Capturing Facial Videos with Kinect 2.0: A Multithreaded Open Source Tool and Database

Daniel Merget Tobias Eckl Martin Schwoerer Philipp Tiefenbacher Gerhard Rigoll
Institute for Human-Machine Communication, TUM, Germany
daniel.merget@tum.de

Abstract

Despite the growing research interest in 2.5D and 3D video face processing, 3D facial videos are actually scarcely available. This work introduces a new open source tool, named FaceGrabber, for capturing human faces using Microsoft's Kinect 2.0. FaceGrabber permits the concurrent recording of various formats, including the raw 2D and 2.5D video streams, 3D point clouds and the 3D registered face model provided by the Kinect. The software is also able to convert different data formats and playback recorded results directly in 3D. In order to encourage research with Kinect 2.0 face data, we publish a new public video face database which was captured using FaceGrabber. The database comprises 40 individuals, performing the six universal emotions (disgust, sadness, happiness, fear, anger, surprise) and two additional sequences.

1. Introduction

The field of 2.5D and 3D face recognition and manipulation has grown rapidly since the advent of affordable recording technologies such as structured light and time-of-flight sensors. Yet more importantly, computer hardware became fast enough to process such data in realtime, making way for a wide field of novel 3D applications such as video face recognition [13], video emotion recogntion [6, 22], movie production [2], video face replacement [7, 12], face reenactment [17], videoconferencing [10] or virtual avatar control [3, 15]. Having suitable data at hand is inevitable in order to compare and evaluate novel methods in these fields. For example, it is reasonable to compare 2D with 3D performance for face recogntion [1, 4]. Although a number of 2.5D and 3D face databases already exist, there are apparently only four which provide actual video sequences: BU-4DFE [19], BP4D-Spontaneous [21], 3DMAD [9] and KinectFaceDB [14].

The BU-4DFE database [19] is the transition from the static BU-3DFE database [20] to dynamic 3D video. Just like the BP4D-Spontaneous dataset [21], the data includes live performances of facial expressions and is captured by a 3D laser scanner. Due to the data being of a very high quality, however, the database is unsuitable for mimicking typical real-world scenarios. A consumer, for example, can probably not afford to buy an expensive laser scanner. 3DMAD [9] is recorded with Kinect 1.0, targeting spoofing attacks for face authentication using 3D masks. Since it is recorded in a sterile and very specific setting, other applications are fairly limited. For example, 3DMAD does not include facial expressions.

Lastly, KinectFaceDB [14] is captured with Kinect 1.0 and provides emotion as well as head scan sequences, including difficult lighting and occlusion scenarios [14]. In their work, Min et al. were able to show that 3D face and emotion recognition performance increases significantly with the quality of the sensor. The major shortcoming of KinectFaceDB is indeed that the Kinect 1.0 is outdated, not very accurate and has a comparatively low depth resolution of only 320x240. The depth sensor of Kinect 2.0 is much more robust to noise, especially at foreground-background boundaries. Furthermore, the color and depth resolutions are roughly 7 and 3 times higher, respectively. A comprehensive comparison between Kinect 1.0 and 2.0 is given in [11].

Our main contributions are 1) FaceGrabber, an open source capturing tool for Microsoft's Kinect 2.0, as well as 2) a video face database recorded with FaceGrabber. The database improves on KinectFaceDB mainly by using the more accurate Kinect 2.0 sensor. Furthermore, we hope that FaceGrabber encourages other researchers to record or extend human face video databases. The FaceGrabber source code and database are both available at: `www.mmk.ei.tum.de/facegrabber/`

2. FaceGrabber

Alongside the higher resolution and noise resistance, another advantage of the Kinect 2.0 over its predecessor is that it comes with facial correspondences based on an active ap-

Name	Structure
2D scene[a]	1920x1080 RGB & 512x424 depth
3D scene	512x424[b] colored point cloud
3D face	$\approx 3-6$k colored point cloud
2D model	1347 facial landmarks in 2D
3D model	1347 facial landmarks in 3D

[a] raw Kinect 2.0 outputs
[b] excluding z-buffered points, see Section 2.1

Table 1: Overview of the capture types.

pearance model [16]. Despite the fact that these correspondences are not as accurate as common landmark detection methods, e.g., [18, 23, 24], they are helpful for face detection, head pose estimation, or initialization for further algorithms. The core idea of FaceGrabber is therefore simple: The user should be able to capture face video sequences while also being able to store the model points generated by the Kinect. The full set of capture types is listed in Table 1. A secondary objective for FaceGrabber was to have a convenient way to examine the recorded data. Finally, for performance reasons, it should be possible to convert the data format *after* it was captured, e.g., from binary to ASCII.

