Argus Vision: A Tracking Tool for Exhibition Designers

Moritz Skowronski[1], Daniel Klinkhammer[1], Harald Reiterer[1]

Human-Computer Interaction Group, University of Konstanz[1]

name.surname@uni-konstanz.de

Abstract

Contemporary exhibitions are increasingly staged using extensive and often interactive media. To create such exhibitions, exhibition design companies employ professionals from a wide range of different disciplines. The support of interdisciplinary exhibition designers in the design process is one goal of research in Human-Computer Interaction. This includes the deployment of Do-It-Yourself (DIY) Tools that enable professionals from all disciplines involved to design and create interactive media themselves. In this paper, we will present Argus Vision, a DIY Tool, which allows exhibition designers the use of camera-tracking to rapidly prototype and develop immersive exhibitions and interactive installations. We successfully used Argus Vision in an exhibition as well as in a proof-of-concept study in our lab. Additionally, we conducted expert interviews with exhibition designers, investigating its usefulness for them.

1 Introduction

In contemporary exhibitions, artifacts are no longer solely presented in an exhibition room but are staged in order to create a thematically coherent narrative for visitors (Kossmann & de Jong 2010). To achieve this, exhibition designers use a variety of instruments such as light, sound, graphic design, and interactive media (Hughes 2010). The huge number of techniques available makes it necessary for exhibition designers to work in interdisciplinary teams or to outsource the implementation of parts of the exhibition (Schwarz 2001). Especially the realization of interactive exhibitions is very expensive, allowing only large exhibition design companies to design and deploy such exhibitions. One approach in recent research in Human-Computer Interaction (HCI) is to provide Do-It-Yourself (DIY) Tools to enable exhibition designers from all disciplines and with varying programming skills to implement interactive exhibitions themselves (Kubitza et al. 2015; Maye et al. 2014; Petrelli et al. 2013). The need for such tools is highlighted by the example of MESO Digital Interiors, a leading exhibition

Veröffentlicht durch die Gesellschaft für Informatik e. V. 2018 in
R. Dachselt, G. Weber (Hrsg.):
Mensch und Computer 2018 – Tagungsband, 02.–05. September 2018, Dresden.
Copyright (C) 2018 bei den Autoren. https://doi.org/10.18420/muc2018-mci-0291

design company that provides the DIY Software Tool vvvv[1], a visual programming environment, which is widely used for the rapid prototyping and developing of interactive media and explicitly made for people with basic programming skills. By conducting five semi-structured interviews with exhibition designers of leading exhibition design companies in Germany we elicited requirements for the design of such DIY Tools. They have to be easy to use and therefore reduce the required programming skill to a minimum, provide direct visual feedback on interaction and should support rapid prototyping and the fast changing of settings on site. Furthermore, these tools should be specifically designed for common interactive use-cases in exhibitions. These include interactive walls (MESO 2011), tabletops (Atelier Brückner 2016), tangible interaction (ART+COM 2010) or kinetic installations (ART+COM 2008), all of which are research areas in HCI and therefore well explored. A new trend in many modern exhibitions is to immerse the visitors, casting them as actors instead of mere spectators. In such exhibitions, visitors have an actual influence on parts of the exhibition explicitly by touching the exhibits or by body movement or implicitly by entering a specific area in the exhibition (Kortbek & Grønbæk 2008; Snibbe & Raffle 2009). These immersive interaction concepts are by and large implemented using implicit and explicit proxemic interactions, which are difficult to design (Marquardt et al. 2011).

Our main contribution is the provision of the DIY Tracking Tool Argus Vision. It allows exhibition designers with basic or no programming skills the design and implementation of implicit and explicit proxemic interactions by abstracting complex tracking algorithms using multiple Kinect version 2 cameras into a simple user interface. We support the rapid prototyping and developing of proxemic interaction concepts and the change of settings on site by the use of Triggerzones, virtual three-dimensional areas, which detect the presence of visitors. To evaluate whether the tool is applicable in exhibitions and usable by exhibition designers, we used the tool in two case studies and conducted five expert interviews with exhibition designers.

2 Related Work

The implicit and explicit interaction with exhibits and installations through natural body movement plays an integral role in the creation of immersive exhibitions. The exhibitions use proxemic data and can be categorized based on their application of five proxemic dimensions as described by Greenberg et al. – distance, orientation, movement, identity, and location (Greenberg et al. 2011). In the majority of the immersive experiences in exhibitions however, only the proxemic dimensions distance and location are used. The interactive media is either controlled by the position of a person in relation to an installation, which is measured as a continuous value (MESO 2011; Snibbe & Raffle 2009) or by detecting whether a person is entering a part of the exhibition, which in contrast can be measured as a simple binary value (Gestalten & Hanschke 2011). Therefore, it is especially important for DIY Tracking Tools to provide easy access to these measurements.

