Chapter 1

Evaluation of linear combination of views for object recognition on real and synthetic datasets

Vasileios Zografos and Bernard F. Buxton

Department of computer science,
University College London,
Malet Place, London, WC1E 6BT
{v.zografos,b.buxton}@cs.ucl.ac.uk

In this work, we present a method for model-based recognition of 3d objects from a small number of 2d intensity images taken from nearby, but otherwise arbitrary viewpoints. Our method works by linearly combining images from two (or more) viewpoints of a 3d object to synthesise novel views of the object. The object is recognised in a target image by matching to such a synthesised, novel view. All that is required is the recovery of the linear combination parameters, and since we are working directly with pixel intensities, we suggest searching the parameter space using a global, evolutionary optimisation algorithm combined with a local search method in order efficiently to recover the optimal parameters and thus recognise the object in the scene. We have experimented with both synthetic data and real-image, public databases.

1.1. Introduction

Object recognition is one of the most important and basic problems in computer vision and, for this reason, it has been studied extensively resulting in a plethora of publications and a variety of different approaches[a] aiming to solve this problem. Nevertheless accurate, robust and efficient solutions remain elusive because of the inherent difficulties when dealing in particular with 3d objects that may be seen from a variety of viewpoints. Variations in geometry, photometry and viewing angle, noise, occlusions and incomplete data are some of the problems with which object recognition systems are faced.

In this paper, we will address a particular kind of extrinsic variations: variations of the image due to changes in the viewpoint from which the object is seen. Traditionally, methods that aimed to solve the recognition problem for objects with varying pose relied on an explicit 3d model of the object, generating 2d projections from that model and comparing them with the scene image. Such was the work

[a] For a comprehensive review of object recognition methods and deformable templates in particular, see Refs. 1–4.

by Lee and Ragnarath.[5] Although 3d methods can be quite accurate when dealing with pose variations, generating a 3d model can be a complex process and require the use of specialised hardware. Other methods[6,7] have thus tried to capture the viewpoint variability by using multiple views of the object from different angles, covering a portion of, or the entirety of, the view sphere. If the coverage is dense these methods require capture and storage of a vast number of views for each object of interest. Quite recently, new methods have been introduced that try to alleviate the need for many views while still working directly with 2d images.They are called *view-based* methods and represent an object as a collection of a small number of 2d views. Their advantage is that they do not require construction of a 3d model while keeping the number of required stored views to a minimum. Prime examples are the works by Bebis et al.[8] and Turk and Pentland.[9]

Our proposed method is a view-based approach working directly with pixel values and thus avoids the need for low-level feature extraction and solution of the correspondence problem such as in Ref. 8. As a result, our model is easy to construct and use, and is general enough to be applied across a variety of recognition problems. The disadvantage is that it may also be sensitive to illumination changes, occlusions and intrinsic shape variations.[10] We adopt a "generate and test" approach using an optimisation algorithm to recover the optimal linear combination of views (LCV) coefficients that synthesise a novel image which is as similar as possible to the target image. If the similarity (usually the cross-correlation coefficient) between the synthesised and the target images is above some threshold then an object is determined to be present in the scene and its location and pose are defined (at least in part) by the LCV coefficients.

In the next section we introduce the LCV and explain how it is possible to use it to synthesise realistic images from a range of viewpoints. In section 1.3 we present our 3d object recognition paradigm which incorporates the LCV and the optimisation solution, and in section 1.4 we show some experimental results of our approach on synthetic and real imagery. Finally, we conclude in section 1.5 with a critical evaluation of our method and suggest how it could be further improved in the future.

1.2. Linear combination of views

LCV is a technique which belongs in the general theory of the tri- and multi-focal tensors , or Algebraic Function of View (AFoV)[11] and provides a way of dealing with variations in an object's pose due to viewpoint changes. This theory is based on the observation that the set of possible images of a set of landmarks points on an object undergoing 3d rigid transformations and scaling is, under most (i.e. affine) imaging conditions, to a good approximation embedded in a linear space spanned by a small number of 2d images of the landmark points. With the aid of an additional assumption as to how to combine the pixel intensities in the 2d images, it follows that the variety of 2d views depicting an object can be represented by a combination of a small number of 2d *basis views* of the object.

Ullman and Basri[12] were the first to show how line drawings or edge map images of novel views of a 3d object could be generated via a linear combination of similar 2d basis views. More specifically, they showed that under the assumption of orthographic projection and 3d rigid transformations, 2 views are sufficient to represent any novel view of a polygonal object from the same aspect. The proof may easily be extended to any affine imaging condition. Thus, to a good approximation, given two images of an object from different (basis) views I' and I'' with corresponding image coordinates (x', y') and (x'', y''), we can represent any point (x, y) in a novel, target view I_T according to, for example:

$$\begin{array}{c} x = a_0 + a_1 x' + a_2 y' + a_3 x'' \\ y = b_0 + b_1 x' + b_2 y' + b_3 x'' \end{array} . \tag{1.1}$$

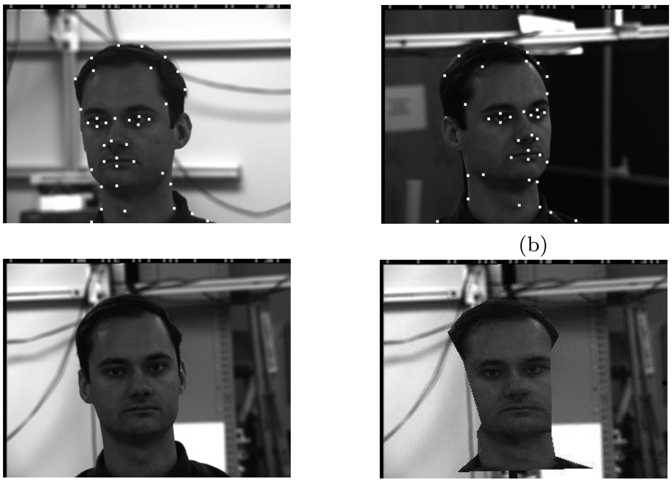
The target view is reconstructed from the above two equations given a set of valid coefficients (a_i, b_j). Provided we have at least 4 corresponding landmark points in all three images (I_T, I', I'') we can estimate the coefficients (a_i, b_j) by using a standard least squares approach. Based on a method for weighting the combination of the intensities (or colours) of corresponding points in the basis views I' and I'', several others have taken this concept further from its initial application to line images and edge maps to the representation of real images I_T.[8,13–15]

