A Tree-Structured Model of Visual Appearance Applied to Gaze Tracking

Jeffrey B. Mulligan

NASA Ames Research Center

Abstract. In some computer vision applications, we may need to analyze large numbers of similar frames depicting various aspects of an event. In this situation, the appearance may change significantly within the sequence, hampering efforts to track particular features. Active shape models [1] offer one approach to this problem, by ”learning” the relationship between appearance and world-state from a small set of hand-labeled training examples. In this paper we propose a method for partitioning the input image set which addresses two problems: first, it provides an automatic method for selecting a set of training images for hand-labeling; second, it results in a partitioning of the image space into regions suitable for local model adaptation. Repeated application of the partitioning procedure results in a tree-structured representation of the image space. The resulting structure can be used to define corresponding neighborhoods in the shape model parameter space; a new image may be processed efficiently by first inserting it into the tree, and then solving for model parameters within the corresponding restricted domain. The ideas are illustrated with examples from an outdoor gaze-tracking application.

1 Introduction

Many computer vision applications consist of analyses of large numbers of similar images; in this paper, we will be concerned with the problem of gaze estimation from images of the eye captured with a head-mounted camera. Assuming the camera platform does not move relative to the head, the images will vary within a restricted subspace. Variation within this subspace will be due both to the parameters of interest (the pose of the eye), and to parameters which are irrelevant for our purposes, such as variations in environmental illumination.

When images of the eye are collected in the laboratory, illumination can be carefully controlled, and the variations in pose are often restricted (e.g., we may only be interested in tracking the gaze within a display screen). In this case, simple methods which search for features known to be present are usually effective. When we attempt to measure gaze in natural behaviors outside of the laboratory, however, we may be confronted with a collection of images in which large gaze deviations cause expected features to disappear. Our inability to control the illumination outdoors during daylight also presents a new set of problems. Figure 1 presents a representative sample from the space of images in our study. Subjects recorded during the day, as in figure 1, generally maintain their eyelids in a relatively closed posture compared to subjects recorded at night, or in a dark laboratory. In many of the images in figure 1, the eyelids appear closed. While some of these are certain to correspond to blinks, others correspond to downward fixations. In spite of the fact that in many of the images we cannot see any of the eyeball itself (and are so prevented from applying traditional eye-tracking methods), the position and shape of the lids are highly informative. When we have highly certain data regarding the position and orientation of the eyeball, the pose of the lids is irrelevant to determination of the line-of-sight, but for other frames the eyelid pose is our *only* observable. We seek a method to tell us which methods to apply to any given image.

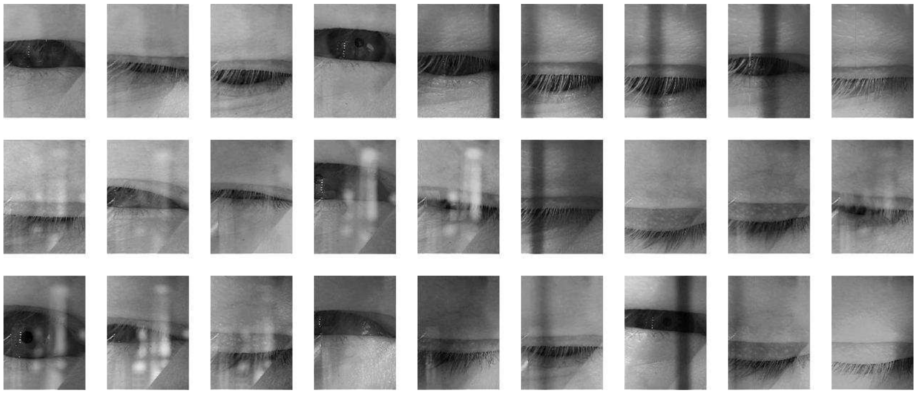

Fig. 1. A collection of images of a human subject's eye, collected during a behavioral experiment conducted outdoors, showing the variety of appearances encountered. In addition to changes in gaze direction, the images vary due to illumination; in this collection we see superimposed bright blobs (resulting from scattering or incomplete reflection at the imaging system's dichroic mirror), and dark vertical stripes, resulting from the shadow of the helicopter rotor blade passing in front of the sun.