[1] https://vvvv.org/, Retrieved July 09, 2018

Several toolkits have been developed with the aim to provide easier access to proxemic data using tracking-devices, albeit with different target user groups. The Proximity Toolkit (Marquardt et al. 2011) and SoD-Toolkit (Seyed et al. 2015) are toolkits for the prototyping and development of ubiquitous spaces. Both toolkits feature GUIs to monitor the tracking results and set up the devices. However, these tools were neither designed for the use in exhibitions nor for people with basic programming skills. A major disadvantage of the Proximity Toolkit is its dependence on motion capturing systems to use its full set of features, e.g., the use of fixed-features to detect whether a person is entering a specific area in the room. As these systems need markers to detect persons, they are not suitable for the use in exhibitions. The Kinect's skeleton tracking can be used to enrich the tracking-data of the motion capturing system. The SoD-Toolkit allows the use of a variety of sensors like mobile devices to detect the orientation of a person or the Leap Motion for gesture detection. To detect the position of people, it uses the skeleton tracking of the Kinect. Since both toolkits rely on the proprietary skeleton tracking of the Kinect it is necessary to position the camera in body height and parallel to the floor, which creates difficulties in hiding the hardware in an exhibition. Additionally, both toolkits provide a range of data and settings for research in proxemic interaction. A majority of these is not necessary for the use in exhibitions and only makes the user interface and the processing of tracking-data more complex. In contrast, TSPS[2] and KinectA (Honauer 2013) are tools specifically designed for the use in creative applications and exhibitions. Both feature simple user interfaces and allow the use of blob or skeleton tracking algorithms to detect objects and persons in the camera image. However, they do not offer support for multiple cameras, thus limiting the exhibition space that can be effectively monitored. Most importantly, they only provide access to continuous tracking-data, which still needs to be processed further (e.g., calculating the position of a blob in relation to an installation) which is a challenging step for people with little programming skills. Thus, even though these tools are developed for the use in exhibitions, they are still intended for designers with a background in programming.

In contrast to the related work, we provide an easy and fast way for exhibition designers to use camera-tracking that is still complex enough to implement a wide range of contemporary installations and interactive exhibits. At the same time, we support the rapid prototyping of such media remotely and on site. We especially focused on enabling designers with few or no programming skills to make use of the aforementioned continuous and binary measurements of the proxemic dimensions distance and location.

3 Argus Vision

In this section, we will introduce the DIY Tracking Tool Argus Vision. First, we will explain the architecture of the tool to show how the tool can be used in exhibition spaces. Then, we will describe its main features – communication, user detection, Triggerzones, and the user interface and specifically highlight their benefit for the usability of the tool. The source code

[2] http://www.tsps.cc, Retrieved July 09, 2018

of Argus Vision is available to the public[3] and written in Processing[4], a programming environment and language commonly used for creating interactive media. This allows exhibition designers to participate in further development of the tool.

Architecture. Argus Vision can be regarded as middleware, providing all the necessary tracking-data to be used in animations, installations, and other interactive media. Argus Vision itself is divided into two applications (see Figure 1), which are communicating via LAN – Argus Control and Argus Kinect. The latter performs the tracking using the Kinect's raw depth data. This division makes it possible to use multiple Kinects, each controlled by one computer running an entity of Argus Kinect. Furthermore, it ensures that for every Kinect the tracking algorithm performs at the same speed, which is especially important in applications where tracking-data should be visually represented without glitches, e.g. if the tracked silhouette is shown in a projection. Argus Kinect forwards the processed tracking data to Argus Control, which bundles the data from all Argus Kinect-entities and sends these results to one client, for example a programming environment for interactive media (vvvv, Processing), a game engine (Unity) or DIY Hardware Tools (Arduino).

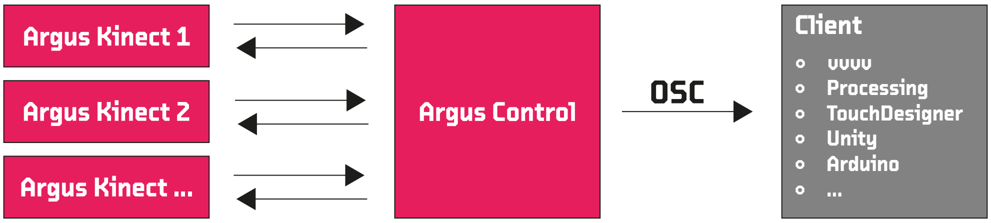


Figure 1: The architecture of Argus Vision.

