The Use of Attention and Spatial Information for Rapid Facial Recognition in Video

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 with this approach is that objects superficially dissimilar to the target are given the same consideration in recognition as similar objects. In video, objects recognized in previous frames at locations distant to the current fixation point are given the same consideration in recognition as objects previously recognized in locations closer to the current target of attention. Due to the continuity of smooth motion, objects recently recognized in previous frames at locations close to the current focus of attention have a high probability of matching the current target. Here we investigate rapid pruning of the facial recognition search space using the already-computed low-level features that guide attention and spatial information derived from previous video frames. For each video frame, Itti & Koch's bottom-up visual attention algorithm is used to select salient locations based on low-level features such as contrast, orientation, color, intensity, flicker and motion. This algorithm has shown to be highly effective in selecting faces as salient objects. Lowe's SIFT object recognition algorithm then extracts a signature of the attended object, for comparison with the facial database. The database search is prioritized for faces which better match the low-level features used to guide attention to the current candidate for recognition or those that were previously recognized near the current candidate's location. The SIFT signatures of the prioritized faces are then checked against the attended candidate for a match. 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 facial recognition in video footage.

1. Introduction

Bottom-up visual attention is the process by which primates quickly select regions of an image that contain salient or conspicuous 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 [13, 10, 12]. 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 [9]. 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 [15].

Previous work with attention-guided object recognition utilizes bottom-up visual attention models that only compute static features such as color, orientation, and intensity [13, 10, 12]. These experiments explore the use of features such as flicker and motion in computing visual saliency. This technique is intended to focus attention on temporally dynamic objects in video.

Dynamic pruning of complex search spaces has been proved to improve processing rate without a corresponding loss in accuracy in speech recognition [14]. In the visual domain, however, the value of rapidly pruning the facial recognition search space using the already-computed low-level features that guide attention has been under-explored. We also investigate whether information about a previously identified face's location can aide in recognition. From frame to frame, a person's location typically varies by a small amount. In the matching process, faces recently identified in a location close to the current target face's location should be ranked ahead of those recently seen at distal locations.

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 for each video frame [4, 3]. 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 "$\ominus$" 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 for frame n, I_n, is obtained by averaging the red, green, and blue color channels as $I_n = (r_n + g_n + b_n)/3$. A Gaussian pyramid, $I_n(\sigma)$, is created from I_n with $\sigma \in [0..8]$ as the scale. Four color channels are computed for each frame: red $R_n = r_n - (g_n + b_n)/2$, green $G_n = g_n - (r_n + b_n)/2$, blue $B_n = b_n - (r_n + g_n)/2$, and yellow $Y_n = (r_n + g_n)/2 - |r_n - g_n|/2 - b_n$. These channels are used to create four Gaussian pyramids $R_n(\sigma)$, $G_n(\sigma)$, $B_n(\sigma)$, and $Y_n(\sigma)$. The orientation channels are obtained from I_n using oriented Gabor pyramids $O_n(\sigma, \theta)$, 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 ($\ominus$) 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 $\mathcal{I}_n(c, s)$, with $c \in \{2, 3, 4\}$ and $s = c + \delta, \delta \in \{3, 4\}$:

$$\mathcal{I}_n(c, s) = |I_n(c) \ominus I_n(s)| \tag{1}$$

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

$$\mathcal{RG}_n(c, s) = |(R_n(c) - G_n(c)) \ominus (G_n(s) - R_n(s))| \tag{2}$$

$$\mathcal{BY}_n(c, s) = |(B_n(c) - Y_n(c)) \ominus (Y_n(s) - B_n(s))| \tag{3}$$

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

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

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

For experiments using attention with flicker and motion, additional temporal flicker (onset and offset of light intensity, combined) and four oriented motion (up, down, left, right) feature maps are computed using center-surround differences at six different spatial scales to total 72 feature maps [5]. A flicker pyramid $F_n(\sigma)$ is computed using the absolute difference between the intensity I_n of the current frame and that I_{n-1} of the previous frame. The motion pyramids are computed using spatially-shifted differences between Gabor pyramids from the previous and current frame. The same four Gabor orientations as in the orientation channel and only shifts of one pixel orthogonal to the Gabor orientation are used to create one shifted pyramid $S_n(\sigma, \theta)$ for each Gabor pyramid $O_n(\sigma, \theta)$. These are then used to compute the motion pyramid $R_n(\sigma, \theta)$:

$$R_n(\sigma, \theta) = |O_n(\sigma, \theta) * S_{n-1}(\sigma, \theta) - O_{n-1}(\sigma, \theta) * S_n(\sigma, \theta)| \tag{5}$$

