The Media Façade Toolkit: Prototyping and Simulating Interaction with Media Façades

Sven Gehring, Elias Hartz, Markus Löchtefeld, Antonio Krüger
German Research Center for Artificial Intelligence (DFKI)
Campus D3 2, Saarbrücken, Germany
{sven.gehring, elias.hartz, markus.loechtefeld, krueger}@dfki.de

ABSTRACT
Digital technologies are rapidly finding their way into urban spaces. One prominent example is media façades. Due to their size, visibility and their technical capabilities, they offer great potential for interaction and for becoming the future displays of public spaces. To explore their potential, researchers have recently started to develop interactive applications for various media façades. Existing development tools are mostly tailored to one specific media façade in one specific setting. They usually provide limited means to incorporate interaction by a user, and the applications developed are limited to running on only one particular media façade. In this paper, we present a flexible, generalized media façade toolkit, which is capable of mimicking arbitrary media façade installations. The toolkit is capable of running interactive applications on media façades with different form factors, sizes and technical capabilities. Furthermore, it ensures application portability between different media façades and offers the possibility of providing interactivity by enabling user input with different modalities and different interaction devices.

Author Keywords
Media façades; simulation; prototyping; interfaces

ACM Classification Keywords
H.5.2 Information interfaces and presentation: User Interfaces.

General Terms
Design, Human Factors.

INTRODUCTION
Nowadays, urban landscapes are more and more dominated by digital installations such as situated public displays [23], video walls and so-called media façades [4, 29]. The Bugis+ Illuma Shopping Center in Singapore[1] and the ARS Electronica Center[2] in Linz, Austria are only two examples out of hundreds of such façades. The façade of a building is much more than just the outer shell, separating and protecting the interior from the outside. It is the visual representation of a building in the public space. It shapes both the building's societal and cultural role, determining how the building and the surroundings are perceived [5]. As such, it is one of the most important design points for architects. With the term media façade, we generally denote the idea of turning the outer surface of a building into a gigantic public screen by equipping it with light emitting elements [5, 12, 28]. Haeusler defines a media façade as "a façade into which dynamic communication elements are embedded" [15]. The general nature of this definition results from the broad spectrum of existing media façades. They strongly vary in construction, technology, capabilities and scope, which leads to development processes and tools that are tailored to one specific setting. Furthermore, media façades vary in size, location, possible viewing angles and form factor. Since they are embedded into the architecture of the hosting building, in contrast to situated public displays, media façades can have arbitrary 3-dimensional (3D) form factors, as can be seen in Figure 2. Contrary to embedding light emitting elements into the architecture of a building, Scheible and Ojala turned arbitrary objects in urban environments into media façades, utilizing a projector and mobile device to interact with the projection [27]. The form factor or shape of a media façade is a critical issue when developing digital content (see Figure 1).

With the integration of digital technologies, such as projectors, light emitting or display elements, building façades in

[1]http://www.bugis-plus.com.sg
[2]http://www.aec.at

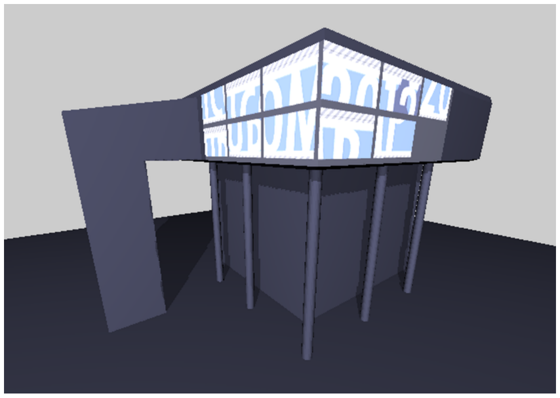


Figure 1. The media façade toolkit displaying a simulated scene. An interactive application is running in real-time on a simulated media façade within a 3D environment.

urban spaces offer new possibilities for interaction. Furthermore, the architectural and technological design of the façades raises the need for new approaches to plan and construct such façades, as well as to develop novel interactive installations to exploit the enormous potential of media façades. According to Schoch [28], they are paving the way for new ambitious concepts to provide digital content and interaction. A media façade is a permanent part of the urban space; its context and surroundings affect how content is perceived and experienced as well.

