

# Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences

M.A. Giese and T. Poggio

Center for Biological and Computational Learning  
Massachusetts Institute of Technology, E25-206 / 218  
Cambridge, MA 02142, USA  
Tel.: 617 253 0549 / 5230, FAX: 617 253 2964

E-mail: [giese@ai.mit.edu](mailto:giese@ai.mit.edu)  
[tp@ai.mit.edu](mailto:tp@ai.mit.edu)

Paper for the IEEE Workshop on  
Multi-View Modeling & Analysis of Visual Scene  
June 21-23, 1999  
Fort Collins, Colorado

# Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences

M. A. Giese and T. Poggio

Center for Biological and Computational Learning  
Massachusetts Institute of Technology, E25-206 / 218  
Cambridge, MA 02142, USA

## Abstract

*The linear-combination of prototypical views has been shown to provide a powerful method for the recognition and analysis of images of three-dimensional stationary objects. In this paper, we present preliminary results on an extension of this idea to video sequences. For this extension, the computation of correspondences in space-time turns out to be the central theoretical problem, which we solve with a new correspondence algorithm. Using simulated images of biological motion we demonstrate the usefulness of the superposition of prototypical sequences for the synthesis of new video sequences, and for the analysis and recognition of actions.*

*Our method permits to impose a topology over the space of video sequences of action patterns. This topology is more complicated than a linear space. We present a new method that is based on the structural risk minimization principle of statistical learning theory, which permits to exploit this knowledge about the topology of the pattern space for recognition.*

## 1 Introduction

The two-dimensional views of three-dimensional objects change continuously with the orientation of the object in space. This permits to represent intermediate views of an object by interpolation between prototypical views. This idea has been extensively exploited by Vetter and Poggio [14] who demonstrated that it is possible to define linear spaces over sets of views of the same three-dimensional object. For this purpose, the correspondence vector fields that result from the calculation of the correspondences between a set of prototypical views of the object, and a reference image, were used as basis vectors of a linear vector space. Such correspondence vector fields can be linearly superpositioned, resulting in a smooth morphing between the different prototypical views of the

object. This permits to represent a whole class of similar images of an object in a very compact way through the weights of this linear superposition. Since correspondences can be calculated also between images of slightly different objects, e.g. faces of two different individuals, the same technique permits to morph between different faces. Additionally, complex image transformations, like a change of the pose of an object, can be represented as a nonlinear transformation in the space of the weights. Such mappings can be represented with radial basis function networks, and can be learned from a set of example images [1].

The linear combination of prototypical views has been successfully applied to a number of different problems, like the view-invariant recognition of faces [1], for view morphing [7], and for the generation of animation sequences [5]. The underlying strategy has been also generalized to three-dimensional images [13].

These promising results from the application of linear object classes to stationary images has motivated us to investigate if similar ideas can be transferred to the the recognition and synthesis of motion patterns, or actions in video sequences. If it would be possible to represent image sequences by adequately defined linear superpositions of prototypical sequences, the advantages of the linear combination of prototypes could potentially transferred to the recognition an synthesis of movements patterns and actions. It is the aim of this paper to give a first evaluation of this new concept in the context of a simple example. We analyzed the synthesis and the analysis of motion patterns using simulated images of biological motion.

We believe that the proposed method provides a new interesting alternative to existing methods for action recognition, that are usually either model-based (e.g. [10, 4, 15]), or relying on the extraction of adequate spatio-temporal features (e.g. [9, 4, 3]). We assume that our method is also interesting for computer graphics applications since it permits to morph between different actions or movements.

## 2 Linear object classes for stationary images

To set up a framework that will be helpful for the discussion of the theoretical problems that are associated with the generalization of the concept of linear object classes to image sequences, we start with a short review of the ideas of Vetter and Poggio [14].

How can an image be represented by a linear superposition of a set of prototypical example images? The linear combination of the brightness values of images on a pixel-by-pixel basis provides no useful definition for such a superposition. The superposition of the images would look like two transparent objects. Only when the shapes of the two-dimensional projections of the object in the images are exactly the same the superposition would only look natural.

