Layered Active Appearance Models*

Eagle Jones
eagle@cs.ucla.edu

Stefano Soatto
soatto@ucla.edu

Computer Science Department, University of California, Los Angeles, CA 90095

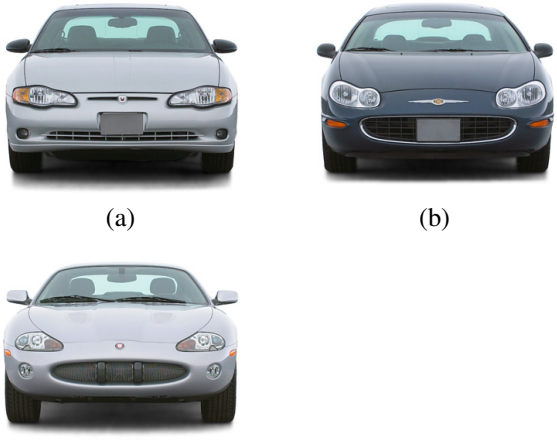
Abstract

Active appearance models (AAMs) provide a framework for modeling the joint shape and texture of an image. An AAM is a compact representation of both factors in a conditionally linear model. However, the standard AAM framework does not handle images which have missing features, or allow modification of certain structures in the image while leaving neighboring ones undeformed. We introduce the layered active appearance model (LAAM), which allows for missing features, occlusion, substantial spatial rearrangement of features, and which provides a more general representation that extends the applicability of the Active Appearance Model.

1. Introduction

Active appearance models [6] and related statistical models of shape and texture have received a great deal of attention in recent years. In the traditional AAM framework, the model is constructed from a collection of input images and corresponding landmark points (typically hand-labeled, although see [10, 4] and references therein for recent extensions that allow automatic registration) on each image. These landmark points correspond to important shared "features", such as the eyes, nose, and mouth in a model of faces. The mean position of these landmarks is found, and all input images are warped to bring corresponding landmarks into identical positions. The resulting "shape-free" images are then processed using principal component analysis (PCA). Since the relevant features of the shape-free images are in correspondence, the model produced in this manner is superior to one generated without the initial warping step. PCA is also applied to the shape parameters (the positions of the landmark points), and the results for both shape and texture are combined in a single model through a final PCA step.

*Research supported by AFOSR F49620-03-1-0095 and ONR N00014-03-1-0850.



(a) (b) (c)

Figure 1. *Examples of images which cannot be modeled by a standard AAM due to non-diffeomorphic warps or missing features. In (a) the license plate partially overlaps the grill; in (b) the license plate is completely inside the grill; in (c) it is completely absent.*

This framework yields good results when applied to images such as faces. In such settings, the warps involved are relatively small and do not move one feature into or across another. More importantly, all features are present in all images. We wish to emphasize the distinction between features and landmarks — although a particular landmark, or set of landmarks, may be absent or occluded [9], it is generally assumed that all objects in all images have the same parts. [1]

However, other classes of objects, such as cars, are not handled well by the standard active appearance model. Some instances of such objects may not have all possible features. It is not that they are occluded — they simply

[1] In this discussion, "features", "parts", and "layers" are used informally and interchangeably, and are distinct from the very specific meaning of "landmarks" [2] (two-dimensional points which define salient feature locations).

do not exist, and the surrounding region continues into the space occupied by that particular feature. A standard AAM has no way to model this. Others may have features with strong position changes, such as a license plate that is outside of, partially overlaps, or is completely inside a grill (fig. 1). Such position changes make it impossible to bring features into correspondence through a diffeomorphism, so the AAM fails. Even if all features are present, and a diffeomorphism is sufficient to model the movement of features, performance of the standard AAM tends to be very poor when landmarks undergo large position changes.

We propose the layered active appearance model (LAAM) to address these shortcomings. In a LAAM, each feature is considered a layer, which may or may not be present in a particular instance of the object. Each layer has a position, which is an additional parameter for the model, treated similarly to shape and texture, and the position of landmarks within a particular layer are relative to that layer's reference frame. Layers occlude each other, but are considered to be defined everywhere within their particular domain. The resulting model-building problem with missing data (due to both occlusion and missing features) is solved using an Expectation-Maximization (EM) [8] algorithm.

