Towards Adaptive Models for Classification of Technical Objects

Astrid Laubenheimer and Norbert Link*

Fraunhofer Institut für Informations- und Datenverarbeitung (IITB)
Fraunhoferstrasse 1, 76131 Karlsruhe, Germany
Email: lau@iitb.fraunhofer.de

Abstract

The recognition of the specific type of a technical object under consideration is often crucial in understanding its capabilities and its potential behaviour. Reliable automatic object type recognition from digital imagery is therefore demanded in many applications like reconnaissance. In this paper the problem of simultaneous estimation of object pose and type-characteristic shape parameters from images is attacked by the use of CAD models. In order to cope with the usually incomplete and even erroneous model data set (base models), we build from these a flexible adaptive model which forms an orthogonal basis. For this purpose a metamodel is constructed by volume-based metamorphosis from the base models, which are then inserted by mesh optimization methods to give the final flexible model. The model and pose parameter estimation is based on the correspondence of image and model edge elements to avoid illumination influence. An objective function for robust estimation is used and optimized with respect to the parameters by a Levenberg-Marquard method. Preliminary results with civil aircraft show the applicability of the methods presented.

1 Introduction

In safety-critical aerial surveillance, object detection and classification systems must have robust performance with very high reliability. While predictable automatic screening systems for technical objects are available, automatic classification of detected objects still has insufficient reliability. As a matter of fact, trained aerial image analysts are experts and their abilities are unlikely to be achieved by computer-vision systems in the near future. On the other hand, in a technical context, human classification skills concentrate on discrete features, while continuous or slightly differing features, such as aspect ratios or geometric deformations, are hardly distinguished by human perception. Hence, fine classification of technical objects resp. verification of hypotheses demands to be supported by artificial intelligence.

Earlier work [1] has shown, that desired robustness of classification, can rarely be achieved without incorporating three-dimensional a-priori knowledge about the observed objects.

On the other hand, in classification tasks, the observed object is unknown. One could forego this problem, by applying pose estimation to all models which are similar to hypotheses and then deciding for the model with the highest confidence value. In practice, this strategy often fails because providing models with the required geometric precision and variety is impossible.

Thus, our approach is based on the use of data-driven and automatically built adaptive linear models, where each model is elastic and therefore can represent a number of similar objects, instead of providing a single model for each object. The shape and pose parameters are then estimated simultaneously by maximization of a robust objective function, which is describing the correspondence between model and image features. Since shape parameters are orthogonal projection coefficients on the model basis, they are coding the observed object uniquely and therefore can serve as features for classification.

Blanz und Vetter formerly proofed [2, 3], that adaptive linear models can be powerful tools for reconstruction and identification of human faces and thus motivated our work. One of the reasons, why

*Fachhochschule Karlsruhe - Hochschule für Technik, Moltkestrasse 30, 76133 Karlsruhe, Germany;
email: norbert.link@fh-karlsruhe.de.

their approach cannot be applied straight forward to our problem, is the fact, that their model-building step requires objects to be star shaped around an axis or to be at least projectable to a plane. Many object classes, that are of interest in the context of aerial surveillance, e.g. commercial and military aircraft, do not have this property. Shelton [4] redesigned the model-building step, leading to more general object classes. As this approach is based on calculating point correspondences by energy minimization, where the energy term is based on distances between points and surfaces, this will result in collapsing thin object parts, auch as aircraft wings.

There are many alternative approaches on metamorphosis, which is the key point to be solved in order to automize modelbuilding. In [5] a characterization of different metamorphosis techniques can be found. These include aspects of the two main methods, namely boundary and volume based approaches.

Our model construction process uses warping and combines volume and boundary based metamorphosis techniques, which allows modelling of sharp features as well as thin object parts. No assumption on projective properties of the objects are made. As mentioned before, our adaptive model is data-driven, which means, it is generated from only a small set of *commercial off the shelf* models, called the base models. Thus, we do not depend on a complete or precise data base, but use some representatives instead. In order to yield an orthogonal model, we use principal component analysis.

By now, we restrict our object classes to commercial airplanes, but nevertheless, the approach can be applied for many other types of technical objects, as long as the boundaries of the objects are topologically equivalent to a sphere and as the geometry between base objects does not differ in sence of discrete features.

2 Adaptive Models

In this section, we give a brief description of our model construction process, starting out with a definition for our linear adaptive models. In the following, we denote our base models with $\mathcal{M}_0, \ldots, \mathcal{M}_N$, $N > 0$, and all of them are assumed to be given in a boundary representation.

