Using Kinect for Facial Expression Recognition under Varying Poses and Illumination

Filip Malawski[1] and Bogdan Kwolek[1] and Shinji Sako[2]

[1] AGH University of Science and Technology, 30-059 Krakow, Poland
{fmal,bkw}@agh.edu.pl
[2] Nagoya Institute of Technology, Japan
sako@mmsp.nitech.ac.jp

Abstract. Emotions analysis and recognition by the smartphones with front cameras is a relatively new concept. In this paper we present an algorithm that uses a low resolution 3D sensor for facial expression recognition. The 3D head pose as well as 3D location of the fiducial points are determined using Face Tracking SDK. Tens of the features are automatically selected from a pool determined by all possible line segments between such facial landmarks. We compared correctly classified ratios using features selected by AdaBoost, Lasso and histogram-based algorithms. We compared the classification accuracies obtained both on 3D maps and RGB images. Our results justify the feasibility of low accuracy 3D sensing devices for facial emotion recognition.

Keywords: Facial Image Analysis; Depth Maps Analysis.

1 Introduction

The use of emotion recognition in the age of smartphones with front cameras is a new concept. Through the specialized software the camera can record and then transmit the user facial expressions to a remotely located analysis center. The analysis center equipped with emotion recognition software will be able to perform analysis and recognition of user emotions. What it means is that as people see a news item, or watch a TV show or see an advertisement, it will be possible to know how they are feeling or seeing the current media coverage. Through visual analysis of the face articulations the future technology will be able to decipher the user's facial expressions and to tell whether he/she is happy, sad, angry, tense, relaxed, or depressed. In consequence, it will be possible to infer about the relevance of message or information selection for a specific user.

Emotion recognition offers a new direction to media analysis as well as human machine communication. It will allow finding out what the customers are truly feeling about the delivered messages, and not what they say about their feelings. This is because many research studies have shown that people may hide their true emotions during surveys or filling the questionnaires. Emotion recognition technology should not depend upon what the customers say they are feeling, but it should capture their facial expressions to find out their true emotions.

As media analysis progresses further, in the near future it might be possible to analyze the true emotions that the targeted news and movies are generating.

The potential applications of facial expression recognition (FER) concern not only media technology but also include service robotics, virtual reality, games etc. Most of the previous work focused primarily on 2D domain, see survey [1]. 2D domain-based facial expression classification algorithms have demonstrated remarkable performance in controlled conditions, particularly in constant lighting conditions and with small head pose variations. On the other hand, the 3D data based approaches are invariant to changes mentioned above and therefore current research focuses on 3D modalities [2]. One of the most popular methods for 3D FER is based on the distances between certain facial landmarks and their changes that occur during facial articulations. The focus on such approaches is because the facial geometry is invariant to illumination and imaging conditions.

As demonstrated in another survey [3], a considerable attention has been drawn on 3D FER after publication of the BU-3DFE dataset. The discussed dataset is in fact a testbed for benchmarking the 3D approaches to FER. It was captured using 3DMD setup, and consists of 3D models with 20000 to 30000 polygons depending on the face size. As mentioned in [2], all publicly available databases for 3D-based FER were recorded in controlled conditions with the use of similar setups, i.e. using devices that allow data recordings with high precision and accuracy, and with very low noise level. In this context it is worth mentioning that most databases were recorded using 3D acquisition systems, which are based on structured light technologies, such as the Minolta Vivid 900/910 series. For instance, [4] presents a FER system, which is capable of operating at several frames per second using data acquired from a precise 3D scanner.

Automatic 3D FER recognition from image sequences of low resolution or video quality that is offered by current consumer cameras, smartphone cameras or service robot cameras is very challenging research problem, with many potential applications in media technology. However, little work has been done in the area of 3D face analysis using 3D consumer cameras. One exception is work by Li et al. [5], who recently demonstrated that on the basis of RGBD images acquired by the Kinect sensor it is possible to achieve high face recognition rates. To the best of our knowledge, no significant work has been done in the area of facial articulations analysis using RGBD images delivered by currently available low cost 3D sensors, which are now utilized or will be utilized in modern game consoles, smart TV, service robots, etc., or even in future smartphones.

