
Object Correspondence as a Machine Learning Problem

Bernhard Schölkopf
Florian Steinke

Max-Planck-Institut für biologische Kybernetik, Tübingen, Germany

Volker Blanz

Max-Planck-Institut für Informatik, Saarbrücken, Germany

BS@TUEBINGEN.MPG.DE
STEINKE@TUEBINGEN.MPG.DE

BLANZ@MPI-SB.MPG.DE

Abstract

We propose machine learning methods for the estimation of deformation fields that transform two given objects into each other, thereby establishing a dense point to point correspondence. The fields are computed using a modified support vector machine containing a penalty enforcing that points of one object will be mapped to “similar” points on the other one. Our system, which contains little engineering or domain knowledge, delivers state of the art performance. We present application results including close to photo-realistic morphs of 3D head models.

1. Introduction

Over the last decade, machine learning has found its way into a number of engineering domains, ranging from machine vision and telephony to bioinformatics. A large and important area where machine learning applications are relatively sparse, however, is the field of computer graphics. Although there have been some learning applications (Beymer & Poggio, 1994; Bregler & Omohundro, 1995; Freeman et al., 2002; Grochow et al., 2004; Saul & Jordan, 1997, e.g.), as well as the use of statistical techniques such as PCA in graphics, it is probably fair to say that when compared to computer vision (its sister discipline), computer graphics has so far seen relatively little applications of state-of-the-art machine learning methods. The present paper deals with a problem which we believe is both important for computer graphics and very well suited to machine learning, making it an ideal sandbox for modern machine learning methods. The techniques which



Figure 1. The images of the two heads (left and right) are linearly superposed (middle), leading to an image which no longer is an admissible head. Under pixelwise addition, the class of head images is not closed and does thus not form a linear space.

we will apply belong to the class of kernel methods, which are considered to be among the most accurate and easy-to-deploy machine learning algorithms (Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002).

Most kernel methods, including the Support Vector Machine (SVM), share the use of a *positive definite kernel* $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Here, \mathcal{X} is the domain in which the empirical data live. Positive definite kernels are characterized by the property that there exists a mapping Φ from \mathcal{X} into a Hilbert space \mathcal{H} (the *reproducing kernel Hilbert space* associated with k) such that for all $x, x' \in \mathcal{X}$,

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle. \quad (1)$$

This paper presents a kernel method for the estimation of correspondences between objects. It is organized as follows. The next section briefly reviews the correspondence problem, including some prior work in the field. Section 3 describes our algorithm for computing deformation mappings between objects. The approach builds on an SV regression method for implicit surface modeling, described in Section 4. In Section 5, we show some application results, and we conclude with a discussion (Section 6).

2. Correspondence and Morphing

The study of object correspondence is one of the basic issues of computer graphics, and is intimately related to the problem of morphing. To illustrate it, consider Figure 1. If we are given two face images I_1, I_2 (for the sake of simplicity, consider b/w images and think of them as scalar functions on $[0, 1]^2$), and we would like to produce a third image which is “in between” the other two, then we might first attempt to simply compute a superposition $\frac{1}{2}I_1 + \frac{1}{2}I_2$. However, as the figure illustrates, this does not lead to satisfactory results. It produces ghost contours caused by the fact that, for instance, the eyes of the two faces were not aligned before computing the superposition. If, on the other hand, we manage to align all relevant facial features in I_1 with the corresponding ones in I_2 , then the two faces are said to be *in correspondence*. Obviously, such an alignment will typically not be possible by, say, an affine transformation of the image plane. Usually, it is characterized by a *warp* or *deformation field* $\tau : [0, 1]^2 \rightarrow [0, 1]^2$, which, given any point in I_1 , specifies to which point in I_2 it corresponds.¹ Assuming that τ is onto, the warped image I'_1 is constructed as follows: the intensity $I'_1(x', y')$ is the mean of all $I_1(x, y)$ satisfying $\tau(x, y) = (x', y')$. Whilst the warped I'_1 will usually not be equal to I_2 , at least the facial features in I'_1 will be aligned with the ones of I_2 , in which case images like $\lambda I'_1 + (1 - \lambda)I_2$ should look like meaningful face images. Once a class of images is thus brought into correspondence, such superpositions will not take us outside the class; it is in this sense that people refer to such classes as *linear object classes* (Vetter & Poggio, 1997; Blanz & Vetter, 1999).