2.1 Definitions

In order to define our elastic model, we need to distinguish topology T and geometric realization $\mathbf{g}$ of a three-dimensional model $\mathcal{M}$. While the topology describes the graph, that yields the information, how the vertices are connected to edges and triangles, the geometric realization is an indexed set of three-dimensional vectors, yielding the corresponding geometric position of each vertex.[1] We follow [2] by writing the geometric realization in a single shape-vector $\mathbf{g} = (x_0, y_0, z_0, x_1, y_1, z_1, \ldots)^T$, where indices coincide with vertex indices. Clearly, the representation of a model $\mathcal{M}$ by $(T, \mathbf{g})$ is not unique, since many topologies can describe the same surface.

An adaptive model, built from base models $\mathcal{M}_0, \ldots, \mathcal{M}_N$, is defined by

$$(\mathcal{T}, \sum_{n=0}^{N} \lambda_n \mathbf{g}_n), \lambda_n \in I\!R \qquad (1)$$

where $\mathcal{T}$ is a topology, called *meta-topology*, and $\mathbf{g}_n$ is a geometric realization, such that the tuple $(\mathcal{T}, \mathbf{g}_n)$ represents the base model $\mathcal{M}_n$, $n = 0, \ldots, N$. This generic model describes infinitely many geometric realizations, coded by few parameters $\lambda_0, \ldots, \lambda_N$. In figure 1 an adaptive model for three base models is shown.

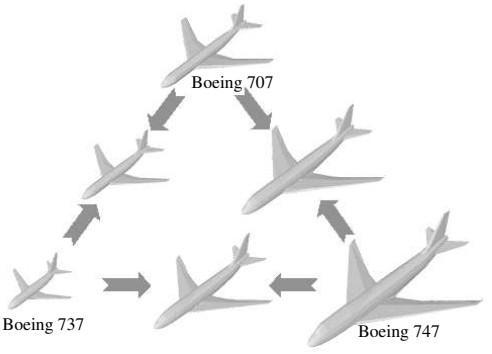

Figure 1: Realizations of an adaptive model, that was built from three base models. The corners of the triangle show representations of the base models and the models inbetween are intermediate shape transformations.

[1] In [6] one can find a mathematical definition of topology and geometric realization, based on concepts of algebraic topology.

Applying principal component analysis (PCA) on $(\mathbf{g}_0, \ldots, \mathbf{g}_N)$, delivers N orthogonal eigenvectors $\mathbf{e}_n$ and a mean vector $\bar{\mathbf{g}}$. With these components, the notation for our orthogonal version of the adaptive model is given by

$$\mathcal{M}_{\delta_1,\ldots,\delta_N} = (\mathcal{T}, \bar{\mathbf{g}} + \sum_{n=1}^{N} \delta_n \mathbf{e}_n), \delta_n \in I\!R,$$

depending on parameter vector $\tilde{\theta} = (\delta_1, \ldots, \delta_N)^T$. In figure 2 the variation along the first eigenvector of an adaptive model is shown. As clearly can be seen, the first eigenvector changes the size of the realizations.

$\delta_1 = -\sigma_1$ $\quad$ $\delta_1 = 0$ $\quad$ $\delta_1 = \sigma_1$

Figure 2: Variation of the first eigenvector of an adaptive model. The objects are realized by setting $\delta_n = 0$, $n = 2, \ldots, N$. σ_1 denotes the standard deviation corresponding to the first eigenvalue and is obtained from the PCA.

2.2 Construction

Now, that the definition of an adaptive model is given, we describe the way of its construction. This requires - according to (1) - the construction of a meta-topology $\mathcal{T}$ and calculation of N geometric realizations $\mathbf{g}_n$. The construction of these components is done in two steps, namely (i) constructing a *meta-model* $(\mathcal{T}, \tilde{\mathbf{g}})$, which defines the meta-topology and an initial geometric realization and (ii) adding the base models by calculating the geometric realizations $\mathbf{g}_n$, $n = 0, \ldots, N$.

For the overall construction process, we make use of the fact, that for a static model $\mathcal{M}$, whose boundary is topologically equivalent to the sphere, there always exists an implicit function, whose iso-surface coincides with its boundary. For constructing such an implicit function, many approaches exists. We directly use the signed distance transformation, since the computationally more attractive approximations, e.g. by radials sums [7], turned out not to be appropriate to model sharp features and thin object parts.