Up to now, there has been a lack of development and prototyping tools for media façades. Due to the characteristics of media façades, existing toolkits are tailored to one specific setting and to the physical and technical properties of the particular media façade. They provide a simplified representation of the façade, its surroundings and the actual hardware utilized in the setting. Besides inhibiting the portability of the developed systems from one media façade to another, this increases the efforts needed for developing and prototyping interactive installations. Dalsgaard and Halskov identified eight key challenges for designing urban media façades [8]. They argue that the urban settings media façades are usually situated in call for new forms of interfaces or alternative assemblies of existing ones. In addition, new installations of media façades need to be integrated into the existing physical structures and surroundings, leading to complex form factors. Furthermore, the developed content needs to suit the medium, meaning the content has to fit the format of the display and the kinds of interaction intended to be supported, as well as shifting light and weather conditions needing to be taken into account. Identifying and overcoming such design challenges is an integral part of the development process. Existing tools for simulating and prototyping interactive systems for urban public environments do not sufficiently support designers and developers facing the aforementioned challenges. Existing tools are mostly tailored to one specific media façade in one specific setting. They neither provide generalized means to display applications on the façade, nor do they offer possibilities to incorporate interactivity with external input devices or ways to simulate changing conditions of the area surrounding the media façade.

In this paper, we present a generalized media façade toolkit for rapid prototyping, which is capable of mimicking large-scale media façade installations, specifically of any size, form factor, technology and hardware. We follow a modularized design approach by strictly separating building model, media façade, application and user interaction. The toolkit is capable of running arbitrary interactive applications on arbitrary media façades. We describe how the modular design of the toolkit ensures application portability between different media façades and the possibility of providing interactivity by enabling user input with different modalities and different interaction devices. With the simulator, we provide a flexible simulation, test and development tool for designers and researchers alike, addressing the diversity of media façade settings. We further report on the feedback from first users developing interactive installations using the media façade toolkit.

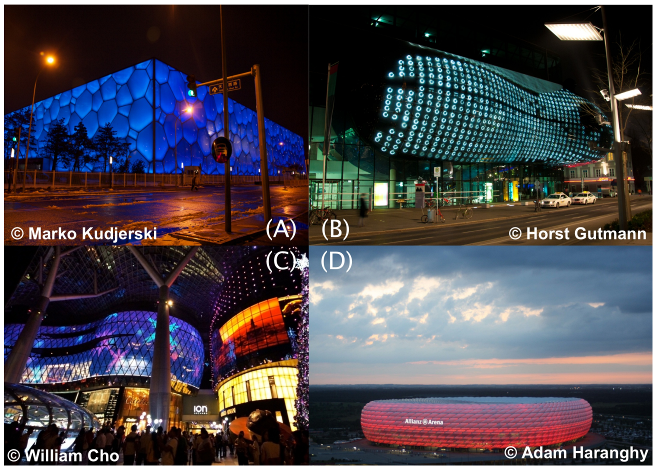


Figure 2. Media façades of different sizes and form factors: (A) The National Aquatics Center, Beijing, China (Photo: Marko Kudjerski, 2012), (B) Kunsthaus Graz in Graz, Austria (Photo: Horst Gutmann, 2011), (C) the ION Orchard building in Singapore (Photo: William Cho, 2009) and (D) the Allianz Arena in Munich, Germany (Photo: Adam Haranghy, 2009).

RELATED WORK

Designing interactive installations for media façades has been extensively explored by researchers and designers alike, providing valuable insights into the design and deployment process, as well as into problems occurring therein. We identified simulation and prototyping tools for digitally augmented environments as a further area related to our work. To follow, we give an overview on the most relevant works from these categories and we discuss how they are related to the work presented in this paper.