The present paper proposes a machine learning method for computing warps between general object classes, focusing on surfaces embedded into 3D. Automated algorithms for computing warps between surfaces have been presented previously. For parameterized surfaces of human faces that were captured with a laser scanner, Blanz and Vetter (1999) proposed a modified optical flow algorithm; cf. also (Cootes et al., 1998) for an approach based on corresponding pairs of landmark points. For shapes such as animals or human bodies, methods have been developed that match each mesh vertex of the first shape to the most similar point on the second mesh (Shelton, 2000; Allen et al., 2003).

¹This is sometimes called a *forward* warp, as opposed to a *backward* warp, where the roles of I_1 and I_2 are interchanged. For continuous images and invertible warps, the backward warp is the inverse of the forward warp; however, if the images are spatially discretized, the warps are not guaranteed to map pixels to individual pixels, and interpolation is necessary.

These methods minimize the distance to the target mesh and maximize the smoothness of the deformation in terms of stiffness of the source mesh (Shelton, 2000) or the similarity between transformations of adjacent vertices (Allen et al., 2003). For matching partially overlapping portions of the same surface, Iterative Closest Point Algorithms (Besl & McKay, 1992; Rusinkiewicz & Levoy, 2001) provide a reliable solution. A variety of methods are available for medical data registration (Audette et al., 2000).

In contrast to deformations defined only on the surface, a volume deformation algorithm based on free-form-deformations with B-Splines between landmark points has been described for MRI scans of human brains (Rueckert & Frangi, 2003). Müller et al. (2004) extend a physically plausible deformation from a set of sample points to the whole object using a Moving Least Squares approach. Cohen-Or et al. (1998) morph two objects into each other by first applying an elastic deformation based on landmark point correspondences and then blending two implicit functions describing the objects into each other.

The algorithm to be described below will belong to the class of volume deformation methods — it will compute a deformation field not only for the object surfaces, but also in a neighborhood around the surfaces, as chosen by the user.

3. Learning Correspondence

Suppose we are given two objects O_1 and O_2 , both being elements of an object class \mathcal{O} . We assume the objects are embedded in a domain \mathcal{X} , which we may think of as a subset of \mathbb{R}^D .

3.1. Locational Cost

The main idea of our approach is that intuitively, a good warp from O_1 to O_2 has the general property that it will map $x \in \mathcal{X}$ to a point $\tau(x)$ such that *relative to x , O_1 looks like O_2 relative to $\tau(x)$* . For instance, it could be that O_1 in the vicinity of x is similar to O_2 in the vicinity of $\tau(x)$, with respect to a specific discrepancy measure.

We formalize this intuition in a cost function

$$c_{loc}(O_1, x, O_2, \tau(x)). \quad (2)$$

Due to its conceptual similarity to the *locational kernels* defined by Bartlett and Schölkopf (2001), we refer to c_{loc} as a *locational cost function*. Note that such kernels or cost functions are actually used in a variety of domains, including bioinformatics (Zien et al., 2000).

Locational cost functions can readily be constructed if, for instance, we are given *feature functions* $f_1, f_2 : \mathcal{X} \rightarrow \mathbb{R}$ capturing relevant properties of O_1, O_2 . For our warping application, a good choice of such functions are those that have the object surfaces as their zero sets, called *implicit surfaces*; in particular, the so-called *signed distance function*: at any given point in space, the modulus of the signed distance function is the distance of the point to the object surface, and its sign is positive outside the object and negative inside. By definition, the zero set of the signed distance function coincides with the object’s surface, thus the signed distance function implicitly describes the surface. In Section 4, we will describe a way to construct an approximation of the signed distance function which we will use in our experiments.