For the inverse process, which is the polygonization of a given implicit function, we use a marching cube implementation [9].

2.2.1 Creating the Meta-Model

Construction of the meta-model is based on typical volume based metamorphosis, which would work as follows: Identifing every base model $\mathcal{M}_n$ with an implicit function f_n, the mean-model could be calculated by applying a polygonizer to

$$\frac{1}{N+1} \sum_{n=0}^{N} f_n. \tag{2}$$

For our object types, this would lead to degenerations of all those object parts, that do not overlap for most of the base models. Thus, a preprocessing warp similar to [10] is implemented, in order to transform all our base models into roughly overlapping positions. The warp needs a small number (e.g. 16) of point correspondences $\{\mathbf{x}_1^n, \ldots, \mathbf{x}_L^n\}$, so called anchor points, which are defined by the user. Actually, this is the only user interaction for the whole model-building step.

From these anchor points, the set of centroids of corresponding points

$$\left\{ \frac{1}{N+1} \sum_{n=0}^{N} \mathbf{x}_1^n, \ldots, \frac{1}{N+1} \sum_{n=0}^{N} \mathbf{x}_L^n \right\} \tag{3}$$

is calculated. Now, for every model $\mathcal{M}_n$ a warping transformation w_n can be applied on $\mathcal{M}_n$, that maps base model specific anchor points $\{\mathbf{x}_1^n, \ldots, \mathbf{x}_L^n\}$ onto points definied in (3). Since this warping process has to be redone later on and the warps are not analytically invertable, we further calculate functions $\hat{w}_n$, with source and destination points being exchanged.

The base models warped by w_n are denoted by $\mathcal{M}_n^w$ and corresponding implicit functions by f_n^w. Substitution of the functions f_n in (2) by f_n^w might still lead to great loss of volume, when a marching cube is applied to the new mean function. This is due to the fact, that marching cube algorithms do not necessarily handle model parts, that lie within the cell raster of the marching cube. Instead of decreasing cell size, which would increase the number of outcoming triangles drastically, we add an offset c, which is exactly half of the size of the marching cube, to our mean function. Adding this offset can

be interpreted as morphological opening, which has another positive effect, described in the next section.

With these definitions, we are now able to define the meta-model $(\mathcal{T}, \tilde{\mathbf{g}})$: A marching cube is applied to our modified mean function

$$f = c + \frac{1}{N+1} \sum_{n=0}^{N} f_n^w \tag{4}$$

and in a postprocessing step, the triangle size is adapted to the local curvature. Therefore, we apply a polygon reduction method [11] on the output of the marching cube and then do local 1:4 subdivisions followed by triangle regularization as proposed in [12]. We use

$$\max_{\mathbf{c} \in \mathcal{C}(T)} \mathbf{n}^T \frac{\bigtriangledown f(\mathbf{c})}{\| \bigtriangledown f(\mathbf{c}) \|} > \cos(\pi/4) \tag{5}$$

as subdivision criterium for a triangle T, where $\mathbf{n}$ is the normal vector of T and $\mathcal{C}(T)$ denotes the set of centroids of the four subtriangles, when T is subdivided 1:4. For our aircraft objects, the resulting meta-model has less than 10 000 triangles.

2.2.2 Adding the Base Models

In this section, we describe how a geometric realization $\mathbf{g}_n$ is computed, such that $(\mathcal{T}, \mathbf{g}_n)$ is a representation of base model $\mathcal{M}_n$.

The geometric realization $\mathbf{g}_n$ is calculated by smoothly moving the initial positions $\tilde{\mathbf{g}}$ towards the surface of base model $\mathcal{M}_n$. Therefore, we extended a technique developed for mesh optimization, published by Ohtake and Belyaev [13] described below. The input of their algorithm is a model, which is thought to be the output of a polygonizer, and a corresponding implicit function. By means of a dual net, a curvature-weighted vertex resampling and an optimization of vertex position are investigated. Optimization of vertex position is done by calculating intersection points of tangential planes. As a consequence, their algorithm delivers an outer approximation of the isosurface (triangles are tangential planes of the isosurface), which is the reason, why good approximizations of sharp features can be obtained.

Since their opimization does not assume the starting vertices to lie on the isosurface, we can use their optimization, for tracking vertices over isosurfaces. The latter correspond to a finite number K of linear interpolants of two implicit functions, where one of them is representing a given model and the other one is defining the destination shape. Unlike [13], we do not apply subdivision during opimization, such that the meta-topology $\mathcal{T}$ is not changed. The overall metamorphosis algorithm is stated in figure 3.