Active Shape Models [1] have been proposed to recognize structures in medical images in the presence of variations in local image structure. In brief, these models work by learning an association between the local image appearance of model feature points, and the overall configuration of the features. Typically this association is learned from a collection of images in which the structures of interest have been hand-labeled. In selecting this training set, there are two important considerations: first, the training set must span the range of possible shapes (i.e., we must be sure to include the most extreme examples); second, the sampling density must be high enough to capture the variations of shape within the image space. Selection of the training set is therefore critical. One approach might be to simply add images until performance becomes acceptable (perhaps adding images chosen from the set of initial failures), but when the number of images is large, it may not be practical to have a human expert review the model fit for every frame, or even to view every image prior to selecting the training set. Therefore, we would like to have an automatic procedure to select a suitable training set.

Methods for sampling a space of images can be found in the field of *Vector Quantization* [2]. Vector Quantization (VQ) refers to a collection of techniques used in signal coding and compression. While many variants have been proposed for different applications, in every case the process begins with the generation of a *codebook*, which is a table of images chosen to coarsely represent the entire space. An arbitrary image is transmitted by simply sending the index of the most similar codebook entry; the receiver, which also possesses the codebook, uses the corresponding codebook entry to approximate the input image. The quality of the reconstruction depends on both the size and structure of the codebook, and many methods of codebook design have been proposed to meet the needs of different applications. While codebook design (which is a one-time computation, done ahead of time) is the most important determinant of reconstruction quality, another area which has received much attention is efficient mapping of arbitrary images into the codebook, which is critical for real-time encoding processes. It is desirable to avoid exhaustive search, in which the input image is compared to every codebook entry; a Tree-Structured Vector Quantizer (TSVQ) can reduce the number of comparisons from a codebook size of N to something on the order of log(N).

In the remainder of this paper, we present a variant of tree-structure vector quantization developed for a large set of eye images collected during helicopter flight tests. We then describe how the resulting codebook an be exploited to improve the performance of active shape models and other tracking procedures, by limiting the range of parameter values that must be searched for new images.

2 Codebook Generation

Given a set of images, we wish to find a subset which spans the entire set, in the sense that *any* image from the set will be ”near” one of the *exemplars* from our special subset. The exemplars are analogous to the codebook entries in VQ, but unlike most VQ applications we are not particularly concerned with insuring that the exemplars are good matches to the nearby images; instead, it is simply sufficient for them to be near enough to point us in the right direction for subsequent processing.

Typical gaze records consist of a series of *fixations* in which the eye is steadily pointed at an object of interest, and *saccades*, which are rapid, ballistic movements from one position to another. In addition to these two types of eye movement, there are also smooth movements which are performed when the eye attempts to follow a moving target. Smooth movements of the eye in the head are also seen when the head moves while the eye is maintaining fixation on a stationary target. The interested reader can find a thorough introduction to the study of eye movements in [3] and [4].

Because gaze behavior typically consists of fixations lasting 250-500 milliseconds, when we process video sequentially it is highly likely that a given frame will be similar to the preceding frame. Therefore our algorithm proceeds as follows: we take the first frame as the first exemplar. Subsequent frames are processed by first computing the similarity to the exemplar chosen to represent the immediately preceding frame. If the "distance" is below a threshold δ, then we accept the exemplar and move on to the next input frame. Otherwise, we search the set of remaining exemplars for one whose distance is less than δ, accepting the first one we find. If none of the current exemplars are within δ of the new image, then we add it to the set of exemplars. This procedure results in a set of exemplars with the following property: every image x in our collection is within δ of at least one of the exemplars, and every exemplar is separated by at least δ from from every other exemplar. Our goal is to choose δ as large as possible (to keep the size of the catalog small), but still small enough that the resulting classifier on the input images provides useful distinctions.