The flicker and motion feature maps are then created using the same center-surround operation as the other feature maps:

$$\mathcal{F}_n(c, s) = |F_n(c) \ominus F_n(s)| \tag{6}$$

$$\mathcal{R}_n(c, s, \theta) = |R_n(c, \theta) \ominus R_n(s, \theta)| \tag{7}$$

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 $\bar{I}_n$, color $\bar{C}_n$, and orientation $\bar{O}_n$, 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:

$$\bar{I}_n = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{I}_n(c, s)) \tag{8}$$

$$\bar{C}_n = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(\mathcal{RG}_n(c, s)) + \mathcal{N}(\mathcal{BY}_n(c, s))] \tag{9}$$

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:

$$\bar{O}_n = \sum_{\theta \in \{0^o, 45^o, 90^o, 135^o\}} \mathcal{N}(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{O}_n(c, s, \theta))) \tag{10}$$

The flicker conspicuity map is computed in the same manner as the intensity conspicuity map [5]:

$$\bar{F}_n = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{F}_n(c, s)) \tag{11}$$

The motion conspicuity map is computed via the creation of four intermediary maps for a given θ in a manner similar to the orientation conspicuity maps [5]:

$$\bar{R}_n = \sum_{\theta\in\{0^o,45^o,90^o,135^o\}} \mathcal{N}(\bigoplus_{c=2}^{4}\bigoplus_{s=c+3}^{c+4} \mathcal{N}(\mathcal{R}_n(c,s,\theta))) \tag{12}$$

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

$$S_n = \frac{1}{3}(\mathcal{N}(\bar{I}_n) + \mathcal{N}(\bar{C}_n) + \mathcal{N}(\bar{O}_n)) \tag{13}$$

The method of attention utilizing flicker and motion similarly normalizes, sums, and inputs the five conspicuity maps in this case, into the saliency map [5]:

$$S_n = \frac{1}{3}(\mathcal{N}(\bar{I}_n) + \mathcal{N}(\bar{C}_n) + \mathcal{N}(\bar{O}_n) + \mathcal{N}(\bar{F}_n) + \mathcal{N}(\bar{R}_n)) \tag{14}$$

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 as 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 [9]. 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 image from the region surrounding the attended point [13]. 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. 1). This region may or may not actually contain a face. Future work might bias attention by modulating the relative weights of the feature maps as they are linearly combined into the saliency map to preferentially fixate on regions whose low-level features match those of faces.

2.2. Facial Recognition

2.2.1 Feature Encoding

Some previous approaches to facial recognition have involved the identification and characterization of prominent facial landmarks or key points [1, 2, 11]. We investigated the effectiveness of a keypoint-based, general object recognition algorithm in facial recognition by using a reimplementation in our laboratories of that of Lowe's. It identifies local, scale-invariant features (SIFT keypoints) and attempts to match these keypoints to those of known objects [7, 8, 6].

To identify candidate keypoint locations, the scale space extrema are found in the difference-of-Gaussian function convolved with each frame n of the video clip, $D_n(x, y, \sigma)$, which is computed from the difference of scales separated by a constant factor k:

$$D_n(x, y, \sigma) = (G_n(x, y, k\sigma) - G_n(x, y, \sigma)) * I_n(x, y) \tag{15}$$

The extrema of $D_n(x, y, \sigma)$ 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 video frame at each scale are then found. The gradient magnitude, $M_{n,i,j}$, and orientation, $R_{n,i,j}$, are computed for each pixel, $A_{n,i,j}$:

$$M_{n,i,j} = \sqrt{(A_{n,i,j} - A_{n,i+1,j})^2 + (A_{n,i,j} - A_{n,i,j+1})^2} \tag{16}$$

$$R_{n,i,j} = atan2(A_{n,i,j} - A_{n,i+1,j}, A_{n,i,j+1} - A_{n,i,j}) \tag{17}$$

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 face with a face already existing in the database (Figure 2). Lowe's algo-