Input:	
K	number of interpolation steps
$\mathcal{M}_s$	source model $(\mathcal{T}, \mathbf{g}_s)$
f_d	destination function
Output:	
$\mathcal{M}_d$	destination model: approximation of isosurface of f_d with topology $\mathcal{T}$.
Algorithm:	
(1)	Provide a function f_s, which corresponds to input model $\mathcal{M}_s$.
(2)	Set $\mathcal{M}^0 := \mathcal{M}_s$.
(3)	For k = 1 to K do: Apply mesh optimization on model $\mathcal{M}^{k-1}$ and function $(1 - \frac{k}{K}) f_s + \frac{k}{K} f_d.$ Denote the resulting geometric realization with $\mathbf{g}^k$ and set $\mathcal{M}^k := (\mathcal{T}, \mathbf{g}^k)$.
(4)	Set $\mathcal{M}_d := \mathcal{M}^K$.

Figure 3: Metamorphosis algorithm.

With these preparations, three steps can be formulated, that have to be done in order to calculate geometry $\mathbf{g}_n$ (see also figure 4).

1. Apply metamorphosis with K interpolations on the meta-model $(\mathcal{T}, \tilde{\mathbf{g}})$ and destination function $c + f_n^w$.
2. Undo base model specific warp, by applying prior calculated transformation $\hat{w}_n$.
3. Apply metamorphosis on the resulting model and destination function f_n, in order to cancel the effect of offset c in (4). Denote the resulting geometry with $\mathbf{g}_n$.

We found out, that it is important to substract offset c not prior than the last iteration. Otherwise, vertex resampling will collapse thin object parts.

2.3 Results

As stated in figure 7, our method delivers good reproductions of base models: The base model features are caught completely and vertex positions have function values less that 10^{-10} (meters). Re-

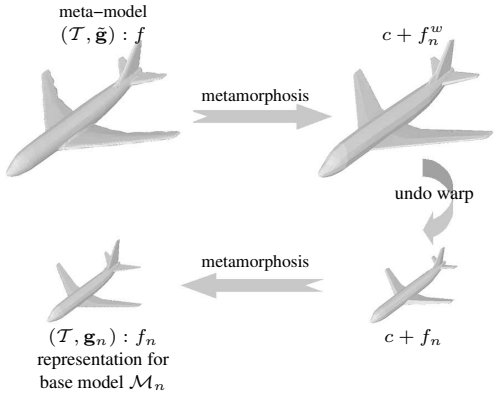

Figure 4: Adding a base model is done by applying metamorphosis twice and undoing warp. Images are labeled with the corresponding implicit functions. The resulting model $(\mathcal{T}, \mathbf{g}_n)$ is a representation for the base model $\mathcal{M}_n$ in the meta-topology.

sulting errors, as shown in figure 7 (f), are due to undersampling, which could have been avoided by decreasing the subdivision thresh in (5). On the other hand, as explained in the next section, we use an objective function, that is robust concerning corrupted model features. For our purpose, it is more important, that the model segments must not be too short in order to be realiable. Hence, we reduce the number of triangles of the adaptive models to approx. $1\,500$. Therefore, we extended the strategy proposed in [11] to multiple geometric realizations.

3 Pose and Shape Estimation

Registration of any three-dimensional model against an image requires three substantial steps, namely (i) extraction of image as well as model features, (ii) defining a cost or objective function and (iii) a minimization resp. maximization algorithm. In this section, these three components are described, where no application specific assumptions are made so far. Thresholding and other tuning parameters will be described in section 4.

3.1 Feature Extraction

Since our model does not include color or intensity values and neither of them can be assumed to be constant over the complete observed object, we built our algorithm on edge segment extraction, which formerly was shown to be a successful strategy in other registration tasks (e.g. [14]).

For image features, we decided to make use of an efficient implementation, that was developed in the context of [14], where edge pixel detection is based on a gradient filter described in [15]. Our implementation is tuned by a single threshold, which is the minimum absolute value of gradient μ, that defines a pixel to be an edge pixel. The output of the feature extraction is a set of image edge segments (described by the endpoint coordinates), denoted by $\mathcal{H}$.

