Combining attention and recognition for rapid scene analysis

J.J. Bonaiuto & L. Itti

Neuroscience Department
University of Southern California
Los Angeles, CA 90089

Abstract

Bottom-up visual attention allows primates to quickly select regions of an image that contain salient objects. In artificial systems, restricting the task of object recognition to these regions allows faster recognition and unsupervised learning of multiple objects in cluttered scenes. A problem is that objects superficially dissimilar to the target are given the same consideration in recognition as similar objects. Here we investigate rapid pruning of the recognition search space using the already-computed low-level features that guide attention. Itti & Koch's bottom-up visual attention algorithm selects salient locations based on low-level features such as contrast, orientation, color, and intensity. Lowe's SIFT recognition algorithm then extracts a signature of the attended object, for comparison with the object database. The database search is prioritized for objects which better match the low-level features used to guide attention to the current candidate for recognition. The SIFT signatures of prioritized database objects are then checked for match against the attended candidate. By comparing performance of Lowe's recognition algorithm and Itti & Koch's bottom-up attention model with or without search space pruning, we demonstrate that our pruning approach improves the speed of object recognition in complex natural scenes.

1. Introduction

Bottom-up visual attention is the process by which primates quickly select regions of an image that contain salient or conspicuous objects [**?**]. Previous models of this process include spatial attention with saliency maps [**?**, **?**], object-based attention [**?**], and modulation of visual processing mechanisms [**?**]. In artificial systems, restricting the task of object recognition to these regions allows faster recognition and unsupervised learning of multiple objects in cluttered scenes [**?**, **?**, **?**]. It has been shown that the saliency of target objects can be increased by biasing attention to the target's known low-level features, allowing them to be quickly and reliably detected [**?**]. This biasing typically consists of modulating the relative weights of different features like color, orientation and intensity in guiding attention. Attention biasing allows animals to restrict their search space to only candidate locations which resemble the desired target. For example, a stop sign might be found quicker by restricting the search to only red objects [**?**].

Here we investigate whether the computational mechanisms which enable attention biasing can also serve to dynamically prune object search space in recognition. Dynamic pruning of complex search spaces has been proved to improve processing rate without a corresponding loss in accuracy in speech recognition [**?**]. In the visual domain, object recognition search space has been shown to be effectively pruned by using attention as a cropping mechanism [**?**, **?**, **?**]. Here we investigate the value of further pruning of the recognition search space by using the already-computed low-level features that guide attention.

2. Approach

2.1. Attention

Itti & Koch's saliency-based attention system is used to select highly salient points and pre-attentive, low-level feature descriptors for these points. Here we present a summary of the published description of this model (for implementation details, see [**?**, **?**]). Salient points are identified by computation of seven center-surround features: image intensity contrast, red/green and blue/yellow double opponent channels, and four orientation contrasts. Center-surround operations (denoted "Θ" below) are implemented as a difference between the image at two scales: the image is first interpolated to the finer scale, then subtracted point-by-point from the image at the previous scale.

The intensity channel, I, is obtained by averaging the red, green, and blue color channels as $I = (r + g + b)/3$. A Gaussian pyramid, $I(\sigma)$, is created from I with $\sigma \in [0..8]$ as the scale. Four color channels are computed: red $R = r - (g+b)/2$, green $G = g-(r+b)/2$, blue $B = b-(r+g)/2$, and yellow $Y = (r+g)/2-|r-g|/2-b$. These channels are used to create four Gaussian pyramids $R(\sigma)$, $G(\sigma)$, $B(\sigma)$, and $Y(\sigma)$. The orientation channels are obtained from I

using oriented Gabor pyramids $O(\sigma, \theta)$, where $\sigma \in [0..8]$ is the scale, and $\theta \in \{0^o, 45^o, 90^o, 135^o\}$ is the preferred orientation.

