FACE MODEL FITTING WITH GENERIC, GROUP-SPECIFIC, AND PERSON-SPECIFIC OBJECTIVE FUNCTIONS

Sylvia Pietzsch
Chair for Image Understanding, Technische Universität München, Germany
sylvia.pietzsch@cs.tum.edu
Matthias Wimmer
Perceptual Computing Lab, Faculty of Science and Engineering, Waseda University, Tokyo, Japan
Freek Stulp
Group of Cognitive Neuroinformatics, University of Bremen, Germany
Bernd Radig
Chair for Image Understanding, Technische Universität München, Germany

Keywords: model fitting, person-specific, group-specific objective function

Abstract: In model-based fitting, the model parameters that best fit the image are determined by searching for the optimum of an objective function. Often, this function is designed manually, based on implicit and domain-dependent knowledge. We acquire more robust objective function by learning them from annotated images, in which many critical decisions are automated, and the remaining manual steps do not require domain knowledge.

Still, the trade-off between generality and accuracy remains. General functions can be applied to a large range of objects, whereas specific functions describe a subset of objects more accurately. (Gross et al., 2005) have demonstrated this principle by comparing generic to person-specific Active Appearance Models. As it is impossible to learn a person-specific objective function for the entire human population, we automatically partition the training images and then learn partition-specific functions. The number of groups influences the specificity of the learned functions. We automatically determine the optimal partitioning given the number of groups, by minimizing the expected fitting error.

Our empirical evaluation demonstrates that the group-specific objective functions more accurately describe the images of the corresponding group. The results of this paper are especially relevant to face model tracking, as individual faces will not change throughout an image sequence.

1 INTRODUCTION

Model-based image interpretation has proven appropriate to extract high-level information from images. Using a priori knowledge about the object of interest, these methods reduce the large amount of image data to a small number of model parameters, which facilitates and accelerates further interpretation. The model parameters **p** describe its configuration, such as position, orientation, scaling, and deformation. Facial expression interpretation, which is a major topic of our research, is often implemented with model-based techniques (Cohen et al., 2003; Tian et al., 2001; Pantic and Rothkrantz, 2000) Usually, the parameters of a deformable model describe the opening of the mouth, the direction of the gaze, or the raising of the eye brows, as depicted in Figure 1.

Model fitting is the computational challenge of determining the model parameters that best match a given image and this process usually consists of two components: the objective function and the fitting algorithm. The *objective function* evaluates how well a model fits to an image. In this paper, lower values represent a better model fit. These functions are usually designed manually by selecting salient image features and by mathematically composing them, see Figure 2 (left). Their appropriateness is then verified with test images. If the results are not satisfying, the objective function is tuned or redesigned from

Figure 1: In image sequences for recognizing facial expressions changes between the images are small.

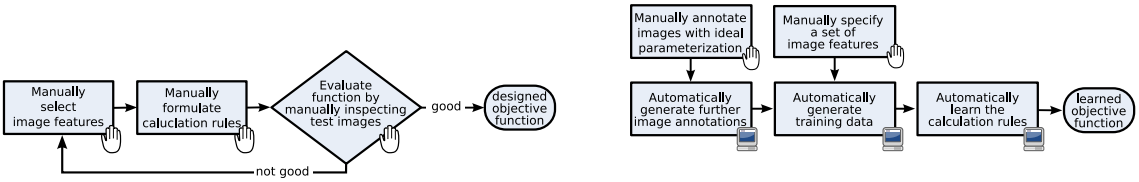

Figure 2: The traditional procedure for designing objective functions (left), and the proposed method for learning objective functions from annotated training images (right).

scratch. This approach is time-consuming and highly depends on the designer's intuition and his knowledge of the application domain. The *fitting algorithm* searches for the model parameters that constitute the global minimum of the objective function.

