Face Alignment by 2.5D Active Appearance Model Optimized by Simplex

A. Sattar, Y. Aidarous, S. Le Gallou and R. Seguier

abdul.sattar@supelec.fr, sylvain.legallou@supelec.fr, yasser.aidarous@supelec.fr, renaud.seguier@supelec.fr
Team SCEE, SUPELEC/IETR, Avenue de la Boulaie, 35576 Cesson Sévigné, France

Abstract. In this paper we propose an efficient algorithm to align the face in real time, based on Active Appearance Model (AAM) in 2.5D. The main objective is to make a robust, rapid and memory efficient application suitable for embedded systems, so they could align the pose rapidly by using less memory. Classical AAM is a high memory consumer algorithm, consequently transfer of this stored memory in an embedded system makes it a time consuming algorithm as well. Our 2.5D AAM is generated by taking 3D landmarks from frontal and profile view and 2D texture only from frontal view of the face image. Moreover we propose Nelder Mead Simplex technique for face search. It does not require large memory, thus becoming suitable for embedded systems by eliminating the excess memory and access time requirements. We illustrate 2.5D AAM optimized by Simplex for pose estimation and test it on three databases: M2VTS, synthetic images and webcam images. Results validate our combination of simplex and AAM in 2.5D.

1 Introduction

Our team works in Human Machine Interface (HMI) for embedded systems and specially in face analysis i.e. face alignment, recognition, feature extraction and expression detection. Not only a solution to this interface is required but at the same time that solution should be efficient and robust. Talking about efficiency and robustness, the constraints of time and memory are also to be tackled specially when face analysis methods are implemented on embedded HMI systems (e.g. Mobile phone, game console etc).

Generally global face analysis and synthesis system is composed of three phases (see figure 1). Initially, the face is detected. Followed by face alignment with respect to the coordinates. Finally this aligned face can be used for several man machine interactions. Our main focus is on the second phase of the global face analysis system i.e. face alignment.

Many methods have been proposed in this respect. [1] presented the idea of Active Appearance Model (AAM) in which shape and textural information is included into the model and regression matrices (RM) are used as a search algorithm. RM are build during the learning phase of AAM taking into the account all the faces of the given database. Convergence occurs by varying model

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

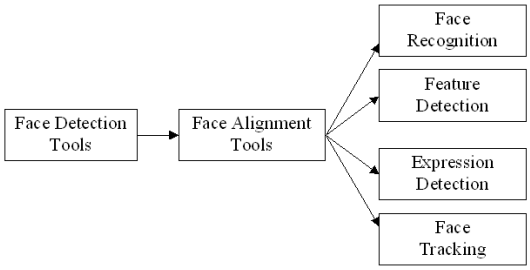


Fig. 1. Face Analysis System

coefficients and minimizing the error between model and an object. Since AAM was first introduced by regression matrix therefore we call it as a classical AAM. We have chosen this AAM but without its classical search algorithm with which it becomes high memory consumer and eventually makes it a bad choice for our embedded system applications. This RM also weakens generalization property of our proposed system since it inhibits only those faces of the database from which it has been created. [2] performed pose prediction by using 3 AAMs, one frontal view and two profile views, which is not efficient enough for our system because of three times increase in data transfer between processor and memory.

Now we present the work of other teams dealing with the problem of memory and robustness. In [3] simple gradient descent method was used as a search algorithm for face tracking by AAM. [4] used another fitting algorithm, inverse compositional image alignment algorithm [5], which is again an extension of the gradient descent method. In our work we restrict those algorithms (Gradient descent) which require the computation of function derivatives. Moreover it is more complex and slow for our system.

[6] applied 3D model of head, obtained from a 3D scanner, to track the face in a stream of a video by varying appearance parameters obtained by AAM. Although they used Nelder-Mead Simplex as an optimization technique but their model generation procedure is quite cumbersome. They applied Mutual Information and Correlation Ratio as similarity metrics.