The feature maps are computed using center-surround differences (Θ) between a fine (center) scale c and a coarser (surround) scale s. Six different pairs of center and surround spatial scales are used to compute the intensity feature maps $\mathcal{I}(c, s)$, with $c \in \{2, 3, 4\}$ and $s = c + \delta, \delta \in \{3, 4\}$:

$$\mathcal{I}(c, s) = |I(c)\Theta I(s)| \tag{1}$$

The color channels are used to construct double-opponent color feature maps. The maps $\mathcal{RG}(c, s)$ account for red/green and green/red double opponency and $\mathcal{BY}(c, s)$ for blue/yellow and yellow/blue opponency:

$$\mathcal{RG}(c, s) = |(R(c) - G(c))\Theta(G(s) - R(s))| \tag{2}$$

$$\mathcal{BY}(c, s) = |(B(c) - Y(c))\Theta(Y(s) - B(s))| \tag{3}$$

Orientation feature maps, $\mathcal{O}(c, s, \theta)$, represent local orientation contrast between the center and surround scales:

$$\mathcal{O}(c, s, \theta) = |O(c, \theta)\Theta O(s, \theta)| \tag{4}$$

A total of 42 feature maps are computed: six for intensity, 12 for color, and 24 for orientation.

A map normalization operator, $\mathcal{N}(.)$, is used to promote maps in with a small number of strong activity peaks, while suppressing maps with numerous similar activity peaks. The operator $\mathcal{N}(.)$ consists of three steps:

1. normalize the map values to a fixed range $[0..M]$;
2. find the location of the map's global maximum M and compute the average $\bar{m}$ of all other local maxima; and
3. globally multiply the map by $(M - \bar{m})^2$.

The feature maps are then combined into three conspicuity maps, intensity $\bar{I}$, color $\bar{C}$, and orientation $\bar{O}$, at the saliency map's scale ($\sigma = 4$). These maps are computed through across-scale addition, "$\bigoplus$", where each map is reduced to scale four and added point-by-point:

$$\bar{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{I}(c, s)) \tag{5}$$

$$\bar{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(\mathcal{RG}(c, s)) + \mathcal{N}(\mathcal{BY}(c, s))] \tag{6}$$

To compute the orientation conspicuity map, four intermediary maps are created by combining the six feature maps for a given θ. These intermediary maps are then combined into a single orientation conspicuity map:

$$\bar{O} = \sum_{\theta \in \{0^o, 45^o, 90^o, 135^o\}} \mathcal{N}\Big(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{O}(c, s, \theta))\Big) \tag{7}$$

The three conspicuity maps are then normalized and summed into the input S to the saliency map:

$$S = \frac{1}{3}(\mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O})) \tag{8}$$

The saliency map maximum defines the most salient image location, to which attention should be directed to, at any point in time. The saliency map is modeled a two-dimensional layer of leaky integrator neurons, consisting of a single capacitance which integrates the synaptic input charge, a leakage conductance, and a voltage threshold. The saliency map is input into a two-dimensional winner-take-all (WTA) network, where lateral inhibition ensures that all units are silenced except for the most active one. This most active unit guides attention by providing the fixation point for recognition. Local inhibition is also transiently triggered in the saliency map at the current fixation point, allowing the next most salient point to become the winner in the next attention shift.

The low-level feature descriptors are 42-dimensional vectors obtained from sampling the already computed feature maps at the currently attended point [**?**]. These feature descriptors serve to provide a representative description (red/green, blue/yellow, intensity, and four orientations at six center-surround scales) of a highly salient region of an object.

Walther et al.'s shape estimator then extracts the object image from the region surrounding the attended point [**?**]. The feature map with the highest contribution to the saliency of the currently attended location is segmented using a flooding algorithm with adaptive thresholding. The segmented map is used generate a binary mask to select the region around the fixation point most likely to encompass the attended object (Fig. **??**).

2.2. Object Recognition

2.2.1 Feature Encoding

The object recognition algorithm used here is a reimplementation in our laboratories of the algorithm of Lowe's. It identifies local, scale-invariant features (SIFT keypoints) and attempts to match these keypoints to those of known objects [**?**, **?**, **?**].