Model tracking represents the very similar challenge, where the model is repeatedly fit to a sequence of images. As changes from image to image are small, fitting results of previous images in the sequence constitute prior knowledge, which is often used to bias the fitting process in subsequent images. In our approach, the stationarity assumption is that the appearance of the face will not significantly change within the image sequence, e.g. a bearded man will not suddenly lose his beard. Knowing that the visible person has a beard, beard-specific model fitting increases fitting accuracy and processing speed throughout the remainder of the image sequence. In this paper, we propose to make the objective function the specific part and use standard model fitting strategies, such as Gradient Decent, CONDENSATION, Simulated Annealing, etc. We show how to learn generic and person-specific *objective functions*. (Gross et al., 2005) conduct a similar investigation comparing generic to person-specific Active Appearance Models.

The contributions of this paper are threefold. First, we demonstrate how to learn objective functions from manually annotated training images in order to avoid the shortcomings of the design approach. We automate many critical decisions, and the remaining manual steps hardly require domain-dependent knowledge, which simplifies the designer's task and makes it less error-prone. Second, we make the objective functions specific to one person by restricting the set of training images. This approach makes them highly appropriate for tracking a model through a sequence of images. We present an empirical evaluation that shows that person-specific functions are, as expected, more accurate than generic ones. Third, since these functions cannot be learned for the entire human population in advance, we are partitioning the set of training images such that the persons within each partition look similarly. Now, we learn partition-specific objective functions and we show the increase of accuracy, again. Since these functions are learned in advance, they have potential to improve face model fitting also for previously unseen persons.

This paper elaborates on face model applications but the insights presented are relevant for any other model-based scenario as well.

The remainder of this paper is organized as follows: Section 2 describes how to learn objective functions from annotated images. In Section 3, we elaborate on learning person-specific objective functions. Section 3 describes our approach to automatically determine the optimal partitioning for learning partition-specific objective functions. Section 4 compares model fitting with the generic and specific functions. Section 6 summarizes our approach and gives an outlook to future work.

2 LEARNING GENERIC OBJECTIVE FUNCTIONS

An objective function $f(I,\mathbf{p})$ is either computed directly from the image I and the model parameters $\mathbf{p}$ or as a sum of *local* objective functions $f_n(I,\mathbf{x})$, as in Equation 1. These local functions consider the image content in the vicinity of the model's contour point $\mathbf{c}_n(\mathbf{p})$ only. They are easier to design than global ones and therefore, this approach is widely used in current model fitting research (Cristinacce and Cootes, 2006; Romdhani, 2005; Hanek, 2004; Cohen et al., 2003). As their main advantage, their low-dimensional search space $\mathbf{x}\in\mathbb{R}^2$ facilitates minimization. For a more elaborate discussion, we refer to (Wimmer et al., 2007).

$$f(I,\mathbf{p}) = \sum_{n=1}^{N} f_n(I,\mathbf{c}_n(\mathbf{p})) \qquad (1)$$

So-called *ideal* objective functions have two properties: First, their global minimum corresponds to the best model fit. This implies that finding the global minimum is sufficient for fitting the model. Second, the objective function must have no local minimum apart from the global minimum. This implies that any minimum found corresponds to the *global* minimum,

which facilitates search. An example of an ideal local objective function is shown in Equation 2, where the preferred model parameters $\mathbf{p}_I^\star$ denote the best model fit for a certain image I, according to human judgment. Since $\mathbf{p}_I^\star$ is not known for unseen images, $f_n^\star$ cannot be used in fitting applications. Instead, we take this ideal objective function to generate training examples for learning a further objective function f_n^ℓ. The key idea behind our approach is that the ideal objective function $f_n^\star$ is used to generate the training data, from which f_n^ℓ is learned. Since $f_n^\star$ has these two properties of idealness, f_n^ℓ will approximately have them as well. Figure 2 illustrates the five-step procedure of learning objective functions.

$$f_n^\star(I,\mathbf{x}) = |\mathbf{x} - \mathbf{c}_n(\mathbf{p}_I^\star)| \qquad (2)$$

Different images contain faces of different sizes. Distance measures, such as the return value of $f_n^\star$, must not be biased by this variation. Therefore, we convert all distances in pixels to the interocular measure by dividing them by the pupil-to-pupil distance.

