DOI: 10.5604/01.3001.0009.5442

Acquisition of databases for facial analysis

Filip Malawski
AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunication
Department of Computer Science
al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: fmal@agh.edu.pl

This article describes guidelines and recommendations for acquisition of databases for facial analysis. New devices and methods for both face recognition and facial expression recognition are constantly developed. In order to evaluate these devices and methods, dedicated datasets are recorded. Acquisition of a database for facial analysis is not an easy task and requires taking into account multiple issues. Based on our experience with recording databases for facial expression recognition, we provide guidelines regarding the acquisition process. Multiple aspects of such process are discussed in this work, namely selection of sensors and data streams, design and structure of the database, technical aspects, acquisition conditions and design of the user interface. Recommendations how to address these aspects are provided and justified. An acquisition software, designed according to these guidelines, is also discussed. The software was used for recording an extended version of our previous facial expression recognition database and proved to both ensure correct data and be convenient for the recorded subjects.

Keywords: facial expression recognition, face recognition, facial analysis, database acquisition

Introduction

Automatic facial analysis is currently a very active area of research. Face recognition, which allows to identify or verify a person based on the video or image data, has applications in many fields. Regarding security, it is employed for instance for surveillance in public spaces, such as airports, or checking people crossing the borders. Biometric applications include authorization of access to secure places or devices. Social media adopted face recognition mechanisms for tagging people in the photographs.

Facial expression recognition aims at enhancing human-computer interaction, by allowing us to communicate with machines in the most natural way - by expressing emotions. Applications of such systems include immersive video games or interaction with robots. Modern cameras are able to automatically take photos once they recognize a smile. Analysis of customers reactions to certain products or materials is a possible application in the marketing.

Both new methods and new devices are constantly developed, which allow to make the facial analysis more accurate and more robust. An important aspect of such research is the acquisition of proper data, needed for development and evaluation of new methods. Although several facial expression datasets exist, novel devices as well as more advanced usage scenarios lead to creation of new ones.

Acquisition of a dataset for facial analysis is not an easy task. Multiple aspects must be considered when designing the structure of the recorded data, acquisition conditions and the recording software. The acquisition process requires usually large amount of time and resources, therefore it is crucial to make it efficient, as well as to ensure that the dataset will be recorded correctly. Otherwise, the time and other resources required to fix errors may be significant, particularly if they are found later rather than sooner.

Based on our experience with recording a facial expression dataset, we propose a number of guidelines and recommendations, which we believe can prove useful for anyone attempting a similar task. An example of a recording software based on this recommendations is also discussed.

Background

Multiple facial analysis methods have been proposed in recent years. A survey on 2D and 3D face recognition is presented in [1]. Authors of [2] analyze infrared based approaches. In [3] methods for 3D face recognition under expression variations are discussed. For facial expression recognition authors of [4] enumerate approaches based on edges, textures, color, motion, deformable models and appearance. Both static and dynamic 3D facial expression recognition is discussed in [5]. It is worth mentioning, that most recent works often employ consumer level depth sensors, such as Kinect, both for face recognition [6] as well as facial expression recognition [7].

Methods proposed in both areas are often verified using some publicly available datasets [2, 5]. Popular 2D datasets include, for instance, Cohn-Kanade database [8] and JAFFE database [9]. More recently, a number of 3D datasets have been proposed. Texas database [10] provides 3D data, acquired with high-resolution stereo cameras, from 106 adults with 2 expressions. Bosphorus database [11] provides data from 104 adults, with 6 basic expressions and various occlusions, recorded with a high-quality 3D scanner. In [12] a database of high-resolution, spontaneous 3D dynamic facial expressions is presented. Despite such a variety of available resources new databases are constantly being developed [13, 14].

Aforementioned papers describe fully the structure of recorded data and in some cases discuss methods of evoking emotions. Often less details are provided regarding the recording process itself and interaction with the acquisition system. Issues specific to designing and recording a database are usually of less importance for researches, whose only goal is to use the particular database for development and evaluation of their methods. Nevertheless, as the devices and methods become more advanced, new challenges are addressed, hence new databases are needed. For instance, research on spontaneous facial expressions is still very limited, and further progress in this area is soon to be expected. Therefore, we believe that guidelines and recommendations for recording a dataset for facial analysis may prove useful for many researchers.

