Magic Mirror: I and My Avatar - A Versatile Augmented Reality Installation Controlled By Hand Gestures

Alexander K. Seewald[1] and Alexander Pfeiffer[2]

[1]*Seewald Solutions, Lärchenstraße 1, 4616 Weißkirchen a.d. Traun, Austria*
[2]*Danube University Krems, Dr. Karl Dorrek Straße 30, 3500 Krems, Austria*
alex@seewald.at, alexander.pfeiffer@donau-uni.ac.at

Keywords: Augmented Reality, Depth Cameras, Hand Gesture Recognition, People Counting, Shopping Window Rating, Gaze Direction Heat Map, Person Height Measurement

Abstract: Since 2012 we have been building the augmented reality system *Magic Mirror* based on Kinect V1 and V2's native API. It relies on the magic mirror metaphor, where a large screen shows a mirrored camera view with overlaid graphical elements. In our case, it shows a different face mesh over the person's face which reliably tracks face poses in real time while leaving the eyes and mouth of the person visible for interaction and to improve immersion; replaces the background with images that may be changed, smoothly zoomed and dragged; and allows to take screenshots which are automatically printed out on photo cards with an unique QR code linking to its digital twin. Control of the system is primarily via easily learned hand gestures very similar to multitouch screen gestures known from mobile phones and tablets. We have demonstrated the system to the public as well as in private over a wide variety of settings, faces and backgrounds. Here, we explain the challenges inherent in creating high-quality face meshes and textures from 2D images, and how we solved them; describe the different versions of the system, how they differ and their limitations; and demonstrate the usefulness of our system in several applications from people counting and tracking to obtaining height measurements without storing or processing personal data.

1   INTRODUCTION

In 2012 – when Microsoft's Kinect for Windows, which enabled using their XBOX depth camera on PCs, finally became available in Europe – my co-author Alex P. dropped a unit on my desk and challenged me to build a hand gesture control for his presentations using its – at that time – unprecedented body pose tracking. The Kinect V1 used an active sensor and was one of the first commercially available low-cost depth cameras. It also included a state-of-the-art face tracker as well as a pretrained random-forest-based classifier to automatically detect 3D body positions and poses (Shotton et al., 2011) – including a reasonably accurate position of both hands – so initial tests indicated that it should be feasible.

We implemented a first prototype in about two weeks. We were then invited by Microsoft to the Pioneers Festival 2012 in Vienna and presented a keynote on our gesture control app – of course using the same app. There were many newspapers and magazine reports, and even an interview with a local TV station, W24.[1] We built a downloadable version, *Universal Remote*, in another few weeks and it sold reasonably well. However the customer base of professional presenters was quite small, so we also wanted to build something for everyone to use and to show other uses of our gesture module and the Kinect itself. We additionally believed that the potential of this technology was far beyond our simple first application.

Our main inspiration for Magic Mirror came from gaming theory. (Caillois, 1958) extended the theories by Johan Huizinga with four universal game types. One of these is *Mimicry*. Mimicry means *becoming someone else* and participating in an *illusionary world*. This can occur between the free game type (paida) and the rule-based game type (ludus). Here, the gamer is inside a virtual character that is an Avatar - a concept mentioned for example in Neal Stephenson's Science Fiction novel *Snow Crash* (Stephenson, 1992), where it has been introduced to the public in the context of computer gaming. However already in the 80ies the computer game series *Ultima* introduced

---

[1]See https://w24.k4w.at

the concept of an Avatar - especially in part IV *Quest for the Avatar* where the gamer takes over the role of an Avatar as virtual image. (Gee, 2003) considers the different identities. For him there is the identity of the gamer, the identity of the avatar in the game, and the projected identity. By this he describes transfer effects between gamers and their Avatars.

This naturally led to a *face changer*, which allowed to replace the background as well as wearing a flexible mask which could show facial expressions – basically an avatar – all rendered in real-time and instantly responsive to changes in face and body pose. And so we called the system *Magic Mirror - I and my Avatar*, and defined it as an Autonomous Augmented Reality Art installation where AI and machine learning techniques for gesture control, body segmentation and face tracking (such as Random Forest, Active Appearance Models, Support Vector Machines and Dynamic Programming) are utilized to allow users to wear and intuitively change – by hand gestures – a dynamic virtual carnival mask which tracks detailed face expressions, and also replace their background with other scenes, real and imagined.