Our methods are independent of the model. Here, we use the *Active Shape Model* approach of Cootes et al. (Cootes and Taylor, 1992) to model two-dimensional human faces. The model parameters $\mathbf{p}=(t_x,t_y,s,\theta,\mathbf{b})^T$ consist of translation, scaling, rotation, and a vector of deformation parameters $\mathbf{b}$. The face model contains N=134 contour points that are projected to the surface of the image by $\mathbf{c}_n(\mathbf{p})$ with $1\leq n\leq N$, see Figure 1.

Step 1. Manually Annotate Images. A database of images I_k with $1\leq k\leq K$ is manually annotated with the ideal model parameters $\mathbf{p}_{I_k}^\star$. These parameters allow to compute the ideal objective function $f_n^\star$, see Equation 2. For synthetic images, $\mathbf{p}_{I_k}^\star$ is known, and can be used in such cases. For real-world images, however, $\mathbf{p}_{I_k}^\star$ depends on the user's judgment. Annotating the images represents the only laborious step of the proposed methodology. For our experiments, we manually annotated 500 images, which takes an experienced person around 1 minute per image.

Step 2. Generate Further Annotations. The ideal objective function returns the minimum $f_n^\star(I,\mathbf{x})=0$ for all image annotations, because $\mathbf{x}=\mathbf{c}_n(\mathbf{p}_I^\star)$. This data is not sufficient to learn the characteristics of f_n^ℓ. Therefore, we will acquire image annotations $\mathbf{x}\neq\mathbf{c}_n(\mathbf{p}_I^\star)$, for which $f_n^\star(I,\mathbf{x})\neq 0$. In general, any position within the image may represent one of these annotations. However, it is more practicable to restrict this motion in terms of distance and direction, as is done in (Ginneken et al., 2002)

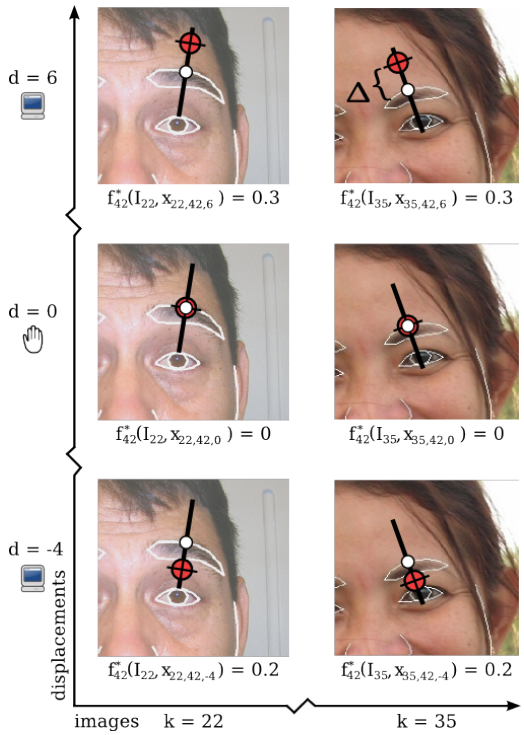

Figure 3: In each of the K images, each of the N contour points is annotated with $2D+1$ displacements. Manual annotation is only necessary for d=0 (middle row). The other displacements are computed automatically. The upper right image shows the learning radius Δ. The unit of the ideal objective function values and Δ is the interocular measure.

Therefore, we generate a number of displacements $\mathbf{x}_{k,n,d}$ with $-D\leq d\leq D$ that are located on the perpendicular to the contour line with a maximum distance Δ to the contour point. This procedure is illustrated in Figure 3. The center row depicts the manually annotated images, for which $f_n^\star(I,\mathbf{x}_{k,n,0})=f_n^\star(I,\mathbf{c}_n(\mathbf{p}_{I_k}^\star))=0$. The other rows depict the displacements $\mathbf{x}_{k,n,d\neq 0}$ with $f_n^\star(I,\mathbf{x}_{k,n,d\neq 0})>0$.

