On Affine Invariant Clustering and Automatic Cast Listing in Movies

Andrew Fitzgibbon and Andrew Zisserman

Visual Geometry Group
Department of Engineering Science, The University of Oxford
http://www.robots.ox.ac.uk/ vgg
http://www.robots.ox.ac.uk/ vgg

Abstract. We develop a distance metric for clustering and classification algorithms which is invariant to affine transformations and includes priors on the transformation parameters. Such clustering requirements are generic to a number of problems in computer vision.
We extend existing techniques for affine-invariant clustering, and show that the new distance metric outperforms existing approximations to affine invariant distance computation, particularly under large transformations. In addition, we incorporate prior probabilities on the transformation parameters. This further regularizes the solution, mitigating a rare but serious tendency of the existing solutions to diverge. For the particular special case of corresponding point sets we demonstrate that the affine invariant measure we introduced may be obtained in closed form.
As an application of these ideas we demonstrate that the faces of the principal cast of a feature film can be generated automatically using clustering with appropriate invariance. This is a very demanding test as it involves detecting and clustering over tens of thousands of images with the variances including changes in viewpoint, lighting, scale and expression.

1 Introduction

Clustering and classification problems abound in the applied sciences, in applications from citation indexing to the study of gene function. The task of clustering is to divide a large amount of data into disjoint subsets or *classes*, such that some measure of distance is minimized within classes, and maximized between classes. In computer vision, several recent advances have incorporated clustering algorithms for the canonicalization of large data sets: selecting exemplars [30], building unsupervised object recognizers [1, 2], texton generation [16,21], and learning in low-level vision [19]. What makes clustering hard is that the measurement of distance between data is invariably corrupted by measurement error in the data themselves. To overcome this, the distance measures must include a model of the noise process underlying the measurement errors, and the clustering algorithms must employ sophisticated search techniques in order to minimize the distance.

In some problems—particularly those arising in vision applications—there is another common source of variation, caused when the observed data undergo a parametrized transformation. For example, changes in scale and rotation of a head in face recognition

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 304–320, 2002.

Fig. 1. Clustering on faces. A standard face detector has been run on 30000 frames of the movie "Groundhog Day", some example detections being shown here. The distance metric developed in this paper is used in a standard clustering algorithm to extract the principal cast list from the several thousand faces detected in a typical feature film.

applications that arise from variations in pose, see figure 1. Clustering algorithms for computer vision must allow for, or be invariant to, such transformations.

Our contribution in this paper is to develop a set of distance functions which take account of affine transformation of the data. These distance functions may either be invariant to the transformation or contain priors based on the transformations parameters. Closed form and numerical iterative solutions are given for these functions. This enables a Bayesian maximum a posteriori (MAP) cluster estimation.

The rest of the paper is arranged as follows. First we specify the problem in detail, and review the related literature. Section 2 develops the affine-invariant distance with and without priors for a specific problem: the registration of 2D point sets. Section 3 develops the general form of the affine-invariant distance for image comparisons, and shows how the prior is incorporated. Then, in section 4, the application to image clustering is developed in the context of automatic cast indexing. This allows us to give details about the implementation for a concrete application. Finally we conclude with a discussion of current and future avenues for research.

The Problem

Before reviewing the literature on transformation-invariant clustering, it will prove useful to formally define the clustering problem we wish to solve. We have a data set S of n **observations** X_i, and want to obtain a **partition** $\mathbb{S} = \{S_1 \dots S_k\}$ such that $S = S_1 \cup \dots \cup S_K$, and a set of **cluster centres** $\{X_c\}_{c=1}^{K}$ such that a cost

$$\sum_{c=1}^{K} \sum_{X_i \in S_c} d(X_i, X_c) \tag{1}$$

is minimized, where $d(X, Y)$ is a **distance function** between X and Y.

Thus a solution to a clustering problem always involves two aspects: (1) choosing a distance function $d(,)$, (2) determining the clusters given $d(,)$. Research into clustering and classification of data has a long literature [8], and many good algorithms

are available. Examples include K-means [7], normalized cuts [24], minimal spanning tree. In this work, we use the "Partition among Medoids" algorithm of Kaufman and Rousseeuw [14], and treat the clustering routine as a black box. Some of these algorithms offer automatic computation of K, the number of cluster centres, and some require it as an input parameter. The distance metric which is the novel contribution of this paper is independent of these issues, and may be used with any clustering algorithm. Furthermore, although $d(,)$ imposes the structure of a metric space on the space of the X, it is a property of the clustering algorithm, not the distance, whether the X should live in a space which is also a vector space.

Background

The problem we investigate in this paper is the following: we wish to cluster a data set under an affine invariant distance function including priors on the affine transformation. Invariant approaches to unsupervised clustering have taken a number of routes. In a vector space, the techniques which have been used for robustification of principal components analysis [5] and to include some transformation invariance [9] could be applied to clustering, but these solutions are expensive to compute, and many interesting computer vision problems do not have data which may be linearly combined.

In a metric space, attention must concentrate on the distance function in order to obtain invariance. Simard *et al.*[26,27] describe the modification required. The key idea is that the clustering will be independent of the transformation parameters if the distance metric is transformation invariant.

In spaces which are not metric, it is possible [22,23] to obtain transformation invariance by artificially introducing transformed copies of each datum into the dataset. For example Schölkopf [23] adds transformation invariance to a support-vector classifier by carefully adding examples to the dataset which are transformed copies of the initially selected support vectors. However, this strategy is impractical—unless the range of transformations is very small indeed—because of the enormous expansion in the size of the feature space, and a consequent increase in the cost of the clustering algorithms. Also, although this technique can be adapted to other classifiers, it cannot easily be made to work for unsupervised clustering.