Below, d is a metric, and we may think of f_1, f_2 as the signed distance functions of O_1, O_2 . We now list some examples of locational cost functions $c_{loc}(O_1, x, O_2, \tau(x))$,

1. preserving signed distances,

$$d(f_1(x), f_2(\tau(x)))^2 \quad (3)$$

(in particular, this measures whether surface points get mapped to surface points),

2. ... and higher order differential properties,

$$\sum_{i=0}^{\infty} \alpha_i d(\nabla^i f_1(x), \nabla^i f_2(\tau(x)))^2, \quad (4)$$

where the α_i are weighting coefficients determining the contribution of the higher order terms,

3. ... over neighborhoods,

$$\sum_{\Delta x \in \mathcal{N}} d(f_1(x + \Delta x), f_2(\tau(x) + \Delta x))^2, \quad (5)$$

where \mathcal{N} is some “neighborhood” set such as the vertices of a regular simplex centered at 0,

4. ... using multiple functions,

$$\sum_i d(f_1^i(x), f_2^i(\tau(x)))^2, \quad (6)$$

where the f^i are different feature functions computed from the objects, e.g., implicit surface approximations computed at different length scales.

5. One can also define locational cost functions from kernels: a locational kernel is a positive definite kernel on $(\mathcal{O} \times \mathcal{X}) \times (\mathcal{O} \times \mathcal{X})$. If Ψ is the associated feature map, we could use

$$\|\Psi(O_1, x) - \Psi(O_2, \tau(x))\|^2. \quad (7)$$

Note that if d is the standard Euclidean metric in the first four examples, then the last example is the most general — it subsumes the other ones as special cases.

3.2. Landmark Training Point Cost

Computer graphics methods for determining correspondence are often based on a set of landmark points. When computing correspondence between images, say, these are typically points that a user manually clicks on in both images, such as the tip of the nose or the corners of the eyes. If available, such pairs of landmark points $(x_1, z_1), \dots, (x_m, z_m)$ (where x_i belongs to O_1 and z_i to O_2) can be incorporated into the objective function using a term $c_p(\tau(x_1), \dots, \tau(x_m), z_1, \dots, z_m)$.

The above joint dependence of the cost on all point pairs allows the incorporation of constraints on the relative position of points. However, for simplicity, we will in our experiments only use an additive cost

$$c_p(\tau(x_1), \dots, \tau(x_m), z_1, \dots, z_m) = \sum_{i=1}^m \|\tau(x_i) - z_i\|^2. \quad (8)$$

3.3. Function Class and Regularizer

Locational costs and pairs of landmark points may not sufficiently constrain the problem to lead to a satisfying overall solution. We thus incorporate an SVM style large margin regularizer, and optimize over a class of warps where each component function is expressed as a linear function in a reproducing kernel Hilbert space \mathcal{H} induced by a kernel k ,

$$\tau_d(x) = x_d + \langle \mathbf{w}_d, \Phi(x) \rangle. \quad (9)$$

Here, Φ is the feature map of k , x_d is the d th component of x (likewise for τ_d), and $\mathbf{w}_d \in \mathcal{H}$.

As a regularizer, we will use

$$\frac{1}{2} \sum_{i=d}^D \|\mathbf{w}_d\|^2, \quad (10)$$

thus regularizing the component functions separately. Better performance may be possible using more sophisticated regression schemes (Weston et al., 2003; Bakir et al., 2004; Micchelli & Pontil, 2005, e.g.), but we do not pursue this in the present paper.

3.4. Objective Function

Putting together the pieces, we end up with an objective function of the form

$$\frac{1}{2} \sum_{d=1}^D \|\mathbf{w}_d\|^2 + \lambda_p \sum_{i=1}^m \|\tau(x_i) - z_i\|^2$$

$$+\lambda_{loc} \int_{\mathcal{X}} c_{loc}(O_1, x, O_2, \tau(x)) d\mu(x), \quad (11)$$

to be minimized over warps τ of the form (9). The parameters $\lambda_p, \lambda_{loc} \geq 0$ determine the relative influence of the error terms, and μ is a measure whose support covers the area in which we want to estimate the deformation field, e.g., the vicinity of the surface of O_1 . To turn this optimization problem into a practical one, we use the empirical measure arising from sampling a set of points in the desired area. The generalized representer theorem (Wahba, 1990; Schölkopf & Smola, 2002) then states that the optimal deformation τ can be expressed as a finite expansion of kernels centered at the training points, i.e., the landmark correspondences and the above sampled points. We thus need to optimize only over the expansion coefficients.