2.1. Challenges

Despite the simplicity of FaceGrabber's concept, a few challenges had to be adressed. First of all, there is an offset between camera and depth sensor. The offset results in occluded points when combining the RGB channels with the depth image in order to construct a point cloud. The Kinect 2.0 API does not account for this occlusion problem which is illustrated in Figure 1. Min et al. [14] aligned the RGB and depth information via camera-space transformations in order to account for this problem. The downside of that approach is the resulting inaccurate mapping between color and depth. While the error is typically not very large, it is problematic for high frequency components. To solve this issue, the occlusions have to be detected and, subsequently, the affected points have to be filtered, since no reasonable value exists. FaceGrabber therefore applies one of the simplest and computationally least expensive techniques to achieve this, that is, z-buffering [5]. In essence, z-buffering constructs a quantized matrix of 2.5D points and stores only the nearest candidates for each cell. Although this approach filters possibly valuable background depth information, it does conserve the actual scene without introducing errors.

A second challenge while developing FaceGrabber was the synchronization of all buffers. Due to the possible asymmetry between the captured data type sizes and the high throughput requirements, several threads are working in parallel. Furthermore, if all capture types are selected, the raw throughput can easily reach several hundred Mbit/s and may even exhaust common solid state drives. Therefore, any I/O is directly bufferd in memory before being written to disk by a separate thread.

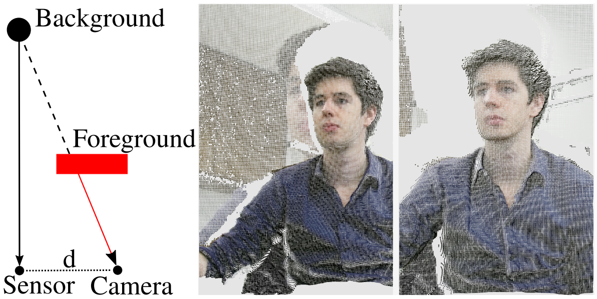


Figure 1: An offset between color and depth sensors results in colored shadows of foreground objects on the background in the 3D scene. These critical areas are filtered via z-buffering.

With the above adaptions, the only remaining performance bottleneck is the CPU. Although we did not analyze the hardware requirements in depth, we were able to consistently reach 30 fps on a Notebook with Core I7-4500U CPU as long as background data was not captured.

2.2. User Interface

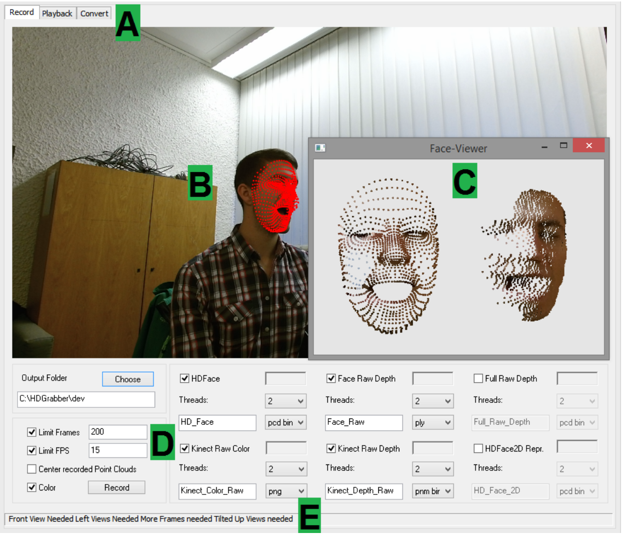
The interface of FaceGrabber is split into three different tabs which are accessible in the top left corner of the GUI:

- the *record tab* (Figure 2)
- the *playback tab* (Figure 3)
- the *convert tab* (not shown)

This division separates the core features of FaceGrabber into intuitive functional units. The GUI runs in a separate thread and is therefore always responsive to user input. The user is given the freedom to manage each capture type separately. Since the required performance during recording may vary depending on the scenario as well as the available hardware, the user can choose the number of threads per capture type separately. For example, a lot of data has to be processed in order to capture the background. Thus it may be necessary to dedicate more threads to meet the target frame rate.

3. FaceGrabber Database

We use FaceGrabber to record a database of human faces at 15 fps. The reasons for not recording at the maximum 30 fps are 1) the fact that the Kinect automatically halves the frame rate in dark scenes (twice the exposure time) and 2) the high performance cost of capturing all data streams at the same time without losing synchronization. For the latter reason, although the full 2D scene is captured, we do not capture the full 3D scene but rather only the face area.