The work in this paper concentrates on the common computer vision case, in which the points of interest may be considered to lie in a metric space.

2 Point Sets

In this section we will illustrate the problem for the case of a set of 2D points $X = \{\mathbf{x}_i\},\ i = 1 \ldots N$ in correspondence with another set of of 2D points $X' = \{\mathbf{x}'_i\},\ i = 1 \ldots N$. The correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ are known. We will demonstrate the behaviour of the affine invariant distance function in comparing and clustering shapes.

We are searching for a set of corresponding points $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$ which are close to the supplied points $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, and are exactly mapped by an affine transformation $\hat{\mathbf{x}}'_i = \mathtt{A}\hat{\mathbf{x}}_i + \mathbf{t}$, where $\mathtt{A}$ is a 2×2 matrix and $\mathbf{t}$ a 2-vector. For the moment we will neglect the prior $p(\mathtt{A}, \mathbf{t})$ on the affine parameters. Then our aim is to compute the distance function $d_{\mathrm{A}}(X, X')$,

$$d_{\mathrm{A}}(\{\mathbf{x}_i\},\{\mathbf{x}'_i\})^2 = \min_{\mathbf{a},\{\hat{\mathbf{x}}_i\}} \sum_{i}^{N} \left((x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(x'_i-\hat{x}'_i)^2+(y'_i-\hat{y}'_i)^2\right) \tag{2}$$

where $\mathbf{a}$ is a 6-vector specifying the 6 parameters of the affine transformation. Note, computing d_{A} involves also estimating the points $\{\hat{\mathbf{x}}\},\ i = 1 \ldots N$ and affine transformation $\mathbf{a}$ – a total of $2N+6$ parameters – and the distance is the minimum over all these parameters. We are only interested in the distance, so for our purposes the points $\{\hat{\mathbf{x}}\}$ and transformation $\mathbf{a}$ are "nuisance parameters". It will be seen below that the distance can be computed without explicitly solving for these nuisance parameters.

In a practical situation this problem might arise in computing the affine transformation between two images of a planar object. The points $\{\mathbf{x}_i\},\{\mathbf{x}'_i\}$ would be the measured points on the object in the first and second images respectively.

If it is assumed the measured points are corrupted by additive Gaussian noise in their position, then minimizing the cost in (2) gives the maximum likelihood estimate of the transformation $\mathbf{a}$ and points $\{\hat{\mathbf{x}}_i\},\{\hat{\mathbf{x}}'_i\}$; and the distance $d_{\mathrm{A}}()$ is known as reprojection error [10]. The distance d_{A} is a generalization from similarity to affine transformations of the "Procrustean" distance of Mardia [6].

If we concatenate the set of 2D points into a single $2N$-vector, so that for $\mathbf{X} = (\mathbf{x}_1{}^\top, \mathbf{x}_2{}^\top \ldots \mathbf{x}_N{}^\top)^\top$ etc, then (2) becomes

$$d_{\mathrm{A}}(\mathbf{X},\mathbf{X}')^2 = \min_{\mathbf{a},\hat{\mathbf{X}}} (\mathbf{X}-\hat{\mathbf{X}})^2 + (\mathbf{X}'-\hat{\mathbf{X}}')^2 \tag{3}$$

Note a (2×2) affine transformation on the 2D space $\mathbf{x}$ does not result in a $(2N \times 2N)$ general affine transformation on the $2N$ dimensional space $\mathbf{X}$. Instead the corresponding transformation matrix is block diagonal with the 2×2 matrix $\mathtt{A}$ defining the block. The corresponding points $\hat{\mathbf{X}} \leftrightarrow \hat{\mathbf{X}}'$ are constrained to lie on a six-dimensional hyperplane (a subspace) in $\mathbb{R}^{2N}$. Figure 2 gives a geometric picture of the vectors involved and the constraint surface.

We now turn to computing the distance. It will be seen that the distance may be computed in closed form by a simple algorithm.

Algorithm for computing the distance $d_{\mathrm{A}}(\{\mathbf{x}_i\},\{\mathbf{x}'_i\})$**:**

1. Translate the point set $\{\mathbf{x}_i\}$ so that its centroid is zero, i.e. compute the centroid $\frac{1}{N}\sum_i \mathbf{x}_i$ and subtract it from each $\mathbf{x}_i$.
2. Similarly translate the point set $\{\mathbf{x}'_i\}$ so that its centroid is also zero.
3. Form the $N\times 4$ matrix $\mathbf{M}$ with rows $(\mathbf{x}_i{}^\top, \mathbf{x}'_i{}^\top) = (x_i, y_i, x'_i, y'_i)$.
4. Form the Singular Value Decomposition of $\mathtt{M} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$, where $\mathtt{D}$ is a diagonal matrix with diagonal elements $\sigma_1,\sigma_2,\sigma_3,\sigma_4$ (the singular values) followed by zero's.
5. Then the required distance $d_{\mathrm{A}}(\{\mathbf{x}_i\},\{\mathbf{x}'_i\})^2 = \sigma_3^2+\sigma_4^2$.
6. If required the points $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i$ may be obtained as

$$\begin{pmatrix}\hat{\mathbf{x}}_i\\ \hat{\mathbf{x}}'_i\end{pmatrix} = \left[\mathtt{I}_{4\times 4} - \mathbf{v}_3\mathbf{v}_3{}^\top - \mathbf{v}_4\mathbf{v}_4{}^\top\right]\begin{pmatrix}\mathbf{x}_i\\ \mathbf{x}'_i\end{pmatrix}$$

where $\mathbf{v}_i$ are the columns of $\mathtt{V}$.