A more useful definition for the linear combination of prototypical images is based on the correspondences between the images. Assume that the image features, e.g. the pixels, are characterized by a vector  $\mathbf{x}$ , and that a reference image is chosen that is characterized by the vector  $\mathbf{x}_0$ . The image can then be characterized by a correspondence vector  $\boldsymbol{\xi} = \mathbf{x} - \mathbf{x}_0$  that describes the spatial shifts between corresponding points of the image and the reference image. Such correspondence vectors can be calculated with a usual optic flow algorithm.

Given a set of prototypical images that are characterized by the feature vectors  $\mathbf{x}_p$ , and by the correspondence vectors  $\boldsymbol{\xi}_p$  with respect to the reference image  $\mathbf{x}_0$ , one can define a nicely interpolating linear combination of images by linearly combining the spatial shift vectors  $\boldsymbol{\xi}_p$  in the form:

$$\boldsymbol{\xi} = \sum_{p=1}^P c_p \boldsymbol{\xi}_p \quad (1)$$

The constants  $c_p$  determine the contributions of the individual prototypical images. From the obtained new correspondence vector  $\boldsymbol{\xi}$  a new image feature vector  $\mathbf{x}$  can be calculated, just by adding the shift vector to the reference vector:  $\mathbf{x} = \boldsymbol{\xi} + \mathbf{x}_0$ . This is a warping of the reference image that is specified by the new correspondence vector  $\boldsymbol{\xi}$ . By applying this method the topology of a linear space is defined onto the set of images. Of course, the prototypical images and the represented image must be sufficiently similar, so that an interpolation between the prototypes makes sense.

The defined linear space of images can be used for the synthesis of new images, and for analysis or recognition. For synthesis prototypical images from different objects, e.g. faces from different persons or

view angles are linearly combined. This leads to a smooth morphing between the prototypical images. New images can be characterized by the relatively low-dimensional coefficient vector  $\mathbf{c} = [c_1, c_2, \dots, c_P]'$ . Complex image transformations, like rotations of a three-dimensional object, can be expressed by the associated changes of the coefficient vector  $\mathbf{c}$ . Transformations of the image space correspond to the application of nonlinear mappings on the coefficient vector  $\mathbf{c}$ . Such mappings can be represented by radial basis function networks [11], that can be trained with a set of example images for which the true transformation is known [1].

For analysis, the correspondence between a new image and the reference image is calculated. The resulting correspondence vector  $\boldsymbol{\xi}$  is then approximated by a linear combination of the prototypical correspondence vectors  $\boldsymbol{\xi}_p$ , e.g. by least squares fitting. The resulting coefficient vectors can be used to estimate the pose parameters of three-dimensional objects. For this purpose, a radial basis function network is trained with image examples that represents the mapping from the coefficient vector  $\mathbf{c}$  onto the pose parameters [1].

## 3 Linear object classes for image sequences

To transfer the idea of a linear combination of prototypes to whole image or video sequences one might first think about a relatively simple solution. The correspondences between the image sequences could be defined by calculating the correspondences between the images for each fixed point in time on a frame-by-frame basis. The linear combination of the prototypical sequences would then be defined simply by a "time-indexed" version of equation (1):

$$\boldsymbol{\xi}(t) = \sum_{p=1}^P c_p \boldsymbol{\xi}_p(t)$$

The disadvantage of this simple procedure is that image sequences with identical spatial structure, but slightly different timing result in large spatial deviation vectors, almost like two sequences that show different movements. Assume that two video sequences show the same arm movement of two different persons, one of which moves his arm faster in the beginning, and slower at the end of the movement than the other. If only spatial displacements enter the correspondence process, these two image sequences would be matched with very strong spatial displacements between the

corresponding image points. A linear combination of prototypes with such large spatial displacement vectors would likely lead to distortions that prevent a smooth interpolation between the arm movements of the two persons.

A useful definition for the "linear combination" of image sequences must, therefore, result in a good interpolation between sequences with slightly different temporal structure. To interpolate smoothly between sequences with similar spatial, but different temporal structure we must admit also *shifts in time* between the corresponding image points in the sequences. This implies that we must calculate *spatio-temporal correspondences*.