Such results suggest that it is possible to use LCV for object recognition in that target views of an object can be recognised by matching them to a combination of stored, basis views of the object. The main difficulty in applying this idea within a pixel-based approach is the selection of the LCV coefficients (a_i, b_j). In particular, as described in the next section, synthesis of an image of a novel view from the images of the basis views, although straightforward, is a non-linear and non-invertible process.

1.2.1. *Image synthesis*

To synthesise a single, target image using LCV and two views we first need to determine its geometry from the landmark points. In principle we can do so by using (1.1) and n corresponding landmark points (where $n \geqslant 4$), and solving the resulting system of linear equations in a least squares sense. This is straightforward if we know, can detect, or predict the landmark points in image I_T. Such methods may therefore be useful for image coding and for synthesis of target views of a known object.[13,14] For pixel-based object recognition in which we wish to avoid feature detection a direct solution is not possible, but we instead use a powerful optimisation algorithm to search for and recover the LCV coefficients for the synthesis.

Given the geometry of the target image I_T, in a pixel-based approach we need to synthesise its appearance (colour, texture and so on) in terms of the basis images I' and I''. Since we are not concerned here with creation of a database of basis views of the objects of interest, we may suppose that a sparse set of corresponding landmark points $(x'(j), y'(j))$ and $(x''(j), y''(j))$ may be chosen manually and offline in images I' and I'' respectively and used to triangulate the images in a consistent manner. An illustration of the above can be seen in Fig. 1.1.



(c) (d)

Fig. 1.1. Example of real data from the CMU PIE database. The two basis views (a) and (b) and the target image (c). The synthesised image (d) is at the correct pose identified by our algorithm. Note that in (d) the face is missing some information (around the ears for example). This is because these areas do not appear in both basis views due to self-occlusion and cannot be modelled by the two images alone.

Given a set of hypothesised landmark points $(x(j), y(j))$ in the target image we can, to a good approximation, synthesise the target image I_T as described in Refs. 10,13,16 from a weighted linear combination:

$$I_T(x,y) = w'I'(x',y') + w''I''(x'',y'') + \epsilon(x,y) = I_S(x,y) + \epsilon(x,y), \tag{1.2}$$

in which the weights w' and w'' my be calculated from the LCV coefficients to form the synthesised image I_S. Essentially this relies on the fact that, in addition to the multi-view image geometry being to a good approximation affine, the photometry is to a good approximation affine or linear.[17] The synthesis essentially warps and blends images I' and I'' to produce I_S. It is important to note therefore that (1.2) applies at all points (pixels) (x, y), (x', y') and (x'', y'') in images I_S, I' and I'' with the dense correspondence defined by means of the LCV equations (1.1) and a series of piecewise linear mappings[18] within each triangle of the basis images. If (x', y') and (x'', y'') do not correspond precisely to pixel values, bilinear interpolation is used.[13,14] The same idea may be extended to colour images by treating each spectral band as a luminance component (e.g. I_R, I_G, I_B).

1.3. The recognition system

In principle using the LCV for object recognition is easy. All we have to do is find the LCV coefficients in an equation such as (1.1) which will optimise the sum of squared errors ϵ from (1.2) and check if it small enough, or our synthesised and target images I_S and I_T are sufficiently similar, to enable us to say that they match.

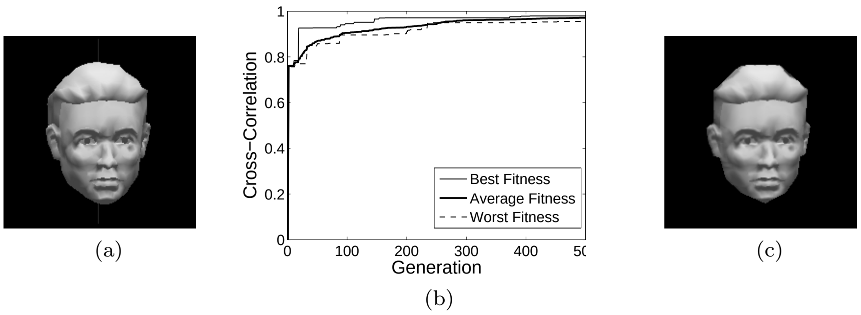


Fig. 1.2. Example of a synthetic target image used for testing (a). The average test results are shown in (b). The image produced by the LCV method to match the target (a) is shown in (c).