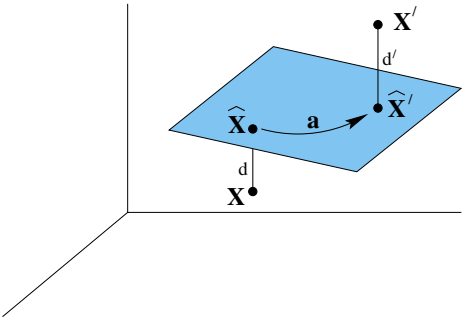


Fig. 2. Affine invariant distance measure (AIDM). The points $\mathbf{X}$ and $\mathbf{X}'$ in $\mathbb{R}^{2N}$ are the supplied vectors. We seek the points $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}'$ which are closest to $\mathbf{X}$ and $\mathbf{X}'$ respectively, but are related exactly by an affine transformation. The hyperplane $\boldsymbol{\pi}$ represents all the points that can be reached by the affine action on $\hat{\mathbf{X}}$ (or equivalently on $\hat{\mathbf{X}}'$). It is the orbit of the affine group, specified by six parameters. The distance $d_{\mathrm{A}}(\mathbf{X}, \mathbf{X}')^2 = d^2 + d'^2$. It is evident that the distance is minimized by finding the hyperplane spanned by the affine action which minimizes the perpendicular distance to the supplied points $\mathbf{X}$ and $\mathbf{X}'$, and that the points $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}'$ are the projections of $\mathbf{X}$ and $\mathbf{X}'$ respectively onto this plane.

The proof uses the property of the SVD that the closest fitting rank 2 column space to `M` is given by the first two columns of `U`, but is omitted here for lack of space. The result is closely related to the factorization algorithm [28], though here applied to a single view, instead of multiple views, and for structure restricted to a plane [12].

2.1 Application to Shape Matching

Suppose we wish to cluster the six polygonal shapes in figure 3(b). Each polygon (2 examples each of a unit square, a rotated and scaled square, and a triangle) is defined by four points, and the correspondence between the points of each polygon is known. If Euclidean distance is measured between the points (after registering the shapes' centroid) then there are three pairings which have a small distance, and these are the pairing of the square with the square, triangle with triangle and rotated square with rotated square (see table 1). However, if affine invariant distance is measured between the shapes then all four squares may be clustered, but the triangles are still distinct. Indeed if there is no noise added to the points then the affine invariant distance between the square and its rotated and scaled version is zero.

2.2 Including Priors on the Transformation

Of course, in practical computer vision applications, one is rarely concerned about invariance to all of the transformations which a particular parametrization admits. For example, in digit recognition one wishes to maintain invariance to small differences in rotation, scaling and translation of letters, but not to allow a "6" to be rotated into a

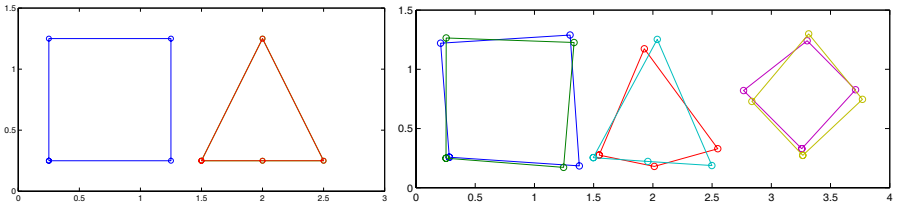


Fig. 3. Shape space. (a) Two shapes ("exemplars") defined by four points. (b) Affine observations of the exemplars: the observations are generated by applying an affine transformation (which is the identity in four of the cases shown) and then adding Gaussian noise with $\sigma = 0.05$ independently to each point. There are six observations, shown in three superimposed pairs. The square and rotated square are in the same affine equivalence class.

"9" [27]. Thus we wish to include *a priori* knowledge in the form of prior probability distributions $p(\mathbf{a})$ on the transformation parameters in the distance function.

If the generative model which produces the point sets is an affine transformation followed by additive Gaussian noise on the point's location, then the distance $d_{\mathrm{A}}(\mathbf{X}, \mathbf{X}')$ is the (negative) log-likelihood that $\mathbf{X}$ and $\mathbf{X}'$ are observations of the object $\hat{\mathbf{X}}$ (which represents the affine equivalence class). As this is a log-likelihood, a prior must also be included in the distance function as a (negative) log-likelihood, i.e. as $-\log p(\mathbf{a})$. In this way, by Bayes, the distance is related to the posterior probability that the two observations are in the same affine equivalence class given a prior on the transformation between observation pairs.

If we can use a zero-mean Gaussian prior on $\mathbf{a}$, we may write $p(\mathbf{a}) = \exp(-\mathbf{a}^\top \Lambda \mathbf{a})$. Although the process need not be zero-mean, we assume for simplicity that it is. Given this prior then, the distance function of equation (3) is extended to

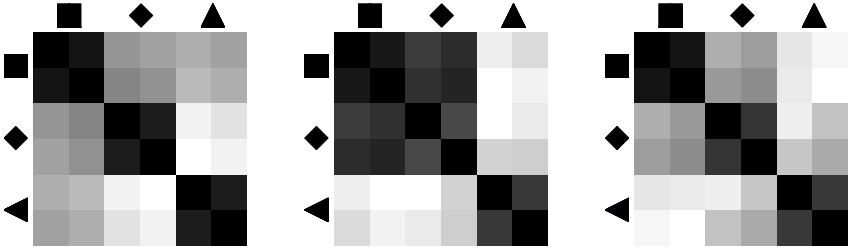
$$d_{\mathrm{A}}(\mathbf{X}, \mathbf{X}')^2 = \min_{\mathbf{a}, \hat{\mathbf{X}}} (\mathbf{X} - \hat{\mathbf{X}})^2 + (\mathbf{X}' - \hat{\mathbf{X}}'))^2 + \mathbf{a}^\top \Lambda \mathbf{a} \tag{4}$$