These considerations lead to the idea to define not only a spatial displacement vector  $\xi(t)$ , but also a temporal displacement parameter  $\tau(t)$  for each point in time. Instead of the linear superposition equation (1) one obtains the pair of equations:

$$\begin{aligned}\xi(t) &= \sum_{p=1}^P c_p \xi_p(t) \\ \tau(t) &= \sum_{p=1}^P c_p \tau_p(t)\end{aligned}\quad (2)$$

Each correspondence vector has thus a spatial and a temporal component, which are linearly combined using the same weighting coefficients. In the rest of this article, we demonstrate that this definition of a linear combination of image sequences leads to useful results.

## 4 Correspondences in space-time

We have, so far, not specified how spatio-temporal correspondences image sequences can be calculated. The spatio-temporal correspondence algorithm must assign to each spatial point in the feature space at a certain point in time in one image sequence a, usually spatially and temporally shifted point in the another sequence.

It is important to understand that finding spatio-temporal correspondences is an *ill-posed problem*. Assume for a moment that the feature space is only one dimensional. In this case the image sequences can be characterized by two one-dimensional time functions  $x_1(t)$  and  $x_2(t)$ . Assume now, that the two functions specify two movements that have exactly the same spatial structure, but which have a different timing. Let, for instance, the movement of the trajectory  $x_2$  first be slower, and then faster than the movement of  $x_1$ . In this case we could assign a whole continuum of

spatio-temporal shifts that map the two curves continuously onto each other. We could for instance try to minimize the temporal shifts and assign  $\tau(t) = 0$  to each point in time. The deviations between the two functions would then be captured by the spatial shifts. If one, however, favors the other extreme, and imposes no restrictions on the temporal shifts at all, one can obtain zero spatial shifts everywhere just by warping the trajectory  $x_1$  onto the other trajectory in time. (Since we assumed that the two trajectories have the same spatial structure this is always possible.) From this can be concluded that a unique solution for the spatio-temporal correspondence problem requires the inclusion of a priori information about the trade-off between temporal and spatial shifts. This trade-off determines the regimes over which the obtained representation interpolates in space and time.

We have developed a correspondence algorithm that resolves this ambiguity. The algorithm should assign to each point  $\mathbf{x}_2(t)$  on the trajectory  $\mathbf{x}_2$  a corresponding point on the trajectory  $\mathbf{x}_1$ , which not necessarily has the same time value. We introduce, therefore the modified time  $t'$ . The pair of corresponding point to time  $t$  is thus given by the points  $\mathbf{x}_2(t)$  and  $\mathbf{x}_1(t')$ . The corresponding points can also be uniquely characterized by their spatial and temporal shifts that are indexed by the continuous time parameter  $t$ . Mathematically, these shifts are defined by the equations:

$$\begin{aligned}\mathbf{x}_2(t) &= \mathbf{x}_1(t') + \xi(t) \\ t' &= t + \tau(t)\end{aligned}\quad (3)$$

Our correspondence algorithm determines the temporal and spatial shifts by minimizing an error that is the weighted sum of the quadratic spatial and temporal deviations over the whole image sequence. In the time-continuous case, this error is given by the integral:

$$E_c[\xi, \tau] = \int [|\xi(t)|^2 + \lambda \tau(t)^2] dt \quad (4)$$

This error is a functional of the spatial displacement function  $\xi(t)$ , and of the temporal displacement function  $\tau(t)$ . The error has to be minimized under the additional constraint that the mapping between the time variable  $t$  and the modified time  $t'$  for the trajectory  $\mathbf{x}_1(t')$  must be continuous, one-to-one, and monotonically increasing, in order to define unique temporal warping of the sequence  $\mathbf{x}_1$ . This implies for the function  $\tau(t)$  the following constraints:

$$d\tau/dt > -1 \quad (5)$$

$$\tau(0) = \tau(t_{\max}) = 0 \quad (6)$$

For the minimization of the error  $E_c$ , we have developed an algorithm that combines dynamic programming and parametric optimization. (The technical details are discussed in appendix A.)