Step 3. Specify Image Features. We learn a mapping from I_k and $\mathbf{x}_{k,n,d}$ to $f_n^\star(I_k,\ \mathbf{x}_{k,n,d})$, which is called f_n^ℓ. Since f_n^ℓ has no access to $\mathbf{p}_I^\star$, it must compute its value from the image content. However, we do not directly evaluate the pixel values but apply a feature-extraction method, see (Hanek, 2004). The idea is to provide a multitude of features, and let the learning algorithm choose which of them are relevant to the calculation rules of the objective function.

Our approach takes Haar-like image features (Viola and Jones, 2001) of different styles and sizes, which greatly cope with noisy images. They are not only computed at the location of the contour point it-

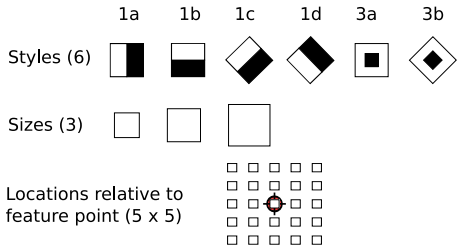

Figure 4: A set of A=6·3·5·5=450 features is used for learning the objective function.

self, but also at several locations within its vicinity, see Figure 4. This variety of $1 \leq a \leq A$ image features enables the objective function to exploit the texture of the image at the model's contour point and in its surrounding area. Each of these features returns a scalar value, which we denote with $\mathbf{h}_a(I, \mathbf{x})$.

Step 4. Generate Training Data. The result of the manual annotation step (Step 1) and the automated annotation step (Step 2) is a list of $K(2D+1)$ image locations $\mathbf{x}_{k,n,d}$ for each of the N contour points. Adding the corresponding target value $f_n^\star$ yields the list in Equation 3.

$$\left[\quad I_k, \quad \mathbf{x}_{k,n,d}, \quad f_n^\star(I_k, \mathbf{x}_{k,n,d}) \right] \quad (3)$$

$$\left[\; \mathbf{h}_1(I_k, \mathbf{x}_{k,n,d}), \ldots, \mathbf{h}_A(I_k, \mathbf{x}_{k,n,d}), \; f_n^\star(I_k, \mathbf{x}_{k,n,d}) \right] \quad (4)$$

$$\text{with } 1 \leq k \leq K, \; 1 \leq n \leq N, \; -D \leq d \leq D$$

Applying each feature to Equation 3 yields the training data in Equation 4. This step simplifies matters greatly. We have reduced the problem of mapping high-dimensional image data and pixel locations to the target value $f_n^\star(I, \mathbf{x})$, to mapping a list of feature values to the target value.

Step 5. Learn Calculation Rules. The local objective function f_n^ℓ maps the feature values to the result value of $f_n^\star$. Machine learning infers this mapping from the training data in Equation 4. Our proof-of-concept uses model trees (Witten and Frank, 2005; Quinlan, 1993) for this task, which are a generalization of decision trees. Whereas decision trees have nominal values at their leaf nodes, model trees have line segments, allowing them to map features to a continuous value, such as the value returned by $f_n^\star$. These trees are learned by recursively partitioning the feature space. A linear function is then fitted to the training data in each partition using linear regression. One of the advantages of model trees is that they tend to use only features that are relevant to predict the target values. Currently, we are providing A=450 image features, see Figure 4. The model tree selects around 20 of them for learning the calculation rules.

After these five steps, a local objective function is learned for each contour point. It can now be evaluated at an arbitrary pixel $\mathbf{x}$ of an arbitrary image I.

3 LEARNING SPECIFIC OBJECTIVE FUNCTIONS

Without having any specific knowledge about the given image, generic objective functions as presented in Section 2 are able to provide an acceptable model fit. This is a considerable challenge, because there is an immense variation between the images, e.g. due to gender, hair style, etc. The training images must contain a high range of these conditions in order to yield robust objective functions. In contrast, the image content between two consecutive frames does not greatly change in image sequences. The model must be fitted to each single image within the sequence, but the image content is not arbitrary, because many image and model descriptors are fixed or only change gradually, such as illumination, background, or camera settings.