Designing Interaction for Media Façades

One of the first systematic analyses of the design of interactivity for media façades was presented by Dalsgaard et al. [9]. They state that the location of the façade, its scale and the technologies utilized have an impact on how the displayed content is experienced by a user. Dalsgaard and Halskov expanded this work [8], identifying the aforementioned eight key challenges when designing applications for media façades. Boring et al. describe how they applied Touch Projector [2] to allow multiple users to simultaneously interact with a media façade through live video on mobile devices [3]. They point out that due to the lack of suitable testing and prototyping tools, their initial implementation performed poorly on the target façade in the real-world setting for variable viewing distances and changing lighting and weather conditions. They needed to perform additional design revisions to solve those issues. Böhmer et al. confirm that prototyping for media façades is a critical part in the design process and the lack of generalized tools makes it impossible to reproduce installations in a controlled setting [1]. Böhmer et al. developed a dedicated virtual simulator tailored to the specific façade to embed their content into a 3D model for testing. Wiethoff and Gehring report on their experiences utilizing Lightbox [31] — a miniature hardware toolkit that we will discuss in the section “Simulation & Prototyping Toolkits“— for reproducing the hardware setup of the façade to simulate and test the interplay of a user's mobile device and the hardware interfaces of the media façade [32]. They report that reproducing an installation on a small scale might be suitable for prototyping the hardware interplay, but comes with the danger of missing important factors like the visibility of the model. Possible viewing angles may differ from the real-world counterpart and the technology of the setup and conditions on site are difficult to replicate to a full extent. Fischer et al. present Spread.gun, an urban media intervention tool allowing users to shoot text messages onto a projected media façade [11]. They report on how they misjudged visibility and the appearance of colors during testing in a controlled lab setting. They outline the discrepancies between testing in a controlled setting and deploying an application in the wild. They further explored the spatial settings of media façades to provide a better understanding [10].

The authors of the aforementioned work report on their experiences designing interactive installations for media façades. They state that the lack of prototyping and testing facilities for controlled settings and the enormous discrepancies between a controlled lab setting and the actual on-site conditions complicate the design process and that they are an important issue in the design process. By introducing the media façade toolkit, we take a first step towards addressing these issues by providing a prototyping and simulation toolkit for interactive applications for media façades.

Simulation & Prototyping Toolkits

Jacobs introduces the concept of *rapid prototyping* and describes the level of accuracy that can be reached, providing large benefits to designers at low cost [17]. Nakanishi shows that physical and digital representations can also be combined, although he remarks that this approach comes with the supplementary issue of synchronizing the virtual and physical parts of the toolkit [21]. Nakanishi et al. also provide a framework for hybrid prototyping, called the *City Compiler* [22]. It aims at bridging the gap between the two realms. Providing designers with an impression of the application's performance in a virtual environment and allowing "interactive trial-and-error" testing, the City Compiler supports an iterative design process. However, current tools to completely execute an application on a simulated façade are still very limited. They are tailored to one specific façade and they neither support interactivity, nor do they provide opportunities to model and simulate the surroundings of the media façade. Marquardt et al. introduced the *Proximity Toolkit* [20] to simplify the exploration of interaction techniques by supplying fine-grained proxemic information between people, portable devices, large interactive surfaces, and other non-digital objects in a room-sized environment. Their toolkit supports rapid prototyping of proxemic-aware systems and it includes different tools to observe, record and explore proxemic relationships within a 3D space. It is designed in a modularized manner and it separates sensing hardware from the data model. Hence, different sensing technologies can be substituted or combined to derive proxemic information.