To identify candidate keypoint locations, scale space extrema are found in a difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, computed from the difference of scales separated by a constant factor k:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{9}$$

The extrema of $D(x, y, \sigma)$ are found by comparing each point with its neighbors in the current image and adjacent scales. A point is selected as a candidate keypoint location if it is the maximum or minimum value in its neighborhood.

The image gradients and orientations at each pixel of the Gaussian convolved image at each scale are then found. The gradient magnitude, M_{ij}, and orientation, R_{ij}, are computed for each pixel, A_{ij}:

$$M_{ij} = \sqrt{(A_{ij} - A_{i+1,j})^2 + (A_{ij} - A_{i,j+1})^2} \tag{10}$$

$$R_{ij} = atan2(A_{ij} - Ai + 1, j, A_{i,j+1} - A_{ij}) \tag{11}$$

Each key location is assigned an orientation determined by the peak of a histogram of previously computed neighborhood orientations.

Once the orientation, scale, and location of the keypoints have been computed, invariance to these values is achieved by computing the keypoint local feature descriptors relative to them. The local feature descriptors are 128-dimensional vectors obtained from the precomputed image gradients and orientations around the keypoints. For each keypoint, the orientation values (relative to the keypoint's orientation) of all pixels within circle of radius 8 pixels around the keypoint's location are inserted into 8 orientation planes. These planes are sampled over a 4x4 grid of orientations, with linear interpolation for intermediate orientations. This gives a total of $8 * 4 * 4$ or 128 samples to form the keypoint's local feature descriptor.

2.2.2 Matching

The model then attempts to match a new object with an object already existing in the database (see Fig. **??**). Lowe's algorithm does not specify an order in which various stored object representations should be checked against a new image. In our study, the pruning of search space is achieved at this stage, by prioritizing objects in the database, and keypoints within each object. Keypoint prioritization occurs only once for each object during its initial recognition or addition to the database, but object prioritization must be performed for each database object at each new fixation point. This introduces added computational costs, but we will show that these are offset by the reduction in keypoint feature vector comparisons gained by object prioritization.

The object prioritization orders objects by their similarity to the low-level features of the current candidate for recognition. Each object in the database is ranked in ascending order according to the square of the Euclidean distance of its 42-dimensional pre-attentive feature vector to that of the target object. In the following formulation, f denotes this squared Euclidean distance, FV_1 is the 42-dimensional feature vector of the database object, and FV_2 is the feature vector of the target object.

$$f = (\sum_{i=1}^{42} FV_1[i] - FV_2[i])^2 \tag{12}$$

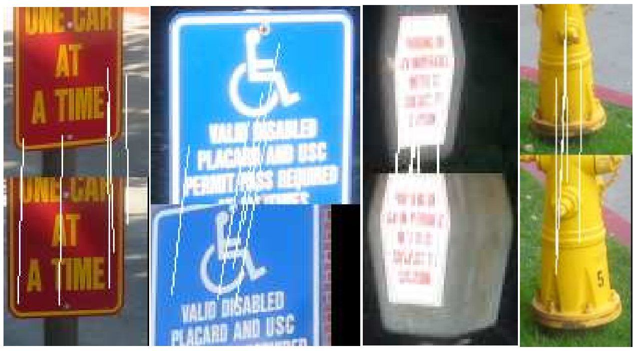

Figure 2: Examples of correct matches. Object images in each pair were extracted from different original images. White lines connecting the the two images in each pair show where keypoints from each object were successfully matched.

Each database object's ranking determines the order in which it will be compared to the target object. This serves to ensure that the most superficially similar objects in the database are checked first.

The keypoint prioritization sorts each object's keypoints in the order of descending scale-space extrema before matching. Each keypoint of the database and target objects are ranked in descending order by the magnitude, M of the scale space extrema they were found at (section 2.2.1). This ranking determines the order in which the keypoints of the database object will be compared to those of the target object. By checking the keypoints at the most extreme points first, the most unique keypoints can first be checked for consistency.

We use a simple matching scheme where for each keypoint, the two keypoints with the smallest Euclidean distance between their feature vectors are found. If the smallest keypoint distance is less than 60% of the second smallest, a keypoint match is declared.