1.3.1. *Template matching*

The first component of our system is the two stored basis views I' and I''. These are rectangular bitmap images that contain gray-scale (or colour), pixel information of the object without any additional background data. The images are obtained from basis views chosen, as indicated earlier, so that the viewpoint from which the target image I_T is taken lies on the view sphere between or almost between the basis views from which I' and I'' are taken. It is important not to choose a very wide angle between the basis views since this can lead to I' and I'' belonging to different aspects of the object and thus to landmark points being occluded[b]. Having selected the two basis views, we pick a number of corresponding landmark points in particular lying on discontinuity boundaries, edges and other prominent features. When the appropriate number of landmarks have been selected we use constrained Delaunay triangulation to produce consistent and corresponding triangular meshes of all the images. The above processes may be carried out during an offline training stage and are not examined here. The recognition system involves choosing the appropriate LCV coefficients (a_i, b_j), synthesising an image I_S and comparing it with the target image I_T, using some similarity or dissimilarity metric. The synthetic image of the object is compared from (1.1) and (1.2) only over the region covered by the object. However, in order to make a probabilistic interpretation of the match, the

[b] It is still quite possible to synthesise novel images at wider angles and remove any self-occluded triangles, although we do not address this problem here, see Ref. 14.

synthesised image must be superimposed on the background as shown in Fig. 1.1(d) and all the pixels compared, such as in Ref. 19. The background must therefore be known as it is in the CMU PIE database,[20] or very simple (e.g. a uniform black background as in the COIL-20 database[21]) or itself calculated from an appropriate model. Making the comparison over all pixels belonging to both the foreground and background in this way means that either a dissimilarity metric such as the sum of squared differences (SSD) or a similarity measure such as the cross-correlation coefficient $c(I_T, I_S)$ may be used, without generating spurious solutions for example, when the area of the foreground region covered by the object shrinks to zero.[22] We have used the latter because when applied to the whole image it is invariant to affine photometric transformations.[22] The choice of LCV coefficients is thus determined by maximising the cross-correlation coefficient:

$$\min_{a_i, b_j}(1 - c(I_T, I_S)). \tag{1.3}$$

Essentially we are proposing a flexible template matching system, in which the template is allowed to deform in the LCV space until it matches the target image.

1.3.2. *Optimisation*

To find the LCV coefficients (a_i, b_j) we need to search a high-dimensional parameter space using an efficient optimisation algorithm. For this purpose, we have chosen a hybrid method, which combines a global (albeit slower) stochastic algorithm with a local, direct search approach. The idea is that when we find a good-enough solution with the stochastic method (usually after a pre-determined number of function evaluations) and we are inside the basin of attraction of the optimal solution, we can switch over to the local method to refine the results and quickly reach the optimum.

The stochastic method used is a recent evolutionary, population-based optimisation algorithm that works on real-valued coded individuals and is capable of handling non-differentiable, nonlinear and multi-modal objective functions. It is called Differential Evolution (DE) and was introduced by Storn and Price.[23] Briefly, DE works by adding the weighted difference between two randomly chosen population vectors to a third vector, and the fitness of the solution represented by the resultant is compared with that of another individual from the current population. In this way, in DE we can deduce from the distances between the population vectors where a better solution might lie, thereby making the optimisation self-organising. In addition, it is efficient in searching high-dimensional spaces and is capable of finding promising basins of attraction[22] early in the optimisation process without the need for good initialisation.

For the local method, we have selected the algorithm[c] by Nelder and Mead,[24] since it is very simple to implement and its use does not require calculation (or

[c] Also known as the downhill simplex method or simplex method. It is not to be confused with the simplex algorithm for the solution of the linear programming problem.

approximation) of first or second order derivatives. A simplex is a polytope of N+1 vertices in N dimensions with each vertex corresponding to a single matching function evaluation. In its basic form (as described by Nelder and Mead) the simplex is allowed to take a series of steps, the most common of which is the *reflection* of the vertex having the poorest value of the objective. It may also change shape (*expansion* and *contraction*) to take larger steps when inside a valley or flat areas, or to squeeze through narrow cols. It can also change direction (*rotate*) when no more improvement can be made in a current path. Since the simplex is a local, direct search method, it can become stuck in local optima and therefore some modifications of its basic behaviour are necessary. The first modification we introduced was the ability of the simplex to *restart* whenever it stalled inside a local optimum. The restart works as follows. After a specific number of function evaluations where there has been no change in the value of the tracked optimum, we keep the best vertex P_0 and we generate n new vertices P_i using the formula:

$$P_i = P_0 + \lambda e_i, \tag{1.4}$$

where e_i are n random unit vectors, $i = 1, .., n$ and λ is a constant that represents the step-size. The idea is that by restarting the simplex close to the best point P_0 we can escape a local optimum but without jumping too far away from the last good solution that we have found. We soon discovered that any fixed step λ will eventually become too big as the algorithm progresses, and the simplex will keep jumping in and out of a good optimum without making any significant improvement for the remaining function evaluations. We therefore allowed λ to reduce as the algorithm progressed using the reduction schedule (typically met in Simulated Annealing):

$$\lambda = \lambda_0 C^{(k-1)} \tag{1.5}$$

where k is the current function evaluation, and C is the "cooling rate". In this way, when the algorithm first stalls, it makes big jumps to attempt to escape from the local optimum and as it progresses the jumps become smaller and smaller, so that the algorithm tries to "burrow" deeper into the basin of attraction. As a result, the algorithm keeps on improving the location of the optimum unlike the fixed-step version which stalls early in the optimisation process. We can see both these two methods illustrated in Fig. 1.3.

As mentioned above, the reason for using a hybrid approach as opposed to the global, stochastic method alone, is that we can get very close to the optimum solution in many fewer iterations. This is because evolutionary methods, although they find a "fairly-good" solution early in the optimisation process, they spend the remainder of the function evaluation "budget" carrying out small improvements in the recovered optimum. If we therefore switch to the local, deterministic method once a fairly good solution is recovered by the stochastic method, we can get to a near-globally optimal solution much earlier. The comparison between a stochastic-only and hybrid optimisation methods can be seen in Fig. 1.4.