1.1. Previous Work

Many variations on the active appearance model have been developed [3]. The most relevant are those which were proposed to deal with occlusion, typically due to viewpoint change. In [5], Cootes et. al introduced a model composed of multiple distinct AAMs, where occluded landmarks due to pose change do not appear in some models, and the appropriate model is selected based on pose. A similar technique could be applied to our problem in the case where some features are absent, essentially defining separate subclasses of objects, which are described by a different model. The layered active appearance model provides a richer framework, in which all information is available in a single model, and objects do not have to be separated by subclass. Furthermore, there is a natural hierarchy in layer structures, which allows for the automatic inpainting [1] of lower regions when upper regions are removed.

More recently, Gross et. al proposed [9] a single-model solution to the problem of occlusion. Their approach to building the model using PCA with missing data is similar to ours, but they still assume that all objects do have the same features, some of which are just not visible in the images. The single model does incorporate all available data. In contrast to the multiple model approach mentioned above, there is no concept of object subclasses or variability in which features an object actually has, meaning the model does not describe objects where the features are not supposed to be there. Also, neither approach mentioned here addresses the problems of AAMs with large shape changes or violations of the diffeomorphic constraint.

2. Model Construction

We begin with a training set of images I, labeled with a given set of landmarks S on each object in the set. Although extensions to automatic registration and segmentation of regions are conceivable, these are well beyond the scope of this paper and will not be addressed. Instead, we assume that landmark points are divided into G groups, and each group determines (via convex closure or interpolation) a compact region of the image Ω, corresponding to various "features". We construct a set of layers, Φ, where each layer is associated with one feature, and in particular with the landmarks S_ϕ, the compact region of 2-D space Ω_ϕ which they identify, the intensity image or "texture" T_ϕ defined on the domain Ω_ϕ, and a local coordinate system with origin at X_ϕ.

The ordering of layers is fixed, so feature ϕ_1 is always in front of (and may occlude) feature ϕ_2 and so on. If features are missing from a particular example, or if landmarks, pixels, or entire features are occluded in an image, we record that information in weight vectors with entries for every component of S, X, and T. We denote the corresponding weights as W^S, W^X, and W^T, respectively. (In this work, we simply enter a 1 to indicate the presence of a quantity and 0 to indicate its absence. As the weighted PCA method we describe is valid for completely general weights, one can imagine more interesting weighting schemes, indicating uncertainty given missing or inaccurate labels, noisy images, transparency, or weights generated by an automatic outlier detection scheme [7].) Thus, each layer has a position, shape, texture, and weights. Since each layer occludes those behind it, we know we will have zero weights at least for the pixels in areas of a layer hidden by another. Figure 2 shows the layers for the car model developed in section 3.

2.1. Warping, Normalization, and Weighting

The first few steps of model construction follow the standard AAM formulation closely; the major difference is that most operations are done on each layer separately rather than for the entire labeled image. First, we find the mean shape for each layer:

$$\bar{S_\phi} = \frac{\sum_I W^S_{i,\phi} S_{i,\phi}}{\sum_I W^S_{i,\phi}}. \quad (1)$$

Then, for each image i and layer ϕ, we generate a warping for the domain $\Omega_{i,\phi} \rightarrow \tilde{\Omega}_{i,\phi}$ such that $\tilde{S}_{i,\phi} = \bar{S_\phi}$; that is, we warp each layer to the mean shape for that layer. T

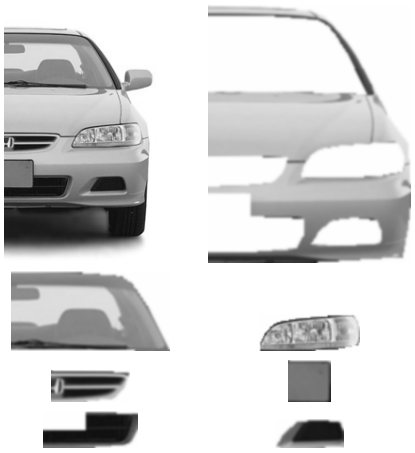


Figure 2. *The layers of an example from the car LAAM developed in sec. 3. For comparison, the original image is shown first. Note the missing data from the lower grill where it is overlapped by the license plate, and from the body of the car where all other parts occlude it.*

and W^T are defined on the domain Ω, so are both warped in this step. Since our warps are generally defined by only the boundary landmarks and not internal points, we have more freedom than in standard AAMs to choose the warping method. Although we employ a simple triangulation scheme here, techniques such as thin plate splines [2] would be much more useful in a LAAM than a standard AAM. At the same time, it remains possible within the LAAM to define additional, internal landmarks to anchor a particular "sub-feature", in which case triangulation may yield superior results.