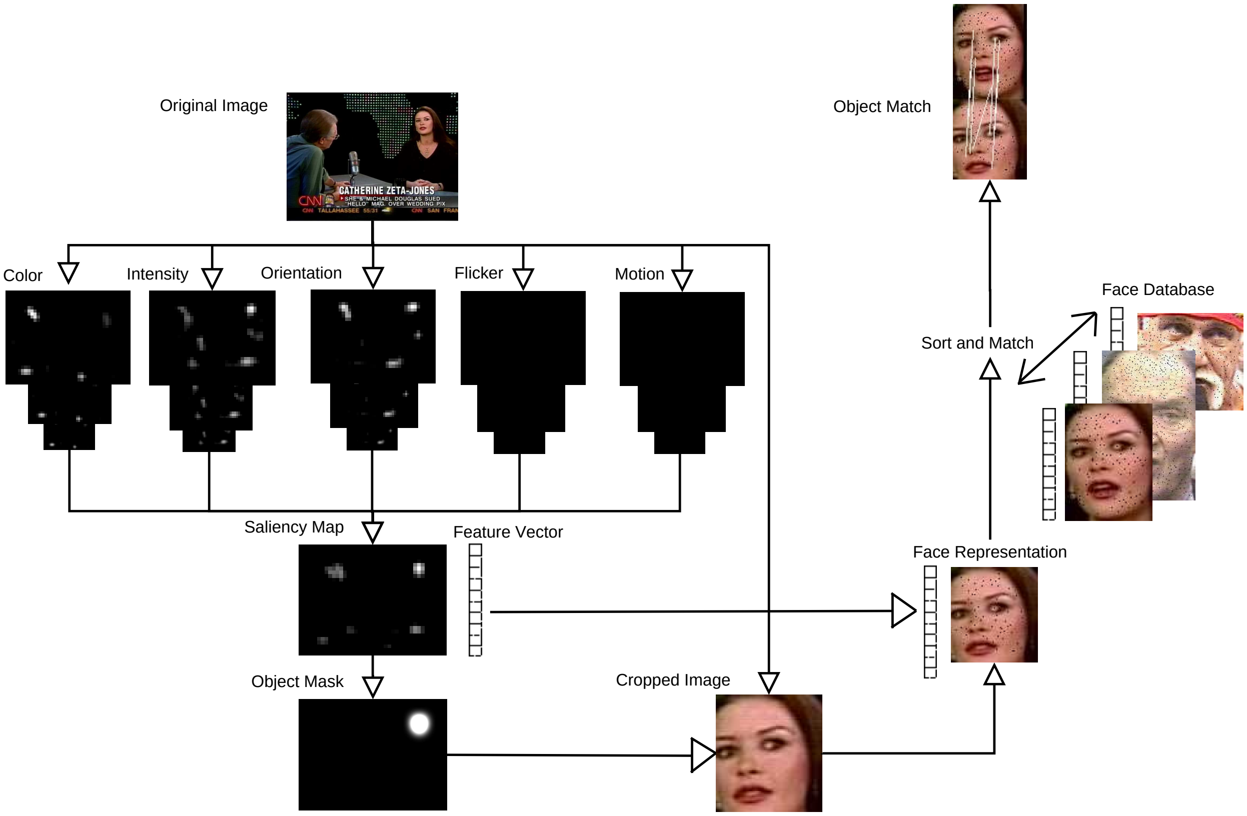

Figure 1: System diagram. The image is split into channels computing color, intensity, orientation, flicker, and motion at several spatial scales. These are combined into a saliency map, and used to extract feature vectors characterizing salient locations. The shape estimator creates an object mask at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

rithm 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 faces in the database in two ways. Each method of prioritization determines the order in which faces in the database are compared to the target object. Face prioritization by feature orders faces by their similarity to the low-level features of the current candidate for recognition. Each face in the database is ranked in ascending order according the the Euclidean distance of its 42-dimensional pre-attentive feature vector to that of the target object. This serves to ensure that the most superficially similar faces in the database are checked first. Face prioritization by location sorts faces in ascending order by the distance from the location they were last recognized at to the location of the current target face. By checking the faces recently identified at the locations closest to the current recognition candidate's location, the most likely faces can first be checked for a match. To evaluate the effectiveness of these two sorting strategies, we compare them with a control condition that uses a random sorting order.