Our main objective is to align the face efficiently in real time by consuming less storage memory. Three dimension alignment of face required 6 pose parameters i.e. three translational parameters t_x, t_y, *scale* and three rotational parameters θx, θy, θz. To achieve this objective we propose 2.5D Active Appearance Model by using simplex as a search algorithm. In training phase 2.5D AAM is constructed by combination of i) 3D landmarks taken from frontal and profile view of a face image ii) texture of only frontal view of a face image. In testing phase pose is estimated by simplex method, implemented in such a way, to reduce the requirement of high memory and time. These two aspects of our application make the embedded systems to have less memory and save time to transfer the data between processor and memory. Along with that it do not require a-priori knowledge of the learning phase as in the case of RM, again saving

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

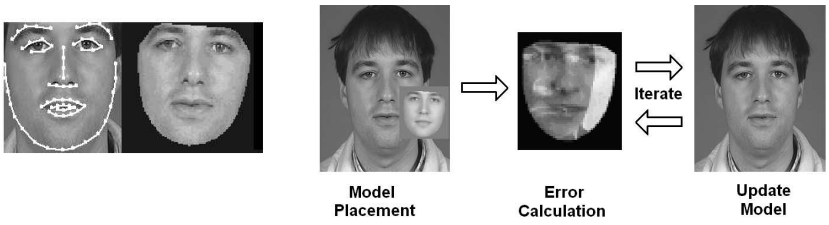
the time in learning phase. In addition to this, absence of this RM give us the ability of generalization as well.

Remaining of the paper is organized as follow. Section 2 explains the creation of 2.5D Active Appearance Model. Section 3 explains how the face is aligned by 2.5D AAM using Nelder Mead Simplex method. Section 4 illustrates the results obtained during experiments. And section 5 concludes the paper.

2 2.5D Active Appearance Model

2.1 AAM

In 2D AAM, the model contains both shape and texture variations. In the training phase (see figure 2(a)) landmarks are marked on all the faces of the database either manually or automatically. Shape model is obtained by calculating mean of these shapes followed by PCA. This mean shape is used to extract the frontal view of all the faces of the database to acquire mean texture. Texture model is obtained by performing PCA. By combining above two models, appearance model is obtained. By varying the parameters of this appearance model one can reshape the model to any face. Matching is performed by projecting the model on the image and reducing the error by varying the model parameters. Convergence occurs after a number of iterations and match is obtained as shown in figure 2(b).



(a) Training Phase (b) Testing Phase

Fig. 2. AAM Training and Testing Phases

2.2 2.5D AAM

Our aim is to build 2.5D AAM using only the frontal view image. 2.5D AAM is somewhat different and it requires little more steps than 2D AAM, which are.

3D Landmarks. In our 2.5D AAM 68 points are marked manually as in 2D AAM as shown in figure 3. Although profile view of the person is used to make a 3D shape but only the frontal view image is taken for the texture database. If

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

there are N number of images in the database then the vector representation of these shapes is

$$s_N = [x_1, x_2, ...x_{68}, y_1, y_2, ...y_{68}, z_1, z_2, ...z_{68}] \tag{1}$$

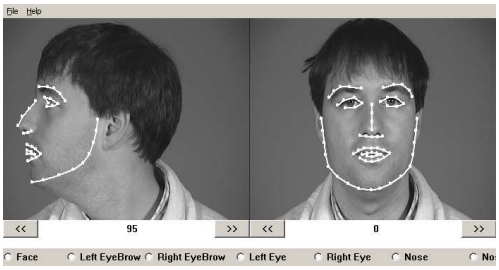


Fig. 3. Landmark Placement

Shape Model and Parameters. All the landmarks obtained in the previous step are resized and aligned in three dimension using procrustes analysis [7, 8]. Mean of these 3D landmarks is calculated which is called mean shape. Principal Component Analysis (PCA) is performed on these shapes to have shape parameters with 95% of the variation stored in them.

$$s_i = \bar{s} + \phi_s * b_s \tag{2}$$

where s_i is the synthesized shape, $\bar{s}$ is mean shape, ϕ_s are eigen vectors obtained during PCA and b_s are shape parameters.

Texture Model and Parameters. 3D Mean shape obtained in the previous step is used to extract and warp (based on the Delaunay triangulation) the frontal views of all the face images. Only two dimensions of the mean shape is used to get 2D frontal view textures. This is why we call our model as 2.5D AAM, since it is composed of landmarks represented in 3D domain but only 2D texture is warped on this shape to acquire 2.5D model. Mean of these texture is calculated. Followed by another PCA to acquire texture parameters with 95% of the variation stored in these parameters.

$$g_i = \bar{g} + \phi_g * b_g \tag{3}$$

where g_i is the synthesized texture, $\bar{g}$ is mean texture, ϕ_g are eigen vectors obtained during PCA and b_g are texture parameters.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Appearance Model and Parameters. Both of the above parameters are combined by concatenation of b_s and b_g. And a final PCA is performed to have the appearance parameters.

$$b = [b_s b_g]^T, b = \phi_c * c \tag{4}$$

where ϕ_c are the eigen vectors obtained by retaining 95% of the variation. And c is the matrix of the appearance parameters, which are used to obtain shape and texture of each face of the database.