Methods

We formed the following guidelines based on our previous experience with recording a database for facial expression recognition. Feedback from the recorded subjects was an important factor. Certain lessons were also learned from some shortcomings of our database, which were identified during development of facial expression recognition methods. The first version of our database was used in research presented in [7] and an extended version is currently employed for further research.

Even though some of the discussed advices may seem obvious, our experience shows that often at least some of them are not taken into account when designing the database acquisition process. Hopefully, these can help to limit the time needed to record a proper database. The guidelines are grouped based on different aspects of the acquisition process. Relations between the aspects are presented in Figure 1. Database design influences the selection of sensors as well as defines required acquisition conditions. Sensors provide selected data streams. Apart from these aspects, acquisition software must also take into account the design of the user interface as well as some technical issues. Finally, a database with a given structure is obtained.

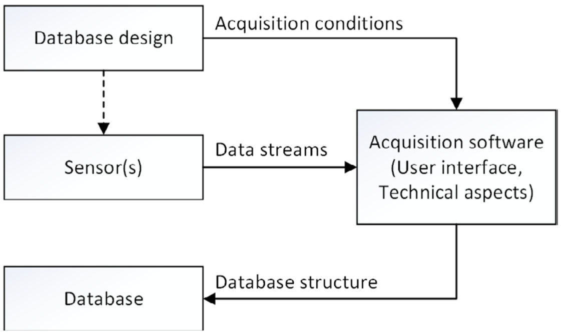


Figure 1. Aspects of the acquisition process and their relations

Sensors

One of the most important decisions in a database acquisition is the selection of a sensor. RGB cameras are available in almost all currently produced notebook devices, hence methods developed with such cameras are easily adapted to practical applications. On the other hand, RGB data is prone to errors related to the influence of lighting conditions, therefore depth sensors are gaining popularity, since they are robust to the illumination changes. High precision depth sensors, such as laser scanners, provide very accurate data, but are unable to work in real-time. For this reason, consumer depth sensors, such as the Kinect are recently applied—they provide relatively good quality and enable real-time processing. It is worth considering selecting a sensor, or even multiple sensors, which provide more than a single modality of data. Even if the primary goal of the research focuses only on one modality, additional data may yet prove to be useful.

Data streams

Employing multiple data streams, particularly with different modalities, has been proved to produce better results [15]. For instance, the Kinect sensor provides multiple data streams, namely RGB data, depth data and skeleton data, including position and rotation of the head. Additional library—Face Tracking SDK—provides positions of 121 facial landmarks based on the Candide-3 face model and the Active Appearance Models (AAM) [16] method. In the case of any sensor or combination of sensors which provide more than a single data stream it is crucial to record all available data, as it is often the case, that researched methods evolve and some information which seemed irrelevant in the beginning may be required later. For instance, it may seem sufficient to record only the selected face area and not the whole background, however, this would have precluded the possibility to use different face extraction or tracking methods and therefore limit the usefulness of the recorded database.

There are two other important aspects to consider, when multiple data streams are employed. Firstly, all of the data streams must be synchronized. In the case of a single sensor, such as the Kinect, the device usually provides synchronized data. In the case of multiple sensors, custom synchronization must be performed. Secondly, coordinates mapping must be recorded. In the case of the Kinect, the SDK provides mapping between depth data, RGB data and skeleton data, however, these computations should be performed during the acquisition process, since the SDK makes it difficult to apply them later. It is worth considering saving raw depth and color data as well as depth mapped to color and color mapped to depth. Also, the skeleton and landmarks positions should be recorded in all available coordinate spaces.

Last, but not least, an important decision is to choose between static (images) and dynamic (videos) data. Recording video data requires much more resources, in terms of storage space and computational power, nevertheless we recommend to consider it, even if the primary objective of the research is to develop methods for static images. By recording the video data the database becomes much more useful and should a decision be made to switch from static recognition methods to dynamic ones, the data will be already available.

Database design

There are multiple issues in the face recognition as well as facial expression recognition areas, which may be addressed only if proper data is available. Researchers often analyze the influence of different lighting conditions, view angles, and occlusions, as well as age, sex and ethnicity of the subjects. In the case of facial expressions, six basic expressions (anger, disgust, fear, joy, sadness, surprise) are usually recorded, and neutral one is often added as well. In some works more detailed expressions are considered. Also, a choice must be made between acted and spontaneous expressions, the latter being much more difficult to evoke.