Communication. In order to make the setup of a connection between Argus Control and Argus Kinect easier, we implemented a broadcasting method, automatically connecting an Argus Kinect-entity to an available Argus Control-entity if they are in the same network. As long as Argus Kinect is not connected to Argus Control, it sends a communication request via LAN in a specific time interval and waits for a response from Argus Control. If it receives a response, Argus Control and Argus Kinect automatically setup a TCP connection for the lossless transfer of settings and a UDP connection to stream images from the respective Kinects. The tracking data itself is sent via OSC, a popular and easy-to-use network protocol in interactive media. This ensures that many applications commonly used in exhibitions are able to receive and process the data.

User Detection. In most exhibitions, the hardware is hidden to not alter the desired aesthetic of the respective exhibition room. Therefore, in contrast to most other GUI-toolkits for the Kinect, which use its proprietary skeleton tracking, we developed our own background subtraction algorithm to track the contours of persons in the view of the camera. This allows the

[3] https://github.com/hcigroupkonstanz/ArgusVision, Retrieved July 09, 2018

[4] https://processing.org/, Retrieved July 09, 2018

use of Kinects from all different positions and angles. In a first step, we create a background image from the current depth values of the Kinect (see Figure 2a). The depth values of each subsequent frame are then compared to the depth values of the background image. If the difference between these values is larger than a user-specified threshold we can assume that a feature in the room has changed, e.g., that a person has moved or entered the field of view of the Kinect (see Figure 2b). In a last step, we use openCV to extract additional features from this image, such as the contour and centroid of each person (see Figure 2c).

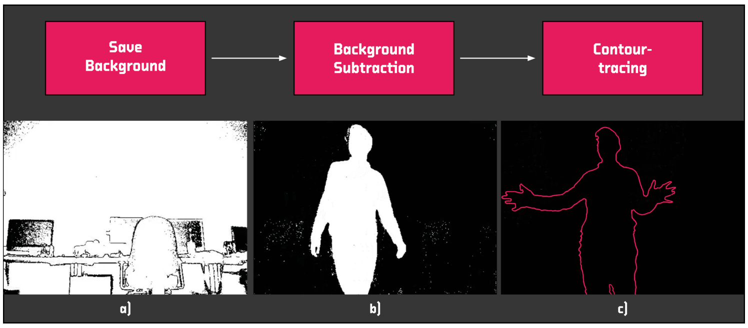
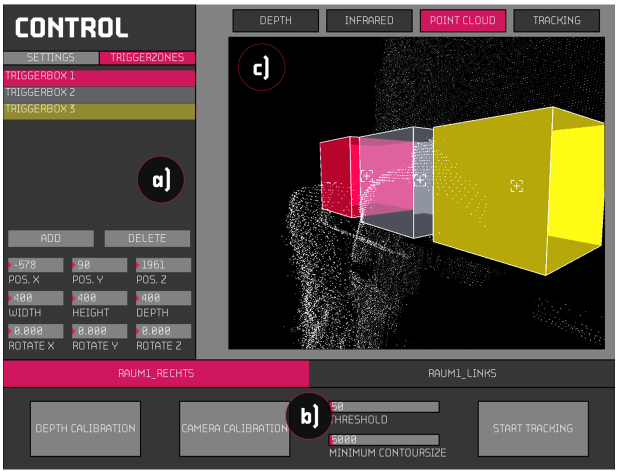


Figure 2: Results of the Detection Process of Argus Vision against an unsaved background (a), a saved background (b), and with activated contour-tracing (c).