In general a warp can be split into a rigid and an elastic transformation. We focus on the latter, as there are numerous algorithms to estimate the rigid part based on some landmark points (Haralick & Shapiro, 1992).

3.5. Optimization

However, the optimization problem (11) remains a difficult non-convex problem with spurious local minima. In such problems, it is helpful to construct a good starting point. If we have landmark points (i.e., if $m > 0$), one way to proceed is to set λ_{loc} to zero initially. In that case, the problem can be decomposed into D problems of the form

$$\underset{\mathbf{w}_d \in \mathcal{H}}{\text{minimize}} \frac{1}{2} \|\mathbf{w}_d\|^2 + \lambda_p \sum_{i=1}^m |\tau_d(x_i) - z_{i,d}|^2. \quad (12)$$

This is a convex quadratic program corresponding to an SVM with a squared loss function. Taking this as an initial solution, we then optimize (11) using gradient descent.

As a kernel, we use the Wu function (Schaback, 1995)

$$k(r) = (1-r)_+^4 (4 + 16r + 12r^2 + 3r^3), \quad (13)$$

where $r = \frac{\|x-y\|}{\sigma}$. It has compact support of size $\sigma > 0$ (leading to sparse kernel matrices) and is in $C^2(\mathbb{R}^3)$.

To further stabilize the optimisation process, we apply a multi-scale scheme. We use wide kernels in order to make sure that the sparse landmark point correspondences lead to a good initial guess in a larger vicinity. For matching detail structures on the surface of the second object, we need enough flexibility in the model as provided by smaller kernels. We iterate the optimization procedure from coarse to fine and approximate the residual errors on the next finer scale.

We stop the refinement process once the kernel width reaches the size of the smallest surface features. As kernel widths, a cascade of $1/2, 1/4, \dots$ of the diameter of the object is used.

The volume \mathcal{X} is sampled at each scale at a resolution necessary to yield on optimisation problem as small as possible. We extract approximately equally spaced points by recursively subdividing an initial bounding box until the boxes reach a resolution smaller than a chosen fraction the kernel width (in our case $\frac{1}{2}$). We then take the center point of those boxes to be a training point in our optimisation problem. Boxes are just subdivided if they lie within the volume \mathcal{X} . This way we efficiently construct a uniform sampling yielding a sparse kernel matrix with a time requirement proportional to the sampling volume.

In the experiments to be shown below, we use Euclidean 2-norms everywhere, and as a locational cost function we use one which depends on signed distances f_1, f_2 as well as their gradients (cf. (4)), where f_i is computed from O_1 as described in the next section. Although preserving the warped points' distance to the surface (and the gradients of the signed distance function) does not completely determine their 3D position, it turns out that together with the regularization, this will lead to rather good overall deformation fields.

4. Surface Reconstruction

Implicit surface modeling is based on the idea that a surface can be described as the set of all $x \in \mathcal{X} \subseteq \mathbb{R}^D$ (D being the dimension of input space) for which a function $f : \mathcal{X} \rightarrow \mathbb{R}$ equals zero. The method we are using is described in detail in (Steinke et al., 2005); thus we presently give only a brief account.

Similar to Walder and Lovell (2003) and Schölkopf et al. (2005), we model the surface as a hyperplane in the reproducing kernel Hilbert space \mathcal{H} , i.e., as the zero set of

$$f(x) = \langle \mathbf{w}, \Phi(x) \rangle + b, \quad (14)$$

where $\mathbf{w} \in \mathcal{H}, b \in \mathbb{R}$. Using SV regression, f can be written as

$$f(x) = \sum_{i=1}^m \alpha_i k(x_i, x) + b, \quad (15)$$

where k satisfies (1). As input for the construction process we use points on the surface of the object as well as surface normals. The regression is trained both on the given surface points (with target values 0) and on especially constructed off-surface points (see below).

We use a modification of standard ϵ -insensitive SV regression by setting the offset b to a fixed value. With

this change and a compactly supported radial basis function (such as the Wu function (13)), the dual problem turns out to be a convex quadratic program with just a box constraint and a sparse kernel matrix. We can solve this problem rather efficiently by applying a coordinate descent method.