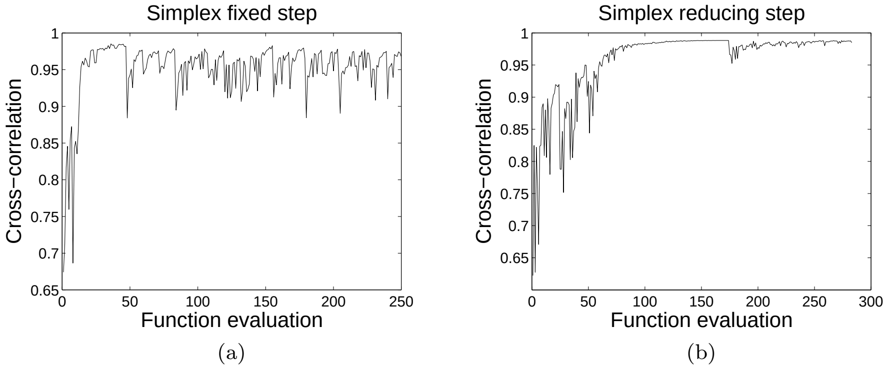


Fig. 1.3. Comparison between a fixed step simplex method (a) and a reducing step variant (b). In the latter, the algorithm keeps on improving the object recognition.

1.4. Experimental results

We performed a number of experiments on synthetic and real images under varying camera angle. The synthetic images were generated by taking 2d snapshots of a 3d object (a human head model) in front of a black background (see Fig. 1.2(a)). Landmarks where manually selected amongst the vertices of the 3d object and their projected positions were automatically calculated in the 2d images. This way we could eliminate the approximation errors associated with manual placement of corresponding landmarks in the two basis views and have control over the projection model (in this case orthographic projection). Our synthetic dataset consisted of a number of pose angles between $\pm 14^o$ about the vertical and $\pm 10^o$ about the horizontal axes. The majority of the target views lay on the portion of the viewsphere between the basis views, but in a few examples the system had to extrapolate away from the great circle on the viewsphere between the basis views, in order to recover the optimal coefficients. In terms of optimisation complexity, these synthetic examples are considered quite simple since we are dealing with an object with diffuse (Lambertian) reflectivity, which is fairly convex (i.e. not self-occluding, at least over the range of angles we are testing), under constant lighting and distance from the camera, and there is no approximation error on the landmarks of the two basis views. In addition, the object is imaged against a constant background, which produces a convex error surface with a wide basin of attraction. The optimum solution in such cases can be easily and efficiently recovered. Therefore, we only needed to carry out a few experiments on this dataset in order to determine whether the method works in principle or not. In total, we ran 10 synthetic experiments and the results are illustrated in Fig. 1.2(b). These results are very encouraging with the majority of the experiments converging to the correct solution with a cross-correlation of > 0.97. The only cases which failed to converge to the correct optimal solution occurred when the target viewpoint was far from the great circle in view space between the basis views. In such cases, the LCV could not synthesise the target view accurately indicating the need to use more than two basis views in order to better represent that portion of the view-sphere.

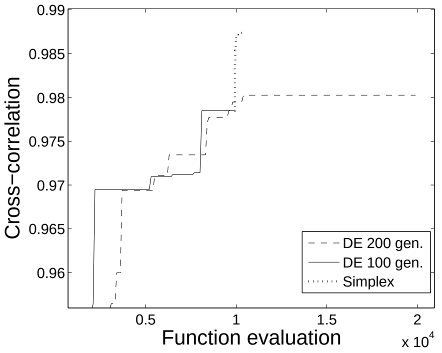


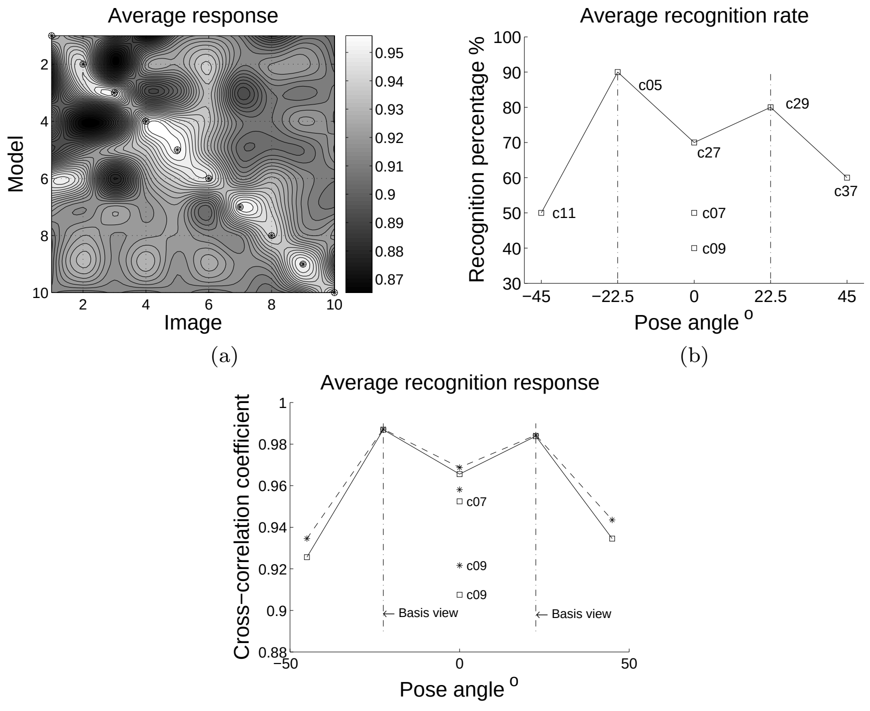
Fig. 1.4. The optimisation results for a DE-only test run (200 generations) compared with that of a DE (100 generations)+Simplex test. It is obvious that we can obtain better results in the latter case with many fewer iterations.

1.4.1. *Experiments on the CMU PIE database*

For the real image experiments, we used two publicly available datasets: the CMU PIE database[20] and the COIL-20 database.[21] The CMU PIE contains examples of facial images from various individuals across different pose, illumination and expression conditions. In our tests, we used pose variation subsets, making sure the manually chosen landmarks were visible in both basis views (see Fig. 1.1). We constructed LCV models from 10 individuals using as basis views the left and right images (*c29*, *c05*) of each individual with a natural expression (see Fig. 1.8(a)). The face, once synthesised, was then superimposed onto the background which is given separately in the database, and the resulting image was compared with a test, target view. Comparisons were carried out against the images of the 10 individuals in the database, while attempting to detect poses from $-45^o, -22.5^o, 0^o, 22.5^0, 45^o$ about the vertical and a limited range about the horizontal axes (images *c09* and *c07*).