Triggerzones. With the introduction of Triggerzones, we greatly reduce the programming effort for exhibition designers. Triggerzones are virtual areas in a room, which detect the presence of visitors and therefore resemble light barriers in their behavior. At the time of writing, we support quad-shaped areas. However, since the algorithm used to detect whether a person is in the area can be used for every convex shape, we will support other shapes, especially radii, in a future version of the tool. The Triggerzones can be positioned and rotated freely in the camera-view by either using the settings menu or by dragging and dropping in a point cloud generated from the Kinect data. Positioning three-dimensional objects like the Triggerzones in common two-dimensional images can be difficult as it is hard to estimate the distance between the objects and the camera. In order to address this, we implemented a point cloud view, which exhibition designers can use to drag and drop the Triggerzones in three-dimensional space using a virtual camera with which they can view the room from all perspectives (see Figure 3c). This makes it easy to position the areas on object surfaces or floors accurately. With this concept, we reduce the results of the camera-tracking to on-and-off-statements, which can be easily processed in other applications through simple if-then-else expressions. We support the rapid prototyping of interaction concepts by eliminating the need to write complex and time-consuming algorithms as well as the implementation of simple installations, e.g. triggering an animation when a person touches an exhibit or enters different areas in a room. Additionally, this enables exhibition designers to design their installations remotely and to adjust their interaction concepts to any other setting by repositioning, adding or deleting Triggerzones on site.

Argus Control. All of the functions available in Argus Vision are bundled into the user interface of Argus Control. It is divided into three parts: In the Communication and Triggerzone settings (see Figure 3a), the user is able to setup a connection between Argus Control and Kinect or Argus Control and a client. Moreover, one can manage the Triggerzones. The tracking algorithm – specific for every Kinect-entity – is setup in the Kinect controls (see Figure 3b). In the Kinect view (see Figure 3c), the user has access to four different visualizations of the Kinect data.



POS. X: -578
POS. Y: 90
POS. Z: 1961
WIDTH: 400
HEIGHT: 400
DEPTH: 400
ROTATE X: 0.000
ROTATE Y: 0.000
ROTATE Z: 0.000
THRESHOLD: 50
MINIMUM CONTOURSIZE: 5000

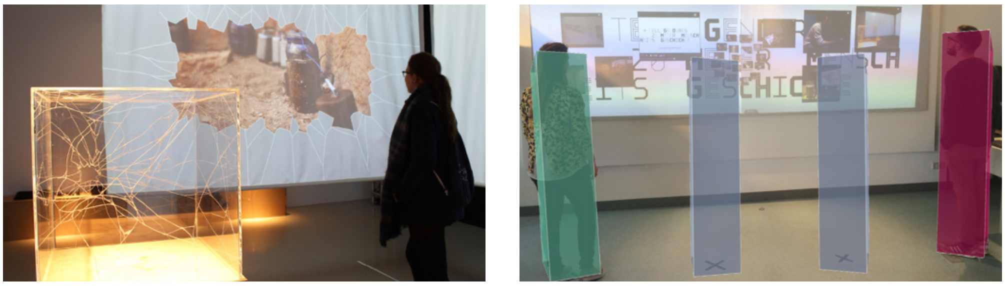
Figure 3: The user interface of Argus Control is divided into Communication and Triggerzone settings (a), Kinect controls (b) and the Kinect View (c). In the latter, three Triggerzones are placed on the edge of a table.

The depth and infrared streams can be used to visualize the room regardless of the room's lighting, while the tracking view shows all detected persons in the room. As described in the previous subsection, the point cloud view is used for managing the existing Triggerzones. By using all of the functions of Argus Vision, users have access to the following data:

- For each person: id, age, centroid, contour, velocity, acceleration
- For each Triggerzone: id, occupancy level, occupancy per person

4 Evaluation

To evaluate whether Argus Vision is applicable in the area of exhibition design, we used the tool in a room-spanning installation at an exhibition and in a proof-of-concept study in our lab. Furthermore, we conducted five expert interviews with exhibition designers to assess the usefulness of the tool for the specific domain.



a) b)

Figure 4: a) A visitor destroyed the virtual glass by getting too close to the projection in the installation "Bruch". b) By entering one of the four Triggerzones in front of the display, two persons can view images at their positions.

Case Study "Bruch". We displayed the interactive installation "Bruch" for six weeks at a local exhibition space. In this installation, we controlled projected animations via the positions of persons in the room. These projections showed virtual glass, mimicking an actual showcase, which was positioned in the middle of an exhibition room (see Figure 4a). The virtual glass either cracked or burst depending on the position and distance of visitors in relation to the projection. With Argus Vision, we were able to implement the glass-animation remotely, using the mouse and keyboard as a substitute for the actual presence of persons. In the exhibition space itself, we mounted all hardware, including three Kinects on a truss on the ceiling facing the floor. To create an interactive, immersive experience for the visitors, we then simply substituted the input devices used in the prototyping phase with the centroid values from Argus Vision. The installation operated maintenance-free during the course of the exhibition.

