom-up effects may yield superior results.

5. Conclusion

In this paper, we presented SegSaliency, a visual attention system designed for coupled oscillator acceleration. Our algorithm extends a widely used bottom-up saliency algorithm by providing top-down modulation and object-based binding via segmentation, making it a useful candidate as a front-end to an object recognition pipeline. Specifically, we extend the pipeline presented in [18] by adding a new top-down channel. We have shown how the feature maps computed by bottom-up saliency can be leveraged along with an image segmentation to determine segment-based top-down biases with little computational overhead. Additionally, a novel technique to identify these exemplar object segments using only bounding boxes was proposed. We demonstrated the benefits of our techniques through evaluation on a cluttered and challenging dataset.

Note that unlike existing saliency methods, ours can be fully accelerated using oscillators, so it can be used to perform efficient object search in embedded settings. Despite our algorithm's relative simplicity, SegSaliency's deployment resulted in substantial gains in performance when bootstrapped to a number of saliency pipelines, illustrating the vital role top-down attention plays in making sense of our cluttered visual world.

In our future work, we will explore the novel idea of dynamically tuning the filters used to produce the bottom-up map based on learned object models. For example, regions containing a suspected car could have their orientation filters adjusted to those known to respond strongly to cars. In this scenario, top-down biases would be implicitly expressed in the bottom-up saliency map.

Acknowledgment: This material is based upon work supported by the National Science Foundation under award number CCF-1317373.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):898–916, 2011. 4, 7

[2] P. Baldi and R. Meir. Computing with arrays of coupled oscillators: An application to preattentive texture discrimination. *Neural Computation*, 2(4):458–471, 1990. 3

[3] B. Barry, C. Brick, F. Connor, D. Donohoe, D. Moloney, R. Richmond, M. O'Riordan, and V. Toma. Always-on vision processing unit for mobile applications. *IEEE Micro*, (2):56–66, 2015. 1

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, 2006. 6

[5] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *Image Processing, IEEE Transactions on*, 24(12):5706–5722, 2015. 2

[6] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):185–207, 2013. 2

[7] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan. Visual saliency model for robot cameras. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2398–2403. IEEE, 2008. 2

[8] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. Mit saliency benchmark. 3

[9] Y. Cao, Y. Chen, and D. Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015. 6

[10] J. A. Carpenter, Y. Fang, C. N. Gnegy, D. M. Chiarulli, and S. P. Levitan. An image processing pipeline using coupled oscillators. In *Cellular Nanoscale Networks and their Applications (CNNA), 2014 14th International Workshop on*, pages 1–2. IEEE, 2014. 2, 3, 5

[11] X. Chen and G. J. Zelinsky. Real-world visual search is dominated by top-down guidance. *Vision research*, 46(24):4118–4133, 2006. 2

[12] Y. Fang, M. J. Cotter, D. M. Chiarulli, and S. P. Levitan. Image segmentation using frequency locking of coupled oscillators. In *Cellular Nanoscale Networks and their Applications (CNNA), 2014 14th International Workshop on*, pages 1–2. IEEE, 2014. 4

[13] S. Frintrop, T. Werner, and G. M. García. Traditional saliency reloaded: A good old model in new shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 82–90, 2015. 4, 8

[14] A. Furnari, G. M. Farinella, and S. Battiato. An experimental analysis of saliency detection with respect to three saliency levels. In *European Conference on Computer Vision Workshops (ECCVW) 2014 Workshops*, pages 806–821. Springer, 2014. 2

[15] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Neural Information Processing Systems (NIPS)*, 2006. 7

[16] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(1):194–201, 2012. 7

[17] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *Image Processing, IEEE Transactions on*, 13(10):1304–1318, 2004. 2

[18] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(11):1254–1259, 1998. 2, 3, 4, 5, 7, 8

[19] B. Jennings, R. Barnett, C. Gnegy, J. Carpenter, Y. Fang, D. Chiarulli, and S. Levitan. Hmax image processing pipeline with coupled oscillator acceleration. *IEEE Workshop on Signal Processing Systems*, 2014. 1, 2, 3, 5

[20] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *IEEE 12th international Conference on Computer Vision (ICCV)*, pages 2106–2113. IEEE, 2009. 2, 3

[21] R. Kasturi, D. B. Goldgof, R. Ekambaram, R. Sharma, G. Pratt, M. Anderson, M. Peot, M. Aguilar, E. Krotkov, D. Hackett, et al. Performance evaluation of neuromorphic-vision object recognition algorithms. In *International Conference on Pattern Recognition*, 2014. 6

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012. 1

[23] D. Kuzum, R. Jeyasingh, S. Yu, and H.-S. Wong. Low-energy robust neuromorphic computation using synaptic devices. *Electron Devices, IEEE Transactions on*, 59(12):3489–3494, Dec 2012. 1

[24] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014. 6

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*, 2013. 1

[26] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, pages 331–340, 2009. 6

[27] R. Murtagh. Mobile now exceeds pc: The biggest shift since the internet began. 1

[28] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision research*, 45(2):205–231, 2005. 2

[29] V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 2

[30] V. Navalpakkam and L. Itti. Top–down attention selection is fine grained. *Journal of Vision*, 6(11):4, 2006. 2, 3

[31] V. Navalpakkam and L. Itti. Search goal tunes visual features optimally. *Neuron*, 53(4):605–617, 2007. 2

[32] D. E. Nikonov, I. A. Young, and G. I. Bourianoff. Convolutional networks for image processing by coupled oscillator arrays. *arXiv preprint arXiv:1409.4469*, 2014. 1

[33] M. G. Quiles, D. Wang, L. Zhao, R. A. F. Romero, and D.-S. Huang. An oscillatory correlation model of object-based attention. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 2596–2602. IEEE, 2009. 3, 4

[34] B. Rajendran. Embedded tutorial-can silicon machines match the efficiency of the human brain? In *International Conference on Embedded Systems*, 2013. 1

[35] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. 5, 6

[36] H. Schuster and P. Wagner. A model for neuronal oscillations in the visual cortex. *Biological cybernetics*, 64(1):77–82, 1990. 1

[37] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems (NIPS)*, 2014. 1

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[39] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[40] D. Wagner and D. Schmalstieg. Making augmented reality practical on mobile phones, part 2. *Computer Graphics and Applications, IEEE*, 29(4):6–9, July 2009. 1

[41] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural networks*, 19(9):1395–1407, 2006. 2

[42] D. Wang and D. Terman. Image segmentation based on oscillatory correlation. *Neural Computation*, 9(4):805–836, 1997. 3

[43] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. Wong. An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *Electron Devices, IEEE Transactions on*, 58(8):2729–2737, Aug 2011. 1

[44] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32, 2008. 2