While image segment extraction is a question of image processing, model segments depend on a projection chain, which represents the mathematical model of physical camera properties. The projection chain is given by a transformation $\mathcal{A}$ from world to camera coordinates and a perspective or parallel projection $\mathcal{P}$ onto the camera plane. While we assume the projection $\mathcal{P}$ to be known, the transformation $\mathcal{A}$ has to be estimated in order to calculate the object pose. The latter depends on three rotation angles ω, φ and κ around the axis x, y and z as well as translation vector $(t_x, t_y, t_z)^T$.

Using the projection chain and the actual geometric realization of the adaptive model as input, a hidden line algorithm delivers model segments in image coordinates. For clarity, model segments are visible intervals of model edges and only edges sharper than a fixed thresh $\tau = \pi/2$ are projected. In some cases, we do not consider inner edges and use silhouette only.

The resulting set of model segments is denoted by $\mathcal{L}$. Since model-fitting includes shape and pose estimation, the set of all model segments depends on the extended parameter vector $\theta = (\delta_1, \ldots, \delta_N, t_x, t_y, t_z, \omega, \varphi, \kappa)^T$. Hence, we denote the set of all model segments by $\mathcal{L}_\theta$.

3.2 Objective Function

In order to cope with many-to-many edge correspondences, the objective function is definied in the same manner as formerly done by Krüger [14] for the purpose of robust map-to-image registration. In the following, his objective function is shortly reviewed.

Given the projection $l_i = (\mathbf{q}_{i1}, \mathbf{q}_{i2})$ of a model segment with endpoints $\mathbf{q}_{i1}$ and $\mathbf{q}_{i2}$ and an image segment $h_j = (\mathbf{p}_{j1}, \mathbf{p}_{j2})$, we define $\mathbf{n}_j$ to be a unit normal vector of image segment h_j and

(a) (b)

Figure 5: Extracted features for the model-fitting step: (a) image edge segments with minimum absolut value of gradient $\mu = 3$ and (b) model segments.

$\mathbf{e}_j = (\mathbf{p}_{j2} - \mathbf{p}_{j1})/\|\mathbf{p}_{j2} - \mathbf{p}_{j1}\|$ the normalized direction vector. The distance of the two segments is defined as a function of overlap o_{ij} and the orthogonal distances d_{ij}, where

$$\begin{aligned} o_{ij} &:= |\,[\,0, \|\mathbf{p}_{j2} - \mathbf{p}_{j1}\|\,] \\ &\quad \cap\,[\,\mathbf{e}_j^T(\mathbf{q}_{i1} - \mathbf{p}_{j1}), \mathbf{e}_j^T(\mathbf{q}_{i2} - \mathbf{p}_{j1})]\,|, \\ d_{ij}^2 &:= d_{ij1}^2 + d_{ij2}^2 \\ &= \|\mathbf{n}_j^T(\mathbf{p}_{j1} - \mathbf{q}_{i1})\|^2 + \|\mathbf{n}_j^T(\mathbf{p}_{j2} - \mathbf{q}_{i2})\|^2. \end{aligned}$$

Based on Tukey's biweight ρ [16], which is the loss function of a hard redescending M-estimator, Krüger derives the function

$$\bar{\rho}(d) = \begin{cases} (1-d^2)^3 & |d| \le 1 \\ 0 & |d| > 1 \end{cases}$$

via $\bar{\rho}(d) = 1 - 6\rho(d)$. For a known scale parameter σ the goodness of fit is then measured by the product $o_{ij}\bar{\rho}(d_{ij}/\sigma)$.[2] The objective function can now be formulated in dependence of parameter vector θ as sum of goodness over all possible correspondences:

$$\chi^2(\sigma, \theta) = \sum_{(l_i, h_j) \in \mathcal{L}_\theta \times \mathcal{H}} o_{ij}(\theta)\bar{\rho}\left(\frac{d_{ij}(\theta)}{\sigma}\right). \qquad (6)$$

As mentioned in [14], maximizing this function solves feature matching and parameter estimation simultaneously and makes no hard distinction between correct and incorrect correspondences. Furthermore, we strongly take advantage of its robustness concerning both types of segments, model segments as well as image segments.

[2] Obviously, the goodness of fit is not symmetric. On the other hand, only $\mathbf{q}_{i1}$ and $\mathbf{q}_{i2}$ depend on the parameter vector θ, which leads to nice analytical derivatives that can be evaluated efficiently.

Actually, the usage of $\bar{\rho}$ can be interpreted as a model-to-image segment-distance low-pass filter applied to all image segments. The cut-off and the steepness of the filter is controlled by scale factor σ.