In total we carried out 700 experiments across pose and constructed a $10 \times 10 \times 7$ "confusion array" of model×image×pose. Each 10×10 pose-slice of this array contains information about the recognition responses (cross-correlation) of our tests, the highest being along the main diagonal, where each individual's model is correctly matched to that individual's image. The recognition response should be less when comparing a specific model with images of other individuals. This behaviour, averaged over pose can be seen as a "heatmap" in Fig. 1.5(a) whilst the pose-dependent recognition rate and recognition response (averaged over the 10 models) are shown in Fig. 1.5(b) and (c) respectively. We can see from the high values (white) along the leading diagonal of the averaged "heatmap" Fig. 1.5(a) that for all 10 experiments, the calculated cross-correlation $c(I_S(i), I_T(j))$, where $I_S(i)$ is the image synthesised from the model of the ith object (i.e. its basis views) and $I_T(j)$ a target image of the jth object, is generally greatest when $i = j$. The response usually falls off for $i \neq j$, with some individuals being more similar than others (grey areas in the heatmap). For the average recognition rate, we checked to see if the highest response corresponded to a correct match between the model and an image of that individual at every pose. As expected, we had the highest recognition rates at the basis views ($\pm 22.5^o$) when no interpolation is necessary, slightly lower rates when we had to interpolate to find the frontal view (0^o) and still lower rates when extrapolation was required to synthesise the correct view at ($\pm 45^o$). The same reduction in recognition rate also applies to pose variation about the vertical axis (*c07,c09*). Examination of the average recognition response in Fig. 1.5(c) shows the expected "M-Shaped" curve, with the highest response being at the basis views, slightly lower for the frontal image (interpolation) and still lower for images taken from "outside" the basis views (extrapolation). The solid line shows the response for the tests in the leading diagonal of the confusion array (i.e. where the correct solutions lie) and the dashed line shows the maximum response in each column of the image×pose slice of the confusion array. Where there are large discrepancies between the solid line and the dashed-line, the recognition rate will be low, whilst there will be no difference if the recognition rate is 100%. Thus, at the two basis views where we have a high recognition rate, there is little difference between the solid and dashed lines, there is a slight difference for the frontal image and a bigger difference for the target images where we need to extrapolate beyond the basis views.

In general, the results are quite pleasing with the correct person identified the vast majority of times when the target view was between the basis views and no extrapolation was required. It is important to note that the recognition rate is not 100% at the two basis views as we might have expected it to be. There are many reasons for this, mainly the fact that we have carried out only 10 experiments per individual per pose, and therefore a single failure reduces the recognition rate to 90%. Also, the landmarking of the face images (and other objects to be discussed later in the COIL-20 database) is quite sparse (Fig. 1.1), resulting in particular errors around the extremal boundaries of the face images. Additionaly, the chosen objects (facial images) are quite similar to each other (facial features, skin tone, hair colour and so on) and in this case, unlike for the synthetic examples, we are dealing with a much more complex optimisation problem, partially caused by the cluttered background.[22] The lower than expeced recognition rates thus are a combination of the sparse landmarking, the limited number of experiments and the occassional failure of the optimisation algorithm to converge to the correct solution. It should be possible to increase the recognition rates by using more landmark points and views during the modelling stage, constructing models for a larger set of objects and thus making additional tests against more target images, possibly taken from a greater number of viewpoints, and also initialising the optimisation algorithm closer to the basin of attraction of the optimal solution.



(a) (b)

(c)

Fig. 1.5. A heatmap (a) showing the entries in the confusion array averaged over pose. (b) shows the recognition rate as a function of pose angle averaged over all models and (c) the recognition response at the same angles as in (b). Solid symbols are used for the best match and open squares for the correct match with i=j.

1.4.2. *Experiments on the COIL-20 database*

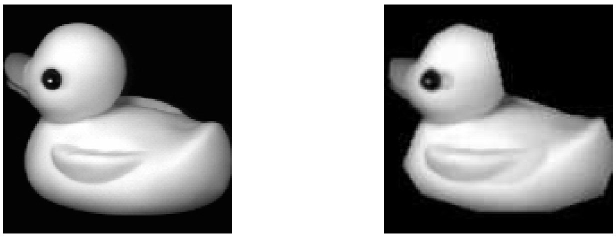
The COIL-20 database contains examples of 20 objects imaged under varying pose (horizontal rotation around the view-sphere at 5^o intervals) against a constant background, with the camera position and lighting conditions kept constant (see Fig. 1.8(b)). In this case, we created two “confusion arrays”, one using the cross-correlation coefficient and the other using the negative of the mutual information:

$$M(I_S, I_T) = -\sum p(I_S, I_T) \log \frac{p(I_S, I_T)}{p(I_S)p(I_T)}, \qquad (1.6)$$

where $p(I_S, I_T)$ is the joint p.d.f. and $p(I_S)$, $p(I_T)$ the marginal p.d.f.s of the synthesised and target images I_S and I_T respectively. In these calculations, the probability distributions were approximated by histograms computed from the images, akin to Ref 25. Mutual information was used to see if there was any advantage in using a matching measure that is known not to be dependent on a simple, direct mapping between the pixel attributes. For the mutual information measure, a low score (i.e. a negative number of larger magnitude) indicates a better match. In these experiments, we selected two basis views from images of half of the objects and tested the matching across all objects both those modelled (here labeled 1-10) and unmodelled (here labeled 11-20) and across 7 poses. The image confusion array was thus of dimensions 10×20×7 (model×image×pose). For the pose samples, we selected the basis images at -20^o and 20^o about the frontal view at 0^o, and tested between -30^o to 30^o at 10^o intervals. As a result, we have 3 images where we need to interpolate between the basis views $(-10^o, 0^o, 10^o)$ and two where extrapolation is required $(-30^o, 30^o)$.