Pose Parameters. 2.5D Model can be translated as well as rotated with the help of translational and rotational parameters. Therefore we have three angles of rotation and three translational parameters named as pose parameters P.

$$P = [\theta_x, \theta_y, \theta_z, t_x, t_y, scale]^T \tag{5}$$

where θ_x correspond to the face rotating around x axis (shaking head up and down), θ_y to the face rotating around y axis (semi profile views) and θ_z to the face rotating around z axis. t_x, t_y are the offset values from the supposed origin and *scale* is the magnification of the model. Figure 4 shows the face rotating around y axis making left and right semi profile views.

Fig. 4. Snapshots of rotating 2.5D AAM

2.5D Active Appearance Model. In order to search a face by 2.5D AAM, it is necessary to acquire mean shape and mean texture along with the limits of appearance and pose parameters. The model is rendered by the following steps:

- 3D mean shape is rotated and translated to the desired value of pose parameters.
- Mean texture is overlaid on this rotated shape.
- Appearance parameters are varied to fine tune the shape as well as texture to obtain each individual of the database.

In this way we can have frontal, half profile and all the remaining views of each individual of the database as shown in figure 4. In the testing phase this model is projected on the unknown image and face is searched.

Generalization comes from the fact that while varying the appearance parameters we can have such faces that do not exist in the database. In fact this face is the mixture of the faces of the given database. Therefore controlled variation

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

of the appearance parameters can create new faces. This generality increases as the number of faces in the given database increases.

3 2.5D AAM Pose Estimation

3.1 Image Similarity Metrics

Along with optimization technique of simplex, similarity metrics also plays an important role in matching. In each iteration similarity between model and an object is calculated and residual error value is returned to simplex to re-evaluate the parameters of the model to get a closer match. In order to choose the best similarity metrics for our application we have performed experiments on three known methods i) Euclidean Distance (EUC) defined as a distance between two vectors, ii) Correlation Coefficient and iii) Mutual Information.

Correlation Coefficient (CC). It is a linear relationship between two random variables. It is obtained by dividing the covariance of the two variables by the product of their standard deviations.

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{6}$$

where a_i and b_i are the gray levels of the images to be matched for errors. $\bar{a}$ and $\bar{b}$ are the means of these images and n is the total number of pixels in an image.

Mutual Information (MI). It is more complex and slow as compare to EUC and CC. It is calculated in terms of individual and joint entropy of the two images to be matched [6].

$$MI = E(h_a) + E(h_b) - E(h_{a,b}) \tag{7}$$

$$E(h) = -\sum h(x_i) log h(x_i) \tag{8}$$

where $E(h_a)$ and $E(h_{a,b})$ are individual and joint entropy whereas h_a and $h_{a,b}$ are 1D and 2D histograms respectively.

3.2 Simplex

Nelder Mead Simplex algorithm [9] is used to optimize the pose parameters to estimate the pose. As discussed earlier there are six pose parameters of which three are rotational and three are translational. The target is to find out the best possible values of these parameters giving minimum error value between model and the face. Limits are applied on these parameters and seven $(number of parameters + 1)$ possible solutions are chosen randomly within these limits. Simplex starts the face search and calculates the best values of pose

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

parameters corresponding to minimum error obtained. Simplex works in an iterative manner to search for minimum values. In each iteration parameters are modified (reflected, expanded, contracted or shrunk [9]) based on the previous value of error. It converges when no further good solutions are possible. Fixed number of these iterations are applied for each simplex executions. This number is either fixed by number of convergences or processing time. In return it provides seven solutions among them first is the best possible.

Simplex initialization is in random and to avoid divergence, limits of parameters are introduced, which forces it to search minimum value within these limits. It is obvious that as we narrow down our limits to the best solution, we have better results. All the limits are narrow enough for one simplex except for profile angle. Profile angle spans 90 degree which is more for one simplex, thus profile angle value is divided in M parts each is optimized by one simplex, resulting in M number of simplex executions. As the number of simplex executions increases as the system gets more sluggish. Therefore M is to be chosen carefully to achieve a good trade-off between accuracy and time.