## 5 Application for the synthesis of new image sequences

To test the appropriateness of our idea of a linear combination of prototypical image sequences, we generated a set of synthetic image sequences that showed a stick figure that performs three different walking styles (walking, running and limping). Using a 3D-model for the stick figure, we generated for each of the three walking styles two-dimensional image sequences from five different view angles (one view was directly from the side; for the additionally views, the "camera" axis was rotated either  $\pm 25$  deg up and down, or 25 deg right or left. The prototype for "limping" was obtained from the image sequence for "walking" by rewarping the image sequence for walking in time, by first slowing the movement down, and then increasing its speed in order to keep the cycle time of the movement constant.) In total, we used 15 different prototypical sequences (five different view angles for each of the three walking styles), each sampled with only 21 discrete time steps.

We used the two-dimensional joint positions of the figure as feature vectors  $\mathbf{x}(t)$ . The algorithm described above was applied for the calculation of the correspondences between the prototypical patterns and a reference sequence, which was the side view of walking. The obtained spatio-temporal correspondence fields of the prototypes were then linearly combined according to equation (2) using weights that fulfilled the conditions  $c_p \geq 0$  and

$$\sum_{p=1}^P c_p = 1.$$

The resulting new correspondence vector field was then used to warp the reference sequence  $\mathbf{x}_1(t)$  to a new synthetic sequence that is defined by the equation:

$$\mathbf{x}(t) = \mathbf{x}_1(t + \tau(t)) + \boldsymbol{\xi}(t) \quad (7)$$

Our tests showed that the synthesized new image sequences interpolated smoothly between the prototypical sequences of the *same* walking style. The quality of the resulting interpolated motion sequences was comparable to the quality of the prototypical sequences themselves. Interestingly, also linear combinations of *different* walking styles looked relatively

natural, as long as the view angles were similar. A combination of "walking" and "running" with equal weights looks like slow running or quick walking. A mixture of "limping" and "running" looks like a weak form of limping. This shows that, at least with the stick figures, the proposed method permits to interpolate smoothly between different views of a single walking style, and also between different action pattern. In particular, the smooth interpolation between walking and limping, that differed only with respect to their temporal structure, shows that the method fulfills the constraint that was formulated above: it interpolates smoothly between patterns that differ with respect to their temporal structure.

When two different walking styles with different view angles were combined the resulting motion sequences showed distortions. This shows that not all motion patterns can be linearly combined equally well. The combination of some patterns, like prototypes of the same walking style, or different walking styles with the same viewing parameters, leads to useful interpolated patterns. Combinations of prototypes that differ in both, view angle and walking style seem not to form useful linear combinations. More abstractly, this can be interpreted as evidence for the existence of a topology over the set of image sequences. In this topology, some prototypical patterns seem to be close, like prototypes for the same walking style, or different walking styles with the same view parameters, and an interpolation between them led therefore to useful results. Other sequences, like running and limping from different view angles seem to further apart, so that an interpolation between them leads not to useful results. Interestingly, the mixture of all prototypical patterns with equal weights leads to a natural pattern, potentially because the characteristic extreme properties of the different prototypes are averaged out. We are presently investigating if similar results can also be obtained with real image sequences.

## 6 Analysis of image sequences

To test if the linear combination of prototypical sequences can also be used for the recognition of walking styles and their view parameters, we simulated new synthetic image sequences that showed the walking styles from many different view angles. We tried to recover the walking style, and the view parameters (rotation angles of the camera).