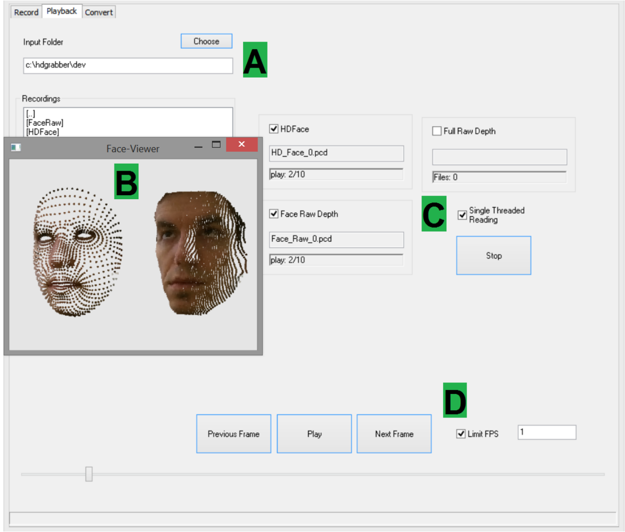
Output Folder: C:\HDGrabber\dev

☑ Limit Frames 200
☑ Limit FPS 15
☐ Center recorded Point Clouds
☑ Color

☑ HDFace — Threads: 2 — HD_Face — pcd bin
☑ Face Raw Depth — Threads: 2 — Face_Raw — ply
☐ Full Raw Depth — Threads: 2 — Full_Raw_Depth — pcd bin
☑ Kinect Raw Color — Threads: 2 — Kinect_Color_Raw — png
☑ Kinect Raw Depth — Threads: 2 — Kinect_Depth_Raw — pnm bir
☐ HDFace2D Repr. — Threads: 2 — HD_Face_2D — pcd bin

Front View Needed Left Views Needed More Frames needed Tilted Up Views needed

Figure 2: The *record tab* of FaceGrabber. (A) The tabs are accessible at the top. (B) The 2D output of the Kinect with overlaid landmarks is shown live. The horizontal axis is flipped in order to imitate a mirror. (C) The 3D landmarks and 3D face are also shown live in a separate resizable window. The 3D views (translation, rotation, zoom) can be changed separately with the mouse on the fly. (D) Recording options can be customized. (E) The user receives feedback if data is missing for accurate tracking.



Input Folder: c:\hdgrabber\dev

Recordings:
[..]
[FaceRaw]
[HDFace]

☑ HDFace — HD_Face_0.pcd — play: 2/10
☐ Full Raw Depth — Files: 0
☑ Face Raw Depth — Face_Raw_0.pcd — play: 2/10
☑ Single Threaded Reading
☑ Limit FPS 1

Figure 3: The *playback tab* of FaceGrabber. (A) Folder selection. (B) The 3D data to be played back is displayed in a separate resizable window. The number of views adapts to the number of captures being selected. Again, the 3D views (translation, rotation, zoom) can be changed separately with the mouse on the fly. (C) The frames are buffered in memory before playback in order to ensure a constant frame rate. (D) The user can customize the frame rate and navigate inside the data using either the buttons or the slider.

3.1. Capturing Conditions

Nowadays, 2.5D sensors like the Kinect are very cheap and they are used in many day-to-day applications. Therefore, one of the main objectives was that the recordings should be as natural as possible in order to mimic a real-world scenario. Consequently, the recordings were done in a regular office environment and no special background restrictions or lighting conditions were imposed. On the contrary, indirect lighting from the sun as well as direct room lightning were varied on purpose. The approximate capture distance was 0.8m, which results in about 3000 to 6000 3D points for each face. This number may seem low compared to other 2.5D databases reporting up to about 9000 points per face using Kinect 1.0 [14]. In contrast to them, however, we do not resample or interpolate the depth values provided by the Kinect, which would introduce errors. Nevertheless, the possibility of artificially increasing the depth resolution is left open by the fact that the raw Kinect outputs are provided.

3.2. Data

Overall, the database consists of 40 different individuals (33 male) aged between 18 and 28. The participants are asked to perform the six universal emotions (disgust, sadness, happiness, fear, anger, surprise) [8] in a two-second time window. Additionally, the database includes a six-second head scan sequence with neutral expression, where the individuals rotate their heads in front of the camera, in order to potentially allow for a 3D reconstruction of the head. Finally, the individuals were invited to perform any movements or facial expressions they wanted to for ten seconds, however, with the restriction of facing the camera ($\pm 45°$) and not performing too fast head motions, both to avoid loss of tracking. In contrast to KinectFaceDB [14], we do not capture occlusion sequences because this would interfere with our idea of capturing the Kinect 2.0 face model.