In this synthetic example, we assume a more complex distribution for the affinities. The prior used is $p(\mathtt{A}, \mathbf{t}) = \exp(-|1 - \det \mathtt{A}|)$, where $\mathtt{A}$ is the 2×2 matrix of the affine transformation $\hat{\mathbf{x}}_i' = \mathtt{A}\hat{\mathbf{x}}_i + \mathbf{t}$, computed between the point sets. The prior in this case simply measures the relative change in areas of the shapes. Table 1 shows the effect of applying the distance function (3) between the pairs of shapes of figure 3b. Without the prior the square and 'rotated and scaled square' are in the same equivalence class, but including the prior breaks the symmetry.

3 Images

For many vision applications, the primary datum is the vector of greylevels representing an image or an image patch, with an associated parametrized transformation, to which invariance is sought. Given an image represented as a column vector $\mathbf{I}$, the image mapped

Table 1. Image intensity represents the distance measured between pairs of shapes from figure 3b, where dark indicates a low distance. (Note the shapes are first translated so that their centroids are coincident). (a) **Euclidean distance**. The only clusters are on the diagonal, for example the two noisy versions of the square form a cluster, but the square and rotated square do not. (b) **Affine invariant distance** d_{A} measured between the shapes. Note that distances cluster into the two equivalence classes. (c) **Affine distances with priors** on the transformation parameters (see text). The effect of the prior is to break the symmetry (the affine equivalence of the sets). The distances now cluster again into three equivalence classes.



(a) Euclidean (b) Affine (c) Affine with priors

by the transformation with parameters $\mathbf{a}$ is written $T(\mathbf{a};\mathbf{I})$. Note that under many common transformations, this transformation is linear in the image graylevels, despite being nonlinear in $\mathbf{a}$.

As in (2) we seek to estimate an image $\mathbf{I}_0$ which maps exactly under this induced transformation but minimizes the distance to the measured images $\mathbf{I}, \mathbf{I}'$.

$$d_{\mathrm{A}}(\mathbf{I},\mathbf{I}')^2 = \min_{\mathbf{a},\mathbf{a}',\mathbf{I}_0} (\mathbf{I} - T(\mathbf{a};\mathbf{I}_0))^2 + (\mathbf{I}' - T(\mathbf{a}';\mathbf{I}_0))^2 \tag{5}$$

where we have specified here the sub-space by a point $\mathbf{I}_0$ which is transformed to estimate $\hat{\mathbf{I}}$ by one transformation parametrized by $\mathbf{a}$, and is transformed to estimate $\hat{\mathbf{I}}'$ by another transformation parametrized by $\mathbf{a}'$. In order to cluster a set of images, we will compute d_{A} pairwise in the set, and apply a standard metric-space clustering algorithm.

Note that this formulation of the distance is essentially that of estimating the affine transformation which best aligns the two images, and reporting the squared error between the aligned and original images. This is a problem with a long history, and two basic approaches may be discerned: *direct* [13] and *feature-based* [29]. The distance metric in this paper is a purely direct method without any feature correspondences.

3.1 AIDM: Affine Invariant Distance Measure

In general, the transformation will not be linear in the parameters $\mathbf{a}$, so we shall linearize about the origin. Expanding in terms of derivatives of $\mathbf{I}$ with respect to the 6 parameters of the affine transformation, we calculate a $N\times$ 6 Jacobian matrix $\mathtt{D} = \nabla_{\mathbf{a}} T(\mathbf{a};\mathbf{I})$, and similarly from $\mathbf{I}'$ calculate $\mathtt{D}'$. Then to a first order approximation (5) becomes

$$d_{\mathrm{A}}(\mathbf{I},\mathbf{I}')^2 = \min_{\mathbf{a},\mathbf{a}',\mathbf{I}_0} (\mathbf{I} - (\mathbf{I}_0 + \mathtt{D}\mathbf{a}))^2 + (\mathbf{I}' - (\mathbf{I}_0 + \mathtt{D}'\mathbf{a}'))^2 \tag{6}$$

Collecting the parameters $\mathbf{I}_0, \mathbf{a}$ and $\mathbf{a}'$ into a single vector of unknowns $\mathbf{x} = [\mathbf{I}_0^\top, \mathbf{a}^\top, \mathbf{a}'^\top]^\top$, this is equal to

$$\begin{aligned} d_{\mathrm{A}}(\mathbf{I}, \mathbf{I}')^2 &= \min_{\mathbf{x}} \left(\begin{bmatrix} \mathbf{I} \\ \mathbf{I}' \end{bmatrix} - \begin{bmatrix} 1 & \mathtt{D} & 0 \\ 1 & 0 & \mathtt{D}' \end{bmatrix} \mathbf{x} \right)^2 \\ &= \min_{\mathbf{x}} \|\mathbf{m} - \mathtt{M}\mathbf{x}\|^2 \end{aligned} \tag{7}$$

Where $\mathtt{1}$ is an $N \times N$ identity matrix. Finally, differentiating with respect to $\mathbf{x}$ and solving gives the solution for the minimizing $\mathbf{x}$ as $\mathbf{x} = \mathtt{M}^+\mathbf{m}$, where $\mathtt{M}^+ = (\mathtt{M}^\top\mathtt{M})^{-1}\mathtt{M}^\top$ is the Moore-Penrose pseudoinverse of $\mathtt{M}$. Substituting this $\mathbf{x}$ into (7) gives the desired distance.