After warping, the texture data for each layer is intensity-normalized to minimize the effects of lighting and color variation. We correct the median to the median of the layer amongst all images. Note that this is done on a layer-by-layer basis, not for entire images at a time. Thus, we are able to correct for different layers varying in intensity relative to each other in a single image. This is useful, for example, when we want to normalize dark and light cars, and both already have the same colored headlights.

Before we can build a combined model of shape and appearance, we need to define another set of weights which will compensate for the differing effects of shape, texture, and position on images generated by the model. Various approaches have been used for this step in the AAM literature; Cootes et. al. [6] originally proposed varying the shape parameters and measuring the RMS change of the generated images. This is laborious and is probably not as robust as we desire since it involves image derivatives. We have identified a simpler technique that works well by comparing the variances of each parameter type amongst the entire training set. Intuitively, weighting with the variances as follows approximately equalizes the dynamic range of the differing data types, giving them equitable influence on the result.

$$W^{TS} = \frac{var(TW^T)}{var(SW^S)}, W^{TX} = \frac{var(TW^T)}{var(XW^X)}. \quad (2)$$

One nice benefit of our weighted PCA approach is that we can combine theses weights with the weight vectors we already use rather than having to combine them into the data itself only to factor it out later. Finally, we form vectors A and W:

$$A_i = \begin{bmatrix} S_{i,\phi_1} \\ \vdots \\ S_{i,\phi_G} \\ \tilde{T}_{i,\phi_1} \\ \vdots \\ \tilde{T}_{i,\phi_G} \\ X_{i,\phi_1} \\ \vdots \\ X_{i,\phi_G} \end{bmatrix}, W_i = \begin{bmatrix} W^{TS}W^S_{i,\phi_1} \\ \vdots \\ W^{TS}W^S_{i,\phi_G} \\ \tilde{W}^T_{i,\phi_1} \\ \vdots \\ \tilde{W}^T_{i,\phi_G} \\ W^{TX}W^X_{i,\phi_1} \\ \vdots \\ W^{TX}W^X_{i,\phi_G} \end{bmatrix}, \quad (3)$$

where $\tilde{T}$ corresponds to the warped and normalized texture data and $\tilde{W}^T$ indicates the warped texture weights.

2.2. Weighted PCA

At this point, the typical active appearance model builder would apply principal component analysis to the combined shape and appearance vectors for all images in the training set. We need to do the same thing, but since we have no information for occluded areas of the images, we have to solve a PCA problem with missing data. We present an expectation-maximization [8] (E-M) based approach, following the lines of Roweis [11], but with inspiration from Skocaj and Leonardis [12]. (There are other approaches to PCA with missing data; any could be used in our framework.) We summarize the process here; the reader is referred to the original papers for a detailed derivation. For the discussion that follows, we make the usual assumption for PCA that the (weighted) mean has been removed from our vectors A.

We begin by recalling that PCA finds bases U and coefficients C which minimize the reconstruction error of the training set A, in the least-squares sense:

$$\epsilon = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(a_{ij} - \sum_{l=1}^{k} u_{il}c_{lj}\right)^2, \quad (4)$$

where there are M components to each vector, N vectors, and we seek a PCA result with k basis vectors.

Instead, we would like to minimize

$$\epsilon = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\left(a_{ij} - \sum_{l=1}^{k} u_{il}c_{lj}\right)^2, \quad (5)$$

therefore weighting the contribution of each element of each vector differently in the total error. Elements with a zero weight should have no effect on our minimization.

We can solve a missing data problem of this type using an E-M algorithm:

- Initialize U and C with estimates (from standard PCA).
- Repeat until convergence:
  - Expectation Step: Given the vectors A and the bases U, find the coefficients C which will minimize eq. (5).
  - Maximization Step: Given the vectors A and the coefficients C, find the bases U which will minimize eq. (5).

For the E step, let us temporarily ignore the weights, and we can easily see how to find the coefficients given the vectors and the bases. For each vector A_j, we have an overdetermined linear system, with one equation per vector element:

$$a_{ij} = \sum_{l=1}^{k} u_{il}c_{lj}, i = 1 \ldots M. \quad (6)$$

We solve for the coefficients corresponding to each vector one at a time, in a least squares sense:

$$C_j = U^{\dagger}A_j, \quad (7)$$

where $U^{\dagger}$ denotes the pseudo-inverse.