Affective computing is the study of systems and devices that can recognize, interpret, process, and simulate human affects. A motivation for such research is desire of simulating and utilizing empathy. In general, the machine assisting humans in daily activities should interpret their emotional state and adapt its behavior to them, giving an appropriate response considering context and emotions. An example of utilization of emotions in context of robotics is robot Kismet, which has been developed in MIT [6]. The robot Nao is a recently developed humanoid robot, which is equipped with two cameras. Among others, the robot is capable of connecting with the Internet and searching the requested

content, for instance latest news. Our aim is to equip this service robot with ability of analysis of facial expressions during presentation to the user the messages or news, which have been found in Internet in response to he/she requests.

2 Face detection and tracking

Active Shape Models (ASMs) are statistical models of the shape of objects, which iteratively deform to fit to an example of the object in a new image [7]. The shape of an object is represented by a set of points. The ASM model is trained using contours (surfaces in 3D) in training images. It finds the main variations in the training data using Principal Component Analysis, which enables to automatically decide if a contour is a good object contour. ASMs are frequently used to analyze 2D and 3D images of faces. In face tracking application the user should collect training images and then represent all shapes with a set of landmarks in order to form a Point Distribution Model (PDM). The ASM works by alternating two stages, where in the first one it generates a suggested shape by looking in the image around each point for a better position for the point, whereas in the second one it conforms the suggested shape to the PDM. Figure 1a depicts locations of 116 facial points, which were determined by ASM.

Depth is very useful cue to attain reliable face detection and tracking since face may not have consistent color and texture but has to occupy an integrated region in space. Kinect sensor provides both color and dense depth images. It combines structured light with depth from focus and depth from stereo. The sensor consists of infrared laser-based IR emitter, an infrared camera and a RGB camera. The IR camera and the IR projector compose a stereo pair with a baseline of approximately 75 mm. A known pattern of dots is projected from the IR laser emitter. Since there is a distance between laser and sensor, the images correspond to different camera positions, and this in turn allows us to use stereo triangulation to calculate each spec depth. It captures the depth and color images simultaneously at a frame rate of about 30 fps. The RGB stream has size 640×480 and 8-bit for each channel, whereas the depth stream is 640×480 resolution and with 11-bit depth. The field of view is 57° horizontally and 43° vertically, the minimum measurement range is about 0.6 m, whereas the maximum range is somewhere between 4-5 m.

Together with the sensor it is delivered Kinect for Windows SDK and the Face Tracking SDK, which enable developing applications capable of tracking human faces in real time. The face tracking engine determines 3D positions of semantic facial feature points as well as 3D head pose. It tracks the 3D location of 121 points, which are depicted on Fig. 1b. Additionally, the Face Tracking SDK fits a 3D mask to the face. The 3D model is based on the Candide 3 model [8], which is a parameterized 3D face mesh specifically developed for model-based coding of human faces. This 3D model is widely used in head pose tracking [9]. As already indicated, Kinect sensor allows low cost sensing with high capture speed. However, the 3D maps provided by Kinect are very noisy and have relatively low resolution in comparison to typical devices utilized in

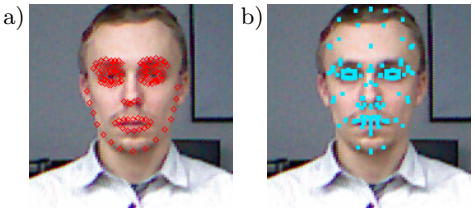

Fig. 1. Locating facial features, a) on gray images using ASM, b) on depth maps using Kinect Face SDK.

facial expression recognition. In consequence, many important fiducial points such as eye and mouth corners are not too precisely locatable. Even more, some fiducial markers undergo occlusion, particularly the points that are located close to the nose. To the best of our knowledge, there exists only one work [5] that was published recently, in which noisy images acquired by Kinect have been used for face analysis.