Case Study "smartPerform". In a proof-of-concept study in our lab, we integrated Argus Vision in an already existing DIY Tool, smartPerform[5], which allows the creation of interactive presentations and tabletops. smartPerform is used in exhibitions as well. Here, we simulated a public display that interacts with the movement of passers-by. To direct the attention of people walking by, we split the display into four vertical areas. As long as persons stood in front of one of these areas, images and videos could be watched on the screen. If the area was empty, the pictures disappeared. This is a typical scenario for the use of Triggerzones. We positioned four Triggerzones on the floor in front of the respective areas on the screen and used these to control the display of the pictures (see Figure 4b). smartPerform itself uses a node designer, where functions are abstracted into connectable nodes, to enable the implementation of interactive scenarios. We were able to use this node designer to filter messages from Argus Vision using regular expression, effectively eliminating the need to write code. However, since smartPerform does not support OSC, we had to use vvvv as broker, which received the OSC messages from Argus Vision and forwarded them as UDP messages to smartPerform. As in the case study "Bruch", we were able to design the content of the presentation in advance

[5] http://www.smartperform.de/, Retrieved July 09, 2018

by simulating interaction through simple mouse controls and setup the actual installation in a short amount of time.

Expert Interviews. In addition to the case studies, we conducted five expert interviews with exhibition designers from leading exhibition design companies. Since large exhibition design companies typically outsource the implementation of interactive media (Schwarz 2001), the exhibition designers did not use the tool themselves, but were shown the functions and user interface of Argus Vision during the interview. We chose this approach because we anticipated quick and valuable feedback from experienced exhibition designers. In further research, we aim to provide the tool to smaller exhibition design companies to see how they apply the tool in real world exhibitions. Since the experts did not to use the tool themselves, we asked questions about the perceived usability of the tool, possible use cases and possibilities for further development. The interviewees judged the user interface, its comprehensibility, as well as the comprehensibility of the user detection algorithm positively. Moreover, multiple experts could envision abstract use cases for the Triggerzones. One interviewee specifically highlighted its importance for rapid prototyping, especially in a domain "[…] where the content and presentation comes first […]"[6]. Another assumed "a huge demand" for Argus Vision "for simple installations, especially in temporary exhibitions and events". Moreover, several interviewees could envision that exhibition design teams would be able to make use of the tool. The experts also suggested possibilities for further development. These include the combination of multiple cameras to one virtual camera by blending the intersections of their field of view, the possibility to activate a skeleton or gesture tracking and the development of a frontend-editor, which allows the control of common interactive media like animations, lights, or sound in the user interface of Argus Control. One interviewee described the tool as "one component in an ecosystem of different technologies used in exhibitions". He continued that "if exhibition designers knew what they could do with it", Argus Vision alone "enables sensational possibilities". Consequently, this expert does not regard the further development of Argus Vision as the most important future work, but "the strategic and long-term isolation of the components needed in the museum of the future" and thus, the deployment of similar DIY Tools for each of these components.

5 Results and Future Work

We demonstrated that Argus Vision can be used in exhibitions by outlining the successful integration of the tool in an installation at an exhibition as well as in a proof-of-concept study in our lab. In the latter, we had to use another DIY Tool as middleware to convert the OSC messages of Argus Vision to UDP. To enable a wider range of applications to receive messages from Argus Vision, we will provide additional network protocols in Argus Vision, such as TCP, UDP and TUIO. The five expert interviews with exhibition designers gave us important feedback on the tool and its usefulness for the target group. We were able to gather ideas for further development of Argus Vision. Overall, the interviewees judged the tool favorably and

[6] All citations are translated by the authors

could envision that especially small exhibition design companies would use Argus Vision. In the future, we will provide the tool to exhibition designers of such exhibition design companies to evaluate the tool in a real-world setting. Furthermore, we believe that the concept of Triggerzones should be extended. The Triggerzones' size and position are only set once, at the implementation of an installation. Additionally, the current version of Argus Vision only allows the use of quad-shaped areas, yet other shapes can be of interest to both exhibition designers and researchers. One example scenario would be the modeling of proxemic interactions using circle-shaped Triggerzones that follow the tracked users. In another scenario, one could imagine the size of the Triggerzones changing depending on the number of people in the vicinity. While these are interesting applications, the design space of such applications and their applicability in a real-world context still needs to be explored. Moreover, it should be noted that adding more complex functions to the tool directly contradicts our approach to making the tool easily usable for exhibition designers with no background in programming.