In the case of facial expression recognition, it can be assumed, that the identity of the person is fixed. Therefore, the appearance of the face only changes slightly from frame to frame. For model fitting, it is sufficient to apply an objective function that is specific to this person. As the advantage of this approach, person-specific objective functions are much more accurate than generic ones. Table 1 summarizes and compares the properties and capabilities of generic and person-specific objective functions. Note that the learned function is highly accurate for images of the specific person, but it yields arbitrary and potentially bad results for images of different persons.

In this paper, we obtain person-specific objective functions by slightly altering Step 1 of the machine learning methodology explained in Section 2. The here utilized image database does not consist of arbitrary face images any more, but face images of one specific person. Nevertheless, it is important that these images still contain a considerable variation w.r.t. further image conditions, such as illumination, background, and facial pose. Section 4 demonstrates the increase of fitting accuracy comparing generic and person-specific objective functions.

3.1 OPTIMAL PARTITIONING

Unfortunately, it is not possible to learn person-specific objective functions for each individual of the entire human population. Therefore, we acquire images of R persons and we propose to learn objective

	generic objective function	person-specific objective function
uses knowledge about person	no	yes
appropriate for	single image	image sequence
learned with	images of different persons	images of a single person
accuracy	any person: moderate accuracy	specific person: very high other persons: undefined, rather low
effort for learning	learned once	learned for every person separately
number of partitions G	$G = 1$	$G = R$

Table 1: Comparing the properties and capabilities of generic and person-specific objective functions.

functions for groups of people, comprising similarities, e.g. gender, age, beard, hair style. Dividing the set of persons into G partitions, the number of partitionings possible is described by the Stirling numbers of the Second Kind $S(R,G)$, see Equation 5.

$$S(R,G) = \frac{1}{G!}\sum_{i=0}^{G}(-1)^i\binom{G}{i}(G-i)^R \qquad (5)$$

Setting G=1 or G=R, there is only one partitioning, because $S(R,1) = 1$ and $S(R,R) = 1$ respectively. In the case of G=1, one partition contains all persons. The partition-specific objective function is equivalent to a generic objective function. In the case of G=R, each partition contains images of one person only. The partition-specific objective function is equivalent to a person-specific objective function. Setting $1<G<R$, the level of specificity of the partition-specific objective functions is between the generic and the person-specific objective function. Higher values of G lead to more specific and accurate, but to less general objective functions. Note that in this case, there are several partitionings, because $S(R,G) > 1$.

```
1 foreach partition in partitions do
2     foreach image in partition do
3         Perform model fitting by applying the
          correct partition-specific objective function
          (e.g. f^(AB) for images with persons A and B)
4         Determine the fitting error on image by
          considering the manual annotations
5     end
6     Compute mean error over all images in partition
7 end
8 Compute λ, the mean error over all partitions
```

Algorithm 1: Computing the error measure λ.

Accurate partition-specific objective functions cannot be learned for every partition. Therefore, the feasability of this method depends on the number of partitions G and the partitions created. We compute an error measure λ for each partitioning, see Algorithm 1. The optimal partitioning is minimizes λ. The challenge is to determine the partitioning with the *minimum* error. It is obtained by exhaustive search for small values of G only. Determining the best partitioning is performed off-line, but it is computationally expensive, especially when R and G are large.

In order to integrate partition-specific objective functions into real-world applications, the correct partition of a persons must be determined on-line. This allows the execution of the correct partition-specific objective function. Selecting the wrong function leads to a much lower accuracy than selecting the generic objective function. In order to determine the correct partition-specific objective function, we are using state-of-the-art person identification, see (Nefian and Hayes, 1999).

4 EXPERIMENTAL EVALUATION

This paper proposes to adapt the objective function to particular persons or groups of persons in order to facilitate model tracking. In this section, we inspect the increase of accuracy that is achieved with these specific objective functions. Furthermore, we evaluate the applicability of the partitioning method proposed in Section 3. All tests are performed using a two-dimensional, deformable, contour model of a human face that is build according to the Active Shape Model approach (Cootes and Taylor, 2004).