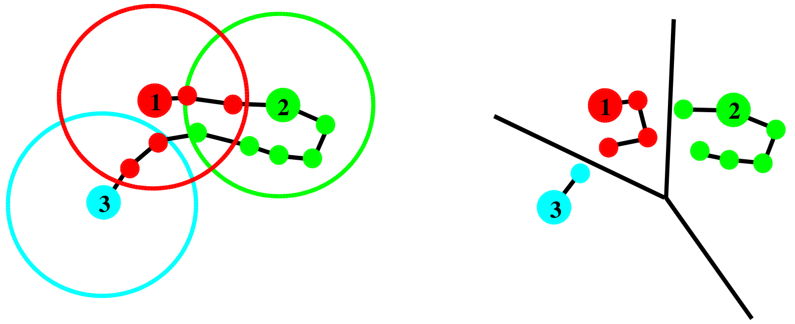

Fig. 2. Left panel: A two-dimensional cartoon of the codebook generation process. Each dot represents an image, with the connecting links indicating temporal order. The first image becomes the first exemplar (the large dot labeled 1), successive images are associated with it as long as their distance remains less than a threshold (δ) indicated by the circle. When a new image falls outside of this circle, a new exemplar is created, and successive images are associated with *it* until they leave its δ-neighborhood. Right panel: partition of the image space induced by the set of exemplars discovered by the codebook generation process. Each image is associated with the nearest exemplar; dark lines indicate boundaries between neighborhoods (Voronoi regions). Note that images which are linked in the sub-regions may not be temporally contiguous.

The left panel of figure 2 shows a two-dimensional cartoon of this process. The circles represent neighborhoods of radius δ centered at each exemplar, and the colors show how each image is associated with the exemplar representing the previous frame as long as the new distance is less than δ. We adopt this heuristic for two reasons: first, because the temporal sequence is continuous, and the behavior contains many stationary intervals, the previous exemplar is usually the best choice, and we can save time by simply accepting it. Second, during our sequential scan of the data there will be many occasions when the exemplar which will ultimately be found to be the nearest neighbor has not been scanned yet. Thus the purpose of the initial pass is simply to find a set of exemplars which completely covers the image space with δ neighborhoods. The association of each image with the nearest exemplar is accomplished by a second pass over the entire data set, performed after the catalog has been generated. This process is effectively a nearest-neighbor classifier [5, 6], where the "classes" are defined implicitly by the exemplars. The results of the second pass are shown in the right-hand panel of figure 2.

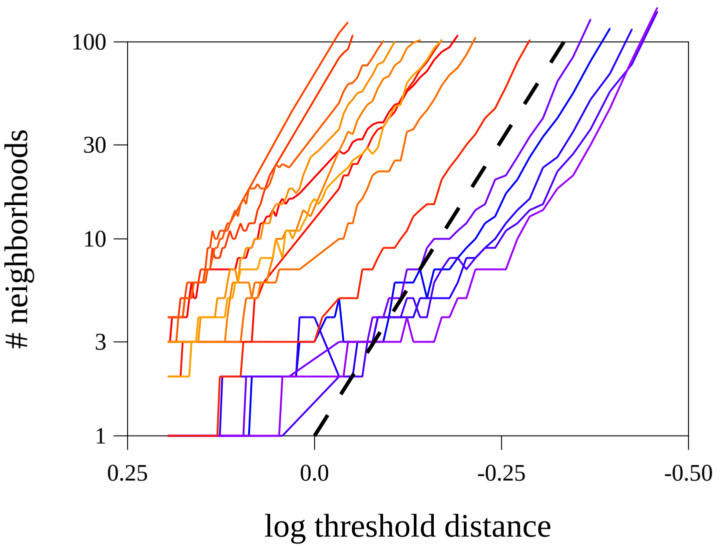

Fig. 3. Plot shows the number of neighborhoods formed for different values of the (angular) distance threshold. The heavy dashed line indicates a slope of 6 on the log-log axes; to the extent that this matches the slope of the curves generated from the empirical data, this suggests that the image manifold has an intrinsic dimensionality of 6. Each data curve represents a recording of approximately 100,000 frames. The five curves to the right of the dashed line represent images collected at night for which the only illumination was provided by the apparatus.

The number of exemplars N found by this procedure depends on the choice of δ, and the dependence tells us something about the intrinsic dimensionality of the image manifold. The discovery of interesting structure from the topology of the manifold is the subject of a relatively new field of study known as *manifold learning*, exemplified by [7]. In figure 2, where we have represented the images as points in a two-dimensional space, the number of exemplars would be expected to grow in inverse proportion to the square of δ; in practice, the dimensionality is much greater. Figure 3 shows a plot of the number of exemplars generated as a function of δ for 15 individual video recordings from a head-mounted eye camera, each consisting of around 100,000 frames. The heavy dashed line indicates a slope of 6 on the log-log plot, which provides a reasonable fit to the asymptotic slope of the data graphs. The five records to the right of the dashed line correspond to the five night flights, for which the illumination is relatively constant. While there is a good deal of horizontal dispersion, each data set shows an asymptotic slope near 6, suggesting that the images vary in 6 dimensions. Four of these can be accounted for by physiological variables: two dimensions of gaze direction, and one each for pupil dilation and degree of eyelid closure.