3.3 Pose Estimation

In our application pose is estimated by rendering 2.5D AAM by initializing parameters and projecting on an unaligned face image. Main objective is to calculate the values of six pose parameters. Limits has been defined on these parameters so as to avoid the occlusion of the face. Steps of the pose estimation are as follow:

1. First of all test image is loaded, along with the location of center of gravity (COG) of the unknown face.
2. 2.5D mean and frontal view AAM is build with zero appearance parameters $c = 0$ and zero pose parameters $P = 0$ for the initialization. COG values are used to initialize only translational parameters.
3. Pose is estimated considering limits as tabulated in table 1.
4. Profile angle spans 90 degrees (left semi profile to right semi profile) which is more for one simplex execution to search a face.
5. We divide these 90 degrees in M parts, each of $\frac{90}{M}$ degrees, in return we have M simplex executions instead of one.
6. Best pose parameters of each M simplex executions are stored.
7. Best value among these M simplex executions is selected to estimate the pose.

4 Experiments

4.1 Selection of Similarity Metrics

Before conducting the experiments it is necessary to observe the behavior of the error obtained while estimating the pose. Three types of error behavior is

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Table 1. Limits of pose parameters

Pose Parameters	Min. Value	Max. Value
Y axis rotation	-45 degree	45 degree
X axis rotation	-5 degree	5 degree
Z axis rotation	-5 degree	5 degree
X axis translation	-20 pixels	20 pixels
Y axis translation	-10 pixels	10 pixels
Scale	-0.05	0.05

logged while rotating the model on frontal view of the face image of M2VTS [10] database. Median value is taken upon performing the above error calculations 20 times for each similarity metrics. Medians of these 20 iterations of each error are calculated followed by their normalization between 0 and 1 to be able to compare with each other. Figure 5 shows the results corresponding to errors of CC, EUC and MI. In the context of our pose estimation application we observed that CC and EUC behavior is almost the same while MI curve has more noise near the global minimum. Moreover it is more complex and slow for our application. We choose EUC method for our experiments since it is less complex as well as fast as compared to CC.

4.2 Best value of M

In our application we divide profile angle in M parts, resulting in M simplex executions. More the simplex executions more the result would be accurate but at the same time system becomes time consuming. To analyze the behavior of M, we should have exact value of difference between estimated and actual profile angles. In this case only synthetic image can provide us the exact value of the profile angles. Therefore it is tracked by varying M from 1 to 9 and mean of the error between actual and estimated profile angle is calculated for each value of M. The results shown in figure 6 eventually enlighten the behavior of error with respect to M. After analyzing we choose M equal to 5 which gives 3.5 degrees of mean error of profile angle.

4.3 Results

The proposed algorithm (2.5D AAM, Simplex and EUC as a similarity metrics) was tested on three database images M2VTS, synthetic image and webcam images. Ten experiments were performed on each database image in order to obtain the median values of results. Performing median of these values serves as a low pass filter, which rejects divergent results caused by inappropriate initializations of simplex.

M2VTS Images. Figure 7 shows the result of pose estimation of M2VTS image with mean 2.5D AAM. All of the images from left half to frontal and to

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

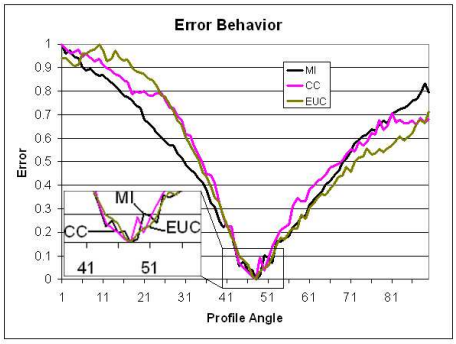


Fig. 5. Comparison of EUC, CC and MI

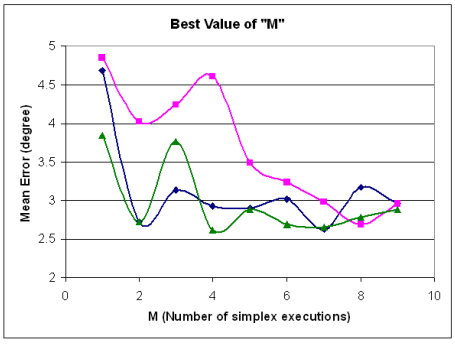


Fig. 6. Behavior of M

right half profile are tracked and their respective profile angles are logged. It shows that pose is estimated correctly.

Synthetic Images. Figure 8 shows the result of pose estimation of synthetic image with 2.5D AAM. In the case of synthetic image we can have the exact value of the profile angle, therefore we can compare pose estimation results with the exact values of profile angle. Linear curve shows actual profile angle values.