Within *Magic Mirror - I and my Avatar* we initially provided a mixture of characters from the well-known Massively Multiplayer Online Role-Playing Game *World of Warcraft*[TM] and contemporary Austrian politicans and presented it at a gaming conference (FROG 2012). It was an instant hit.

Afterwards, to make the installation more accessible to the general public, we successively extended the mask set. At present we have many different masks based on seasonal variations (Easter Bunny, Santa Claus and *Christkind*, Halloween Characters), European politicians and other persons of interest (such as the Pope and Edward Snowden). Our installation allows up to six users to each take over the role of a virtual figure or a politician in parallel and enables each one to change his face, reposition and resize the background, and make a snapshot, by simple hand gestures.

One disturbing variant of the installation is to give everyone the same unchanging face. Do you still feel like yourself?

2 RELATED RESEARCH

(Osokin, 2018) describes a system for body pose estimation from RGB cameras that works bottom-up and therefore scales to high numbers of persons – contrary to the Kinect with its top-down approach that restricts the number of tracked persons to at most six. It reconstructs roughly the same number and types of body parts as the Kinect. It works reasonably well according to real-life tests[2] and needs low computational resources, comparable to the Kinect V1. However no 3D positions of body parts are obtained, therefore our current hand gesture system cannot be applied directly to its output.

(Castro-Vargas et al., 2019) describe a system to directly learn four hand gestures (down, up, left, right) via 3D convolutional neural networks trained directly on depth camera images. While an interesting idea, their quoted accuracy of 73% is not high enough for a practical system.

(Ferrari et al., 2019) describe a system for 3D face reconstruction from combined color and depth camera (RGB-D) data. While the quality of the 3D face construction is very good and comparable to (Smolyanskiy et al., 2014), it has the disadvantage of needing a sequence of RGB-D frames to work with – rather than a single frame – and has not been tested with a single frame at all. So it is likely an application would not track face expressions sufficiently fast to be considered real-time – which is however a primary constraint of our system.

3 HISTORY

In this section we give an overview of the different releases as well as important components of the Magic Mirror. Our claim that it is a single system is supported by the fact that all described variants can be created with different preprocessor defines from a single C/C++ source code project. Magic Mirror was publicly demonstrated at many different locations.[3] We also presented it at each annual lecture Future Media by Alex S. at the Danube University Krems and at various other non-public events.

3.1 CREATING FACE TEXTURES

The creation of high-quality face textures and meshes proved to be a major challenge. We initially thought that the Kinect V1 would only accept real-life faces, which would have made it hard to get e.g. Angela

---

[2]We tested it ourselves at ICAART 2019, together with all the other participants of the session.

[3]Future and Reality of Gaming (FROG) conferences (2012,2013,2014,2017), Vienna City Hall, Austria; Subotron Shop (2014), Museumsquartier Vienna, Austria; International Broadcasting Conference (IBC 2015), Amsterdam RAI, Netherlands; Mastercard Ad Campaign, Museumsquartier Vienna, Austria (2016); Oberbank Wels, Austria (2016,2017); Danube University Krems (twice 2016,2017); Welios Wels, Austria (2016,2017)

Figure 1: Face tracking with projection and high-quality face texture. Original image (left), Face texture w/ mesh (right)

Merkel and then-US-president Obama's face textures. However, by conducting extensive tests we found that Kinect V1's Face tracker relies mostly on the color information and only uses depth information to ensure the correct size of the face w.r.t. distance. As far as we know there is no publication that describes the Kinect V1 face tracking algorithm but from our observation we can safely assume it was a 2D/3D Active Appearance Model similar to the one described in (Cootes et al., 1998) with minor filtering by depth information, where the initial face position is taken from the head position of the estimated body pose.

So by printing out a 2D face image in approximately real size (face centered and scaled to an A4 page) and vertically positioning within in a sufficiently body-similar setting (we fixed the face image on a calibration rig with chessboard pattern and put it on a office chair – see Fig. 1) it was possible to make the Kinect V1 detect it as a face by slowly turning the office chair left and right. In hard cases we would stand next to the office chair until our body and face was detected and then quickly move behind the chair, which in most cases was sufficient for our face mesh to be transferred to the printed face image.