Evaluation data. The experiments require a data base of several images of various persons. In order to learn a generic objective function the training set needs to contain a representative variation of human faces. We extract an image sequence for $R = 45$ different persons from news broadcasts on TV. They comprise of news anchormen and politicians as well as passers-by giving short interviews. The image sequences cover a large variation of environmental aspects as well as faces with different properties, such as beards, glasses, gender. Within the image sequences,

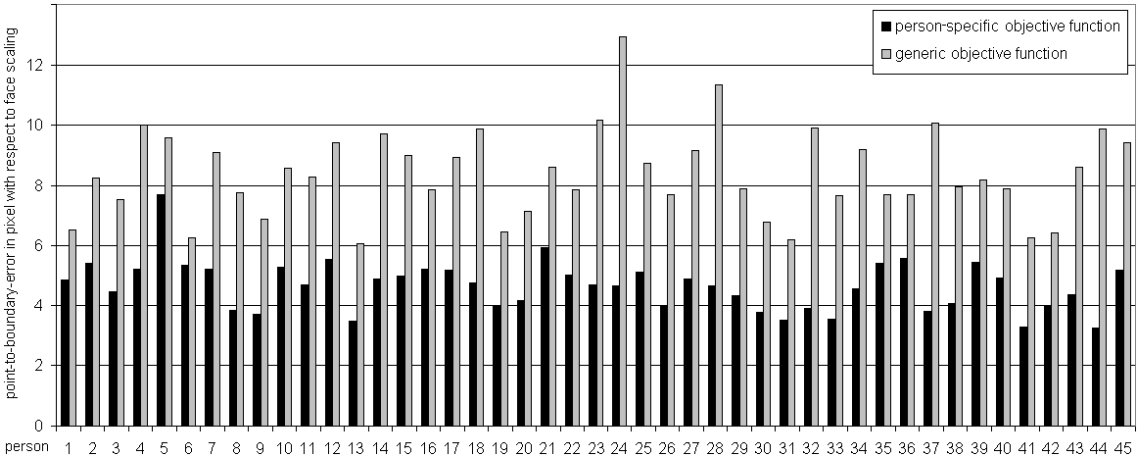

Figure 5: Point-to-boundary error for model fitting using a generic (gray) and person-specific objective functions (black).

persons move their head and show facial muscle activity. We annotate ten images of each person with the ideal model parameters, which amounts to 450 annotated images and split this set up into training (70%) and test images (30%).

Generic vs. person-specific objective functions. According to the description in Section 2, a generic objective function f^{ℓ} is learned from the annotated images of all R persons. Furthermore, we create R person-specific objective functions f^{r} with $1 \leq r \leq R$. Our evaluation fits the face model to all test images of each person using either objective function. Accuracy of fitting is quantified as the average point-to-boundary error, which is the minimum distance between the contour points $\mathbf{c}_n(\mathbf{p})$ and the contour line of the manually annotated model $\mathbf{p}_I^{\star}$. This distance is converted to the interocular measure. Figure 5 illustrates these values for the generic and person-specific objective functions of all persons. The x-axis denotes the person's ID and the y-axis indicates the mean point-to-boundary error. It is clearly visible that learning objective functions for a specific person improves the process of fitting the face model to the images that show this person. Also, Table 2 illustrates the same evaluation for three example persons P26, P42, and P44. As expected, the fitting error is very low for images of the person the objective function is specific to.

P26 P42 P44

Figure 6: An example image of each of the three persons used for learning partition-specific objective functions.

Partition-specific objective functions. The previous experiment investigates the increase of accuracy comparing person-specific objective functions over generic ones. But as described in Section 3, this cannot be used beneficially in real-world applications and we propose a further method that partitions the set of persons within the training images. In the remainder of this section, we verify two issues concerning partition-specific objective functions: First, the feasibility of the partitioning is shown by means of a selective example. Second, the gain in accuracy holds for partition-specific objective functions as well.