3.3 Robust Estimation

In [14], it is shown, that under the assumption of constant overlap, maximizing (6) is equivalent to solving a re-weighted least-squares problem. Properties of collinear projection are used and further assumtions are made, to finally arrive at a linear least-squares problem. Since our projection chain is far away from linearity and the starting values of parameters can be far away from the optimum, we cannot make these assumptions. Instead, we apply Levenberg-Marquard [17] directly on (6) and thus solve the maximization task in a non-linear manner.

In order to apply Levenberg-Marquard, the objective function has to be turned into a sum of squares. Thus, we take roots over summands of (6) and resquare them, which is possible, because summands are nonnegative. Our objective function is continuous and analytical derivatives exist, since overlap is assumed to be constant.

4 Application

In this section, we concentrate on our application, which is - as mentioned in the introduction - an aircraft-type classifier for aerial surveillance tasks. While the model-building step described in section 2 could be realized in a global manner, parameter estimation cannot be realized without incorporating application specific assumptions. This is due to the fact, that model-fitting is an ill-posed problem, e.g. the scale of the objects (determined by parameter t_z) is highly correlated with the model parameters, as can be seen in figure 2. Hence, we assume the size of the aircraft to be roughly known (e.g. small, medium or large) in order to initialize the shape parameters. Furthermore, parameter estimation has to be done in an application specific manner, which is described in the following section.

4.1 Application Specific Estimation

Due to the hard redescending property of Tukey's biweight, our objective function is not convex, and thus has local maxima. As can be seen in figure 6,

this effect is enhanced, when the scaling factor σ is small. On the other hand, an optimal estimation can not be obtained with large scales, because too many outliers influence the objective function. Hence, we realize every estimation in a coarse-to-fine strategy by repeated application of the Levenberg-Marquard optimization, starting with a large scale, which is successiveley decreased until the desired precision is obtained. Clearly, the scales have to be choosen in dependence of objects size in the image plane, which is up to 500 pixels in our case. Therefore, we work with five scales $\sigma = \sqrt{2}\sigma'$, where $\sigma' \in \{2, 5, 10, 50, 100\}$.

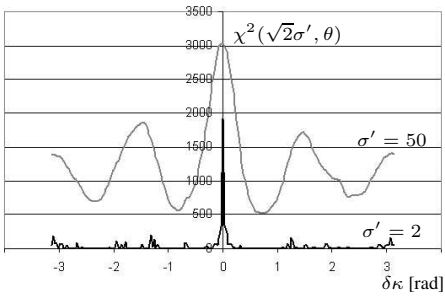

Figure 6: An objective function for two different scales (high scale is denoted in a gray contour and the low scale in black). In order to improve visualization, the lower function is scaled by factor 5. The functions correspond to figure 5 (b), where all parameters are fixed as shown there, except for the rotation angle $\kappa = \kappa_0 + \delta\kappa$. κ_0 is the rotation angle corresponding to the position shown in figure 5 (b).

For our application, we have to deal with views, that are close to nadir. Thus, we start our estimation algorithm with a coarse pose estimation, where only nadirs are evaluated (ω and ϕ are fixed in a nadir view). The estimation is realized with a high threshold for the minimum absolute value of gradient ($\mu = 12$), silhouette model segments and the three highest scalings of the objective function. As stated in figure 5, even for high scales, we have to avoid stepping into one of the local maxima, that are due to the cross shape of our objects. Thus, we sample the parameter space equidistantly, run six corse estimations with initial rotation angles $\kappa_i = i\frac{\pi}{3}$, $i = 0, \ldots, 5$, and decide for the one with the highest objective value.

After the position is roughly found, a fine estimation is applied. Therefore, a lower threshold for the image segments is used ($\mu = 3$) as well as the lowest three scalings of the objective function. Shape and position are now refined iteratively, where the first iteration starts with estimating the weights of two eigenvectors and the number of eigenvectors incorporated is increased after each iteration. Alltogether, this is an automatic algorithm for parameter estimation, where the pose parameters are automatically initialized by subsampling the parameter space and the shape parameters have to be initialized roughly by the user, as mentioned above.

4.2 Results and Conclusion

With the approach described above, we made first experiments on a small set of images, using our adaptive model for aircrafts. Some results are shown in figure 8, which clearly show the applicability of the method presented. On average, the time required for parameter estimation was approx. one minute on a Pentium IV 2 GHz.