If the affine transformation is small, then this first order approximation is often sufficient. In general though we are seeking the true solution to (5). This solution may be obtained in the standard manner by a pyramid search and the linearization employed on the various levels of the pyramid. An alternative is to use a good non-linear minimizer (such as Levenberg-Marquardt) to search for the global solution of (5) directly. In the next section we shall show how including the priors on the transformation parameters is analogous to using a Levenberg-Marquardt-like algorithm.

Advantages of Including the Prior

The transformation priors may be written $\mathbf{x}^\top \Lambda_{\mathbf{x}} \mathbf{x}$, so if priors are included the minimization (7) becomes instead

$$d_{\mathrm{A}}(\mathbf{I}, \mathbf{I}')^2 = \min_{\mathbf{x}} \|\mathtt{M}\mathbf{x} - \mathtt{m}\|^2 + \|\Lambda_{\mathbf{x}}^{\frac{1}{2}}\mathbf{x}\|^2 \tag{8}$$

A linear solution may be obtained as above by differentiating with respect to $\mathbf{x}$ and solving to give $\mathbf{x} = (\mathtt{M}^\top\mathtt{M} + \Lambda)^{-1}\mathtt{M}^\top\mathbf{m}$. Now the numerical advantages of including the prior become apparent: if $\mathtt{M}$ is poorly conditioned, the Newton step $\mathbf{x} = \mathtt{M}^+\mathbf{m}$ can place $\mathbf{x}$, and hence $\mathbf{I}_0$, far from the data points $\mathbf{I}, \mathbf{I}'$, and assign a large value to $d(\mathbf{I}, \mathbf{I}')$. Whether one computes the pseudoinverse directly from the Moore-Penrose formula or via the SVD, poor conditioning of $\mathtt{M}^\top\mathtt{M}$ will generally imply an erroneous step. With the prior, the computation inverts $\mathtt{M}^\top\mathtt{M} + \Lambda$ rather than $\mathtt{M}^\top\mathtt{M}$. We assume that the information matrix Λ is positive definite (as it is the inverse of a covariance matrix), so the smallest eigenvalue of the matrix to be inverted is bounded away from zero. Hence the computed step is actively constrained to lie close to the initial estimate.

A useful comparison is with schemes for nonlinear optimization. The computation of (5) under the tangent distance approximation of [26,31] is effectively a Gauss-Newton update for the hidden parameters $\mathbf{I}_0$, $\mathbf{a}$ and $\mathbf{a}'$. The convergence properties of such schemes are well known [3,4,20]: where the approximation is good, it gives excellent convergence; but where the linearization is not valid, the update can be drastically wrong. Modern optimization techniques invariably regularize the Gauss-Newton update using *trust region* strategies [3,4], and thus gain improved convergence. Including the prior in (8) confers similar advantages on AIDM, as will be shown in the applications which follow, and in the distance matrices in figure 6.

Fig. 4. Test sequence. A selection of the 251 faces detected on a 2000 frame sequence from the film "Groundhog Day". The start and end frame of the sequence are shown in figure 7. There are four principal cast members in this sequence, and some significant distractors. If the face detector fires on background clutter (e.g. the television on the bottom left) it will generally do so reliably, giving a large consistent cluster. Lighting and head orientation changes are significant, so the distance metric must be robust to these effects.

4 Application

As an example of the application of the the proposed invariant distance function, we consider the problem of automatically extracting the principal cast members from a movie sequence. This application is a challenging analogue to the clustering of digits problem [15], and requires the full power of the affine invariant distance as well as the incorporation of motion priors in order to achieve useful results. This is because, although movies are generally well photographed, the actors' faces tend to be seen in quite varied positions and under lighting conditions that are not typical of more traditional mugshot or security applications. Figures 1, 4 and 5 show some examples.

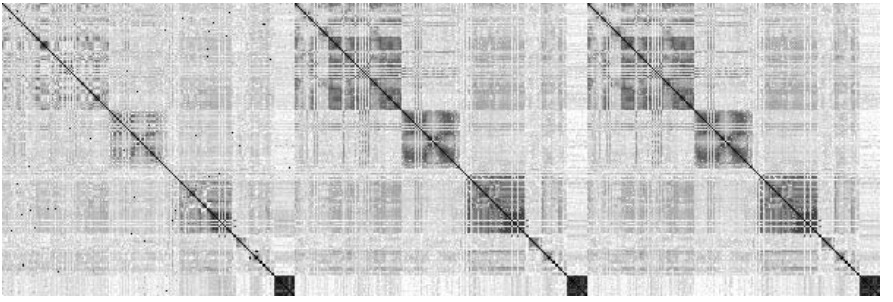
To give an idea of the magnitude of the task: a feature length film contains of the order of 150K frames and a principal cast of perhaps 20, with each character appearing in 1000s of frames. In the examples shown here, we generally subsample temporally by a factor of five, so that we are dealing with datasets of the order of 30000 frames.

The strategy of the algorithm is to detect faces in each frame of the movie, and cluster the detected faces to obtain a representative set which summarizes the faces of the cast, and of course, associates the cast members with the scenes in which they appear.

In the following sections we will describe the steps involved in clustering for a test sequence of 2000 frames with 251 detected faces from the film 'Groundhog Day". See figure 4.