As the size of the object representation database grows, it may become infeasible to compare every database entry with the new image to find the best match. Systems requiring rapid responses to stimuli may benefit from terminating the search process once a ”good enough” object match has been found. In such a system, the order in which object database entries are processed has a great impact on recognition speed by affecting the search space size. Our system implements this match fitness threshold by terminating the search once an object with enough keypoint matches can be found. In additional, unpublished experiments we have found that the optimal range for the keypoint match threshold for object recognition in this system is 5-7. This value is for attention-guided recognition. The optimal keypoint match threshold for object recognition without attention was found to be about twice this value. While we do

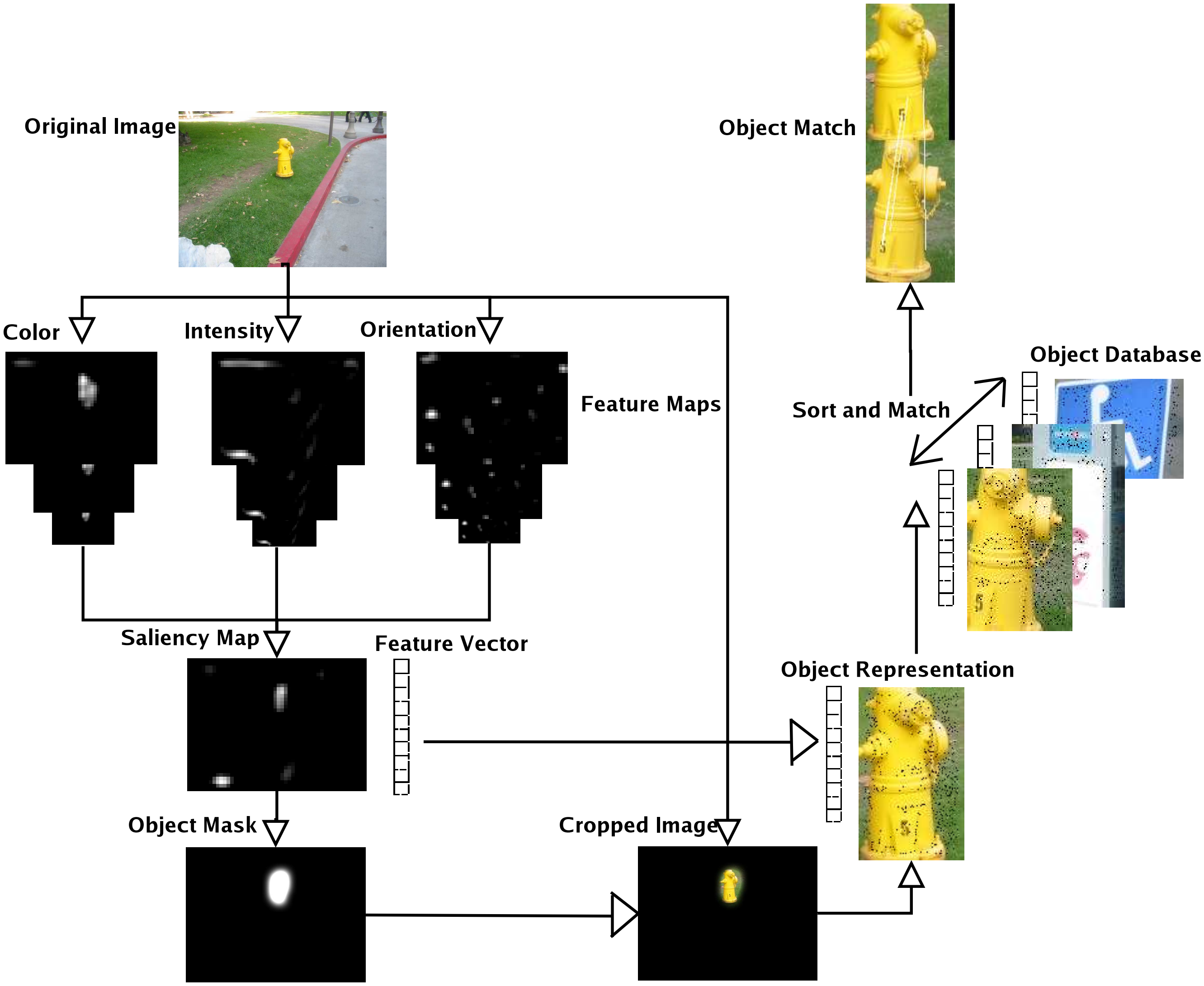

Figure 1: System diagram. The image is split into channels computing color, intensity, and orientation at several spatial scales. These are combined into a saliency map, and used to extract feature vectors characterizing salient locations. The shape estimator creates an object mask at each attended region which is used to crop the image. The cropped image is then processed to extract keypoints. The object's representation in the database consists of the image, the keypoints extracted from the image, and the feature vector for the object's most salient location. The black dots on the object images denote the location of the object's keypoints. The object database is sorted according to the new object's feature vector, and matching is attempted. The white lines in the object match picture show where keypoint matches were found between the two object images.

not analyze these results here, we feel that this is further evidence that the use of attention facilitates the speed of object recognition. In the experiments below, we require five keypoint matches for a successful object match in conditions with and without attention. If a suitable object match could not be found, the new object is added to the database.

A more complex matching scheme could be used where a model of the object transform could be constructed from a subset of keypoint matches based on the probability of a correct match and the other matches could be checked for consistency with this model. This method would improve the accuracy of this system, but it would not benefit the time it takes for a successful object match. Keypoint matches must first be found in order to construct the object model and check the consistency of the matches. Here we attempt to increase the speed of keypoint matching by pruning the object search space, and keypoint search space. A method to enforce consistency on these matches would therefore be beneficial in an implemented system, but is not necessary to conduct our experiments.