The data shown in figure 3 are useful in two ways: first, they allow us to choose an initial value of δ which is small enough to generate more than one exemplar, but not so small that the number becomes unmanageable. Secondly, knowledge of the manifold's intrinsic dimension tells us how to decrease δ when we apply the procedure recursively to the neighborhoods generated by the first pass. If N is the intrinsic dimensionality of the image manifold, then to obtain k child nodes, we should divide δ by the Nth root of k.

3 Computational Efficiency

When δ is small relative to variation in the input set, many images are added to the catalog, and as the size of the catalog grows, the cost of testing a new image against the entire catalog grows apace. Can we reduce the number of tests which must be performed? Provided we retain the distances between the exemplar images, the answer is yes, by judicious application of the triangle inequality (see figure 4). Imagine we are processing a new image $\mathbf{x}$, which we have just compared to the exemplar $\mathbf{e}_i$ corresponding to the previous frame. We call the distance between these two images $d(\mathbf{x}, \mathbf{e}_i)$, which we assume to be greater than δ. There are two classes of exemplar which we can exclude from further testing: if $\mathbf{e}_i$ is far from $\mathbf{x}$, then we can reject any exemplars which are sufficiently near to $\mathbf{e}_i$; conversely, if $\mathbf{e}_i$ is near to $\mathbf{x}$, then we can reject any exemplars which are sufficiently far from $\mathbf{e}_i$. These notions are illustrated in figure 4. To reject a candidate exemplar (like $\mathbf{e}_k$ in figure 4) for being too far away from $\mathbf{e}_i$, we apply the triangle inequality to the lower triangle in figure 4, and find that we can reject $\mathbf{e}_j$ if $d(\mathbf{e}_i, \mathbf{e}_j) - d(\mathbf{e}_i, \mathbf{x}) > \delta$. Similarly, we can apply the triangle inequality to the upper triangle in figure 4, and see that we can reject $\mathbf{e}_j$ if $d(\mathbf{e}_i, \mathbf{x}) - d(\mathbf{e}_i, \mathbf{e}_j) > \delta$. The first test rejects all exemplars falling outside the largest circle in figure 4, while the second test rejects all exemplars falling inside the inner circle. Exemplars falling in the annular region bounded by the two concentric circles cannot be rejected, and must be compared directly to $\mathbf{x}$. The two cases can be combined into a single test: reject $\mathbf{e}_j$ if $|d(\mathbf{e}_i, \mathbf{x}) - d(\mathbf{e}_i, \mathbf{e}_j)| > \delta$.

Each time the distance between the new image $\mathbf{x}$ and an exemplar is computed, we can apply the test indicated above to all the remaining exemplars which have not yet been rejected. The computational savings resulting from this procedure cannot be predicted without knowledge of how the input images are distributed relative to one another. In our data set, the number of tests is reduced by more than half, compared to exhaustive search. Offset against this savings is

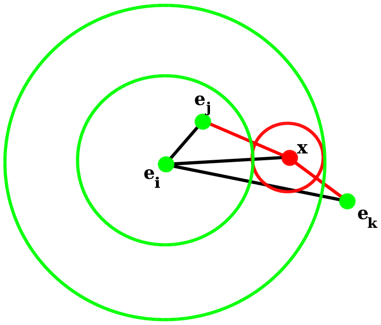

Fig. 4. Two-dimensional illustration of the use of the triangle inequality to cull unnecessary distance tests: the new input is $\mathbf{x}$, which has been tested against the exemplar chosen for the previous frame $\mathbf{e}_i$. We assume that the entire matrix of inter-exemplar distances is available. Exemplar $\mathbf{e}_j$ can rejected for being too close to $\mathbf{e}_i$, while $\mathbf{e}_k$ can be rejected for being too far. Exemplars falling in the annular region between the two large circles cannot be rejected and must be tested against $\mathbf{x}$.

the fact that we must maintain the symmetric matrix of distances between all the exemplars; when a new image is added to the catalog, we must tabulate the distances to all the other images in the catalog.