The best solution for face mesh creation proved to be to project the 2D face triangles from the detected face onto the original face image by reverse projection. This was done by measuring the corners of the printed face image page at subpixel accuracy – yielding an irregular quadrangle – and projecting each 2D triangle corner point within the quadrangle back to the high-resolution face image. This yielded satisfactory results. Fig. 1 shows such a sample. In this case the resolution of the face texture may be arbitrarily high. The limitation is mainly the face tracker, which only outputs integer point values (i.e. does not track at subpixel resolution).
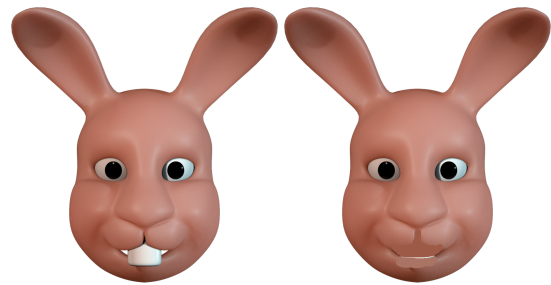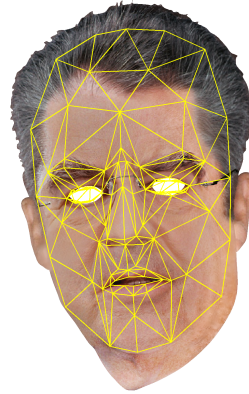

Figure 2: Initial rabbit face that could not be tracked (left), changed rabbit face that could be tracked (right).

In some cases – e.g. the colorful Easter Bunny faces – synthetic faces were not sufficiently face-like to be detected. Minor modification were usually able to recover meshes from these faces as well, and the meshes could in most cases be applied to the original image with only minor modifications. See Fig. 2 for an example.

Unfortunately, the Kinect V2 used a different face tracker described in (Smolyanskiy et al., 2014) which could not be deceived in this way. In extensive experiments we could never make it recognize and track a flat 2D face print. We did however manage to feed the color image data of Kinect V2 into the Kinect V1 face tracker and thus had slightly better and higher resolution tracking. This needed a PC with both frameworks installed as well as both Kinect V1 and V2 physically connected, so it was unfortunately quite cumbersome to use. Also, the new face tracker had about ten times the tracked points of the old face tracker, and these did not exactly correspond to points of the old face tracker. We could still achieve satisfactory but not perfect results by mapping each point of the old face image mesh to a well-chosen point on the new face tracker mesh.

Both the Kinect V1 and V2 face tracker only track the face itself and not the head, which means that parts
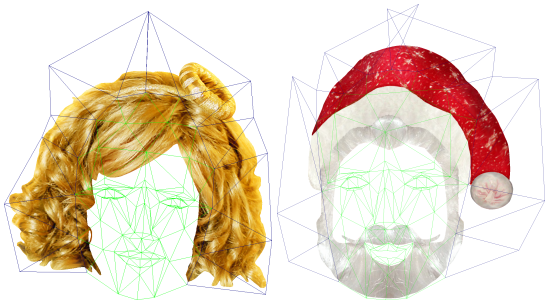
Figure 3: Two face samples with face mesh (green lines) and extended mesh (blue lines) that show masks with see-through face (white=background, transparent). Right mask is a 3D model mask from an earlier AR project, which was rendered to 2D.

Figure 4: Sample image for Xmas Theme installation.

outside the face – forehead, hair, ears, and beards – cannot be properly displayed on the rendered face. To also render these parts, we extended the outermost 3D face triangles by vector arithmetic, generating both new points and new triangles from the face texture mesh and the real-time face mesh, and applied the projection to these extended meshes. This method worked surprisingly well, and also enabled the use of extended masks which are transparent for most of the face area. Fig. 3 shows two examples. The right example is actually a 3D mask projected to 2D which was taken from an earlier unpublished Augmented Reality project. These were used for the X-Mas Theme version, see Fig. 4.