Face detection: Face detection is an area that has seen significant progress in recent years, and impressive systems have been built [11]. In this application we used a well-engineered local implementation [18] of the Schneiderman and Kanade [22] face detector. Parameters were set to obtain a true positive rate of about 80% of frontal faces, which induces a false detection rate of about 0.01 faces per frame. This detector obtains scale and translation invariance by oversampling. The output comprises the image plane translation of the face template centre, and the scale at which the maximum response was

Fig. 5. Pre-processing. Upper set: original (raw) extracted faces. Lower set: faces after bandpass filtering and feathering. Variations in lighting across the image are removed, and boundary effects are diminished.



(a) (b) (c)

Fig. 6. Distance matrices for the test sequence. For this sequence, the faces are naturally clustered, and large dark blocks in the distance matrix indicate the clusters. (a) Multiresolution tangent distance [31]. (b) AIDM without priors. (c) AIDM with deformation prior. In this case, the distance computed by the Newton methods (b,c) is visibly better defined than (a). The improvement due to deformation priors (c) over (b) is not visually obvious, but amounts to a small percentage reduction in the number of divergence failures (individual bright pixels in all matrices).

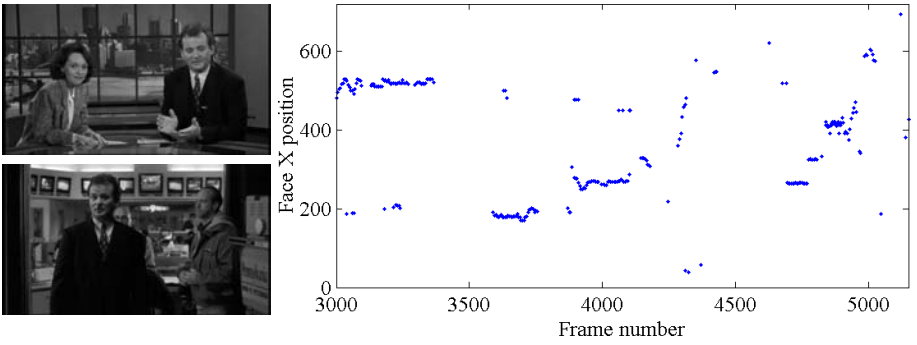


Fig. 7. Temporal coherence. (Left) Two example frames from an 80 sec extract from the test movie. (Right) Image-plane x coordinate of detected faces over that range. Although there are two actors in shot for much of the sequence, they are frequently not detected because the face goes into a profile view. However we can see that, starting at frame 3000, there are two faces in shot, "A" at $x \approx 200$ and "B" at $x \approx 500$ (to the right of the frame). For the first 400 frames, B faces the camera, giving a long consistent trace, and A alternates her gaze between B and the viewer. From 3400 to 3550, both talk to each other in profile, so there are no detections, and then A takes over the straight-to-camera slot.

recorded. The detected faces are rescaled to 81×81 pixels, and stored with their frame number and position. Examples of the output are shown in figure 1. Plots of the recovered image-plane position (shown in figure 7) illustrate the strong temporal coherence of the detector outputs, a constraint we will incorporate via priors into the clustering.

Variations in pose and lighting: The principal sources of variation in this application arise from changes in illumination and in facial pose. Images can be normalized for illumination by applying a bandpass filter to the extracted face regions, and scaling the windows' intensities to have mean zero and variance 1. In addition, faces are feathered by multiplying them with a Gaussian centred at the centre of the image. In detail the filters used are:

- Bandpass: $B = I * G_{\sigma=1} - I * G_{\sigma=4}$
- Feather: $F(x, y) = B(x, y) \exp - \left(\frac{r(x,y)}{.5}\right)^2$

Examples of this pre-processing are shown in figure 5.

Affine invariant clustering: The distance was computed using a multiscale technique [31] on a Gaussian pyramid, with the finest scale corresponding to $\sigma = 2$ pixels. The affine-invariant distance was computed by iterating the solution of (8), with up to five iterations allowed. Implemented in unoptimized MATLAB, evaluation of the distance for a single pair of faces takes about half a second. With a typical movie returning several thousand face detections, computation of the complete distance matrix would be impractical. In order to accelerate the process, we pruned the face set using a Euclidean k-medoids algorithm [14] on the affinely distorted faces, with k chosen to return a few hundred

faces. Then computation of the affine-invariant distance matrix on the reduced set requires compute time only of the order of a few hours—certainly less than the initial face detection stage. Finally, the k-medoids algorithm is applied to the affine-invariant distance matrix and the top K cluster centres extracted, for a user-specified K.

Figure 6 shows some example distance matrices, computed with the tangent distance approximation to affine invariance, and our new Newton methods. In this example, the new metrics improve the signal-to-noise ratio of the clusters, as evidenced by the dark squares in the new distance matrices. Examples of the effect on the clustering output are shown in figure 9.

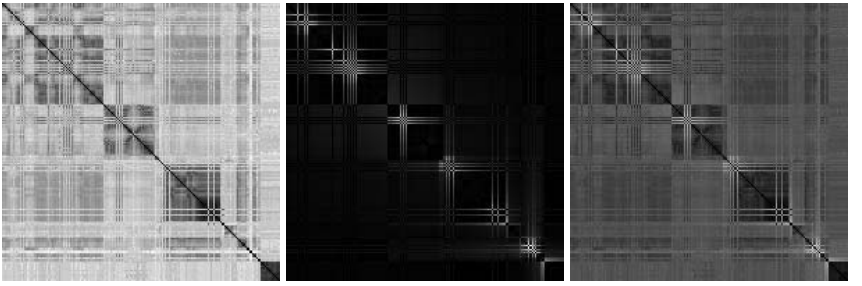
4.1 Clustering Including Priors

Two classes of priors will be included: priors on the transformation between any pair of frames; and priors on the transformation between contiguous frames. We will first describe these two classes, and how they are learnt from images, and then show their effectiveness in improving clustering. The resultant distance matrices are shown in figure 8.