4 Distance Metrics

To this point, we have been deliberately vague about what we mean by the "distance" between two images; a common approach is to treat the images as points in an N-dimensional space (where N is the number of pixels, and the value of each pixel is the coordinate), and compute the standard Euclidean distance, as is done in [8]. This metrics, however, does not capture our intuitive idea of visual similarity under variable illumination. Scaling an image by a constant factor does not generally affect the visual appearance, but can result in a large distance. Normalized cross-correlation is often used to compare images when we wish to ignore scale changes of this sort:

$$r(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} \; . \tag{1}$$

The normalized correlation itself does not obey the triangle inequality; however, recalling that the dot product is related to the angle between two vectors by $\mathbf{x}.\mathbf{y} = |\mathbf{x}||\mathbf{y}|\cos(\theta)$, we use the correlation to compute an angular distance measure:

$$d(\mathbf{x},\mathbf{y}) = \arccos(\; r(\mathbf{x},\mathbf{y}) \;) \; . \tag{2}$$

We can visualize this for the case of three dimensional vectors: if each vector is projected to a point on the unit sphere, and the angle between the two vectors corresponds to the arc length of the great circle joining the two points. Because the triangle inequality holds on the sphere, the culling algorithm described above can be used with this distance measure.

5 Eye State Model

The primary parameters of interest in our application are the angles describing the rotational state of the eye within the orbit, which together with the position and orientation of the head determine the gaze vector in the world. The primary features we will use to determine these angles are the inner and outer margins of the iris, known as the *pupil* and the *limbus*, respectively. While the radius of the limbus is constant for a particular subject, the radius of the pupil varies in response to ambient light level, and is subject to continuous fluctuations due to the under-damped nature of the neural control system. Here we assume that pupil and limbus are concentric circles in the plane of the iris.

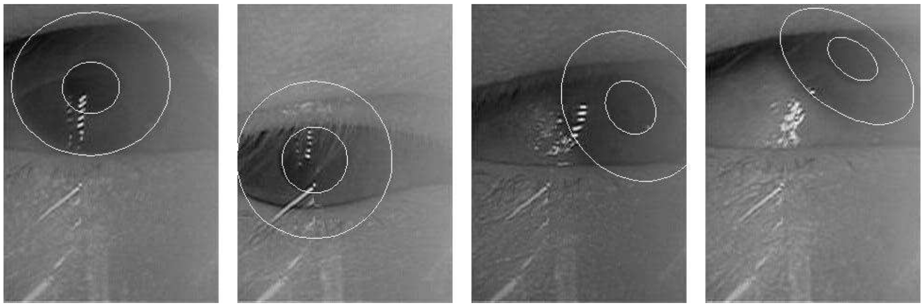

Fig. 5. Images of the eye with a rendering of a hand-tuned pupil-limbus model superimposed. The model has 5 global parameters (fixed for all images from a given run), and 3 parameters set on a frame-by-frame basis, consisting of two gaze angles and the pupil radius.

The problem is complicated by the fact that the pupil is not viewed directly, but lies behind the cornea, which is the eye's primary refracting surface. The effect of this is to increase the apparent size and reduce the apparent distance of the pupil relative to its physical location [9]. We approximate the appearance by a model with no refraction, with a tunable parameter for the depth separation between the planes containing the pupil and limbus. Other parameters which have a fixed value for all the images from a given subject are the limbus radius, and the distance of the plane of the limbus from the center of rotation. When the optical axis of the eye is directed toward the camera, the pupil and limbus appear as concentric circles in the image, and the location of their common center provides another pair of parameters, which are constant as long as there is no relative motion between the camera and the head. As gaze deviates from this direction, the images pupil and limbus are foreshortened and are well-fit by ellipses; lines drawn through the minor axes of these ellipses all intersect at the center point, which we use to find the center in our hand-labeling procedure. Figure 5 shows several images labeled with the pupil-limbus model. In addition to the iris parameters, we also label the eyelid margins in a separate labeling procedure.

The direction of gaze is the variable of primary interest for our application; while the tree-structure imposed on our set of images was based on overall image similarity (as captured by the correlation), it is our intuition that images which are similar will have similar gaze directions, and so the leaves of our tree can be associated with compact neighborhoods in gaze space. Note that the converse is not true: there can be images corresponding to the same direction of gaze which are very dissimilar, either because of lighting variations or a change in eyelid posture.