2.3. Experimental Setup

We used 120 color images of natural outdoor scenes with 800×400 resolution. This set of images was split into a training and test sets of 60 images each. The top five salient locations of each image in the training set were identified. For each location, if its object mask had an area greater than zero, keypoints were extracted from the region, and the database was trained on the object. This yielded a database containing 300 objects. Because the training and test sets contained images of similar scenes at slightly different illuminations and angles, all 300 objects from the database could potentially be recognized in the test set images.

In conditions with attention, the top five salient locations of each image in the test set were then identified. Recognition was then attempted on each salient location whose object mask had an area greater than zero. In conditions without attention, recognition was attempted on the entire image. A minimum of five keypoint matches were required to declare a successful object match (see Section 2.2.2 for discussion of this choice of keypoint match threshold). Four different model variations were tested: recognition with no attention, recognition with attention, recognition with attention utilizing object prioritization, and recognition with attention utilizing object and keypoint prioritization.

3. Results

3.1 Recognition Accuracy

Table **??** shows the number of successful and correct matches for each method tested. Successful object matches were defined when any two objects share five successfully matched keypoints. Correct object matches were defined when successful matches actually did correspond to the same object identity. The correctness ratio is the number of correct object matches divided by the number of objects in the database. This value represents the success of the model in terms of correctly identifying objects in the images in the test set. The accuracy ratio is the number of correct object matches divided by the number of successful object matches. This represents the success of the method in terms of returning only correct object matches and producing no false positive matches. The method score is correctness ratio multiplied by the accuracy ratio. An ideal method would have a method score of 1.0 by correctly recognizing all objects without any false positive matches.

All methods received relatively low correctness and accuracy ratios and method scores. We believe that this is due to the simplified matching method used and the choice of keypoint match theshold. While a more sophisticated matching process and higher keypoint match threshold would improve these values over all methods, the key contribution of this system is the improvement in recognition speed between methods discussed below in section 3.2.