We use a simple keypoint 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 face 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” match has been found. In such a system, the order in which object

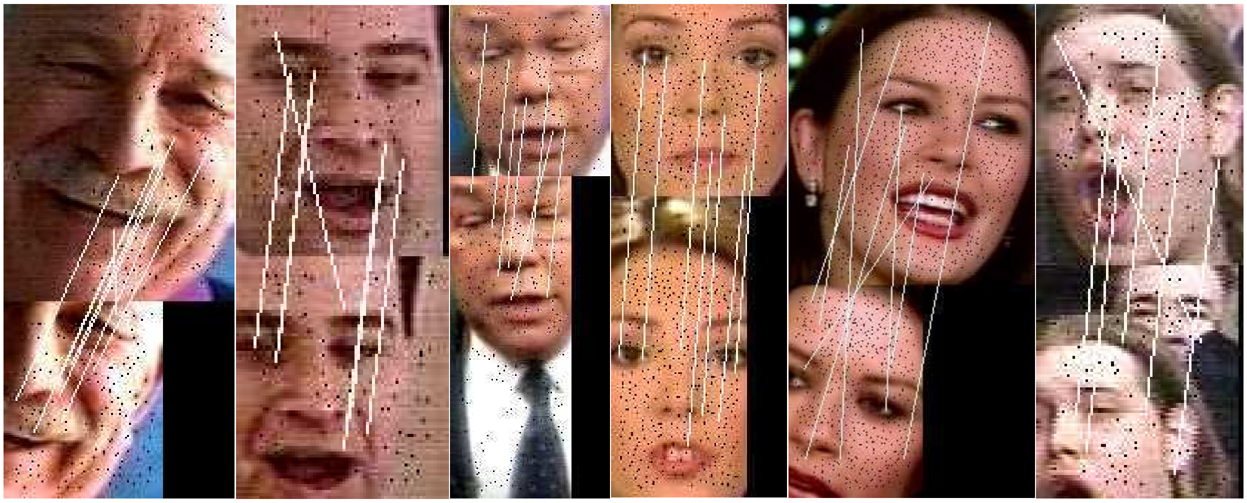

Figure 2: Examples of correct matches. Face images in each pair were extracted from different frames. The top row contains faces from the database. The bottom row contains attended regions of video clip frames from the test set that the system identified as salient. White lines connecting the the two images in each pair show where keypoints from each face where successfully matched.

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 a face with enough keypoint matches can be found. In additional, unpublished experiments we have found that the optimal range for the keypoint match threshold for facial recognition in this system is 5-7. This value is for attention-guided recognition. The optimal keypoint match threshold for facial recognition without attention was found to be about twice this value. While we do 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 eight keypoint matches for a successful match in conditions with and without attention.

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 decrease the occurrence of incorrect face matches, but it would not benefit the time it takes for a successful 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 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 23 color video clips taken from television programs. For each of 31 individuals present in the video clips, 2-3 frames from various clips were cropped to include only the individual's face. This set of images was used as a training set. For each training image, keypoints were extracted, and the database was trained on the face. This yielded a database containing 77 faces.

The 23 video clips were used as a test set. The top salient location of each frame in each video clip in the test set was then identified. Recognition was attempted on each salient location whose object mask had an area greater than zero. A minimum of 8 keypoint matches were required to declare a successful face match. This process was repeated for 2 models (recognition with attention and recognition with attention using flicker and motion) and 3 database sorting methods (random, by feature, and by location). These conditions were compared to a control model of recognition without attention using random database sorting to total seven different model variations.

3. Results

3.1. Facial Detection Efficiency

The bottom-up visual attention model of Itti & Koch proved very effective in finding faces as particularly salient. Although regions of images that did not contain faces were occasionally attended to, these instances were too small in number to have any effect on the efficiency (averaging only a few non-facial fixations per 50 frames actually containing

Table 1: The number of successful face matches (obtained when any two faces shared eight successfully matched keypoints) and the number of correct face matches (obtained when successful matches actually corresponded to the same individual), for each method tested. Incorrect face matches (when eight keypoints were successfully matched but between two actually different individual's faces) is the difference between successful and correct matches.