Let us assume first that the walking style is already known. In this case the view parameters can be recovered by approximating the spatial and temporal shifts

$\xi(t)$  and  $\tau(t)$ , that are associated with the new sequence, by a linear combination of the shifts of the prototypical sequences of this walking style, according to equation (2). This can be easily achieved by minimizing a composite error that is a weighted sum of the squared errors of the spatial and the temporal deviations of the approximation:

$$E_a(\mathbf{c}) = \int \left[ \xi(t) - \sum_{p=1}^P c_p \xi_p(t) \right]^2 + \lambda_a \left( \tau(t) - \sum_{p=1}^P c_p \tau_p(t) \right)^2 dt \quad (8)$$

The parameter  $\lambda_a$  determines the trade-off between spatial and temporal deviations. The coefficient vector  $\mathbf{c}$  that minimizes the error  $E_a$  can be found by solving the linear equation system

$$\mathbf{K} \mathbf{c} = \mathbf{a} \quad (9)$$

where the elements of the  $P \times P$  matrix  $\mathbf{K}$  are given by

$$K_{pq} = \int [\xi_p(t)' \xi_q(t) + \lambda_a \tau_p(t) \tau_q(t)] dt \quad (10)$$

and the elements of the vector  $\mathbf{a}$  by

$$a_p = \int [\xi(t)' \xi_p(t) + \lambda_a \tau(t) \tau_p(t)] dt \quad (11)$$

for  $1 \leq p, q \leq P$ . This equation system was solved using a singular value decomposition method. The coefficient vectors were then used as input signals for a radial basis function network with gaussian basis functions that maps the coefficient vectors onto the view angles. This network had been trained with 16 image sequences with known pose parameters. The width of the gaussian basis functions was optimized for good interpolation properties of the networks. The view angles were recovered with a precision of about 2 deg in the regime  $\pm 25$  deg for both viewing angles by this procedure.

Recovering the type of the walking style from the weights of the linear combination turned out to be more difficult. The reason for that is that an application of usual least squares techniques for the estimation of the view parameters fails when prototypes from all different walking styles are used at the same time. The reason is the high ambiguity in the possibility to decompose the new correspondence vector in linear combinations of the correspondence vectors of the prototypes. This results in coefficient vectors

that load highly on many different walking patterns, and also on combinations of coefficients that do not specify useful linear combinations of the prototypes (for instance "running" and "limping" from different view angles). This is illustrated in Fig. 1 (left), where the gray levels encode the absolute values of the coefficients  $c_p$ . Along the vertical axis of this plots the coefficient vector elements are ordered according to the walking style of the associated prototype (W: walking, R: running, L: limping). Each segment on the horizontal axis indicates one of 48 test sequences that had to be classified as one of the three different walking styles. The left figure shows the results that are obtained using a least squares method for the estimation of the coefficients. Examples for a certain walking style, like "running", lead to substantial load on the coefficients also of other walking styles, like "limping". This unstable estimation makes a reliable recognition of the walking style from the weights impossible. This is true even though we applied singular value decomposition and regularization techniques in order to stabilize solution of the least squares estimation problem.

The deeper mathematical reason for this instability is that the set of prototypical correspondence vectors defines linear function set for approximation that is too rich (its *capacity* is too large). The resulting instability in the estimation of the coefficients could potentially be resolved by adding more features, that could help to disambiguate the linear estimation problem. We will take another route here that exploits the knowledge that we have about the topology of the space of image sequences. It seems reasonable to require that prototypes that do not specify usefully interpolating intermediate patterns if they are superpositioned, should not contribute at the same time to the linear approximation. This helps to avoid that the motion pattern is approximated by linear combinations that do not correspond to usefully interpretable image sequences.

How can this a priori knowledge be integrated in the estimation of the coefficient vector  $\mathbf{c}$ ? We propose here a numerical technique that is based on the *structural risk minimization (SRM)* principle that has been formulated by Vapnik [12]. The idea of structural risk minimization is to look for a solution with minimal capacity of the associated function set, instead of minimizing only the approximation error in the least squares estimation in (9). We propose here to embed the a priori knowledge into the capacity control, resulting a *modified structural risk minimization (MSRM)* method. To measure the capacity we use the

function:

$$E_s(\mathbf{c}) = |\mathbf{c}'\mathbf{W}\mathbf{c}| \quad (12)$$

When  $\mathbf{W}$  is the unit matrix, this function goes over in the function that is usually applied for structural risk minimization, which results in a solution that minimizes the VC dimension [12]. We modified this term by admitting additional positive values in the symmetric matrix  $\mathbf{W}$ . All elements that correspond to pairs of coefficients that specify meaningless combinations of prototypes were set to high positive values. This leads to a strong suppression of such weight combinations in the solution.