We see Argus Vision as a module for using camera-tracking-technology among a range of other technologies from which interactive installations are created. Therefore, we see our future work in line with the vision of one of our interviewees: We aim to investigate current exhibitions, elicit the technological components used and provide DIY Tools like Argus Vision for all of these different components. These tools should integrate seamlessly, making it possible to combine the functions of different DIY Tools to rapidly prototype and create complex interactive exhibitions.

6 Conclusion

Argus Vision is a DIY Tracking Tool that enables interdisciplinary exhibition designers to design and implement immersive exhibitions using implicit and explicit proxemic interactions. Using Triggerzones, freely positionable virtual areas in the room, we support the rapid prototyping of interaction concepts and last-minute changes of settings on the exhibition site. We successfully used the tool in a real-world case study in an installation as well as in a proof-of-concept study in our lab, proving its applicability in exhibitions. We also conducted expert interviews showing that Argus Vision fulfills the requirements of the domain of exhibition design. By hiding the application of complex tracking algorithms on the depth data of multiple Kinects behind a user interface, exhibition designers can focus on designing engaging exhibitions using interactive installations, without worrying about complex programming or technological possibilities.

References

ART+COM (2010). Salt Worldwide. Retrieved April 03, 2018 from https://artcom.de/en/project/salt-worldwide/

ART+COM (2008). Kinetic Sculpture. Retrieved April 03, 2018 from https://artcom.de/en/project/kinetic-sculpture/

Atelier Brückner (2016). Erlebnis Europa – Europa Experience. Retrieved April 03, 2018 from http://www.atelier-brueckner.de/en/projects/erlebnis-europa-europa-experience

Buchanan, A. (2008). Predicting User Behavior – The Creation of the Immersion Installation. *Computers in Entertainment – Media Arts 6*(3), New York City, New York, USA: ACM. S. 34:1-34:9.

Gestalten & Hanschke, V. (Hrsg.) (2011). A Touch of Code – Interactive Installations and Experiences. Berlin, Germany: Gestalten.

Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R. & Wang, M. (2011). Proxemic interactions: the new ubicomp?. *ACM Interactions 18*(1), S. 42-50.

Honauer, M., (2013). Designing Device-less Interaction A Tracking Framework for Media Art and Design. In: Boll, S., Maaß, S. & Malaka, R. (Hrsg.): *Mensch & Computer 2013 - Workshopband*. München: Oldenbourg Verlag. S. 535-538.

Hughes, P. (2010). *Exhibition Design*. London, UK: Laurence King.

Kortbek, K. J. & Grønbæk, K. (2008). Communicating art through interactive technology: new approaches for interaction design in art museums. In *Proceedings of the 5th Nordic conference on Human- computer interaction: building bridges*. Lund, Sweden: ACM. S. 229-238.

Kossmann, H. & de Jong, M. (2010). *Engaging Spaces: Exhibition Design Explored*. Amsterdam, Netherlands: Frame.

Kubitza, T., Thullner, S. & Schmidt, A. (2015). VEII: A Toolkit for Editing Multimedia Content of Interactive Installations On-site. In *Proceedings of the 4th International Symposium on Pervasive Displays*, Saarbrücken, Germany: ACM. S. 249-250.

Marquardt, N., Diaz-Marino, R., Boring, S. & Greenberg, S. (2011). The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. Santa Barbara, California, USA: ACM. S. 315-326.

Maye, L. A., McDermott, F. E., Ciolfi, L. & Avram, G. (2014). Interactive exhibitions design: what can we learn from cultural heritage professionals?. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. Helsinki, Finland: ACM. S. 598-607.

MESO (2011). Reactive Light Effect for Pedestrian Tunnel. Retrieved April 03, 2018 from https://meso.design/en/projects/city-of-aarau-reactive-light-effect-for-pedestrian-tunnel

Petrelli, D., Ciolfi, L., Dijk, D. v., Hornecker, E., Not, E. & Schmidt, A. (2013). Integrating material and digital: a new way for cultural heritage. *interactions 20*(4), S. 58-63.

Schwarz, U. (2001). Entstehungsphasen einer Ausstellung. In Schwarz, U. & Teufel, P. (Hrsg.): *Museografie und Ausstellungsgestaltung*. Ludwigsburg, Germany: avedition. S. 16-37.

Seyed, f., Azazi, A., Chan, E., Wang, Y. & Maurer, F. (2015). SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. Madeira, Portugal: ACM. S. 171-180.

Snibbe, S. S. & Raffle, H. S. (2009). Social immersive media: pursuing best practices for multi-user interactive camera/projector exhibits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Boston, USA: ACM. S. 1447-1456.