In total, we carried out 2800 experiments, 1400 for each of the error measures $c(I_S(i), I_T(j))$ and $M(I_S(i), I_T(j))$. The results can be seen in Fig. 1.9 and Fig. 1.10. On average, the cross-correlation outperforms the mutual information measure. The results are also better than those obtained for the CMU PIE database since we have less similar objects, a much easier optimisation problem and also we have carried out more experiments. What is interesting to note from Fig. 1.9 is how much more distinctive the leading diagonal is in the heatmaps for the cross-correlation (b) than that for the mutual information (a). This indicates that the true positive responses are quite distinct from those of the true negatives and that there is less chance for miss-recognitions. The same conclusion may be drawn from the second half of the heatmap showing the scores obtained for matching the models (1-10) to images of the other objects (11-20). In this case, there are no very good matches to any of the models and all of the images are likely to be rejected as unrecognised. The point that miss-recognition is unlikely is also reinforced by (c),(d),(e) and (f) in Fig. 1.10 which show the matching scores and recognition rates at different pose angles, averaged over the modelled objects as in Fig. 1.5(b) and (c).

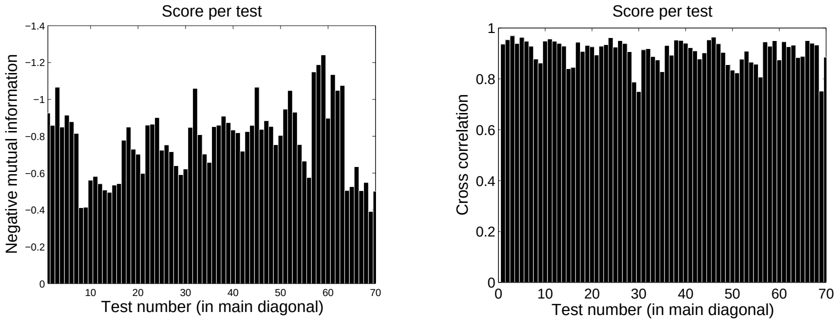
Furthermore, the heatmaps (a) and (b) in Fig. 1.10 show the average response over all the modelled objects at different pose angles. In the case of mutual information, we would expect lower scores (i.e. valley) at a pose of 0^o, which actually occurs, and minima at the basis views at $\pm 20^o$ which do not occur. This is perhaps because when reconstructing at one basis view, there are some "ghosting" (Fig. 1.6) effects from the other basis view that affect pixel intensities even though the geometry is correct. It seems that mutual information is more sensitive to this effect than cross-correlation. Also we notice in Fig. 1.10(a) and (b) some spreading of high and low response values across pose. This spreading is very significant and explains how well a model matches to other objects (including itself) at different poses. For example, a generic looking object (e.g. a box) that can easily "morph" under the



(a) (b)

Fig. 1.6. This figure shows the ghosting effects that might occur from warping the intensities of the two basis views. (a) is the target view that we are trying to reconstruct and (b) is the synthesised image. As we can see the geometry is correct but there are some ghosting effects in (b) from one of the basis views, in particular just behind the eye and behind the wing of the image of the toy duck. This usually occurs when a few triangles cover large, detailed areas of the image. This problem can be usually remedied by using more landmark points, and thus triangles, in those areas.

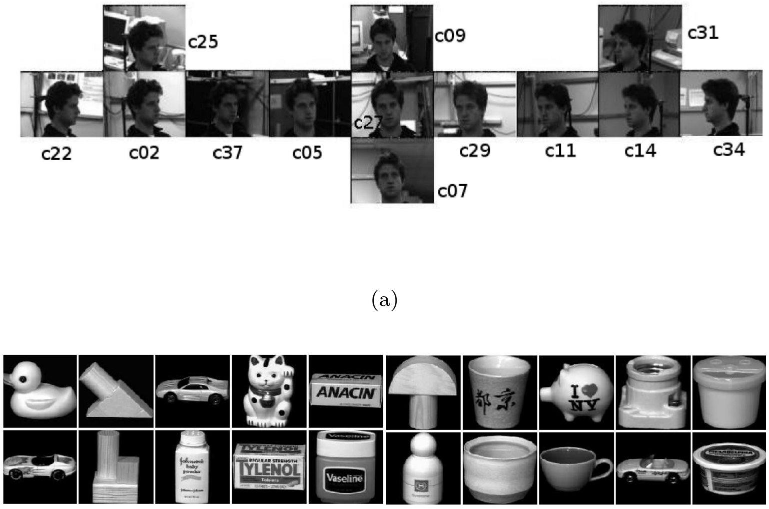
LCV mapping (1.1) to match the shape of the images of many objects from the database, will show a spread of high values (for mutual information) across pose angles. On the other hand, a complex object with unique geometry and intensity will show a spread of low values (again for mutual information).



(a) (b)

Fig. 1.7. Barplots showing the scores per test in the main diagonal, arranged by pose, both for the mutual information (a) and the cross-correlation (b). A "good" measure should exhibit an almost constant value, such as the cross-correlation in (b).