Deformation priors: The prior on the affine transform parameters is learnt by manually selecting the eyes and the centre of the mouth in 200 randomly chosen faces, computing the affine transformations which maps between these, and fitting a single 6D Gaussian distribution to the resulting transformation parameters. This prior is generic to all of the pairwise comparisons, and simply represents the likelihood that the face detector will tend to detect faces in only a small number of poses. Including this prior into the distance metric has only a small effect on the computed distance in almost all cases. However, in cases where the Newton method might diverge, producing an extreme affine correction, this prior will tend to regularize the computation. In the 2000-frame test sequence, this occurs in about 1% of the comparisons, all of which are repaired by the deformation prior.

Temporal coherence prior – speed: An important additional constraint in the analysis of image sequences is that provided by the motion of the detected faces. In movies particularly, the faces generally move little from frame to frame, so that there is an *a priori* restriction on the speed at which objects can move. A typical assumption might be that the image velocity vector $\dot{\mathbf{x}} = d\mathbf{x}/dt$ is drawn from a zero-mean distribution which decays monontonically away from the origin. Figure 7 gives an indication of the stability of the face detector's position reports on a studio-bound shot. In outdoor or crowd scenes the variance is somewhat higher.

To incorporate this knowledge, we can augment the distance computation between two faces. Each 81×81 face window in the pair I, I' is associated with its location $\mathbf{p}$ or $\mathbf{p}'$ in the full frame, and the time (frame number) at which it was detected t or t'. Then, for any pair, a speed estimate is given by $s(I, I') = \|\mathbf{p}' - \mathbf{p}\|/|t' - t|$. This is converted to a log-likelihood via a kernel function E, and added to the image distance, which then contains three terms



(a) (b) (c)

Fig. 8. Distance matrices with and without priors. (a) Affine-invariant distance metric with *deformation priors* only. (b) Matrix of *speed prior* contributions only. This can be precomputed without any iteration as it is based solely on the image locations reported by the face detector. (c) *Combined deformation and speed priors*. The notable contribution of the speed prior in this case (strong narrow white bars) is the increase in dissimilarity between detections which are spatially separated, but temporally close – e.g. two actors in the same scene.

$$d_{\mathrm{A}}(\mathbf{I},\mathbf{I}')^2 = \min_{\mathbf{a},\mathbf{a}',\mathbf{I}_0} (\mathbf{I} - T(\mathbf{a};\mathbf{I}_0)^2 + (\mathbf{I}' - T(\mathbf{a}';\mathbf{I}_0)^2 - \log p(\mathbf{a}^{-1}\mathbf{a}') + E(s(I,I')) \tag{9}$$

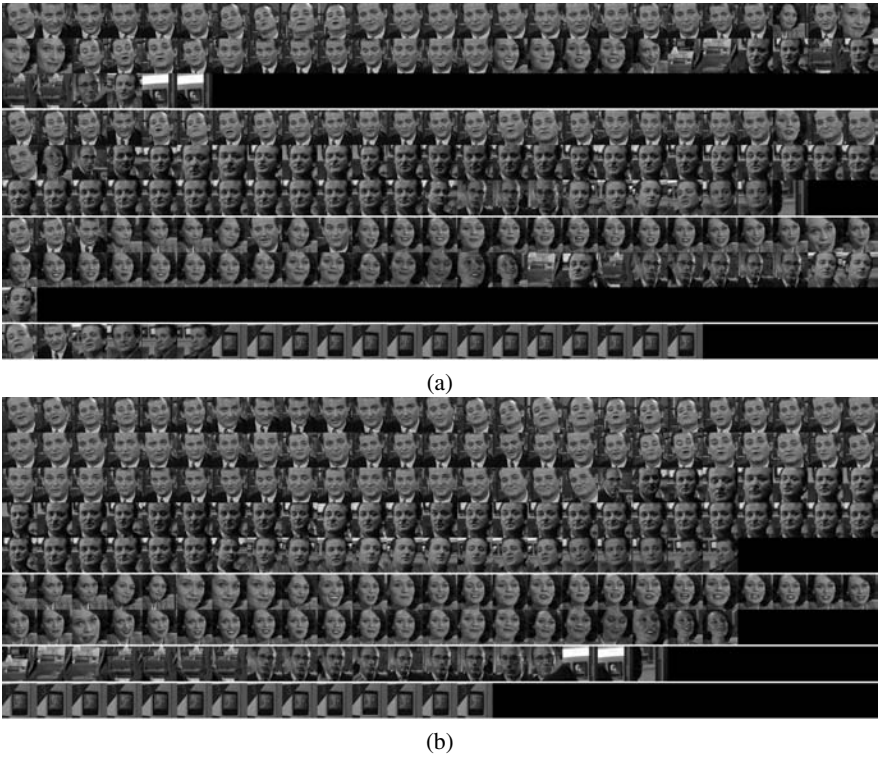
Because the dependence of s on $\mathbf{a}$ is weak in our examples, the motion cost is simply added to the AIDM matrix before clustering. It was empirically determined that an effective form for the error was an arctangent sigmoid $E(x) = \tan^{-1} x$. Figure 8 shows the AIDM matrix before and after incorporating the motion constraint.

An effect of this prior is to increase the distance between faces across shot boundaries, since generally faces are not in precisely the same position in such frames and consequently the speed is large. A prior could also be included on whether the same face appears in contiguous shots.