For video sequences to be of noticably more value than single images, there has to be exploitable temporal context. Hence, we explicitly ask the participants to perform the emotions in a natural and active way, meaning that they neither exaggerate nor freeze like they would for a photograph, but rather run through the full motion. For example, we intentionally allow head movements and tilting, or even leaning back in the chair (e.g., when surprised). On the one hand, this increases the complexity and variety of the performed emotions. On the other hand, it also reinforces the uniqueness of the same emotion among different individuals as well as the different emotions for the same individual. An excerpt from the database is given in Figure 4.

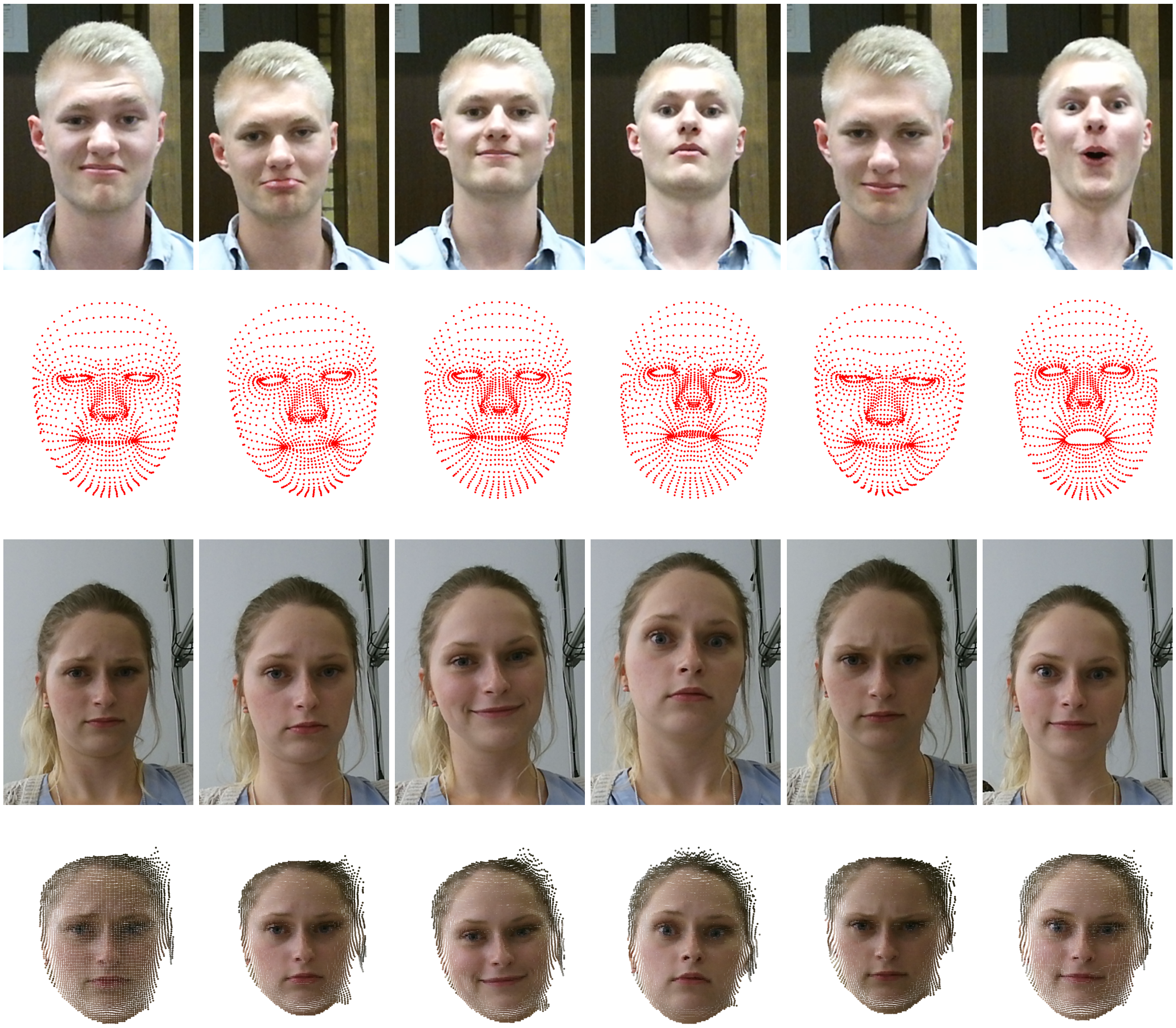


Figure 4: An excerpt from the FaceGrabber database. From left to right: Disgust, sadness, happiness, fear, anger, and surprise. Each image shows the beginning of a two-second time window. The images are cropped to the same region in order to highlight the (intentional) variance in head and body pose. For illustration purposes, the second row depicts the landmarks of the first row, whereas the fourth row depicts the 3D face of the third row.