This is an inefficient way to perform unweighted PCA, but provides a starting point from which to apply the weights for our problem. We simply scale both sides of each equation from (6) by the square root of the corresponding weight:

$$\sqrt{w_{ij}}a_{ij} = \sqrt{w_{ij}}\sum_{l=1}^{k} u_{il}c_{lj}, i = 1 \ldots M. \quad (8)$$

At first glance, the reader may conclude that this step does nothing, but when a different scaling is applied to each equation, the error contributions in eq. (5) are scaled as well. For example, a constraint with weight zero will turn into the equation $0 = 0$ and have no influence on our minimization, just as we hoped. A large weight will cause the error contribution to be multiplied correspondingly. The solution to eq. (8) is:

$$C_j = (\mathrm{diag}(W_j)U)^{\dagger}W_jA_j, \quad (9)$$

where $\mathrm{diag}(W_j)$ denotes a matrix with the weights for vector j on the diagonal and zero elsewhere.

The reasoning for the M step follows a nearly identical process as we find the bases given the coefficients. We wish to emphasize a particularly subtle difference: instead of solving a system of equations for each vector, where each equation corresponds to one element, we now have a system of equations for each vector element, where each equation corresponds to one vector.:

$$\sqrt{w_{ij}}a_{ij} = \sqrt{w_{ij}}\sum_{l=1}^{k} u_{il}a_{lj}, j = 1 \ldots N. \quad (10)$$

The equations are identical to those in (8), but they are grouped into different systems due to varying the other subscript. The solution is:

$$U_i = (\mathrm{diag}(W_i)C)^{\dagger}W_iA_i. \quad (11)$$

3. Experimental Results

We generated a layered active appearance model for a collection of 128 frontal images of passenger cars. Figure 1 shows three typical images from the dataset. Images were transformed to normalize for size and rotation, color data was discarded, and we assumed symmetry, so used only the data for the right half of each image.

We can use the LAAM to reconstruct (or compress) the original dataset, or to generate novel instances similar to the objects in our training set (fig. 3). The four samples shown are images of cars which do not appear in the training set (or the real world, probably for good reasons). Any sample can be generated with or without particular features. The texture of these examples may appear somewhat homogenized, especially when compared to real cars or our experiments showing reconstruction of the training set (fig. 5). This is due to the fact that we do not explicitly model the non-Lambertian nature of cars, which are *de-facto* mirrors. The interior, visible through the front windshield, appears sharp, since we have reduced the dimensionality of our model in that particular layer to only a single basis vector. Varying the dimensionality per layer in this fashion is possible with the layered active appearance model, but cannot be accomplished with traditional AAM, another advantage of the proposed technique.

While we would expect reconstruction and extrapolation from our model or a standard AAM, it is not technically possible to compare our model's performance in these operations to that of the standard AAM on this data set. The AAM is unable to cope with the non-diffeomorphic warps involved. To illustrate directly the shortcomings of the AAM in this type of problem, we have "forced" a standard AAM to model the same dataset, by removing certain

Figure 3. *Novel images generated by randomly sampling from the layered active appearance model. Any sample can be generated with or without particular features; in the first case we omit the fog lights. We do not explicitly model the non-Lambertian nature of cars, so the texture of these samples may appear smoothed.*

(a) (b)

Figure 4. *Attempts at reconstruction and extrapolation of the car dataset using a standard active appearance model. (a) Reconstruction exhibits blurring, tearing, and ghosting. (b) Extrapolation generates impossible geometry (or radically novel designs).*

landmarks and relaxing geometric constraints. As expected, the reconstruction of an image from the training dataset (fig. 4) is poor. It exhibits blurring and warping around the license plate, where the dataset is geometrically unstable,"tearing" between the headlights and grill, where the adjacent polygons were pulled apart and unable to be filled in, and "ghosting" in the interior, where the same region in the LAAM was corrected by adjusting the dimensionality of that layer. Attempts at extrapolation are even worse, frequently generating impossible geometry.

In addition to its descriptive power, the layered active appearance model makes certain useful operations possible. In figure 5, we can see the effect of adding a feature in the synthesis phase, when it was not present in the original image. The model extrapolates the appearance, position, and shape of the missing feature and fills it in. The opposite operation, removing a feature and filling in the occluded area, is seen in figure 6. It is also possible to move features without disturbing the local structure of the image (fig. 7).