The function  $E_s$  has to be minimized under the constraint  $\mathbf{K}\mathbf{c} = \mathbf{a}$ . In practice, this constraint can be often only approximately fulfilled, because of noise and inconsistencies in the data. This makes it necessary to induce a "slack variable" vector  $\boldsymbol{\zeta}$  into the problem that absorbs the deviations from the equality constraint. Instead of the function in (12) we minimize thus the expression

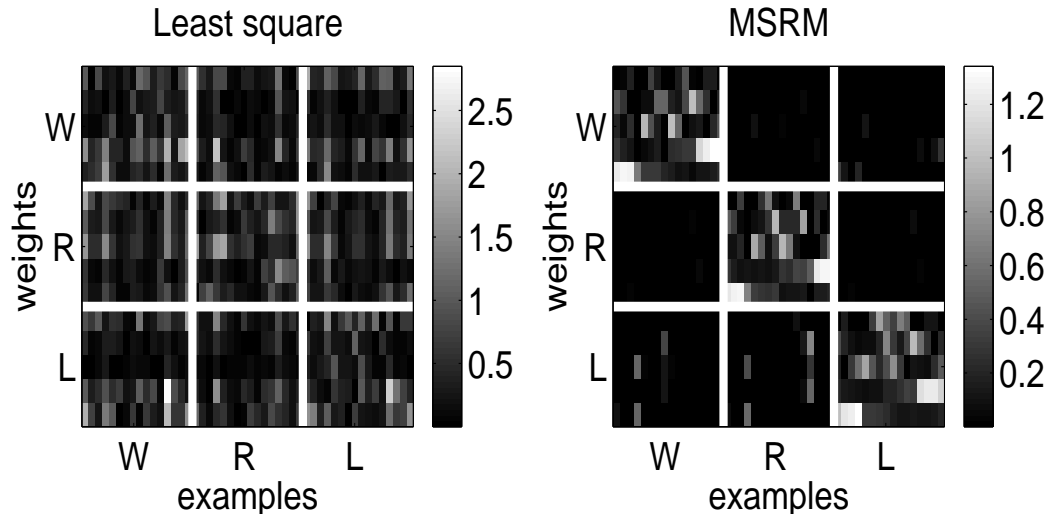
$$\tilde{E}_s(\mathbf{c}) = |\mathbf{c}'\mathbf{W}\mathbf{c}| + C|\boldsymbol{\zeta}|^2 \quad (13)$$

under the constraint  $\mathbf{A}\mathbf{c} = \mathbf{a} + \boldsymbol{\zeta}$ . The large positive constant  $C$  determines how strongly deviations from the equality constraints are punished in the optimization process. It is shown in appendix B that this optimization problem can be transformed into a quadratic programming problem for which effective solution algorithms exist.

Fig. 1 (right) shows that our MSRM method leads to approximations with relatively few non-zero coefficients that are usually corresponding to prototypes of the true walking pattern. This makes it possible to use the linear weights for a classification of the walking patterns. We used the  $L_1$ -norm of the parts of the coefficient vector that belong to the individual walking styles as discriminating function for the classification. For the tested 48 examples the misclassification rate using the MSRM method for the estimation of the coefficients was low ( $< 5\%$ ). Applying the same classification rule to the coefficient vectors that were obtained using least squares fitting lead to very high error rates of the classification (about 30%). This shows that the increased numerical effort that is required for the solution of the quadratic programming problem is justified.

A very interesting result was obtained when a new walking pattern was generated between walking and running, by adequately mixing the coefficients of the 3D-simulation program. In this situation the relative size of the  $L_1$ -norms of the coefficient vectors associated with running and walking prototypes covaried exactly with the relative influence of the two walking styles in the simulation program. This shows that our method permits a gradual classification of action patterns that permits results of the form:  $x\%$  "walking" or "running". The method performs thus a mixture between classification and parametric regression.