Clustering: The distance matrix has been computed, and incorporates both deformation and speed priors. Clustering is now performed using the "Partition among Medoids" algorithm [14]. Because this algorithm has a run time which is greater than quadratic in the number of data, a hierarchical strategy was employed. Given an n-element dataset and a requirement for k clusters, the algorithm divides the set into N subsets, which can be clustered in reasonable time. Each subset is clustered to obtain k medoids, and the process repeated over the Nk cluster centres. Figure 10 shows the results of the algorithm on 2111 faces detected in "Groundhog Day".

5 Discussion

We have presented a new affine invariant distance metric which efficiently manages priors on the transformation parameters, and shown how the use of such priors regularizes



(a)

(b)

Fig. 9. Clusters for the test sequence. (a) Clusters computed using tangent distance. (b) Clusters computed using AIDM with both deformation and speed priors. The number of clusters was set *a priori* at four. The tangent distance clusters are poor, showing very little within-cluster coherence. The AIDM clusters correctly extract the main two characters, and the third cluster center is the third character, although that cluster includes some spurious background.

clustering problems in a controllable way. Previous efficient approaches used *ad hoc* regularizers and did not include priors, while previous approaches incorporating priors were expensive to compute. The new technique is analogous to the use of "trust region" and Levenberg-Marquardt strategies in nonlinear optimization. The power of this metric for unsupervised clustering has been demonstrated by automatically extracting the principal cast from a movie.

The primary difficulty with the current versions of the algorithm is a poor tolerance to changes in expression of the characters. Relaxing the distance threshold will result in merging of clusters containing different characters. Future investigations will include improvements to the tolerance to expression change and other hard cases (see figure 12). In particular we intend to learn the noise distribution in the manner of [25].

Clustering invariant to certain classes of transformations is an interesting strategy to investigate for many vision problems. Here we have investigated face detection which may be considered as a very strongly defined interest region operator. A similar example

Fig. 10. Principal cast list of "Groundhog Day". Cluster centres under AIDM with priors. Duplicates have not been suppressed, so actors can appear multiple times, with different expressions.

Fig. 11. Principal cast list of "The Player". Top 40 cluster centres from 2899 detected faces on every fifth frame (35641 frames).

Fig. 12. Difficult cases. Examples (from "The Player") where the plain AIDM has difficulty. Lighting change is handled reasonably well by our preprocessing. Modification of (5) to include a robust kernel goes some way to handling the problems of occlusion.

would be a building facade detector, where invariant clustering would be expected to determine the sides of the building. More generally, affine invariant clustering on 2D appearance would (partially) remove 3D viewpoint effects and determine clusters corre-

sponding to the aspects of an object. Finally, affine invariant clustering on filter responses will enable viewpoint variant and invariant textons [17] to be learnt from single images of general curved surfaces, instead of, as now, from images of textured planes.

Acknowledgements. We are very grateful to Krystian Mikolajczyk and Cordelia Schmid of INRIA Grenoble for supplying us with the face detection software. David Capel and Michael Black made a number of helpful comments. Funding was provided by the Royal Society and the EC CogViSys project.

References

1. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
2. M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV (2)*, pages 628–641, 1998.
3. R. Byrd, R.B. Schnabel, and G.A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM J. Numer. Anal.*, 24:1152–1170, 1987.
4. A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, 2000.
5. F. De la Torre and M. J. Black. Robust principal component analysis for computer vision. In *Proc. International Conference on Computer Vision*, 2001.
6. I. Dryden and K. Mardia. *Statistical shape analysis*. John Wiley & Sons, New York, 1998.
7. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.
8. D. Fasulo. An analysis of recent work on clustering algorithms. Technical Report UW-CSE-01-03-02, University of Washington, 1999.
9. B. Frey and N. Jojic. Transformed component analysis: joint estimation of spatial transformations and image components. In *Proc. International Conference on Computer Vision*, pages 1190–1196, 1999.
10. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
11. E. Hjelmås and B. K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.
12. M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV*, pages 626–633, 1999.
13. M. Irani and P. Anandan. About direct methods. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 267–277. Springer, 2000.
14. L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, NY, USA, 1990.
15. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
16. T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1010–1017, Kerkyra, Greece, September 1999.
17. T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, December 1999.
18. K. Mikolajczyk, R. Choudhury, and C. Schmid. Face detection in a video sequence – a temporal approach. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

19. B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–9, 1996.
20. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
21. C. Schmid. Constructing models for content-based image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
22. H. Schneiderman and T. Kanade. A histogram-based method for detection of faces and cars. In *Proc. ICIP*, volume 3, pages 504 – 507, September 2000.
23. B. Schölkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In *Articial Neural Networks, ICANN'96*, pages 47–52, 1996.
24. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 731–743, 1997.
25. H. Sidenbladh and M. J. Black. Learning image statistics for Bayesian tracking. In *Proc. International Conference on Computer Vision*, pages II:709–716, 2001.
26. P. Simard, Y. Le Cun, and J. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Info. Proc. Sys. (NIPS)*, volume 5, pages 50–57, 1993.
27. P. Simard, Y. Le Cun, J. Denker, and B. Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Lecture Notes in Computer Science, Vol. 1524*, pages 239–274. Springer, 1998.
28. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, November 1992.
29. P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 278–294. Springer, 2000.
30. K. Toyama and A.Blake. Probabalistic tracking in a metric space. In *Proc. International Conference on Computer Vision*, pages II, 50–57, 2001.
31. N. Vasconcelos and A. Lippman. Multiresolution tangent distance for affine-invariant classification. In *Advances in Neural Info. Proc. Sys. (NIPS)*, volume 10, pages 843–849, 1998.