This experiment considers a rather simple case, but it proves our statement that a decent partitioning does affect the fitting accuracy. We extract $R = 3$ per-

objective function	evaluated on		
	P26	P42	P44
generic:			
f^{ℓ}	7.7	7.9	9.9
person-specific:			
f^{26}	**4.0**	17.0	19.0
f^{42}	15.6	**3.9**	9.8
f^{44}	13.6	11.5	**3.2**
partition-specific:			
$f^{(26,42)}$	**4.8**	**4.4**	12.7
$f^{(42,44)}$	16.1	**4.1**	**3.7**
$f^{(26,44)}$	**4.6**	13.3	**4.1**

Table 2: Point-to-boundary error after model fitting. This error is small for objective functions specific to a certain person or partition (bold numbers).

sons from our image data base, see Figure 6, and we will refer to them as P26, P42, and P44. The set was chosen consciously to contain two persons that look similar (P42 and P44) and one person that differs in the outward appearance. Setting G=2 partitions, we then create all $S(3,2)$=3 partitionings and learn a specific objective function $f^{(r_1,\cdots)}$ for each partition. Note that a partition containing only one person yields an objective function that is equivalent to the person-specific objective function of this person (e.g. $f^{26} \equiv f^{(26)}$).

For evaluation, we fit the model to the test images of these three persons using the pertition-specific objective functions created. Again, the accuracy is represented by the average point-to-boundary error w.r.t the ideal model parameterization $\mathbf{p}_I^\star$. Table 2 illustrates the fitting results applying the partition-specific objective functions created. In all cases, the partition-specific objective function achieves high accuracies for the partition members.

Different partitionings. One of the major points of using partition-specific objective functions is how many partitions to create and to which partitions the persons belong to. Algorithm 1 shows how to calculate an error measure λ that indicates how good a certain partitioning is. Here, we calculate this error measure for the three partitionings of the previous experiment, as depicted in Table 3. The partitioning (P42,P44)(P26) turns out to be best, because it has minimum $\lambda = 3.9$. As expected, our approach clusters the two persons that look similar into one partition and the other person into the other partition.

partitioning	fitting error with correct objective function			λ
	P26	P42	P44	
(26)(42,44)	4.0	4.1	3.7	**3.9**
(42)(26,44)	4.6	3.9	4.1	4.2
(44)(26,42)	4.8	4.4	3.2	4.1

Table 3: Compare the all partitionings of $R = 3$ persons into $G = 2$ partitions by means of the average fitting error λ.

5 DISCUSSION

Learning objective functions instead of designing them manually has several benefits both for the objective function and for the designer. First of all, it automates design decisions which are critical to the robustness of the resulting objective function. The two critical decisions within the designing process are feature selection and their mathematical composition. In our approach, the model tree algorithm automates both, as it tends to use only relevant features, and performs a piecewise linear approximation of the target function with these features. The selection of features is based on objective information theoretic measures, which model trees use to partition the space of the image features, instead of relying on human intuition. A human can only reason about a very limited amount of features, whereas model trees are able to consider (and discard) hundreds of features simultaneously. The resulting objective functions are therefore more accurate and robust, and easier to optimize. Each local objective function $f_n^\ell(I,\mathbf{x})$ uses its own calculation rules and image feature set, because a separate model tree is learned for each contour point. Customizing the calculation rules for each local objective function would also be possible when designing objective functions. However, this is usually not exploited, because it is too laborious and time-consuming.

There are cases in which model fitting with learned objective functions fails to match the face model to the image appropriately. The objective function is only capable of computing an accurate value for locations that are in a certain vicinity of the correct contour point determined by the learning radius Δ. Beyond this radius, the result of the objective function is undefined, because this image content has not been used for learning. In-plane rotations of the face must not be too high, because Haar-like features are not rotation invariant. Other researchers have also faced this issue, and (Jones and Viola, 2003) propose a solution to this shortcoming. Alternatively, integrating rotation invariant features suffices as well.

Designing objective functions requires extensive domain-dependent knowledge about model fitting and feature extraction methods. In our approach, the main remaining manual step is the annotation of images with the best model fit. This annotation is intuitive, and can be performed with little domain-dependent knowledge. The features provided and learning algorithms used are not specific for the application domain. Objective functions can therefore be tailored to different domains simply by using a different model and a different set of images annotated with this model.