3 Automatic Feature Selection

The most popular method in static image-based FER consists in using characteristics distances between certain facial landmarks as well as their changes that occur due to face articulations. For instance, BU-3DFE dataset provides the location of 83 facial points, which together with their distances are widely used in static facial analysis [2]. In [10,11] the classification was done using the distances among all pairs of the available features. An average expression recognition rate was equal to 93.7% and 83.5%, respectively. In a method discussed in [12] six characteristic distances extracted from the distribution of 11 facial feature points from the available points resulted in an average recognition rate of 91.3%. Tang and Huang [13] proposed an automatic feature selection method that is based on maximizing the average relative entropy of the marginalized class-conditional distributions of the features. Tens of the features are automatically selected from a pool determined by all possible line segments between the 83 landmarks. In another work [14] they selected a pool of 96 discriminative features including not only the normalized distances, but also additionally the slopes of line segments connecting a subset of 83 landmarks. In general, the discussed methods rely on features extracted from the locations of facial points provided by 3D databases. In such approaches, typically, a small subset of the selected subset features gives relatively good classification accuracy.

The Kinect sensor provides both depth and color images, and in order to perform a classification of facial expressions the fiducial features should be extracted first. In our approach the location of all facial features was determined using the methods discussed in Section 2. Because of noisy depth maps, all points were subjected to automatic feature selection. In contrast to methods relying on high quality measurements, in the classification we employ relatively large number of

the features. In 3D FER we employ the slopes of line segments connecting 121 points. Thus, the total number of all features is $n_i = 7260$. The slopes were chosen since they give slightly better results in comparison to the distances between the features. Given the estimated pose, the head together with the determined face points were rotated to the canonical frontal pose. The size of each face was then scaled accordingly to its width and the distance to the camera. The faces were captured using Kinect for Windows and Kinect for Xbox 360. The Kinect for Windows supports near mode, which enables the camera to see objects as close as 60 centimeters in front of the device without losing accuracy or precision. For Kinect for Xbox 360 the minimal distance of the object to the sensor is about 1 m. Thus, the faces were captured in two sessions from the distance of about 60 cm and 1 m to the cameras. A typical size of the face on the depth map acquired in such a way was about 280×280 pixels for Kinect for Windows, and 160×160 pixels for Xbox Kinect.

3.1 Histogram-based feature selection

Given a set of training depth images for each considered facial expression, we calculate for each facial point a pool of histograms. A histogram reflects a distribution of the slopes between a given facial point and one of the remaining points within a specific class. This means, that for a given line connecting two facial points the histogram bins are incremented using the training data of a given class. Each histogram consists of 20 bins and is normalized prior the feature selection. For each pair of the emotions we calculate the distances between the corresponding histograms. The distances are then sorted. The larger the distances between corresponding histograms representing a pair of facial expressions, the more discriminative is the equivalent feature. Given such sorted lists of the distances (divergences or ratios) between histograms we choose the assumed number of features together with their corresponding histograms. Such features are then utilized in the classification. It is worth mentioning that in the classification stage we considered also optionally such a pool of the features, additionally extended about their symmetric counterparts. The histograms were compared using:

- histogram intersection
- Kullback-Leibler divergence
- Earth Mover's Distance (EMD)
- Fisher's ratio

Since none of the mentioned above method achieved superior results, we decided to use Kullback-Leibler divergence in comparison of the histograms.

3.2 Feature selection with AdaBoost

The idea of boosting is to select and then combine several classifiers, which are often referred to as weak learners into a more powerful one using a voting

procedure. AdaBoost is a supervised algorithm and it learns such strong classifier by selecting only those individual features that can best discriminate among classes [15]. Although AdaBoost was developed as a method to improve the classification accuracy by combining such weak learners, it has also been utilized for feature selection in detection and classification tasks [16]. During training, incorrectly classified training samples are weighted more to redirect focus of the training on them by subsequent weak learners.