It is important to choose which aspects to include in the database and to what degree. For instance, in the case of view angles some works consider only front view and side view, while others consider multiple directions with multiple degrees of change in the position. The more aspects are included the better, although more aspects require usually considerably more time to record. Therefore, a compromise must be found between the addressed issues and the time and resources available to record the database.

Database structure

The recorded database should be organized in a proper structure in order to facilitate the usage of the data. The structure and the naming convention usually contain important metadata, such as subject id or information regarding the aforementioned aspects, e.g. view angle or expression. It is convenient to create such structure and naming convention that would be easily readable for both computers and humans. In the case of computers it reduces the complexity of code required to load the database, in terms of humans it allows to browse the data manually, which is quite often useful. It is important to avoid redundancy in the naming, e.g. if a folder name includes subject id, the files inside the folder should not. Redundancy in the naming makes the database more difficult to browse manually as well as more complex to process in the code. Also, it can result in very long paths, which in some cases may be longer than maximum length allowed by the operating system.

Another important aspect is creating ids for the samples. While it may seem natural to simply use incremental numbers, these are actually not quite unique. A situation may occur, when a sample is deleted during validation of the correctness of the database and another sample with the same number is later recorded. Meanwhile, a copy of the deleted sample may be stored as backup and in the end two different files with the same name and the same location in the database structure may exist, which can lead to errors. Therefore, we recommend using a timestamp for each recorded sample, which is a truly unique id.

Technical aspects

When recording the database, particularly in the case of multiple data streams, several technical aspects must be considered. First of all, sufficient CPU resources must be provided in order to ensure proper performance. Operations such as synchronization, compression and coordinates mapping between the depth and the color data, can be computationally expensive, therefore a high-end machine may be required. Another limitation is the speed of the storage device. Typical magnetic drives may have insufficient write speed in order to save all the data streams in real-time. The data may be stored temporarily in the internal memory, although in the case of high resolution data it may be difficult to store more than a few seconds of the data streams, due to the large size of the data. A good solution is employing a solid state drive (SSD), which provides much better performance.

In terms of storage it is worth to remember, that sufficient storage space must be provided in order to avoid running out of space during a recording session. It's a good practice to keep track of the remaining free space. Also, the recorded data should be backed up as soon as possible, in order to prevent data loss in the case of a hardware failure.

Acquisition conditions

In order to ensure correct data, repeatable conditions must be provided. Subjects should be recorded in a dedicated setting, with properly positioned lights and in given distance from the sensor. In the case of recording with different view angles a manner of ensuring particular angles should be provided, e.g. by using an inertial sensor attached to the head of the subject. In the case of facial expressions, a proper manner of evoking expressions must be designed, e.g. by presenting to the subjects appropriate images. The acquisition software should monitor as many aspects of the recording session as possible and not allow to record improper data. For instance, in the case of depth cameras it is easy to ensure proper distance to the sensor. The general idea is that preventing errors in the recorded data requires much less effort than finding and fixing them afterwards.

Potential fatigue of the recorded subjects must be taken into account as well—too long recording sessions may result in poorer data, particularly in the case of facial expressions. One of the possible solutions is to ask the subjects to come in pairs, so they can switch during the recording, when the face muscles fatigue is too great. Naturally, it requires dividing the data acquisition into parts, which can be recorded independently.

User interface

While the user interface (UI) may seem to be a rather secondary aspect, it is in fact a very important one. Improper design of the UI may lead to an inefficient recording process or even incorrect data, if a recorded subject becomes frustrated with the software. The UI should be intuitive and understandable, although it is important to keep in mind, that these concepts are relative and can be perceived differently by different persons. Therefore it is a good practice to verify the UI design with multiple persons, before using it in the actual recording sessions.