As above, we use a multi-scale scheme to improve the interpolation properties of the implicit surface function and have enough flexibility at the same time. The off-surface training points are constructed with a method similar to the one used by Carr et al. (2001): We construct training points by displacing the surface points along their surface normals. If those are not directly given from the laser scanner, they can be estimated from nearest neighbor information (provided the sampling is sufficiently dense).

Overall, this procedure efficiently and reliably constructs a signed distance approximation in the vicinity of the surface of the object which can be used as feature function in our warping experiments.

5. Experimental Results

We first show results on synthetic 2D data, illustrating the effect of different locational cost functions. Figure 2 shows some simple geometric 2D outlines which we approximate as zero sets of kernel expansions, using the method of Section 4. We then compute deformation fields which turn the first object into the second one, using different cost functions (Section 3.1). The results show that compared to a simple blending of implicit functions, a cost function preserving distances (3) leads to better results, and a cost function incorporating distances and gradients (cf. (4)) yields a rather good morph, computed by transforming the left image with 50% of the deformation field. This will henceforth be our default cost function.

We next move to 3D data, including head scans from the MPI face database (Blanz & Vetter, 1999), a scan of an artifact, available as a 3D demo object from Cyberware, and two chess pieces taken from <http://www.buckrogers.demon.co.uk>. Figures 3 – 6 show that very realistic results are achieved, with little or no user intervention.

The computational effort to construct these warps depends significantly on how one chooses the warping volume \mathcal{X} and how densely it is sampled. In most cases, we used the vicinity of the surface as our training region and typically constructed around 100k points for the 3D morphs. The runtime using our current implementation, which is not yet optimized for speed, is then about 2h on a recent Pentium processor.

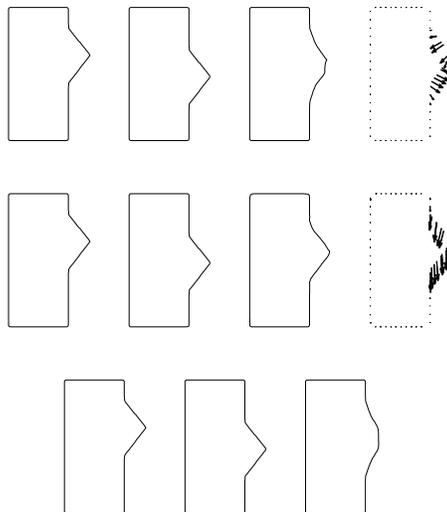


Figure 2. 2D toy example of different morphs between the two left images. **First row:** Using the cost function (3) which attempts to preserve distances to the surface, we obtain a 50% morph (third image) which is reasonable but not optimal. The warp field, shown on the right, reveals that the algorithm tries to shrink the upper part and expand the lower part of the triangular protrusion. **Second row:** If we use a cost function which is also sensitive to gradients of the signed distance function (cf. (4)), then the warp is forced to displace points on the top and bottom sides of the triangle to corresponding points on the shifted triangle. This leads to a satisfactory morph (third image from the left). **Third row:** One might think that a good morph could also be obtained by simply taking a convex combination (with equal weights) of the two signed distance functions, as approximated using our kernel method, and rendering its zero set. The last image shows that this approach already fails for our toy example.

The main parameters of our method are the kernel width σ and the regularization constants λ_p, λ_{loc} (Section 3.4). The former was chosen as described in Section 3.5; the latter were set manually by visually inspecting the results. While it is standard procedure in computer graphics research to have a number of hand-tunable parameters, it would clearly be desirable from a machine learning point of view to develop some automatic means of parameter tuning. This was not yet done in the present work; however, we note that there are several possibilities. (1) In principle, we can use a holdout subset of the unlabelled data $(x_i, \tau(x_i))$ entering into the locational cost term. The advantage of this is that we have an unlimited supply of such data; however, having a small holdout error on this kind of data points is only necessary, but not sufficient for good performance (e.g., it only validates that the warp indeed preserves the signed distance function).



Figure 3. A morph between a male and a female head from the MPI face database. The two heads are of significantly different size, adding to the difficulty of the correspondence problem. We computed a deformation field using our proposed algorithm and the signed distance and the iso-distance normals (cf. (4)) as feature functions. Note that the input to the algorithm are only the 3D models (roughly aligned); no landmark point correspondences were used. Following this, we took the original mesh describing the male head, and transformed its vertices linearly from the initial to the final position along the direction of the deformation field. The color was also linearly interpolated in RGB space between the start and the target object. We thus obtain a sequence of full 3D models, frontal views of which are shown.