Another example is shown in figure 9, which shows the result of fitting an adaptive model to a Boeing 747-200F. The model used, has a Boeing 747-400 as base model but none of the earlier versions of the Boeing 747 series was incorporated. As can be seen, the model catches the differing shape of the Boeing 747-200F. The fact that the turned up nose and the shadow of the aircraft does not mislead the model-fitting, clearly shows the advantages of combining data-driven linear models with our robust estimation: The estimation step workes well, even if parts of the object image features are systematically corrupted. The fact that the model is set up data-driven and elasticity is realized in linear form, prevents the model from getting stuck on wrong edge information and thus keep the aircraft in a realiable shape. This is done without incorporating mathematical constraints, since these constraints are implicitly given by the base models and linearity. Obviously, this restriction on elasticity demands the set of base models to be choosen carefully, such that all desired variations are realizable.

Choosing appropriate base models is even more important, if the observed objects are deformable: When aircrafts are off ground, they are significally deformed, e.g. by lift forces and turned out flaps, which has to be taken into account, when the base models are selected. In other words, tuning elasticity in our case is done by choosing an appropriate

set of base models. The next step towards aircraft classification is to apply our approach to a large set of images, in order to determine the error rate of the model-fitting and to examine the estimated model parameters concerning their separability.

Database: For the formation of our image database, we digitized direct prints from the *Hessisches Landesvermessungsamt*[3] with a $1:13\,000$ scale. Digitization was done with a resolution of $2\,500$ dpi. Thus, the resolution of the digital images is approx. 0.13 meters per pixel. Our model database is a set of models offered by Viewpoint Digital Inc.[4] and 3D ModelWorks [5], where each model has less than $1\,000$ polygons.

Acknowledgements: This work was supported by the German Federal Office of Defense Technology and Procurement (BWB), research project WITMUS II (knowledge-based automatic image interpretation for reconnaissance). The feature extraction modules were kindly provided by our research group Computer Vision at the Fraunhofer IITB.

References

[1] Berger L., Esswein M., Heinze N., Kanthak J., Korn A., Krüger W., Lakatsch E., Laubenheimer A., Link N., Müller M., Schwerdtmann W., Sauer G., Sung C.-K., Wetzel C., and Zimmermann G., "WITMUS - Wissensbasierte, teilautomatische Bildauswertung für die multisensorielle Aufklärung", Final Report, IITB Report No. 10508, 2000.

[2] Blanz V. and Vetter T., "A morphable model for the synthesis of 3D faces", In: *Computer Graphics Proceedings, SIGGRAPH'99*, pp. 187-194, 1999.

[3] Romdhani S., Blanz V., and Vetter T., "Face identification by fitting a 3d morphable model using linear shape and texture error function", In: *Proceedings ECCV 2002*, pp. 3-19, 2002.

[4] Shelton C., "Morphable surface models", *International Journal of Computer Vision*, vol. 38, no. 1, pp. 75-91, 2000.

[5] Lazarus F. and Verroust A., "Three-dimensional metamorphosis: a survey", *The Visual Computer*, vol. 14, no. 8/9, pp. 373-389, 1998.

[6] Popovič J. and Hoppe H., "Progressive Simplicial Complexes", In: *Computer Graphics Proceedings, SIGGRAPH'97*, pp. 217-224, 1997.

[7] Carr J., Beatson R., Cherrie J., Mitchell T., Fright W., McCallum B., and Evans T., "Reconstruction and representation of 3D objects with radial basis functions", In: *Computer Graphics Proceedings, SIGGRAPH 2001*, pp. 67-76, 2001.

[8] Turk G. and O'Brian J., "Shape Transformation Using Variational Implicit Functions", In: *Computer Graphics Proceedings, SIGGRAPH'99*, pp. 335-342, 1999.

[9] Bloomenthal J., "An Implicit Surface Polygonizer", In: *Graphics Gems IV*, pp. 324-329, 1994.

[10] Cohen-Or D., Levin D., and Solomovici A., "Three-dimensional distance field metamorphosis", *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 116-141, 1998.

[11] Garland M. and Heckbert P., "Surface simplification using quadric error metrics", In: *Computer Graphics Proceedings, SIGGRAPH'97*, pp. 209-216, 1997.

[12] Kobbelt L., "$\sqrt{3}$-subdivision", In: *Computer Graphics Proceedings, SIGGRAPH 2000*, pp. 103-112, 2000.

[13] Ohtake Y. and Belyaev A.. "Dual/primal mesh optimization for polygonized implicit surfaces", In: *Proceedings of ACM Solid Modeling Symposium*, pp. 171-178, 2002.