3.2 GESTURE CONTROL

The gesture control was taken from our earlier unpublished project *Universal Remote* and contained three types of gestures: **left/right (one-handed slide)**, **zoom in/out (two-handed)**, and **one-handed drag**. We added another gesture, **two-handed-drag**, in 2016 exclusively for Magic Mirror. Table 1 shows all supported gestures. These could be dynamically assigned, e.g. zoom in/out could be assigned to making a screenshot or to resizing the background image smoothly and fluidly. The basis for this was the body

Table 1: Supported Hand Gestures

Name	Image	Description
move		Slide with your hand to the left or the right to activate left or right move gesture. Either hand will work, only the direction of movement is important.
zoom		Make a diagonal movement with both hands. Push both hands out to the front and move them away from each other (zoom in) or towards each other (zoom out), then retract them again.
one-handed-drag		Raise either hand to press the left mouse button. Afterwards push the other hand out to the front. This controls the mouse cursor. By moving the other (non-raised) hand you can drag the background image around just like normal drag-and-drop.
two-handed-drag		Push both hands out to the front. This controls the mouse cursor. By moving either or both hands, you can drag the background image around just like drag-and-drop. The center point between both hands is used for dragging, so using both hands will make the drag faster.

pose tracking by Kinect V1 and V2 – thankfully very similar – described in (Shotton et al., 2011).

3.3 KINECT V1 (V1.0)

The first version for Kinect V1 only supported one face, but it was already possible to change backgrounds and faces by left/right hand move gestures. A snapshot could be taken using zoom in or out gesture. Fig. 5 shows a sample image. We demonstrated it at the FROG 2012 conference.

The next version V1.1 in 2013 – demonstrated at the FROG 2013 conference – supported the maximum of two faces that was possible for Kinect V1 using the original Microsoft API. We also replaced the local politicans with international politicans, the pope, Edward Snowden, and a Guy Fawkes mask. These masks already had see-through eyes and were fully optimized with all four steps mentioned earlier.

3.3.1 IMPROVING IMAGE QUALITY (V1.5)

Unfortunately the image quality of Kinect V1 was quite bad and image resolution was low, so we combined it with a Hero GoPro 3, using its Mini-HDMI output port and a HDMI frame grabber card to generate a combined real-time video from GoPro image data and Kinect depth data. We treated the Kinect

Figure 5: Kinect V1 – Sample Image. We used a z-filter with adjustable distance here where all objects beyond a certain distance were replaced by background. This accounts for the presence of the chair and parts of the table in this image. Later we implemented z-Skeleton filtering which replaced all background except for recognized bodies (skeletons in Kinect terminology).

V1 and the Hero GoPro 3 as two cameras of a stereo-camera setting and computed intrinsic calibration matrices (K) for both using calibration rigs with a 5x8 chessboard pattern with size A2 and 61.5mm squares, followed by stereo calibration.[4]

Stereo calibration returned the translation matrix T_{stereo} and the rotation matrix R_{stereo}, which transform the world coordinate system of the Kinect into the one of the second camera. We generated a point cloud from Kinect V1's depth data using native API functions, transformed it by $R_{stereo}T_{stereo}$ to the second camera's view frame, and then projected it onto the second camera's focal plane using its intrinsic calibration matrices. This reproduces the almost perfect alignment between Kinect V1 color and depth images, so depth images can be filtered and combined with the color images very easily. We then optimized this mapping – including the initial native point cloud generating function – to a single matrix multiplication which proved sufficient for real-time rendering and yielding almost the same results. The 3D body part positions were transformed using the same matrix.

In 2014 we installed the final system for three weeks behind a shopping window with Easter Bunny face textures and meshes and showed that we could count in/outgoing people as well as compute a *window rating* for shopping window attractiveness and additionally a heatmap of gaze direction. More details can be found in Section 4.1.

This system was also demonstrated at the FROG 2014 conference using the international politicians face set.

---

[4]All algorithms were from OpenCV V1.0.

3.4 KINECT V2 (V2.0)

In 2015 we ported the system to Kinect V2 as the Kinect V1 was no longer available. The API had completely changed, but by extensive use of C/C++ macros it was possible to keep changes to a minimum and still generate all V1 and V2 systems from the same source code project. This was important since we still needed the V1's facetracker to scan new faces. The Kinect V2 allowed up to six faces and had a newer, much more accurate facetracker, but it also had much higher hardware requirements since all color, depth, face and body pose streams were always sent at the highest possible resolution. [5] We demonstrated the system at our stand at the International Broadcasting Conference 2015 (IBC 2015). Fig 6 shows a sample image of the improved system taken at that conference.