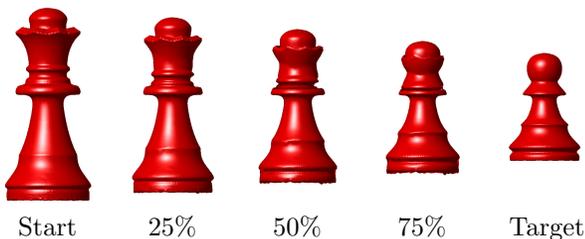


Figure 4. We used the same technique as in Figure 3 to generate a morph between a queen and a pawn.

We are already informally using this method to reduce the search space in our experiments. (2) If a sufficient number of *labelled* training data (i.e., pairs of landmark points on the two objects) are available, they could be used to evaluate the quality of the morphs or to tune parameters. For instance, in Figure 5 D, E, F, the mean Euclidean errors of the pairs of landmark points are 3.87%, 3.33%, and 0.01% of the object diameter, respectively (recall that in Figure 5 F, these points were used in the training of the warp, which explains the low error).

Another method for automatic parameter tuning could arise from a consistency test: if we compute warps in both directions and concatenate them, we should recover the original object.

6. Discussion

We have described a method for estimating a dense deformation field between objects represented by feature functions and, if available, pairs of corresponding surface points. Our algorithm is an SVM which is augmented by a novel penalty term for surface fitting, enforcing the warp to preserve location-dependent fea-

tures of the objects. In contrast to many other morphing methods, our warp is defined on the whole space in which the object lives, not just on the object's surface.

Establishing correspondence between surfaces has become relevant for statistical treatments of classes of objects (Blanz & Vetter, 1999), and we anticipate that the extrapolation in depth may open new fields of shape modeling in the future. For human faces, it could be used for warping additional structures, such as anatomical structures inside of the head, e.g., teeth or bone structures. Our technique could be applied not only for visual effects, but also in medicine or for scientific visualization and modeling of volume data.

On several 2D and 3D examples, we have shown rather encouraging morphing results. These were obtained mainly using implicit surface functions (estimating the signed distance) and their first derivatives (iso-distance normals) as feature functions entering the location-dependent penalty term. We conclude that the signed distance function contains information which is very useful for computing warps. Our conclusions are admittedly somewhat preliminary; other locational cost functions should be explored. From a graphics point of view, an attractive choice would be a cost function incorporating color and texture information; from a machine learning point of view, it would be attractive to have functions that can be computed using the kernel trick (cf. (7)).

Note that presently, all our morphs linearly apply the warp field. It would be interesting to give up this simplifying restriction, e.g., by incorporating model knowledge akin to what has been proposed by Saul and Jordan (1997), morphing along paths of high probability.