In our boosting-based algorithm for feature selection, the finite set of features is considered as the space of the weak learners. The base learner is the decision stump, a one-decision two-leaf decision tree. This means that each feature is considered as a boolean predictor. The learning of the decision stump means selecting a feature and a threshold. Thus, the training of a weak learner simply consists of selecting the one with the minimum error rate. The input of the algorithm is a set of the training examples $\{I_i, i = 1, \ldots, N\}$ and their associated labels $\{l_i, i = 1, \ldots, N\}$, where N is the number of the training examples, and $l_i \in \{0, 1\}$. Each training example is represented by the initial feature set $\{x_i^{(j)}, j = 1, \ldots, n_i\}$. At the beginning, the AdaBoost initializes the weights of the training samples w_i to $\frac{1}{2N_p}$, $\frac{1}{2N_n}$, where N_p and N_n stand for the number of the positive and negative examples, respectively. Afterwards, in each round it selects one feature as weak classifier, the feature that achieves the highest score with respect to the actual weight, and updates the weights of the training examples. It decreases the weights of the examples that were correctly classified by an optimal classifier for the selected feature. In consequence, in the next iteration the classifiers will focus on the examples that were misclassified. Each selected feature forms a weak classifier h_k that is parameterized by a threshold θ_k and output label u_k. The goal is to select T features, which have the best ability to discriminate the considered samples into the desired classes. The error of the classifier is calculated over the entire examples set as a weighted sum of the absolute differences between the output of the threshold-based binary classifier $h(x_i, \theta_i, u_i)$ and the class label l_i in the following manner: $\varepsilon(x, \theta, u, w) = \sum_{i=1}^{N} w_i \left| h(x_i, \theta_i, u_i) - l_i \right|$. Given the feature $x_i^{(j)}$ and the corresponding w_i the error of the classifier depends solely on threshold θ. The AdaBoost selects a feature with the optimal threshold θ^*, which is calculated as follows $\theta^* = \operatorname{argmin}_\theta \varepsilon(x, \theta, u, w)$. The whole feature selection procedure is shown as Algorithm 1.

3.3 Feature selection via the Lasso

The least absolute shrinkage and selection operator, which is known as Lasso, permits computationally efficient feature selection based on linear dependency between input features and output values. It is an L1 penalized regression technique introduced by Tibshirani [17]. Lasso commonly gives sparse solutions due to the L1 penalty so it is an alternative to model or subset selection. Any features that have non-zero regression coefficients can be seen as selected by the

Algorithm 1 AdaBoost algorithm for feature selection

INPUT:
- Training data: $\mathcal{D} = \{(x_i^{(j)}, l_i), i = 1, \ldots, N, j = 1, \ldots, n_i\}$, $x_i^{(j)} \in \mathbf{R}$, $l_i \in \{0, 1\}$
- The initial feature set: $\mathcal{F}$
- Weak learner: $\mathcal{L}$ that learns binary classifier $h(x) : \mathbf{R} \mapsto \{0, 1\}$
- The desired number of features: T

OUTPUT:
- The sequence of selected features: $\{\hat{x}^{(1)}, \hat{x}^{(2)}, \ldots, \hat{x}^{(T)}\}$

Algorithm

Initialize the distribution $\mathcal{D}$: $w_i = \frac{1}{2N_p}, \frac{1}{2N_n}, i = 1, \ldots, N$

$\mathcal{F}' = \mathcal{F}$

for $t = 1$ to T **do**

  Normalize the weights w_i of the examples

  Select a pool of classifiers $H^{(t)}(x)$ from $\mathcal{F}'$

  Train binary classifiers $H^{(t)}(x)$

  Select classifier $h^{(t)}(x)$ having highest evaluation score with respect to weights

  Remove feature $\hat{x}^{(t)}$ corresponding to selected classifier $h^{(t)}(x)$ from $\mathcal{F}'$: $\mathcal{F}' = \mathcal{F}' \backslash \hat{x}^{(t)}$

  Compute the error rates $\varepsilon^{(t)} = \sum_{i=1}^{N} w_i^{(t)} \left| (h^{(t)}(x_i) - l_i) \right|$

  Update the weight as $w^{(t)} = \begin{cases} w^{(t)} \frac{\varepsilon^{(t)}}{1-\varepsilon^{(t)}} & \text{if} \quad h^{(t)}(x_i) = l_i \\ w^{(t)} & \text{otherwise} \end{cases}$

end for

Lasso algorithm. Lasso solves the following regularized optimization problem:

$$min_\beta h(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1, \quad \text{where } \lambda \geq 0 \tag{1}$$

and β is a $p \times 1$ vector, y is a $n \times 1$ vector, and X is a $n \times p$ matrix. The penalty term in (1) is a L1-norm penalty or simply the sum of the absolute values of the components of β. This penalty term encourages sparsity in the components of the solution vector and thus automatically leads to feature selection. Additionally, the penalty term regularizes the solution vector β and hence prevents overfitting. If $p > n$, the Lasso selects at most n variables and the number of selected features is bounded by the number of samples.