As expected, the use of attention to guide recognition significantly reduced the number of incorrect object matches and increased the number of correct matches. This led to larger correctness and accuracy ratios, resulting in an increased method score. Attention guided recognition utilizing object prioritization resulted in more incorrect matches than attention guided recognition alone, but caused a large improvement in the number of correct matches. This increase in correct matches offset the increase in incorrect matches enough to improve the method score. Attention guided recognition utilizing object and keypoint prioritization also caused an improvement in the method score compared to attention guided recognition alone, but this improvement is insignificant when compared with attention guided recognition with only object prioritization.

3.2 Recognition Speed

Recognition speed was analyzed in terms of the mean duration of the matching process (Fig. **??**) and the average number of keypoint comparison operations in the matching process (Fig. **??**). Two-tailed unpaired t-test analyses on mean matching duration and average keypoint comparisons were run between methods. The improvement in recognition accuracy gained by attention-guided object recognition (section 3.1) came at the cost of a significantly large increase in the average number of keypoint comparisons ($p < 0.002$) and in the average matching time ($p < 0.006$, see discussion). Attention- guided recognition with object prioritization drastically reduced the average number of keypoint comparisons ($p < 0.0009$) and the average matching duration ($p < 0.001$) compared to attention-guided recognition. Attention-guided recognition utilizing object and keypoint prioritization also caused drastic improvements in speed compared to attention-guided recognition alone ($p < 0.0008$ for matching duration and $p < 0.0009$ for number of keypoint comparisons), but these improvements did not reach any level of significance when compared with attention guided recognition with only object prioritization.

4. Discussion

4.1 Improving Recognition Accuracy

Our results are consistent with the results of previous studies showing that the use of attention improves the accuracy of object recognition [**?**, **?**, **?**]. The reduction in image area by filtering the surrounding regions from the attended

Table 1: The number of successful object matches (obtained when any two objects shared five successfully matched keypoints) and the number of correct object matches (obtained when successful matches actually did correspond to the same object identity), for each method tested. Incorrect object matches (when five keypoints were successfully matched but between two actually different objects) is the difference between successful and correct matches. The correctness ratio is the number of correct object matches divided by the number of objects in the database. The accuracy ratio is the number of correct object matches divided by the number of successful object matches. The method score is correctness ratio multiplied by the accuracy ratio.

Method	**Successful Matches**	**Correct Matches**	**Correctness Ratio**	**Accuracy Ratio**	**Method Score**
No Attention	295	0	0.0	0.0	0.0
Attention	32	18	0.06	0.5625	0.03375
Attention and Object Prioritize	142	44	0.1467	0.3099	0.04545
Attention, Object Prioritize, and Keypoint Prioritize	142	45	0.15	0.3169	0.04754