We finally include two bar plots comparing the cross-correlation and mutual information measures (Fig. 1.7). These show the response values per test arranged by pose and model, where the model is chosen to be the correct one for the target image, i.e. along the leading diagonal of the confusion array. What we must note here is that an "appropriate" matching measure should give consistently good responses throughout all the tests, both as the viewing angle of the target images for a given object changes and as we change from object to object, resulting in a uniform distribution of the response values. This is in fact the case for the cross-correlation



(a)

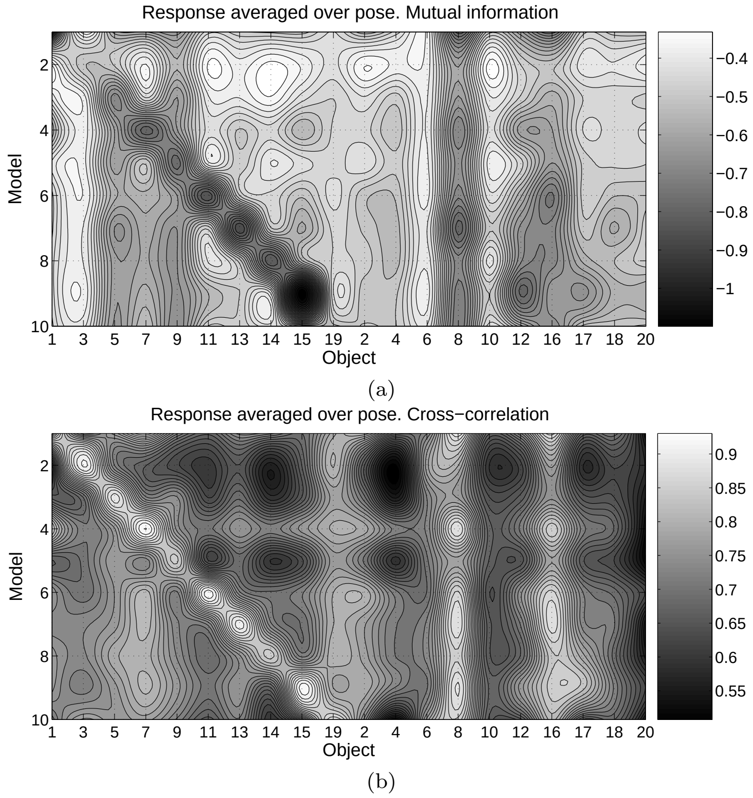
(b)

Fig. 1.8. Image samples from the CMU PIE (a) and COIL-20 (b) databases.

(b) for which the distribution of the matching scores appears quite uniform, but not so for the mutual information (a), a further indication that cross-correlation is the appropriate matching metric for object recognition by image-based template matching.

1.5. Conclusion

We have shown how the linear combination of views (LCV) method may be used in view-based object recognition. Our approach involves synthesising intensity images using LCV and comparing them to the target, scene image. The LCV coefficients for the synthesis are recovered by a hybrid optimisation algorithm, comprised of differential evolution[23] combined with the simplex method.[24] Experiments on both synthetic and real data from the CMU PIE and COIL-20 databases, demonstrate that the method works well for pose variations especially those where the target view lies between, or almost between the basis views. DE plays an important role in our method, by searching efficiently the high-dimensional, LCV space. Such an algorithm can narrow the search space to a promising area within the basin of attraction of a good solution, in which a local optimisation method can be used for finding an accurate solution. For objects from the COIL database, we also compared the use of cross-correlation and mutual information for intensity-based



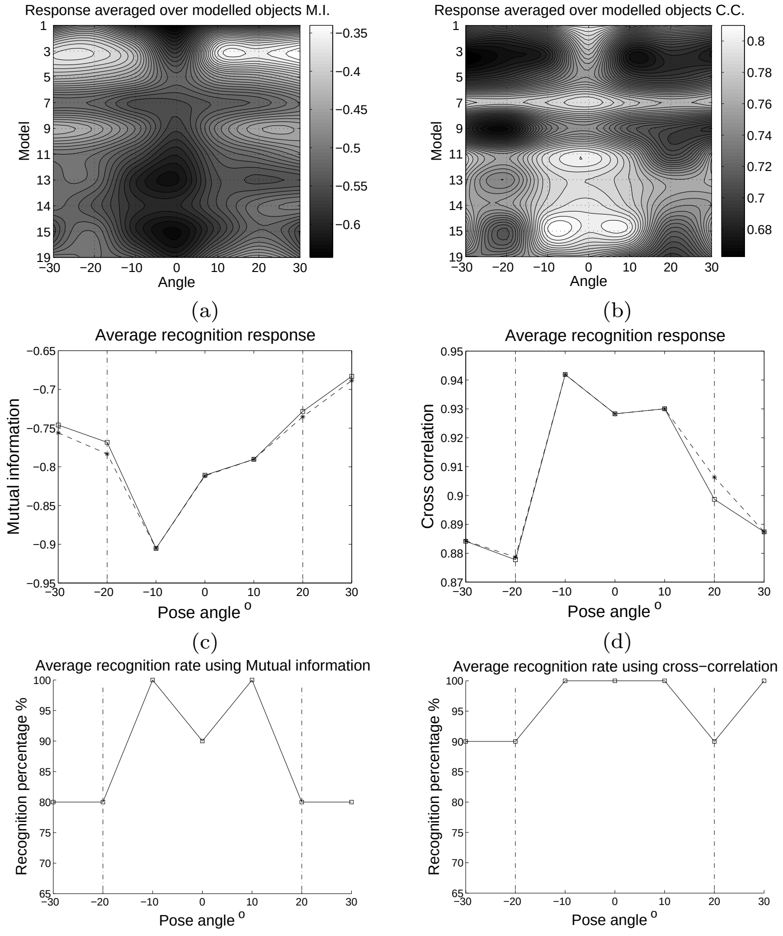
(a)

(b)

Fig. 1.9. Analysis of the results obtained from experiments on the COIL-20 database using cross-correlation and mutual information error measures. (a) shows the 10x20 matrix of responses averaged over the pose for the mutual information measure and (b) shows similar results for the cross-correlation. It is obvious that in the case of the cross-correlation (b) the resulting leading diagonal in the first half of the heatmap is more distinctive than that of the mutual information.

template matching. We have seen from our experiments on real data that the cross-correlation slightly outperforms the mutual information measure. This is possibly because of the type of error surfaces it produces especially around the basin of attraction.