Method	**Successful Matches**	**Correct Matches**
No Attention - Random Sort	5643	162
Attention - Random Sort	138	58
Attention - Sort Faces by Low-level Feature	138	54
Attention - Sort Faces by Location	138	58
Attention with Flicker and Motion - Random Sort	134	45
Attention with Flicker and Motion - Sort Faces by Low-level Feature	133	38
Attention with Flicker and Motion - Sort Faces by Location	134	39

faces). One trend that was evident was a slight increase in non-facial fixations with attention using flicker and motion channels (see discussion). Future models using attentional biasing with the pre-attentive features of faces [9] may optimize the facial detection efficiency of this system.

3.2. Recognition Accuracy

Table 1 shows the number of successful and correct face matches for each method tested. Successful face matches were defined when any two faces shared eight successfully matched keypoints. Correct face matches were defined when successful face matches actually corresponded to the same individual. The accuracy ratio (Fig. 3) is the number of correct face matches divided by the number of successful face matches. This value represents the success of the method in terms of returning only correct matches and producing no false positive matches.

As expected, the use of attention to guide recognition significantly reduced the number of incorrect face matches, but also reduced the number of correct matches. Despite this reduction in correct face matches, the use of attention-guided recognition was able to improve the accuracy ratio. Attention guided recognition utilizing flicker and motion features resulted in a slight increase in successful matches, but a reduction in the number of correct face matches, leading to a decrease in accuracy ratio compared to the same method without flicker and motion.

Neither method of search space pruning by face prioritization had a significant impact on the number of successful or correct matches, and consequently little effect on the accuracy ratio compared to random database sorting. While this may seem discouraging, it should be pointed out that although neither method of prioritization improved recognition accuracy, one method of prioritization improved recognition speed (section 3.3) and neither method caused a significant loss of accuracy.

3.3. Recognition Speed

Recognition speed was analyzed in terms of the average number of keypoint comparison operations in the matching process (Fig. 4), since keypoint comparison is the atomic operation of this process. The improvement in recognition accuracy gained by attention-guided recognition (section 3.2) came at the cost of a large increase in the average number of keypoint comparisons ($p < 4.266 \times 10^{-20}$, see discussion). Attention guided recognition utilizing flicker and motion features had no significant effect on recognition speed compared to the static feature attention model. Face prioritization by low-level feature also had no significant effect on recognition speed while face prioritization by location drastically reduced the mean number of keypoint comparisons needed for a successful match compared to attention-guided recognition alone ($p < 2.748 \times 10^{-26}$ for static feature attention and $p < 2.829 \times 10^{-25}$ for attention using flicker and motion features) without a corresponding decrease in accuracy (section 3.2).

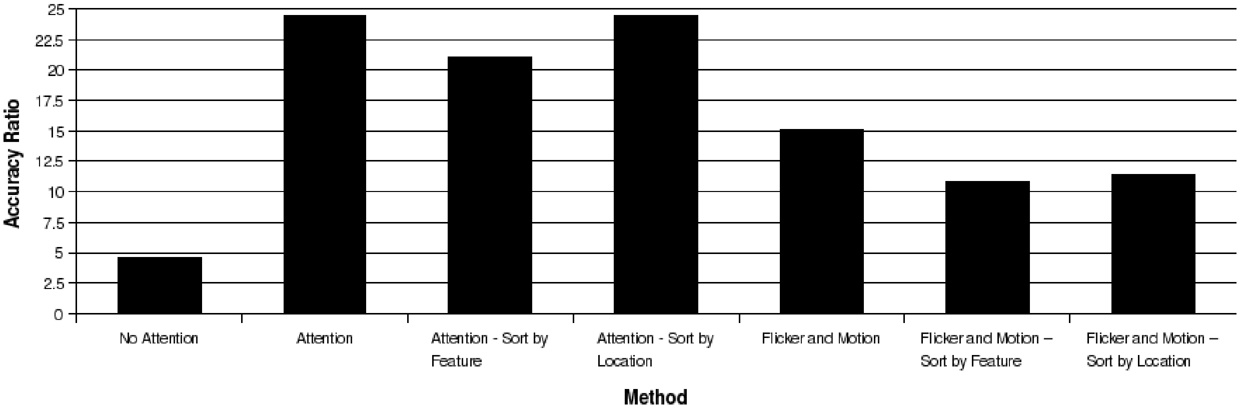

Figure 3: The accuracy ratio for each method tested. The accuracy ratio is the number of correct face matches divided by the number of successful face matches. All methods using attention without flicker and motion received the highest accuracy ratios, while neither technique of face prioritization had an effect on this metric.