In August 2016 we provided a Magic Mirror installation for an ad campaign by MasterCard Austria in the Museumsquartier.[6] For this we extended the system to use the zoom in and zoom out gestures – up to this time considered binary – to continuously zoom within a larger image, and also introduced the two-handed-drag gesture to position the zoomed image with less physical effort by removing the need to keep one hand raised. We also extended the system with an option to automatically print photo cards after each screenshot. This extended version was called V2.1. Running the system outdoors proved a challenge since the Kinect does not work in bright sunshine, so we used a pavillon to create shadow but had to reposition it every other hour. Getting a large monitor which was visible in bright sunshine was on the other hand no problem at all.

In October 2016 we demonstrated a Halloween version of the Magic Mirror (using Halloween faces and background) at Oberbank Wels, and later a Christmas version at Welios Wels and Danube University Krems (both in Austria). We added an unique QR code on each printout which linked to the digital snapshot, thus integrating the visual printout and its digital twin. Before this change, people had often used their phones to make screen snapshots and share them. Now, it was possible to just scan the QR code on the photo card and get to a digital version of the same photo for easy sharing via social media as well as having a physical photo card to remember the event by. We also added rendering improvements such as pseudo alpha-blending (improves the appear-

---

[5]Approximately 5GBit/s via USB 3.

[6]The Museumsquartier is a large plaza in Vienna, Austria with three large museums and several smaller spaces for modern art open to the public.

Figure 6: Kinect V2 – Sample Image

ance of the border between body and background) and z-skeleton filtering (prevents the need to set up the z-distance from which the background image is rendered, by simply rendering just the recognized body shapes – skeletons in Kinect terminology – on the background). This new extended version was called V2.2.

In April 2017 we demonstrated an Easter Bunny version with zoomable background at Welios Wels in Austria, and printed photo cards, again with an unique QR-code that linked back to the image for easy sharing.

4 USES OF THE SYSTEM

Apart from its entertainment value, Magic Mirror can be used in several different ways either for analysis or to add new types of user interaction. Here, we note several ways the system was used in practice.

4.1 TRACKING PEOPLE

As we shortly mentioned before, in 2014 we put the V1.5 system behind a shopping window for 21 days with Easter Bunny faces and background at Museumsquartier Vienna (at the Subotron shop). The Museumsquartier is a wide open plaza in Vienna, Austria with three large museums and several smaller spaces for modern art. In front of this space there is a narrow very long building which most people cross to get into or out of the plaza[7] and this is where we placed the system, perpendicular to this long building. For this installation we added a magnetic polarization filter on the GoPro camera to reduce shopping window reflections. We used a 16:9 computer monitor with one meter / 40 inch diagonal. While the system was mainly intended to provide entertainment, we were also able

---

[7]It is also possible to enter and exit by the sides or at other positions of the narrow building, so we saw only a subset of the people crossing.

to track movements of people, determine how often they looked directly at the monitor, as well as compute aggregate approximate gaze direction into a heat map, without storing any kind of personal data.

4.1.1 COUNTING PEOPLE PASSING

We used the system to count people passing from left to right (Enter) and from right to left (Exit – both w.r.t the Museumsquartier Vienna) by tracking each body Id separately and computing Pearson's correlation coefficient between time and the x position of body center for all body Ids with at least three tracked positions. For a correlation $corr$ with $abs(corr) > 0.9$, we took the sign of the correlation to indicate either left or right directed movement. People where $abs(corr) <= 0.9$ were labeled *Neither*, since in this case no clear linear movement in one direction could be detected (see Eq. 1,2).