4 Experiments

We conducted extensive experiments to evaluate the usefulness of depth map acquired by Kinect for facial expressions recognition. The data for the evaluation of the proposed algorithm were recorded in two sessions using Kinect for Xbox 360 and Kinect for Windows. Each dataset consists of 2520 images of 10 individuals with variations in pose and illumination. The images in the datasets contain three basic facial expressions: normal, smile and anger. Each dataset consists of 90 images in frontal pose and normal illumination (30 images for

each expression), 30 images in frontal pose and dark face (10 images for each expression), 66 images in normal illumination and various poses, and 66 images in non-frontal poses and poor illumination. Figure 2 illustrates some example RGB images that were acquired in the considered illumination conditions using Kinect for Xbox 360. The first three images contain expressions shot in normal conditions, whereas the next ones contain the expressions in poor illumination.

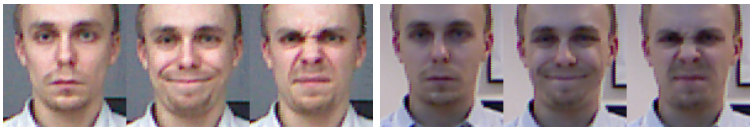

Fig. 2. Facial expressions in two different illumination conditions.

Figure 3 depicts facial expressions in the considered head poses. As we can observe, the change in the head poses is quite considerable. The Face Tracking SDK estimates the user's head pose and returns tree angles: pitch, roll, and yaw, which describe its orientation. Using such angles the head is rotated to the canonical pose as well as is scaled according to its distance to the camera.

Fig. 3. Facial expressions in various poses.

Table 1 shows the correctly classified ratio that was obtained in 10-fold validation on datasets acquired by Kinect for Xbox 360. The features were extracted using histograms with Kullback-Leibler (KL) divergence, AdaBoost (AB) and sparse (SP). The correctly classified ratio was determined using Naïve Bayes (NB), random forests (RF) and Support Vector Machine (SVM). In nonlinear SVM-based classification, the most important parameter is the soft-margin constant c, which controls the trade-off between complexity of decision rule and frequency of error. A smaller value of c allows to ignore points close to the boundary, and increases the margin. In nonlinear SVM, the (Gaussian) radial basis function kernel, or RBF kernel, is a popular kernel function, which usually gives good results. The parameter σ controls how quickly an increased distance causes the value of the kernel to fall toward zero. By the use of cross-validation

and grid-search on c and σ parameters [18], the best prediction results were obtained by SVM with $c = 1$ and $\sigma = 0.01$.

Table 1. Correctly classified ratio [%] using 3D data acquired by Kinect for Xbox 360.

	NB			RF			SVM		
feature sel.	HI	AB	SP	HI	AB	SP	HI	AB	SP
# features	100	800	100	400	200	400	200	400	100
normal	72.8	70.4	73.0	76.4	79.2	78.6	80.6	81.6	72.8
dark	74.8	75.2	78.0	75.0	78.6	75.6	77.8	80.0	77.0
pose	62.2	66.0	71.4	68.8	66.0	66.4	66.6	69.2	69.2
pose dark	62.6	63.4	69.0	70.6	69.0	71.5	66.0	67.8	64.8
average	68.1	67.8	72.9	71.9	73.2	73.0	72.8	74.7	71.0

Using the discussed classifiers we evaluated the classification accuracy of facial expressions for different numbers of the selected features. The best number of the selected features is shown in the third row of Tab. 1. The classification performance was evaluated for normal head pose and normal illumination conditions, dark face, non-frontal pose, and non-frontal pose shot in poor illumination, see subsequent rows in Tab. 1. The average correctly classified ratio is shown in the last row of the table. As we can see, the best classification accuracy was achieved for normal pose and normal illumination conditions via Support Vector Machine classifier, which has been trained on features extracted by AdaBoost. The classification accuracy is equal to 81.6%. Slightly worse correctly classification ratio was obtained in the case of poor lighting (dark). A considerable decrease of the classification performance can be observed for non-frontal head poses. For features selected automatically without extending them by symmetric counterparts the classification accuracy is slightly worse.