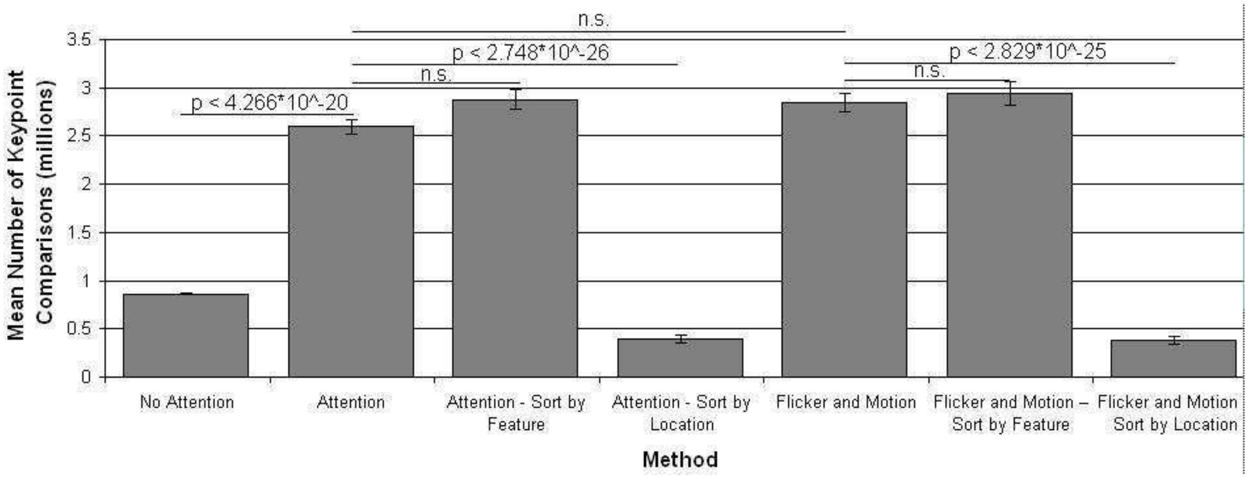

Figure 4: The average number of keypoint comparisons (in millions) in a successful face match (correct and incorrect) for each method tested. Unsuccessful match attempts are not included. Recognition without attention was much faster than attention guided recognition because most matches this method found were incorrect and were among the first faces attempted for recognition. Attention guided recognition was slower because more correct matches were found, but this required a longer search through the database.

The method score (Fig. 5) is computed by the product of the number of correct face matches and the accuracy ratio divided by the sum of the mean number of keypoint comparisons and the standard error. In this way, methods are rewarded for fast, reliable, and correct object matches and penalized for incorrect or slow matches. Attention-guided recognition using face prioritization by location received the greatest method score value by far. This is due to the enormous speed increase afforded by this prioritization technique and increased recognition accuracy provided by attention-guided recognition.

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 [13, 10, 12]. The reduction in image area by filtering the surrounding regions from the attended area reduces the number of spurious keypoints, yielding a increased number of correct keypoint matches and fewer incorrect matches. Recognition without attention failed to find many correct face matches. This is because