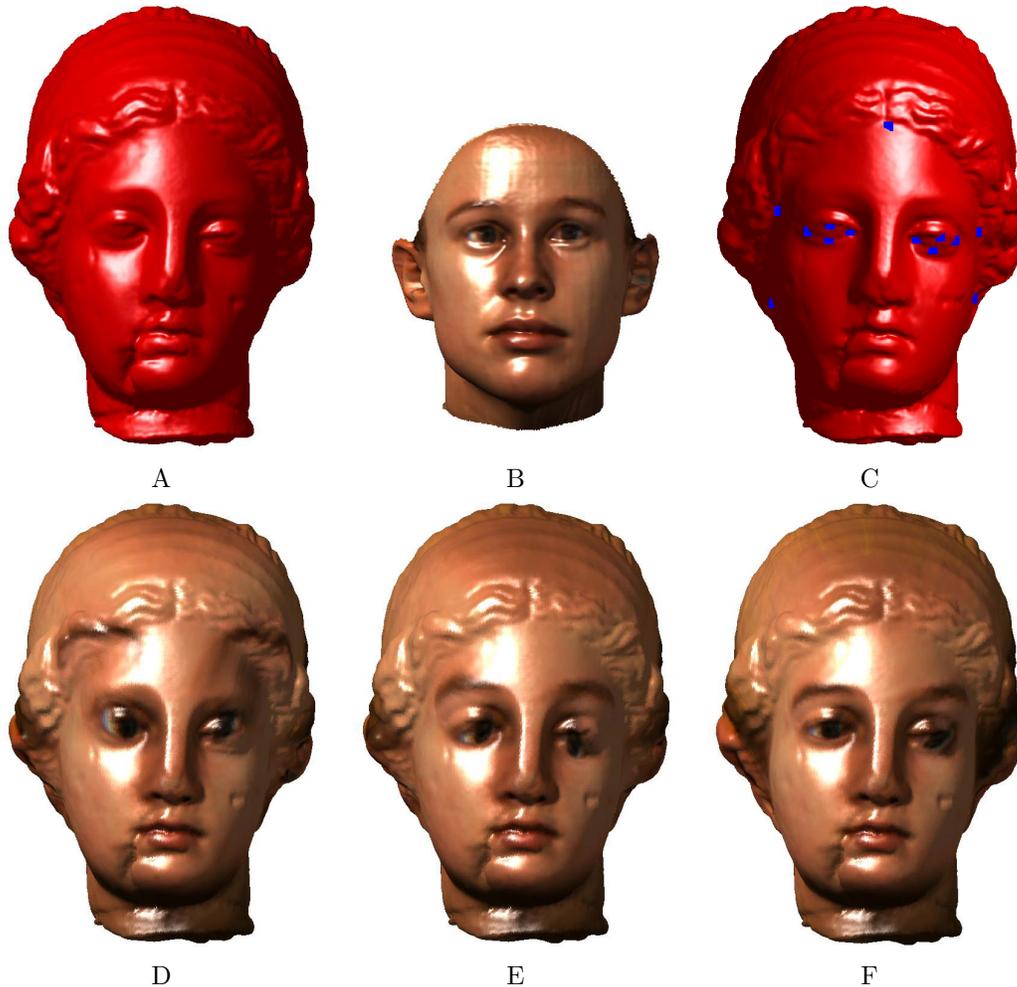


Figure 5. We constructed a correspondence field from the *Igea* artifact (A) to a female face from the MPI face database (B). We calculated the implicit surface functions and the deformation fields. Using these, the color of the face point closest to the transformed vertex of the *Igea* mesh was used to color the artifact (which originally comes without texture). In D, the deformation field was based only on the signed distance (3). Note that the eyebrows are mapped to the top of the forehead. Using additionally gradient information (cf. (4)), the correspondence becomes more accurate, leading to correct eyebrows (E). The best result (F) is obtained if we manually provide 13 pairs of landmark points (Section 3.2) on both objects (shown for the *Igea* in C), taken into account during training the deformation field.

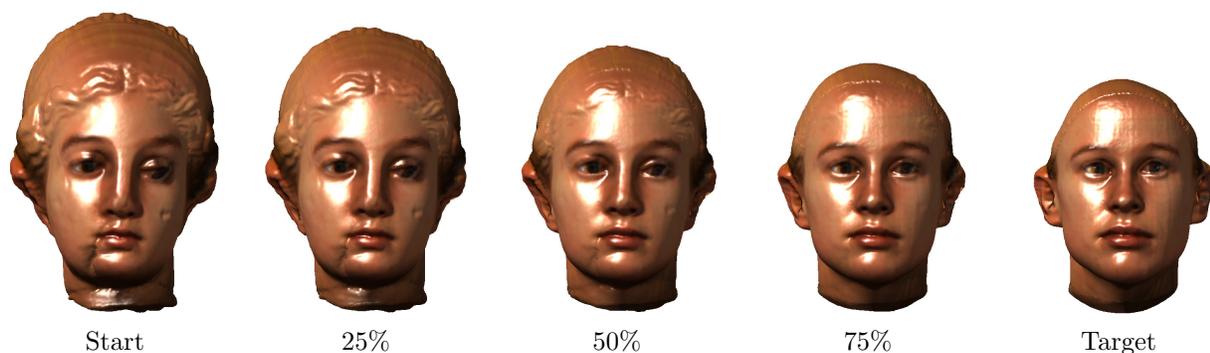


Figure 6. The deformation field underlying the texturing of Figure 5 F is used to morph the artifact into a female head. We again applied the warp to the vertices of object one, and linearly interpolated from the initial to the final position.