Table 2 presents the correctly classified ratio that has been achieved using 3D data acquired by Kinect for Windows. As we can observe, owing to the near mode of the sensor, the results are far better. The SVM-based classification was performed using the same parameters, i.e. $c = 1$ and $\sigma = 0.01$. As we can see, for the discussed sensor the best result was obtained by SVM operating on features selected by sparse-based feature selection. As it was shown in Tab.1, the best result for Xbox Kinect has also been achieved by SVM on the features selected by AdaBoost algorithm. The discussed results demonstrate that the number of features required to achieve favorable classification accuracy is quite high.

Since the SVM gave the best result in the terms of correctly classified ratio we conducted grid-searching on c and σ parameters to achieve even better performance. We found that the best correctly classified ratio is achieved for $c = 100$ and $\sigma = 0.01$, see results shown in Tab. 3. As we can notice, for normal head pose and illumination conditions the best CCR is achieved on features selected by AdaBoost and it is equal to 87%, whereas for normal head pose and

Table 2. Correctly classified ratio [%] on 3D data acquired by Kinect for Windows.

	NB			RF			SVM		
feature sel.	HI	AB	SP	HI	AB	SP	HI	AB	SP
# features	400	800	800	100	200	800	400	400	100
normal	86.2	80.0	85.0	83.2	82.2	83.8	83.2	84.0	85.2
dark	85.8	83.0	86.0	85.2	84.0	81.4	84.6	82.8	87.0
pose	73.4	65.8	71.8	72.8	70.8	72.2	72.4	70.2	75.0
pose dark	74.6	69.0	74.2	73.6	70.4	70.0	73.2	69.6	73.0
average	**80.0**	74.5	79.3	**78.7**	76.9	76.9	78.4	76.7	**80.1**

poor illumination the best CCR is achieved on features selected on the basis of histogram and KL divergence, and it is equal to 85%. The best, averaged CCR is greater than 80%. A recognition rate of 80% using only noisy depth data provided by Kinect is a good starting point for further research in this area. However, a remarkable decrease of the classification performance can be observed for non-frontal head poses. On the other hand, 3D data provided by Kinect are precise enough to cope with the non-frontal face poses and therefore our future work will focus more on this issue.

Table 3. The best correctly classified ratio [%] on 3D data.

	SVM		
feature sel.	Hist.	AdaBoost	Sparse
# features	100	100	100
normal	85.8	87.0	84.2
dark	85.0	84.8	81.4
pose	74.4	74.0	70.2
pose dark	76.6	76.6	70.4
average	80.5	**80.6**	76.6

Figure 4 depicts the selected features, which gave the best correctly classified ratios for NB, RF, SVM with $c = 1$ and $\sigma = 0.01$, and SVM with $c = 100$ and $\sigma = 0.01$. The figures correspond to the best average results, which were achieved by each of the considered classifier. In Tab. 2 and 3 the best results of each classifier were typeset in bold. As we can see, the selection is done accordingly with our intuition. In particular, the selected pairs of the features and their corresponding lines concern the points that during facial articulations undergo significant misalignments.

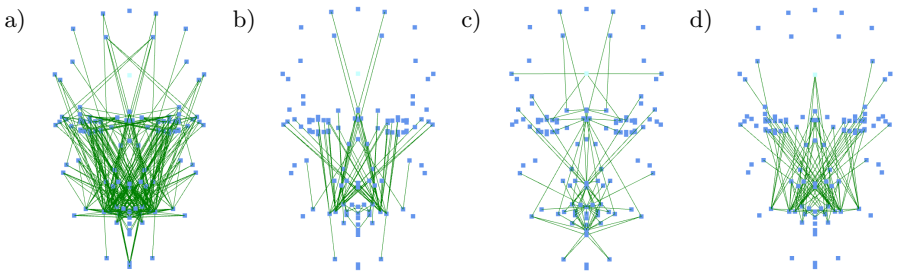

Fig. 4. The selected features, which gave the best CCR for the evaluated classifiers: a) NB, b) RF, c) SVM with $c = 1$ and $\sigma = 0.01$, d) SVM with $c = 100$ and $\sigma = 0.01$.

The depth-based facial expressions classification was compared with classification using RGB images acquired by Kinect. Using the angles between the fiducial points, which were determined by ASM and then selected by AdaBoost, the classification accuracy was about 10% worse for the normal face pose and normal illumination conditions. A small decrease in efficiency was observed for dark pose, whereas the decrease in efficiency for non-frontal pose was about 25%. For non-frontal poses and normal illumination quite similar classification accuracy was obtained via matching of SIFT descriptors. However, in case of illumination change the classification accuracy was worse in comparison to accuracy obtained by ASM.