Fig. 1:



## 7 Conclusions and outlook

In this article we have presented first results that explore the idea of a linear combination of prototypical motion sequences for the synthesis and analysis of video sequences. We are presently trying to generalize the same techniques for an application to real video sequences.

Our results seem to be promising for analysis and synthesis. For synthesis it has to be explored if interpolated sequences with sufficiently high quality can be obtained, for instance by a separate linear combination of texture (brightness) and shape (spatial shifts) [1]. For recognition, potentially higher distortions of the interpolated patterns can be tolerated. Instead of using specific object points as features we are planning to apply optic flow algorithms to real image data (cf. also [14, 1]). Using characteristic key points to simplify the correspondence process may however be helpful in graphics animation applications of our method.

The proposed method for the integration of a priori knowledge in the estimation of the linear weights seems to be an interesting extension of the structural risk minimization principle. Most applications so far, have mainly focused on minimizing the VC dimension of the approximating function set, but did not include problem-specific a priori knowledge about the structure of the admissible function set. The quadratic programming procedure sets irrelevant coefficients, and coefficients that do not fulfill the constraints to zero. This should lead to a high robustness of the described method, in comparison with alternative procedures that are based on weighted least squares estimation.

### Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft.

### Appendix A: Details of the correspondence algorithm

The real implementation deals with image sequences that are sampled equally-spaced in time with sampling time  $T$ , indicated in the following by  $\mathbf{x}_i[n]$ . The algorithm that we used was inspired dynamic time warping methods in speech recognition [8]. (For an application of similar techniques to the time warping of gestures see [2].) Compared to such algorithms, our method tries to minimize the computational effort by minimizing the number of sampled image frames, and by warping between the sampling images. In this way we can emulate a time continuous sequence of images.

Our correspondence algorithm consists of two steps. The first step is based on a dynamic programming algorithm for path optimization. Given the  $N$  frames of the two image sequences  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , that have to be brought into correspondence, we calculate for each frame pairing with discrete times  $n$  and  $n'$ ,  $1 \leq n, n' \leq N$ , the error function value:

$$E_d(n, n') = |\mathbf{x}_1[n] - \mathbf{x}_2[n']|^2 + \lambda(n - n')^2 T^2 \quad (14)$$

The dynamic programming algorithm tries to find a path in the  $n$ - $n'$ -plane with minimum cost, where the cost is the sum values of the function  $E_d$  along the path. The algorithm starts with the index pair  $n = 1$  and  $n' = 1$ . Along the path  $n$  always increases by one. To implement the monotonicity constraint (5), the set of permitted path transitions for  $n'$  is restricted. If we indicate by  $n'[n]$  the value of the index  $n'$  that is associated with the index  $n$ , the possible values of  $n'[n]$  are restricted through the inequality:

$$n'[n - 1] \leq n'[n] \leq n'[n - 1] + 2 \quad (15)$$

To enforce the end-point constraints (6) we introduced the additional restrictions:

$$2n - N \leq n'[n] \leq N \quad (16)$$

The second step of our algorithm determines the exact spatial and temporal shifts by linearly interpolating between the discrete frames of the sequence  $\mathbf{x}_1$ . For this purpose we construct a time-continuous sequence by warping linearly between the frames of the time discrete sequence. The interpolated (quasi) time-continuous image sequence in the time intervals  $I_1 = [(n - 1)T, nT]$  and  $I_2 = [nT, (n + 1)T]$  is given by the equation:

$$\mathbf{x}_1(t) = \begin{cases} \mathbf{x}_1[n] - (n - t/T) \mathbf{d}_1[n - 1] & \text{for } t \in I_1 \\ \mathbf{x}_1[n] + (n - t/T) \mathbf{d}_1[n] & \text{for } t \in I_2 \end{cases} \quad (17)$$

with

$$\mathbf{d}_1[n] = \mathbf{x}_1[n + 1] - \mathbf{x}_1[n]. \quad (18)$$

Introducing this approximation into the error function (4), one can analytically calculate the optimal time shifts within the two time intervals. (The integral was replaced by a sum over the discrete time events that are indexed by  $n$ .) resulting in the expression