$$c_{bId} = \frac{n \sum t * x_{bId,t} - \sum t \sum x_{bId,t}}{\sqrt{n \sum t^2 - (\sum t)^2} - \sqrt{n \sum x_{bId,t}^2 - (\sum x_{bId,t})^2}}$$
(1)

$$d_{bId} = \begin{cases} \text{Neither} & \text{if } |c_{bId}| \leq 0.9 \\ \text{Left} & \text{if } |c_{bId}| > 0.9 \text{ and } c_{bId} < 0 \\ \text{Right} & \text{if } |c_{bId}| > 0.9 \text{ and } c_{bId} \geq 0 \end{cases}$$
(2)

Fig. 7 shows the number of people per hour who entered or exited within the given hour of the day. Computed values were averaged over the whole 21 days of the Easter 2014 installation and show a distinct daily pattern. Colored bands show additive values, e.g. for hour 15 there are on average 11.7 people exiting, 24.2 people entering, and 12.2 people neither entering nor exiting, yielding a total of 48.1 people counted at this hour (averaged over the whole time period). A total of 13,212 people were counted during this period. Note that due to signal attenuation by the shopping window, likely only a subset of people was counted – those who moved close enough to the shopping window (around 2.5m)

4.1.2 SHOPPING WINDOW RATING

We also defined a shopping window rating to determine the attractiveness of the shopping window at different times. Here, the system tracked how long the people watched the screen. For this we simply computed the proportion of tracked faces versus tracked bodies, since the API cannot return any tracked face without a corresponding body pose (see Eq. 3).
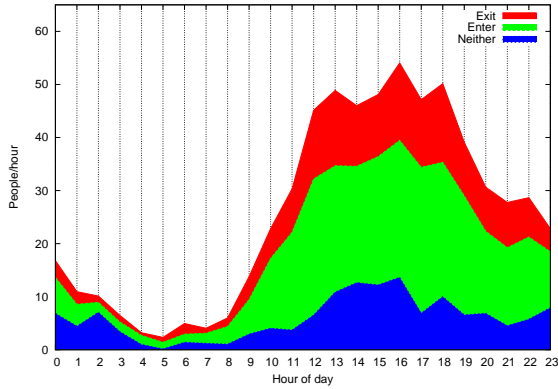
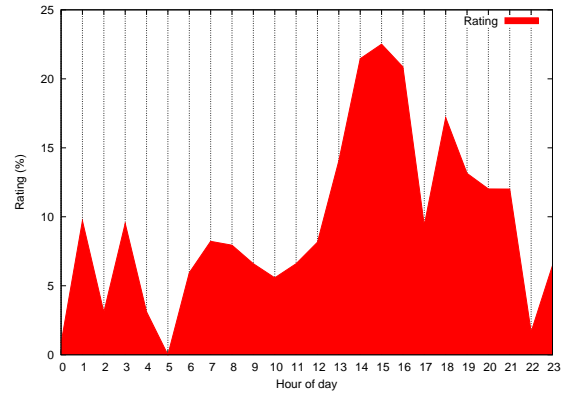Figure 7: People going into / out of Museumsquartier Vienna by time of day, averaged over total timespan.

Figure 8: Window rating for shopping window: Proportion of people looking at the screen (Facetracking success) divided by total number of people detected, grouped by time of day, averaged over total timespan.

$$Rating_{hour} = \frac{1}{|hour|} \sum_{t \in hour} \frac{trackedFaces_t}{trackedBodies_t} \qquad (3)$$

Since the Kinect V1 face tracker reacts very strongly to front faces, very weakly to turned faces (up to $30°$), and not at all to side faces, this was sufficient to detect people looking directly at the screen. Once a face is tracked it can be turned $\pm 30°$ but only slowly, so these values might be slightly overestimated. On the other hand such slow head movements are rarely observed in public.

Of all 13,212 people counted, 1,677 looked at the screen – on average for 15.5 seconds – which gives an overall rating of 12.69%. The system was watched for a total of 12,305 seconds (3.42 hours)

463 people (3.50%) watched for at least 15s, 205 people (1.55%) for at least 30 seconds, and 69 people (0.52%) for at least 60 seconds.

Depending on the hour of day – i.e. at peak hours – high ratings could be obtained, e.g. for hour 15 the rating was 22.51%. This means that of all people standing in front of the screen during this time period, almost a quarter looked at the screen. Rating estimates per hour are not smooth, partially due to small sampling size for some hours but also since people counts are not smooth. Fig. 8 shows the full results per hour, again averaged over all 21 days.