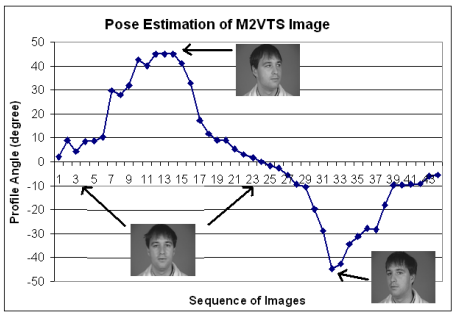


Fig. 7. Pose estimation of M2VTS image

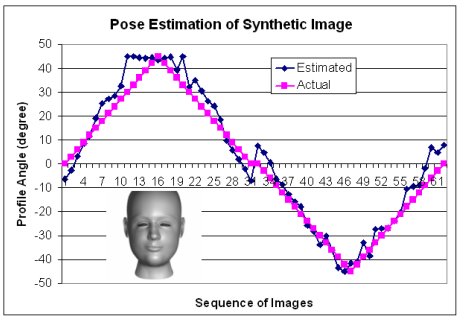


Fig. 8. Pose estimation of synthetic image

Webcam Images. Figure 9 shows the curves of pose estimation of two webcam images with mean 2.5D AAM. In fact the purpose of introducing these images is to proof the generalization of 2.5D AAM. The fabrication of 2.5D AAM do not involve these images therefore they are unknown to the system. Although the curves are noisy but still the pose is well estimated.

5 Conclusions

We have presented Active Appearance Model in 2.5 dimensions. 2.5D AAM is constructed by the combination of 3D landmarks taken from frontal and profile

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

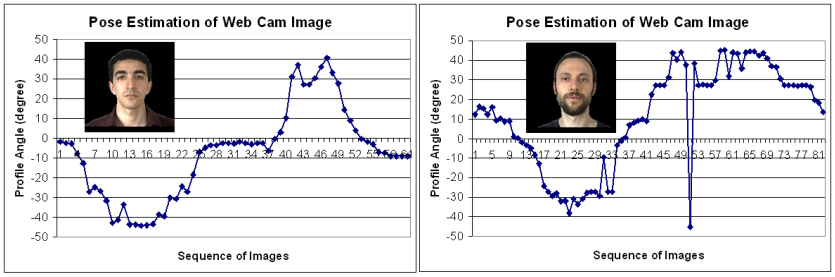


Fig. 9. Pose estimation of webcam images

view and 2D texture of only frontal view. Pose is estimated by simplex method, implemented in such a way, to reduce the requirements of high memory and time. Euclidean Distance (EUC) is used as a similarity metrics. Several experiments have been performed on different databases to verify robustness and rapidness for face alignment. Only one texture of frontal view and 68 landmarks makes the 2.5D AAM very light in memory consumption. Our proposition of 2.5D AAM and simplex is well adapted for embedded systems as it is not only less memory consumer but also saves time to transfer the data between processor and memory.

References

1. T. F. Cootes, G. J. Edwards, and C. J. Taylors, Active Appearance Models. *Proceedings of ECCV'98, European Conference on Computer Vision*, Vol.2, p.484-498, Freiburg, Germany (1998).
2. T. F. Cootes and C. J. Taylor, Statistical Models of Appearance for Computer Vision, *Technical report from Imaging Science and Biomedical Engineering*, University of Manchester, www.isbe.man.ac.uk (2004).
3. Fadi Dornaika and Jörgen Ahlberg, Fast and Reliable Active Appearance Model Search for 3D Face Tracking, *Proceedings of Mirage*, INRIA Rocquencourt, France, (2003).
4. Iain Matthews and Simon Baker, Active Appearance Models Revisited, *International Journal of Computer Vision*, (2004), Kluwer Academic Publishers.
5. S. Baker and I. Matthews, Equivalence and Efficiency of Image Alignment Algorithms, *IEEE Conference on Computer Vision and Pattern Recognition*, (2001).
6. James Paterson and Andrew Fitzgibbon, 3D head tracking using non-linear optimization, *British Machine Vision Conference*, (2003).
7. C. Goodall, Procrustes methods in the statistical analysis of shape, *Journal Royal Statistical Society*, Series B, 53:285-339, (1991).
8. Mikkel Bille Stegmann, Active Appearance Models: Theory, Extensions and Cases. *Master thesis from Informatics and Mathematical Modeling*, Technical University of Denmark, DTU, (2000).
9. J. A Nelder and R. Mead, A simplex method for function minimization, *In Computer Journal*, vol. 7, pp. 308-313, (1965).
10. S. Pigeon, M2VTS, Multi Modal Verification for Teleservices and Security applications, *http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html*, (1996).

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de