6 SUMMARY AND OUTLOOK

Due to the large variations in facial appearances in images, it is challenging to find a general model fitting procedure that fits all faces robustly and accurately. In this paper, we compare specific with generic objective functions, one of the three main components in model

based fitting. These objective functions are learned from annotated images. Generic and person-specific objective functions are learned by training them with all or only images with a specific person in them respectively. In practice, it is infeasible to learn objective function for each person individually. We therefore extend the person-specific approach by first automatically partitioning the set of images into similar partitions before learning, and then learning partition-specific objective functions.

The main application of partition-specific objective functions, is tracking models through image sequences. Although the appearance of a face might change during an image sequence due to lighting etc., the face itself does not. Therefore, once the partition a face belongs to is established, a partition-specific objective function can be used throughout the image sequence.

The empirical evaluation first shows how person-specific objective functions achieve a substantial higher fitting accuracy for the person for which it was trained. We then show the result of applying different partition-specific objective functions on images in and outside of the partition. As expected, partition-specific objective function perform substantially better than generic ones for persons from the partition for which they were trained, but worse on persons not in this partition. Higher accuracy comes at the cost of lower generality. This trade-off is influenced by the number of intended partitions G.

The off-line partitioning for learning partition-specific objective functions is performed automatically. We are currently investigating the use of an automatic classification to determine on-line, to which partition a person belongs, and which objective function should be used.

ACKNOWLEDGEMENTS

This research is partially funded by a JSPS Postdoctoral Fellowship for North American and European Researchers (FY2007) as well as by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition.

It has been jointly conducted by the Perceptual Computing Lab of Prof. Tetsunori Kobayashi at Waseda University, the Chair for Image Understanding at the Technische Universität München, and the Group of Cognitive Neuroinformatics at the University of Bremen.

REFERENCES

Chibelushi, C. C. and Bourel, F. (2003). Facial expression recognition: A brief tutorial overview.

Cohen, I., Sebe, N., Chen, L., Garg, A., and Huang, T. (2003). Facial expression recognition from video sequences: Temporal and static modeling. *CVIU special issue on face recognition*, 91(1-2):160–187.

Cootes, T. F. and Taylor, C. J. (1992). Active shape models – smart snakes. In *BMVC*, pp 266 – 275.

Cootes, T. F. and Taylor, C. J. (2004). Statistical models of appearance for computer vision. Technical report, U of Manchester, Imaging Science and Biomedical Engineering, Manchester M13 9PT, UK.

Cristinacce, D. and Cootes, T. F. (2006). Facial feature detection and tracking with automatic template selection. In *FGR*, pp 429–434.

Ginneken, B., Frangi, A., Staal, J., Haar, B., and Viergever, R. (2002). Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21(8):924–933.

Gross, R., Baker, S., Matthews, I., and Kanade, T. (2004). Face recognition across pose and illumination. In Li, S. Z. and Jain, A. K., editors, *Handbook of Face Recognition*. Springer-Verlag.

Gross, R., Matthews, I., and Baker, S. (2005). Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093.

Hanek, R. (2004). *Fitting Parametric Curve Models to Images Using Local Self-adapting Seperation Criteria*. PhD thesis, Dept of Informatics, TU München.

Jones, M. J. and Viola, P. (2003). Fast multi-view face detection. Technical Report TR2003-96, Mitsubishi Electric Research Lab.

Nefian, A. and Hayes, M. (1999). Face recognition using an embedded HMM. In *Proc. of the IEEE Conference on Audio and Video-based Biometric Person Authentication*, pp 19–24.

Pantic, M. and Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE TPAMI*, 22(12):1424–1445.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California.

Romdhani, S. (2005). *Face Image Analysis using a Multiple Feature Fitting Strategy*. PhD thesis, U of Basel, Computer Science Department, Basel, CH.

Tian, Y.-L., Kanade, T., and Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE TPAMI*, 23(2):97–115.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*, vol 1, pp 511–518, Kauai, Hawaii.

Wimmer, M., Stulp, F., Pietzsch, S., and Radig, B. (2007). Learning local objective functions for robust face model fitting. In *IEEE PAMI*. to appear.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.