[14] Krüger W., "Robust and efficient map-to-image registration with line segments", *Machine Vision and Applications*, vol. 13, no. 1, pp. 39-50, 2001.

[15] Hashimoto M. and Skalansy J., "Multiple-order derivatives for detecting local image characteristics", *Computer Vision, Graphics, and image processing*, vol. 39, pp. 28-55, 1987.

[16] Steward C., "Robust parameter estimation in computer vision", *Siam Review*, vol. 41, no. 3, pp. 513-537, 1999.

[17] Press W., Teukolsky A.,Vetterling W., and Flannery B., "*Numerical Recipies in C: The Art of Scientific Computing*", Cambridge University Press, 2. ed., 1992.

[3] file number 5670-LA 122-67/99

[4] Viewpoint Digital Inc. 498 7th Avenue, Suite 1810. New York, NY 10018

[5] 3D ModelWorks, 620 Discovery Drive, Building2, Suite 250, Huntsville Alabama 35806

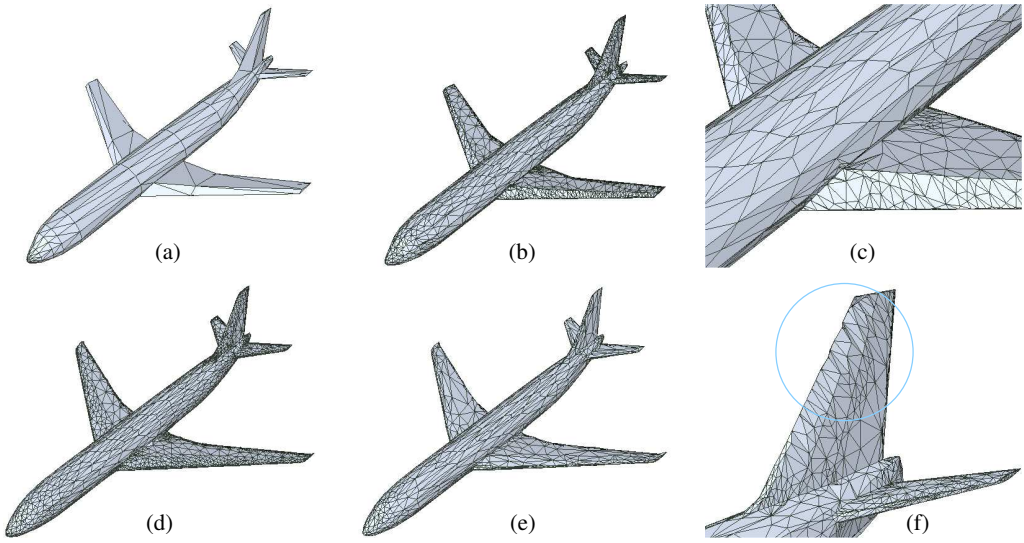

Figure 7: Results of model construction: (a) base model with original topology, (b) base model represented by meta-topology, (c) detail of (b), (d) model in mean realization ($\tilde{\theta} = \mathbf{o}$), (e) model reduced to 1 500 triangles in mean realization, and (f) errors due to undersampling.

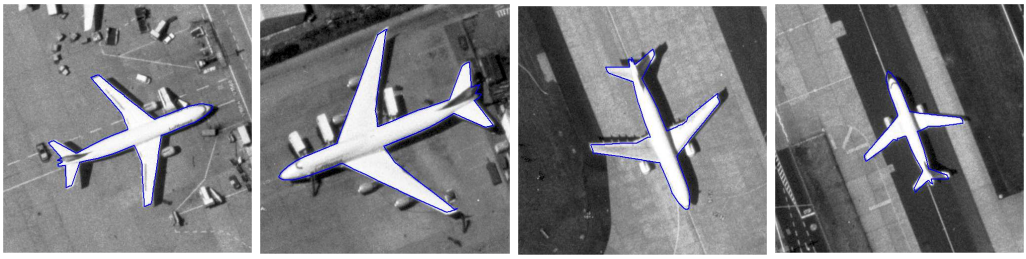

Figure 8: Results of the application specific pose and shape estimation as explained in section 4 (silhouettes given as blue contours). The result in the right image indicates, that the adaptive model used, has a lack of elasticity concerning the length of the fuselage.

Figure 9: Fitting an adaptive model to a Boeing 747-200F (silhouettes given as white contours): (a) result of pose estimation, when model parameters are fixed to represent the base model Boeing 747-400 and (b) result of shape estimation with starting parameters given by (a).