4. Summary and Conclusions

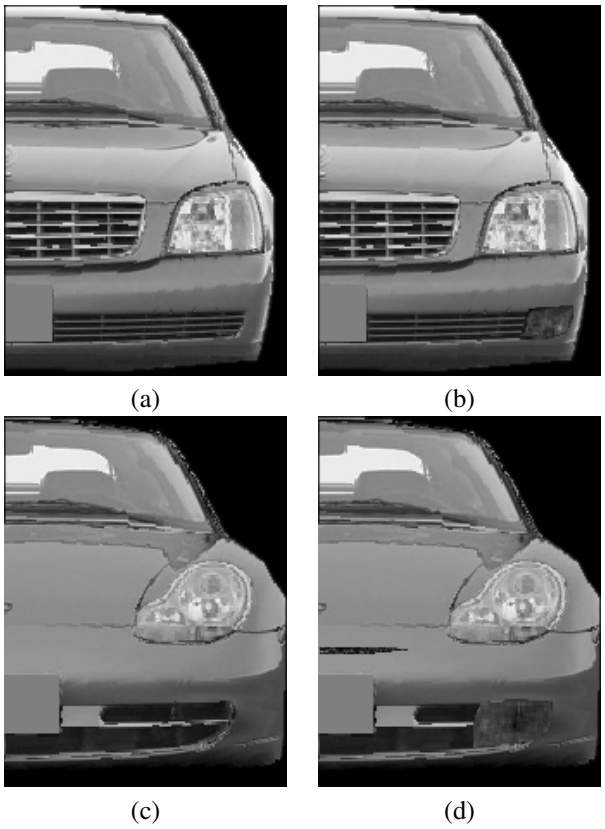
We have presented the layered active appearance model, a generalization of active appearance models which allows for missing features, occlusion, and substantial spatial rearrangement of features. Our model is well-suited to many modeling tasks which are beyond the capabilities of the traditional AAM.

An E-M algorithm has been derived to generate the proposed model from training data, and an example of the resulting model has been shown, along with several images illustrating the capabilities of our approach. While AAMs are often tested with faces, which exhibit diffeomorphic warping, relatively little inter-subject variability, and mostly Lambertian reflection, we have selected a much more challenging dataset, and shown that geometric transformations, including the appearance and removal of parts and motion of one layer on top of others, are correctly captured by our model. Thus, testing on datasets such as faces, where AAMs already work well, is uninteresting; our model naturally overfits in such cases.

As with any robust modeling framework, there are rich possibilities for future work, such as automatic identification of features and landmarks, application to pose change in three-dimensional scenes, and the use of the weights to reflect transparency, noise, or uncertainty.

Acknowledgements

Our thanks to Kerry Connor, who spent many hours labeling hundreds of cars.



(a) (b) (c) (d)

Figure 5. *Adding features to an image. The original car in (a) did not have fog lights. Our model extrapolates their appearance, position, and shape and generates (b). For the sports car in (c), our model extrapolates (d) a small air intake and oversized fog lights matching the shape of the grill.*

Figure 6. *Removing a feature from an image. There is no explicit inpainting in our approach, so the region appears as a "patch". However, the general appearance is correct.*

Figure 7. *Moving a feature without disturbing the local geometry of the image. The license plate is moved down, while the headlights are moved down and toward the inside of the car.*

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 417–424. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[2] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[3] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *Proc. British Machine Vision Conference*, 2002.

[4] T. Cootes, S. Marsland, C. Twining, K. Smith, and C. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Proc. ECCV*, pages 316–327, May 2004.

[5] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *Proc. Int. Conf. on Face and Gesture Recognition*, 2000.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. of the Eur. Conf. on Comp. Vis.*, pages 484–496, 1998.

[7] F. de la Torre and M. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54:183–209, Aug.-Oct. 2003.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. B*, 39:185–197, 1977.

[9] R. Gross, I. Matthews, and S. Baker. Constructing and fitting active appearance models with occlusion. In *Proc. IEEE Workshop on Face Processing*, 2004.

[10] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Computer Vision*, 60(2):135–164, 2004.

[11] S. Roweis. Em algorithms for pca and spca. In *Advances in Neural Information Processing Systems*, 1998.

[12] D. Skocaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *Proc. 9^{th} Int. Conf. on Computer Vision*, 2003.