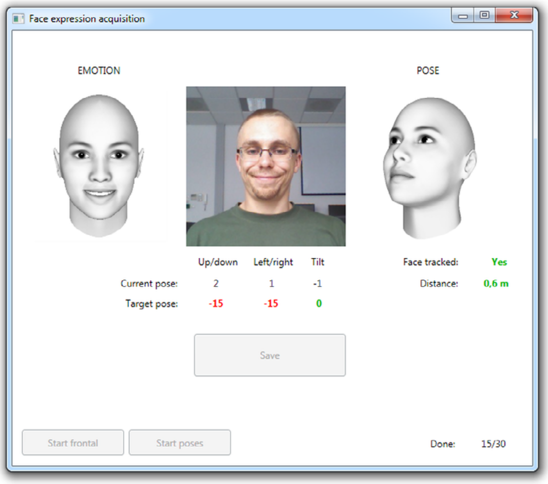
In general, the UI design should address the following issues. The recorded subject should always know exactly what is expected of him/her. For instance, when recording facial expressions the subject should know which expression to show and how—whether they should be intensive or not, spontaneous or posed, with open mouth or closed, etc. Visual guides are always helpful. Secondly, the user should get as much feedback as possible. Seeing a video feed of the recorded data is helpful, but often not sufficient. When different angles of the head pose are recorded and the system is measuring them, the values should be displayed. The subject should be also provided with a clear information how to move his/her head in order to achieve the expected angle. Another important issue to keep in mind, when designing the UI, is that a subject can observe only so many parts of the screen simultaneously. The gaze of the recorded subject is usually focused on one part of the UI, therefore all important information must be displayed in a close neighborhood. Otherwise, for instance, a subject focusing on different facial expressions may fail to notice that a change in the head pose is required. Finally, a good practice is to provide the UI with an option to delete the last recorded sample. Mistakes during the recordings happen from time to time and it is much more efficient to delete the incorrect data at once, rather than later browse all the recorded database for errors.

Acquisition software

In this section we discuss a sample acquisition software, designed according to the aforementioned recommendations. The software was used for recording of a facial expressions database, which extends the one presented in [7].

The Kinect sensor is employed, as it provides 3D data in real-time. Multiple data streams are recorded, namely depth and RGB data, pose of the head and facial landmarks acquired with the Face Tracking SDK. Depth data is saved in both depth and RGB coordinate spaces, and so is RGB data. Landmarks are mapped to depth and RGB coordinates as well. Key frames with given expressions are saved as images. There is currently no video data recording included, since acquisition of the full dynamics of each expression would require considerable amount of time and subjects availability did not permit that. Nevertheless, we consider recording video data in the future.

The software allows to record the 6 basic facial expressions (and the neutral expression) under varying head poses and illumination. The recordings are divided into parts. A single part includes recording frontal or non-frontal poses and light or dark illumination. Frontal pose is recorded with more repetitions as it provides valuable training data, while being relatively quick and easy to record, as compared to the non-frontal poses, which are often problematic for the subjects. The structure of the database is as follows. Each subject has a single directory with the subject id, which contains separate subdirectories for each case (light/dark illumination, frontal/non-frontal poses). Each sample has a name consisting of: sample id (timestamp), pose id, expression id, modality id (depth, RGB, etc.).



Face expression acquisition

	Up/down	Left/right	Tilt
Current pose:	2	1	-1
Target pose:	-15	-15	0

Face tracked: Yes

Distance: 0,6 m

Done: 15/30

Figure 2. User interface

The UI (see Figure 2) is designed to be convenient for the subjects, as well as to ensure correct data. In the centre of the window RGB video feed from the Kinect is displayed, which is a crucial help for the subjects performing given expressions. On both sides, visualizations of currently required expression and pose are presented. Below the video feed there is an information regarding current angles of the head, measured by the Face Tracking SDK, as well as angles required for a particular pose. Red and green colors indicate if a given angle is within acceptable range. We found that such feedback greatly helps the subjects to set a proper head pose. Also, this mechanism is used to ensure correct data—a sample can be recorded only with proper pose. The software displays also information regarding the distance to the sensor as well as whether the face tracking is active. Both are also used to prevent recording incorrect data. It is worth noting, that all of the displayed information remains in close proximity, so the subjects would not miss anything important. Feedback from 10 recorded subjects indicated, that the UI is clear and user-friendly.

Use case: facial expression recognition

The first version of our recording software was employed for facial expression recognition using the Kinect, with focus on robustness to illumination and head pose changes. Detailed description can be found in [7]. We considered 3 basic expressions: neutral, smile and anger. Using geometric features between face landmarks tracked by the Face Tracking SDK, AdaBoost feature selection and SVM classifier we obtained results presented in Table 1.