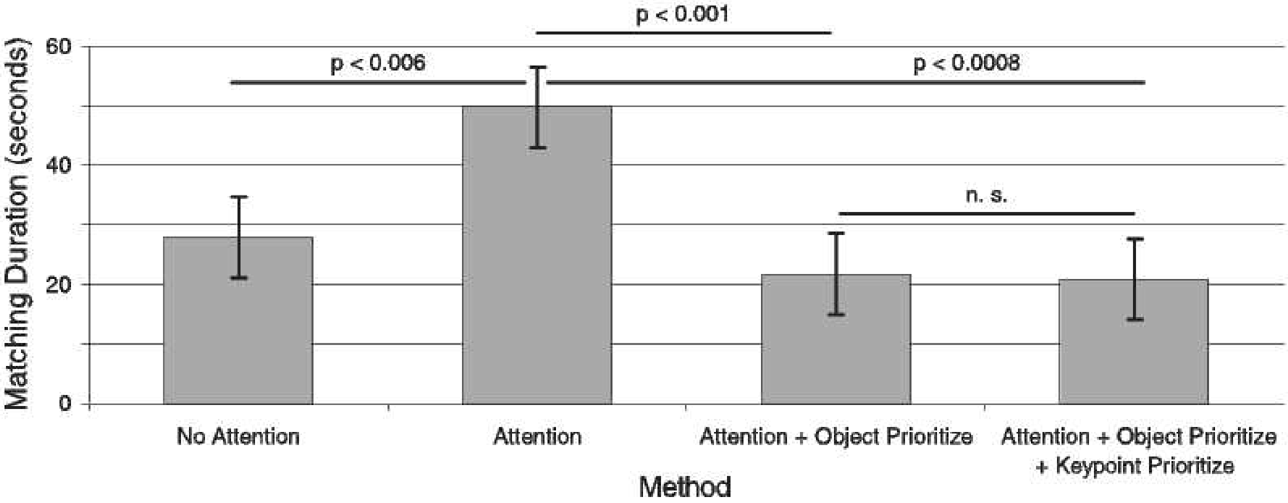

Figure 3: The average time to find an object match in the test image for each method tested. All successful matches (correct and incorrect) are shown. Unsuccessful matches are not included. The results of two-tailed unpaired t-test analyses between methods are also shown. A value of n.s. denotes that this analysis did not reach any level of significance. Recognition without attention was much faster than attention- guided recognition because all incorrect matches were found at the beginning of the object database. Attention-guided recognition was slower because more correct matches were found, but this required a longer search through the database.

area reduces the number of spurious keypoints, yielding a increased number of correct keypoint matches and fewer incorrect matches. The use of object and keypoint prioritization in attention-guided recognition did not significantly improve recognition accuracy, but obviously did not have a decremental effect on it.

4.2 Improving Recognition Speed

We have shown that the use of attention and object prioritization greatly improves object matching speed. Recognition without attention failed to find any correct object matches, but the matches it did find were found much faster than with attention-guided recognition. This is because without attentional selection, a suitable number of keypoint matches were found for the first or second objects in the database, and the search was terminated. Because

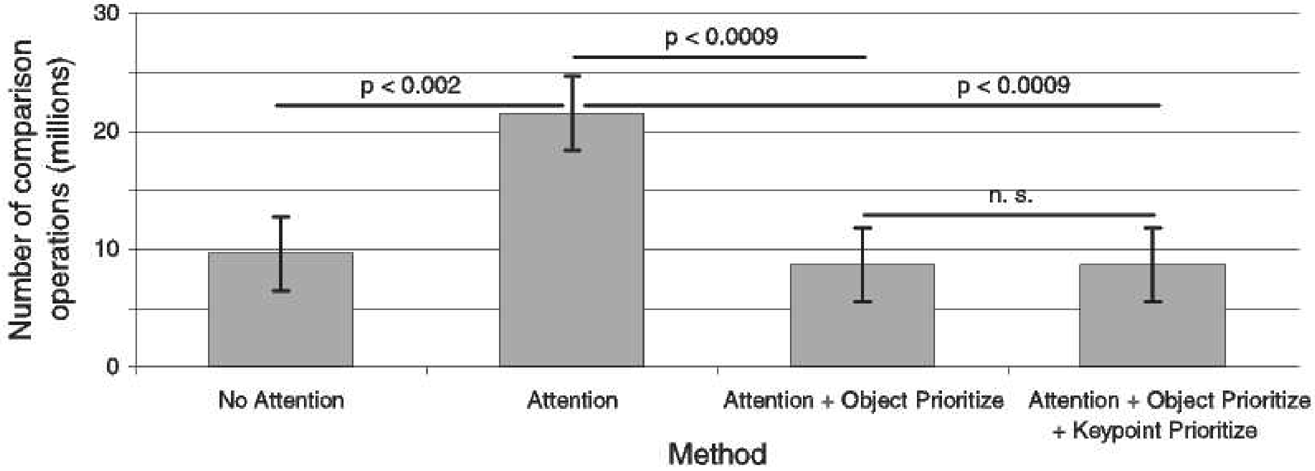

Figure 4: The average number of keypoint comparisons in a successful match (correct and incorrect) for each method tested. Unsuccessful matches are not included. The results of two-tailed unpaired t-test analyses between methods are also shown. A value of n.s. denotes that this analysis did not reach any level of significance.

these matches were incorrect, an incorrect object match was found, but found quickly. Increasing the threshold of keypoint matches would improve the accuracy of recognition without attention, but at the cost of decreased speed.