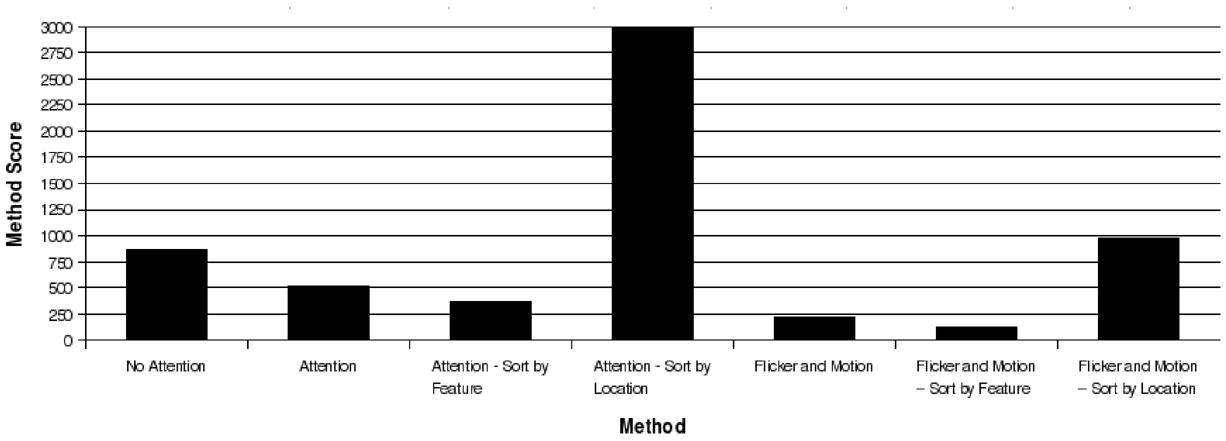

Figure 5: The method score for each method tested. The method score is the product of the number of correct face matches and the accuracy ratio divided by the sum of the mean number of keypoint comparisons and the standard error. This score rewards accuracy, speed, and consistency.

without attentional selection, a suitable number keypoint matches were found for the first or second faces checked in the database and the search was terminated. It is likely that the correct face matches using this method are due to the fact that each training image came from a frame in a video in the test set. The use of flicker and motion in visual attention had a detrimental impact on facial recognition accuracy. This could be attributed to the fact that while the motion feature analysis would bias attention towards a moving mouth, movements such as gestures would tend to distract attention away from the face. Because faces are already found salient by the static visual attention model, the additional information from flicker and motion channels did not have any benefit.

The methods of face prioritization by pre-attentive feature and by location did not improve recognition accuracy compared to random database sorting, but did not have a detrimental effect on it.

4.2. Improving Recognition Speed

By reducing the number of incorrect keypoint matches, attention guided recognition found more correct face matches, but these were found after attempting to match a large number of faces in the database - a lengthy process. Although recognition without attention failed to find many correct face matches, the matches it did find were found much faster than those using attention guided recognition. Because these matches were mostly incorrect, an incorrect face match was found, but found quickly.

Face prioritization by feature did not have any significant effect on recognition speed. This appears to be because all faces may have similar pre-attentional features, invalidating any speed gains that prioritization by these features may provide. This method may be better suited for general object recognition were different objects in the database can be expected to have sufficiently distinctive low-level features.

The method of face prioritization by location, however, caused a significant decrease in the average number of keypoint comparisons needed to find a face match, reducing the number of incorrect face match attempts before a correct match could be found. This method reduced search space by taking advantage of motion continuity and preferentially comparing faces in the database that have recently been identified in locations close to the target face's location.

5. Conclusion

In sum, we have shown that the use of bottom-up visual attention increases the accuracy of facial recognition, but this gain is offset by increased computation time with a large database of faces. However, the technique of bottom-up visual attention coupled with face prioritization by location improves facial recognition speed without negatively impacting recognition accuracy.

References

[1] R. Herpers, M. Michaelis, G. Sommer, L. Witta, Context based detection of keypoints and features in eye regions, Proceedings of the 13th International Conference on Pattern Recognition, Vol 2., 1996, pp. 23-28.

[2] R. Herpers, L. Witta, J. Bruske, G. Sommer, Dynamic cell structures for the evaluation of keypoints in facial images,

International Journal of Neural Systems, Vol. 8(1), 1997, pp. 27-39.

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

[4] 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.

[5] L. Itti, N. Dhavale, F. Pighin, Realistic Avatar Eye and Head Animation Using a Neurobiological Model of Visual Attention, In: Proc. SPIE 48th Annual International Symposium on Optical Science and Technology, Aug 2003, pp. 64-78.

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

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

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

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

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

[11] V. Starovoitov, D. Samal, A geometric approach to face recognition, Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image, Vol. 2, 1999, pp. 210-213.

[12] 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.

[13] 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.

[14] 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.

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