We believe that the present work illustrates the potential of machine learning methods for nontrivial shape processing tasks. The methods described in this paper are a promising starting point, which could be extended in a number of ways, including their application to higher dimensional problems such as the morphing of video sequences.

References

- Allen, B., Curless, B., & Popovic, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. *SIGGRAPH'03*.
- Audette, M. A., Ferrie, F. P., & Peters, T. M. (2000). An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4, 201–217.
- Bakir, G. H., Gretton, A., Franz, M., & Schölkopf, B. (2004). Sequential multivariate regression via stiefel manifold constraints. *Proc. DAGM*.
- Bartlett, P., & Schölkopf, B. (2001). *Some kernels for structured data* (Technical Report). Biowulf Technologies.
- Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Trans. PAMI*, 14, 239–256.
- Beymer, D., & Poggio, T. (1994). Learning and networks: Applications to graphics and very-low bandwidth video-conferencing. *GOMAC*. California.
- Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. *SIGGRAPH'99*.
- Bregler, C., & Omohundro, S. M. (1995). Nonlinear image interpolation using manifold learning. *Advances in Neural Information Processing Systems 7*.
- Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B., & Evans, T. (2001). Reconstruction and representation of 3D objects with radial basis functions. *SIGGRAPH'01* (pp. 67–76).
- Cohen-Or, D., Solomovic, A., & Levin, D. (1998). Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17, 116–141.
- Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II* (pp. 484–498). Springer-Verlag.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge, UK: Cambridge University Press.
- Freeman, W. T., Jones, T. R., & Pasztor, E. C. (2002). Example-based super-resolution. *IEEE Computer Graphics and Applications*.
- Grochow, K., Martin, S. L., Hertzmann, A., & Popovic, Z. (2004). Style-based inverse kinematics. *SIGGRAPH'04*. ACM Press.
- Haralick, R., & Shapiro, L. (1992). *Computer and robot vision*, vol. 2. Reading, MA: Addison-Wesley.
- Micchelli, C., & Pontil, M. (2005). On learning vector-valued functions. *Neural Computation*, 17, 177–204.
- Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M., & Alexa, M. (2004). Point based animation of elastic, plastic and melting objects. *SIGGRAPH/EUROGRAPHICS Symposium on Computer Animation*.
- Rueckert, D., & Frangi, A. F. (2003). Automatic construction of 3-d statistical deformation models of the brain using nonrigid registration. *IEEE Trans. on Medical Imaging*, 22, 1014–1025.
- Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proc. Conf. on 3D Digital Imaging and Modeling* (pp. 145–152).
- Saul, L., & Jordan, M. (1997). A variational principle for model-based morphing. *Advances in Neural Information Processing Systems 9*.
- Schaback, R. (1995). Creating surfaces from scattered data using radial basis functions. In M. D. al. (Ed.), *Mathematical methods for curves and surfaces*. Nashville: Vanderbilt University Press.
- Schölkopf, B., Giesen, J., & Spalinger, S. (2005). Kernel methods for implicit surface modeling. *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.
- Shelton, C. (2000). Morphable surface models. *Int. J. of Computer Vision*, 38, 75–91.
- Steinke, F., Schölkopf, B., & Blanz, V. (2005). Support vector machines for 3D shape processing. *Computer Graphics Forum (Proc. EUROGRAPHICS)*, 24.
- Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
- Vetter, T., & Poggio, T. (1997). Linear object classes and image synthesis from a single example image. *IEEE Trans. PAMI*, 19, 733–742.
- Wahba, G. (1990). *Spline models for observational data*, vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics.
- Walder, C. J., & Lovell, B. C. (2003). Kernel based algebraic curve fitting. *Proc. ICAPR'2003*.
- Weston, J., Chapelle, O., Elisseeff, A., Schölkopf, B., & Vapnik, V. (2003). Kernel dependency estimation. *Advances in Neural Information Processing Systems 15*.
- Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., & Müller, K.-R. (2000). Engineering SVM kernels that recognize translation initiation sites. *Bioinformatics*, 16, 799–807.