$$\tau(nT) = T \left( \frac{n|\mathbf{d}_1|^2 + \lambda T^2 n' \pm \mathbf{d}'_{21}[n', n] \mathbf{d}_1}{\lambda T^2 + |\mathbf{d}_1|^2} - n' \right) \quad (19)$$

where the vector  $\mathbf{d}_{21}$  is given by

$$\mathbf{d}_{21} = \mathbf{x}_2[n'] - \mathbf{x}_1[n].$$

In the interval  $I_1$  the first sign has to be chosen, and  $\mathbf{d}_1 = \mathbf{d}_1[n - 1]$ , whereas for the interval  $I_2$  the second sign is valid and  $\mathbf{d}_1 = \mathbf{d}_1[n]$ . In this way two candidate values for the time shift  $\tau(nT)$  are obtained, one for each

interpolation interval. We selected the value that led to a smaller value of the error function  $E_c$ . Given the calculated optimal time shifts, one can use equation (17) to obtain the associated optimal spatial shifts.

## Appendix B: Quadratic programming problem for the MSRM

By introducing the new non-negative variable vectors  $\mathbf{z}$  and  $\mathbf{z}^*$  one can rewrite the error function in (13) in the form:

$$\tilde{E}_m(\mathbf{z}, \mathbf{z}^*) = [\mathbf{z}' \ \mathbf{z}^{*'}] \mathbf{W}_m \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} + C |\zeta|^2 \quad (20)$$

where the symmetric matrix  $\mathbf{W}_m$  is given by

$$\mathbf{W}_m = \begin{bmatrix} \mathbf{W} & \mathbf{W}_0 \\ \mathbf{W}_0 & \mathbf{W} \end{bmatrix} \quad (21)$$

and where  $\mathbf{W}_0$  by setting the diagonal elements of  $\mathbf{W}$  to zero. (This procedure is important to obtain a non-degenerate quadratic term, since otherwise the matrix has rank zero.)

The constraints have to be reformulated in the form:

$$[\mathbf{K}, -\mathbf{K}] \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} = \mathbf{a} + \zeta \quad (22)$$

$$\mathbf{z} \geq \mathbf{0} \quad (23)$$

$$\mathbf{z}^* \geq \mathbf{0} \quad (24)$$

This is a standard quadratic programming problem with a number of equality and inequality constraints for which efficient solution algorithms exist [6]. The quadratic programming algorithm was part of the MATLAB optimization toolbox. We removed redundant (almost) linear dependent equations from the constraint system  $\mathbf{Kc} = \mathbf{a}$  to ensure a fast convergence of the algorithm.

## References

- [1] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 1996.
- [2] T. J. Darrell, I. A. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. Technical Report 364, Massachusetts Institute of Technology, Cambridge, MA, 1995.
- [3] J. W. Davis and A. F. Bobick. The representation and recognition of action using temporal templates. Technical Report 402, Massachusetts Institute of Technology, Cambridge, MA, 1996.
- [4] I. A. Essa and A. P. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19:(in press), 1997.
- [5] T. Ezzat and T. Poggio. Miketalk: A talking facial display based on morphing visemes. In *Proceedings of the Computer Animation Conference, Philadelphia, PA*, 1998.
- [6] P. E. Gill, W. Murray, and M. H. Wright. *Mathematical optimization*. Academic Press, London, 1981.
- [7] M. J. Jones. *Multidimensional morphable models: A framework for representing and matching object classes*. PhD thesis, Dept. of Computer Science, Cambridge, MA, 1997.
- [8] B. H. Juang and L. R. Rabiner. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [9] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking figures in XYT. Technical Report 223, Massachusetts Institute of Technology, Cambridge, MA, 1994.
- [10] J. O'Rourke and N. I. Badler. Model-based analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2:522–536, 1982.
- [11] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 1990.
- [12] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [13] T. Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28(2):103–116, 1998.
- [14] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(7):733–742, 1997.
- [15] Y. Yacoob and M. J. Black. Parametrized modeling and recognition of activities. *Computer vision and Image Understanding*, (in press), 1999.