The complete FER system was implemented in C++/C#. The recognition performance was evaluated using WEKA software. The system runs in real-time on an ordinary PC with Intel Core i5 2.5 GHz CPU. Computing the 3D positions of semantic facial feature points as well as 3D head pose by Face Tracking SDK is the most computationally expensive part of our algorithm. On ordinary PC/laptop computer the processing time of single frame is 40-50 ms. The classification time is far shorter. For instance, the SVM operating on 100 features takes 0.2 ms, the RF operation on the same number of features requires 0.1 ms, whereas NB operating on 400 features takes 1.7 ms.

5 Conclusions

We have proposed an approach for facial expression recognition using only depth information provided by consumer level 3D image sensor. The best recognition accuracy is equal to 87% and it was obtained using AdaBoost-based feature selection and SVM. In particular, we demonstrated that 3D information is very useful in case of non-frontal head poses. The results suggests that using low-cost 3D sensors a promising recognition accuracy of facial expressions can be obtained even in situation of poor lighting conditions and considerable pose variations. Although depth maps provided by low cost 3D sensors like Kinect are very noisy, they still might be useful for facial expression recognition, particularly in case of non-frontal head poses.

Acknowledgments. A part of this study was supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research) Grant Number 25350666, FY 2014 Researcher Exchange Program between JSPS and PAN, and by the National Science Center (NCN) within the research project N N516 483240.

References

1. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: The state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12) (2000) 1424–1445
2. Sandbach, G., Zafeiriou, S., Pantic, M., Yin, L.: Static and dynamic 3d facial expression recognition: A comprehensive survey. Image Vision Comput. **30**(10) (October 2012) 683–697
3. Fang, T., Zhao, X., Ocegueda, O., Shah, S., Kakadiaris, I.A.: 3D facial expression recognition: A perspective on promises and challenges. In: FG. (2011) 603–610
4. Tsalakanidou, F., Malassiotis, S.: Real-time 2d+3d facial action and expression recognition. Pattern Recogn. **43**(5) (May 2010) 1763–1775
5. Li, B., Mian, A., Liu, W., Krishna, A.: Using Kinect for face recognition under varying poses, expressions, illumination and disguise. In: WACV. (2013) 186–192
6. Brooks, R.A., Breazeal, C., Marjanović, M., Scassellati, B., Williamson, M.M.: The cog project: Building a humanoid robot. In: Computation for Metaphors, Analogy, and Agents. Volume 1562 of LNCS. Springer (1999) 52–87
7. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - their training and application. Comput. Vis. Image Underst. **61**(1) (January 1995) 38–59
8. Ahlberg, J.: Candide-3 - an updated parameterised face. Technical report, Dept. of Electrical Engineering, Linkping University, Sweden (2001)
9. Kwolek, B.: Model based facial pose tracking using a particle filter. In: Int. Conf. on Geometric Modeling and Imaging: New Trends, IEEE Comp. Soc. (2006) 203–208
10. Soyel, H., Demirel, H.: Optimal feature selection for 3D facial expression recognition with geometrically localized facial features. In: Int. Conf. on Soft Computing, Comp. with Words and Perceptions in Syst. Anal., Dec. and Control. (2009) 1–4
11. Sha, T., Song, M., Bu, J., Chen, C., Tao, D.: Feature level analysis for 3D facial expression recognition. Neurocomputing **74**(12-13) (June 2011) 2135–2141
12. Soyel, H., Demirel, H.: Facial expression recognition using 3d facial feature distances. Volume 4633 of Lecture Notes in Computer Science. (2007) 831–838
13. Tang, H., Huang, T.: 3D facial expression recognition based on automatically selected features. In: Conf. CVPR. (2008) 1–8
14. Tang, H., Huang, T.: 3D facial expression recognition based on properties of line segments connecting facial feature points. In: Conf. FG. (2008) 1–6
15. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proc. of EuroCOLT, Springer (1995) 23–37
16. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision **57**(2) (May 2004) 137–154
17. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B **58**(1) (1996) 267–288
18. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan (2010)