4.1.3 GAZE DIRECTION HEATMAP

We were also able to utilize the integrated face tracker to output rough estimates for head position and therefore approximate gaze direction, assuming the person looks straight to the front. To some extent this is quite likely since the face tracker initially only recognizes front faces and later fast movements to the side will make the face instantly lose tracking. We used this

Figure 9: Heatmap for Museumsquartier Vienna Installation (approximate gaze points aggregated over total timespan). Brighter = more accumulated gazes at this point.

data to build a heatmap which was overlaid on an image of the actual shopping window. This shows where people generally look at when viewing the shopping window and enhances the shopping window rating with very specific information on salient objects that is extremely hard to obtain using any other means. Fig. 9 shows the logarithmically scaled heatmap for the whole 21 days. As expected, our installation was the most salient object.

4.2 ESTIMATING BODY HEIGHT

In 2016, we evaluated the use of Kinect V2 to determine people's height. For this, we recorded several hours of body position data, yielding 129 people including 64 children. Body height was estimated as

the distance from the floor plane in meters at the head position (corresponding to the center of the head plus the difference between head and shoulder vertical position, see Eq. 4-6)

$$plane_{floor} = a * x + b * y + c * z + d \quad (4)$$

$$dist_{part} = \frac{|a * part_x + b * part_y + c * part_z + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (5)$$

$$height = dist_{head} + (dist_{head} - dist_{shoulderCenter}) \quad (6)$$

As we did not have ground truth data except for one person, we computed the standard deviation over all body positions with more than 25 consecutive frames, which was 3%. For the one person where ground truth height was available, the estimate from averaging all frames underestimated the true height by 4.3%. However this may be a systematic error that is easy to correct. Even if not, the true height is within the 1.3fold confidence interval of the average and thus not significantly different at 95% confidence level, so a running average of body height should be sufficient to get reasonable height estimates within 1-2s (15-30 samples).

4.3 ARS ELECTRONICA 2019

We submitted a short video – see https://ars.k4w.at – demonstrating our system for Ars Electronica 2019. We did however not win any prizes.

5 CONCLUSION

It was very interesting to work on this project, however we really do not want to continue with Windows as a programming platform, nor do we want to rely on Microsoft hardware in the future.

So one of our plans is to port Magic Mirror to a small platform such as Raspberry Pi 4 or – should it not be sufficiently powerful – small embedded PCs running Linux – and make it work with other depth cameras or perhaps even – one day – with normal 2D color cameras, using appropriate Deep Learning tracking systems at the same functionality level.

We would also like to enable high-quality tracking through window glass (without cutting a hole in it) and outside (without lugging a pavillon around and repositioning it every hour as we did for the Mastercard event). For this specially developed hardware is likely needed.

Every time we put up Magic Mirror, we could count on smiling faces and happy people. So, since the ten-year anniversary of this project is in 2022, we also plan to take all our remaining Kinect sensors and build – as far as we can – all previous installations of Magic Mirror into a single room (just like

a gallery), include free printout of photo cards, and make all of this available for at least several months of that year. If you are interested in visiting, check https://mm.k4w.at for updates or send an email to us.

REFERENCES

Caillois, R. (1958). Théorie des jeux. *Revue de Métaphysique et de Morale*, 63(1):83–102.

Castro-Vargas, J. A., Zapata-Impata, B. S., Gil, P., Garcia-Rodriguez, J., Torres, F., et al. (2019). 3DCNN performance in hand gesture recognition applied to robot arm interaction.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *European conference on computer vision*, pages 484–498. Springer.

Ferrari, C., Berretti, S., Pala, P., and Del Bimbo, A. (2019). 3D Face Reconstruction from RGB-D Data by Morphable Model to Point Cloud Dense Fitting. In *ICPRAM*, pages 728–735.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20.

Osokin, D. (2018). Real-time 2d multi-person pose estimation on CPU: Lightweight OpenPose. *arXiv preprint arXiv:1811.12004*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee.

Smolyanskiy, N., Huitema, C., Liang, L., and Anderson, S. E. (2014). Real-time 3D face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860–869.

Stephenson, N. (1992). *Snow Crash: A Novel*. Bantam Books.