Additional work is required, however. In particular, we would like to reformulate (1.1) by using the affine tri-focal tensor and introducing the appropriate constraints in the LCV mapping process. Formulating (1.1) in term of individual 3d transforms might also help bound the range of the LCV coefficients and make initialisation of the optimisation algorithm more intuitive. Furthermore, we would like to introduce probabilistic weights on the coefficients as prior information about the range of likely views and formulate a Bayesian inference mechanism. This, we believe, will greatly aid the recognition process. At this stage we have only addressed extrinsic, viewpoint variations, but we have indicated how it should be possible to include



(a) (b) (c) (d) (e) (f)

Fig. 1.10. Some additional results from the COIL-20 database. Plots (a) and (b) show the 10x10 matrix of responses averaged over the modelled objects for the mutual information and cross-correlation measures respectively. (c) and (d) show the average recognition response for the two error measures and (e) and (f) the similarly averaged recognition rates.

intrinsic, shape variations (see for example Ref. 10.) and lighting variations on the image pixels.

References

1. A. K. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly, Deformable template models: A review, *Signal Processing.* **71**(2), 109–129, (1998).
2. A. R. Pope. Model-based object recognition. a survey of recent research. Technical Report 94-04, (1994).

3. M.-H. Yang, D. J. Kriegman, and N. Ahuja, Detecting faces in images: A survey, *IEEE Pattern Analysis and Machine Intelligence.* **24**(1), 34–58, (2002).
4. P. J. Besl and R. C. Jain, Three-dimensional object recognition, *ACM Computing Surveys (CSUR).* **17**, 75–145, (1985).
5. M. W. Lee and S. Ranganath, Pose-invariant face recognition using a 3d deformable model, *Pattern Recognition.* **36**, 1835–1846, (2003).
6. Y. Lamdan, J. Schwartz, and H. Wolfson, On recognition of 3d objects from 2d images, *Proceedings of the IEEE International Conference on Robotics and Automation.* pp. 1407–1413, (1988).
7. D. J. Beymer. Face recognition under varying pose. In *Proc. IEEE Conf. CVPR*, pp. 756–761, (1994).
8. G. Bebis, S. Louis, T. Varol, and A. Yfantis, Genetic object recognition using combinations of views, *IEEE Transactions on Evolutionary Computation.* **6**(2), 132–146 (April, 2002).
9. M. Turk and A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience.* **3**(1), 71–86, (1991).
10. M. B. Dias and B. F. Buxton, Implicit, view invariant, linear flexible shape modelling, *Pattern Recognition Letters.* **26**(4), 433–447, (2005).
11. A. Shashua, Algebraic functions for recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **17**(8), 779–789, (1995).
12. S. Ullman and R. Basri, Recognition by linear combinations of models, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **13**(10), 992–1006 (October, 1991).
13. I. Koufakis and B. F. Buxton, Very low bit-rate face video compression using linear combination of 2dfaceviews and principal components analysis, *Image and Vision Computing.* **17**, 1031–1051, (1998).
14. M. E. Hansard and B. F. Buxton, Parametric view-synthesis, *In Proc. 6th ECCV.* **1**, 191–202, (2000).
15. G. Peters and C. von der Malsburg, View reconstruction by linear combination of sample views, *In Proc. British Machine Vision Conference BMVC 2001.* **1**, 223–232, (2001).
16. B. F. Buxton, Z. Shafi, and J. Gilby. Evaluation of the construction of novel views by a combination of basis views. In *Proc. IX European Signal Processing Conference (EUSIPCO-98)*, Rhodes, Greece, (1998).
17. A. Shashua. *Geometry and Photometry in 3D Visual Recognition.* PhD thesis, Massachusetts Institute of Technology (November, 1992).
18. A. Goshtasby, Piecewise linear mapping functions for image registration. **19**(6), 459–466, (1986).
19. J. Sullivan, A. Blake, M.Isard, and J.MacCormick, Bayesian object localisation in images, *Int. J. Computer Vision.* **44**(2), 111–136, (2001).
20. T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination and expression (PIE) database. In *Proc. of the 5th IEEE international conference on automatic face and gesture recognition*, (2002).
21. S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical Report CUCS-006-96, Department of computer science, Columbia University, New York, N.Y. 10027, (1996).
22. B. Buxton and V. Zografos. Flexible template and model matching using image intensity. In *Proceedings Digital Image Computing: Techniques and Applications (DICTA)*, (2005).
23. R. Storn and K. V. Price, Differential evolution - a simple and efficient heuristic for

global optimization overcontinuous spaces, *Journal of Global Optimization.* **11**(4), 341–359 (December, 1997).

24. J. A. Nelder and R. Mead, A simplex method for function minimization, *Computer Journal.* **7**, 308–313, (1965).
25. F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, Multimodality image registration by maximization of mutual information, *IEEE Transactions on Medical Imaging.* **16**(2), 187–198, (1997).

Index

basis views, 3

conclusion, 14

equations
  cross-correlation, 6
  LCV, 3
  mutual information, 11
  new vertex generator, 7
  reduction schedule, 7
  weigthed linear combination, 4
extrinsic variations, 1

image synthesis, 3
intrinsic variations, 16

linear combination of views, 2

piecewise linear mapping, 4

recognition system, 5
  optimisation, 6
    differential evolution, 6
    hybrid optimisation, 6
    Nelder and Mead algorithm, 6
    reduction schedule, 7
    restart, 7
    simplex, 6
  template matching, 5
results, 8
  CMU PIE, 9
  COIL-20, 11
  confusion array, 9
  ghosting effects, 13
  synthetic images, 8

tri-focal tensor, 2, 15

view-based methods, 2