4. Conclusion

In spite of the growing number of depth cameras, 2.5D and 3D video face databases are rare. We have presented a open source face capturing tool, FaceGrabber, as well as a recorded database in order to address the scarcity of 3D face videos. The database strives to achieve realistic caputuring conditions with as few restrictions as reasonably possible. To the best of our knowledge, this is the first public (video) face database using Kinect 2.0, despite the desktop consumer version being sold for close to $1^{1}/_{2}$ years now. We hope to animate other researchers to contribute to this growing database of 2.5D face videos and thereby advance the field of 3D video face recognition, manipulation, and the like.

For future work, we are investigating which further pre- and postprocessing options would be worthwile to integrate into FaceGrabber in order to extend the functionality and make its use even more convenient. Another aspect which could be improved further is achieving a steady frame rate even on lower cost hardware, for example, by means of queueing any 3D conversions for postprocessing rather than using a buffered thread pool. Finally, FaceGrabber is yet lacking a streaming interface which would allow for feeding the captured data into arbitrary algorithms on the fly before (or instead of) writing the data to disk.

References

[1] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28:1885–1906, 2007.

[2] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. The digital emily project: Photoreal facial modeling and animation. In *SIGGRAPH Courses*, pages 12:1–12:15. ACM, 2009.

[3] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40:1–40:10, 2013.

[4] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Computer Vision and Image Understanding*, 101:1–15, 2005.

[5] E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Dissertation, Computer Science Department, University of Utah, 1974.

[6] H. Chen, J. Li, F. Zhang, Y. Li, and H. Wang. 3D model-based continuous emotion recognition. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Transactions on Graphics (TOG)*, 30, 2011.

[8] P. Ekman. Facial expression. *Nonverbal Behaviour and Communication*, pages 97–126, 1977.

[9] N. Erdogmus and S. Marcel. Spoofing in 2D face recognition with 3D masks and anti-spoofing with kinect. In *Proc. Conference on Biometrics: Theory, Applications and Systems*, 2013.

[10] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross. Gaze correction for home video conferencing. *ACM Transactions on Graphics (TOG)*, 31(6), 2012.

[11] E. Lachat, H. Macher, M.-A. Mittet, T. Landes, and P. Grussenmeyer. First experiences with kinect v2 sensor for close range 3D modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, pages 93–100, 2015.

[12] D. Merget, P. Tiefenbacher, M. Babaee, and G. Rigoll. Photorealistic face transfer in 2D and 3D video. In *Proc. German Conference on Pattern Recognition (GCPR)*. Springer, 2015.

[13] R. Min, J. Choi, G. Medioni, and J.-L. Dugelay. Real-time 3D face identification from a depth camera. In *Proc. International Conference on Pattern Recognition (ICPR)*, 2012.

[14] R. Min, N. Kose, and J.-L. Dugelay. Kinectfacedb: A kinect database for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 44(11):1534–1548, 2014.

[15] T. Shiratori, M. Mahler, W. Trezevant, and J. K. Hodgins. Expressing animated performances through puppeteering. *Symposium on 3D User Interfaces (3DUI)*, pages 59–66, 2013.

[16] N. Smolyanskiy, C. Huitema, L. Liang, and S. E. Anderson. Real-time 3D face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32:860–869, 2014.

[17] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[18] M. Uřičář, V. Franc, D. Thomas, S. Akihiro, and V. Hlaváč. Real-time multi-view facial landmark detector learned by the structured output SVM. In *BWILD: Automatic Face and Gesture Recognition Conference (FG) and Workshops*. IEEE, 2015.

[19] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3D dynamic facial expression database. In *Proc. International Conference on Automatic Face and Gesture Recognition (FG)*, pages 211–216, 2008.

[20] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *Proc. International Conference on Automatic Face and Gesture Recognition (FG)*, pages 211–216, 2006.

[21] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, and P. Liu. A high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing*, 2014.

[22] Y. Zhang, L. Zhang, and A. Hossain. Adaptive 3D facial action intensity estimation and emotion recognition. *Expert Systems with Applications*, 42:1446–1464, 2015.

[23] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional neural network. *ICCV workshop on 300 Faces in-the-Wild Challenge*, 2013.

[24] F. Zhou, J. Brandt, and Z. Lin. Exemplar-based graph matching for robust facial landmark localization. In *Proc. International Conference on Computer Vision (ICCV)*, 2013.