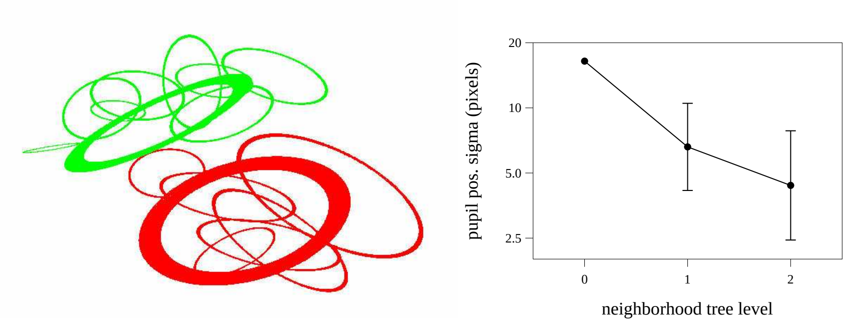

Fig. 6. Left panel: Ellipses depicting the variability in pupil position of child node exemplars for two first-level nodes (thick ellipses), and their corresponding second-level child nodes (thin ellipses). Right panel: Summary of variability in pupil position among subordinate nodes is shown for the first three levels in the hierarchy. The error bars show the standard deviation between the groups at the same level. (There are no error bars for the root node, because there is only one group.) The results validate our intuition that as image similarity increases, variation in gaze direction decreases.

To validate our intuition, we computed the variability in pupil position among the child exemplar images for the first three levels in the hierarchy for a single run (see figure 6). To generate this figure, 653 images were hand-labeled, consisting of the root node and the first three levels of the tree. For each image, we computed the position of the pupil center from the stored model parameters. For each node in the tree, we computed the mean pupil position over that node's children, and the corresponding standard deviations in x and y, and the covariance between the x and y deviations. In the left panel of figure 6, these derived measures are represented as ellipses, showing the scatter (in image space) of the pupil position in subordinate nodes. In the figure, 2 (of 14) first level nodes are represented, along with their all of their subordinate second-level nodes.

A crude univariate measure was formed by taking the Pythagorean sum of the x and y standard deviations, which is plotted for all the nodes on the right side of figure 6. At level 0, there is only a single node (the root of the tree); the average deviation among its children is plotted as the left-most point in figure 6. There are 14 level 1 nodes; for each of these we perform the same calculation over its children; we then compute the mean over the 14 nodes, plotting 1 standard deviation of this mean as an error bar in figure 6. Although the grouping was done on the basis of overall image appearance without regard to pupil position, we see from the data that the gaze directions do become more tightly clustered as we descend the tree.

6 Summary

We have described a method for decomposing a collection of images into subsets based on similarity of appearance; the resulting set of exemplars spans and uniformly samples the original collection, and is useful for application of techniques such as Active Shape Modeling which require hand-labeling of a training set. We have obtained useful results applying this process to a data set of eye images collected outdoors with uncontrolled illumination.

References

1. Cootes, T.F., Taylor, C.J.: Statistical models of appearance for medical image analysis and computer vision. In: Proc. SPIE Med. Imag. 2001. Volume 1. (2001) 236–248
2. A. Gersho and R. M. Gray: Vector Quantization and Signal Compression. Kluwer Academic Publishers (1992)
3. R. H. S. Carpenter: Movements of the Eyes. Pion (1977)
4. E. Kowler: Eye Movements and Their Role in Visual and Cognitive Processes. Elsevier (1990)
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **IT-13** (1967) 21–27
6. R. O. Duda and P. E. Hart and D. G. Stork: Pattern Classification. John Wiley and Sons (2001)
7. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323
8. Christoudias, C.M., Darrell, T.: On modelling nonlinear shape-and-texture appearance manifolds. In Schmid, C., Soatto, S., Tomasi, C., eds.: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). (2005) 1067–1074
9. Ohno, T., Mukawa, N., Yoshikawa, A.: Freegaze: a gaze tracking system for everyday gaze interaction. In Duchowski, A.T., Vertegaal, R., Senders, J.W., eds.: Proc. Eye Tracking Research & Applications Symposium (ETRA), ACM (2002) 125–132