By reducing the number of incorrect keypoint matches, attention guided recognition found more correct object matches, but these were found after attempting to match a large number of objects in the database - a lengthy process. The method of object prioritization caused a decrease in the average number of keypoint comparisons to find a match and the average match time by reducing the number of incorrect object match attempts before a correct match could be found. Our results indicate that it is beneficial to pre-sort the objects in the database by their 42-dimensional pre-attentive feature vectors before SIFT keypoint recognition which involves 128-dimensional feature vectors.

Sorting each object's keypoints by magnitude served to increase the speed of finding correct keypoint matches for each object match attempt. Since this has no affect if there are no successful keypoint matches for two objects, this speed increase only occurs during a successful object match and turned out to be insignificant compared with attention-guided recognition with object prioritization only.

5. Conclusion

In sum, we have shown that the use of bottom-up visual attention increases the accuracy of object recognition, but this gain is offset by increased computation time with a large database of objects. However, the technique of bottom-up visual attention coupled with object prioritization maintains (and slightly improves) object recognition accuracy while greatly improving recognition speed.

References

[1] S. Frintrop, E. Rome, Simulating Visual Attention for Object Recognition, Proceedings of the Workshop on Early Cognitive Vision, Isle of Skye, Scotland, May 2004.

[2] L. Itti, C. Koch, Computational modeling of visual attention, Nature Reviews Neuroscience, Vol. 2(3), 2001, pp. 194-203.

[3] L. Itti, C. Koch, E. Neibur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20(11), 1998. pp. 1254- 1259.

[4] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, Vol. 60(2), 2004, pp. 91-110.

[5] D.G. Lowe, Object recognition from local scale-invariant features, in: International Conference on Computer Vision, 1999, pp. 1150-1157.

[6] D.G. Lowe, Towards a computational model for object recognition in IT cortex, in: Biologically Motivated Computer Vision, 2000, pp. 20-31.

[7] F. Miau, C.S. Papageorgiou, L. Itti, Neuromorphic algorithms for computer vision and attention, Proceedings of SPIE, Vol. 4479, Nov. 2001, pp. 12-23.

[8] V. Navalpakkam, L. Itti, Modeling the influence of task on attention, Vision Research, in press (preprint found at http://iLab.usc.edu/publications/)

[9] U. Rutishauser, D. Walther, C. Koch, P. Perona, Is attention useful for object recognition?, in: International Conference on Computer Vision and Pattern Recognition, 2004.

[10] A.A. Salah, E. Alpaydin, L. Akarun, A Selective Attention-Based Method for Visual Pattern Recognition with Application to Handwritten Digit Recognition and Face Recognition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 3, 2002, pp. 420-425.

[11] Y. Sun, R. Fisher, Object-based visual attention for computer vision, Artificial Intelligence, Vol. 146, 2003, pp. 77-123.

[12] A.M. Treisman, G.A. Gelade, A feature-integration theory of attention, Cognitive Psychology, Vol. 12, 1980, pp.97-136.

[13] J.K. Tsotsos, S.M. Culhane, W.Y. Kei Wai, Y. Lai, N. Davis, F. Nuflo, Modeling visual attention via selective tuning, Artificial Intelligence, Vol. 78, 1995, pp. 507-545.

[14] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, C. Koch, Attentional selection for object recognition a gentle way, in: Lecture Notes in Computer Science, Vol. 2525, Springer, Berlin, Germany, 2002, pp. 472-479.

[15] D.Walther, U. Rutishauser, C. Koch, P. Perona, On the usefulness of attention for object recognition, in: Workshop on Attention and Performance in Computational Vision at ECCV, 2004, pp. 96-103.

[16] S. Wendt, G.A. Fink, F. Kummert, Dynamic search-space pruning for time-constrained speech recognition, in: International Conference on Spoken Language Processing, Vol. 1, Denver, 2002, pp. 377-380.

[17] J. Wolfe, Visual Search, in: Attention (H. Pashler, ed.), London: UCL Press, 1998.