Table 1. Facial expression recognition results

Conditions	Accuracy [%]
Normal	87.0
Dark	84.8
Pose	74.0
Dark pose	76.6
Average	80.6

We considered variations of conditions: with normal and dark lightening and frontal and non-frontal head poses. It is worth noting, that the method proved to be robust to the illumination changes. On the other hand, although head pose variations are handled better than in the case of 2D sensors, they still constitute an important challenge.

As the next step we recorded database with extended set of expressions and our current research focuses on creating advanced face descriptors.

Conclusions

In this work we presented guidelines and recommendations for acquisition of database for facial analysis. These are based on our experience with recording a facial expression recognition database. All discussed issues contribute to the final quality of a recorded database, therefore we believe that they may be useful in such task. The guidelines proved to be helpful for development of the acquisition software, which we used to record an extended version of our previous database for facial expression recognition. Particularly, the design of the user interface allowed to record the data much more effectively, compared to our previous recording software. In our future work we consider addressing the issue of recording spontaneous expressions, which is a difficult, but interesting problem.

References

[1] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, “2D and 3D face recognition: A survey,” Pattern Recognit. Lett., vol. 28, no. 14, pp. 1885–1906, Oct. 2007.

[2] R. Shoja Ghiass, O. Arandjelović, A. Bendada, and X. Maldague, “Infrared face recognition: A comprehensive review of methodologies and databases,” Pattern Recognit., vol. 47, no. 9, pp. 2807–2824, Sep. 2014.

[3] D. Smeets, P. Claes, J. Hermans, D. Vandermeulen, and P. Suetens, “A Comparative Study of 3-D Face Recognition Under Expression Variations,” IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 42, no. 5, pp. 710–727, Sep. 2012.

[4] A. M. Adeshina, S. H. Lau, and C. K. Loo, “Real-time facial expression recognitions: A review,” in 2009 Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009, 2009, no. July, pp. 375–378.

[5] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin, “Static and dynamic 3D facial expression recognition: A comprehensive survey,” Image Vis. Comput., vol. 30, no. 10, pp. 683–697, Oct. 2012.

[6] B. Y. L. Li, A. S. Mian, W. Liu, and A. Krishna, “Using Kinect for face recognition under varying poses, expressions, illumination and disguise,” 2013 IEEE Work. Appl. Comput. Vis., pp. 186–192, Jan. 2013.

[7] F. Malawski, B. Kwolek, and S. Sako, Using Kinect for facial expression recognition under varying poses and illumination, vol. 8610 LNCS. 2014.

[8] T. Kanade and J. F. Cohn, “Comprehensive database for facial expression analysis,” Autom. Face Gesture Recognition, 2000. Proceedings. Fourth IEEE Int. Conf., pp. 46–53, 2000.

[9] M. Lyons and S. Akamatsu, “Coding Facial Expressions with GaborWavelets,” third IEEE Conf. Autom. Face Gesture Recognit., pp. 200–205, 1998.

[10] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, “Texas 3D Face Recognition Database,” Proc. IEEE Southwest Symp. Image Anal. Interpret., pp. 97–100, 2010.

[11] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, “Bosphorus database for 3D face analysis,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5372 LNCS, pp. 47–56, 2008.

[12] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard, “BP4D-Spontaneous: A high-resolution spontaneous 3D dynamic facial expression database,” Image Vis. Comput., vol. 32, no. 10, pp. 692–706, 2014.

[13] D. S. Ma, J. Correll, and B. Wittenbrink, “The Chicago face database: A free stimulus set of faces and norming data.,” Behav. Res. Methods, vol. 47, no. 4, pp. 1122–35, 2015.

[14] S. Escalera, O. Nikisins, K. Nasrollahi, M. Greitans, C. Corneanu, M. O. Simón, Z. Sun, H. Li, Y. Sun, and T. B. Moeslund, “Improved RGB-D-T based face recognition,” IET Biometrics, 2016.

[15] G. Goswami and M. Vatsa, “RGB-D Face Recognition With Texture and Attribute Features,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 10, pp. 1629–1640, 2014.

[16] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active Appearance Models,” Proc. Eur. Conf. Comput. Vis., vol. 2, pp. 484–498, 1998.