A common approach is to build prototyping toolkits using simplified *small-scale* models of the media façade, which comes with further limitations. Since they cannot sufficiently map all features of a media façade, they are usually built for modeling one specific feature. Wiethoff and Blöckner introduced Lightbox [31], a hardware toolkit aiming to provide designers a way to test the hardware interplay of the particular technologies used to assemble a media façade, as well as the input devices, enabling interaction with the façade. As mentioned before, Wiethoff and Gehring [32] applied Lightbox to prototype the hardware interplay of an interactive application for the ARS Electronica center in Linz, Austria. The media façade of the ARS Electronica center is created by equipping every window of the building with DMX controllabe RGB LEDs. This results in an approximately 20x25-pixel media façade. Lightbox can be used to get a general impression of how an application might look on such a low resolution façade and to test the hardware interplay before deployment onto the actual façade. However, its main purpose is simulating hardware and its visual prototyping capabilities are limited to façades with a 2D form factor. Hence, it covers only a small portion of the full media façade. A software-based simulator could avoid many of the issues introduced by a physical toolkit such as Lightbox, or when used in conjunction with physical prototypes like Nakanishi [22] proposes, mitigate them to a certain degree. With a modularized simulator, the prototyping environment could be adapted to a new task with little effort, without sacrificing simulation precision. However, currently available software tools are at least as specifically tailored to the target façade as physical toolkits. Due to their hard-coded nature and the often missing abstraction between application and façade, they force the programmer to fully implement prototypes in the early stages of the development process in order to test them. To be able to use such a simulation tool during the development process, a developer is forced to commit his application to this specific installation. Transferring the application to another façade is not possible without re-writing large parts of the application.

Both low-fidelity prototypes representing simplified and abstracted models and high-fidelity prototypes mapping the target setting as accurately as possible are created from interactive systems during a design process [26]. The choice of prototypes depends on (1) how accurately they need to represent their real-world counterparts, and (2) the details and functionality intended to be addressed. They depend on the aim designers are pursuing with a prototype, which is commonly referred to as the prototype's scope [19]. Such tools help designers to create prototypes early in the design process in a more efficient and cost-effective manner. When designing for media façades, however, there is no common ground for creating experience prototypes. We need to ask ourselves why it is so difficult to build a toolkit powerful enough and flexible enough for the development of interactive installations for media façades, which can be transferred from one media façade to another — independent of the media façade's form factor and its technical specifications — without tailoring the developed application to one specific setup. The answer lies in the huge variety of media façades and the dynamic public context in which they are deployed. While certainly not all features can be sufficiently captured within a prototyping environment, we need to identify the key features that are essential for providing a generalized simulation framework. We build upon the aforementioned ideas to provide a more general, flexible and powerful approach, which supports the integration of interactivity with different modalities and input devices, as well as the full integration of existing applications into a virtual representation of a media façade. In [14], Gehring and Krüger introduce the idea of applying cartographic map projections to create 2D map representations of the 3D surface of a media façade. We integrate this idea into the media façade toolkit to dynamically map 2D application canvases onto arbitrary shaped media façades. By doing so and by designing the toolkit in a modular manner, such that application, façade and 3D model of the building are strictly separated, we ease the portability of applications from one media façade to another.

MEDIA FAÇADE TOOLKIT

When designing interaction in general, Rogers et al. recommend an iterative design approach and the use of low- and high-fidelity prototypes [25]. For regular graphical user interfaces (GUIs), researchers can easily construct a prototype and choose from a variety of tools and approaches [6]. For media façades, this is hardly possible due to their specifications and the highly dynamic and public context they are situated in. However, when designing interactive installations for media façades, the possibilities for prototyping are limited. Due to their technical specifications, media facçades are mostly not visible and active during daylight. This restricts time for testing during the development process. In addition, the outcome of early testing is already visible to a large audience, since media façades are usually situated in urban public spaces. As a result, not many design iterations are feasible on the media façade itself. Hence, researchers and designers are forced to do most of the testing in an artificial lab setting, using projected or regular displays. The physical and technical properties of a media façade prevent building a full-scale replica for development. In general, we have to face the following challenges among others during the development and deployment process of applications for media façades. These have been derived from the design challenges described by Dalsgaard and Halskov [8]:

1. **Various irregular form factors**: Media façades are embedded into the physical and architectural surroundings of an existing building, which results in various 3D form factors that have to be addressed during development.

2. **Robustness and stability**: Since media façades are situated in a public outdoor environment, changes in the environment, such as changing weather and lighting conditions, can influence the visibility and hence the usability of the application.

3. **Limited testing**: Light emitting media façades are usually not visible during bright daylight, which limits the timeframe for testing to a few hours. In addition, all content displayed on the media façade is immediately exposed to a large audience.

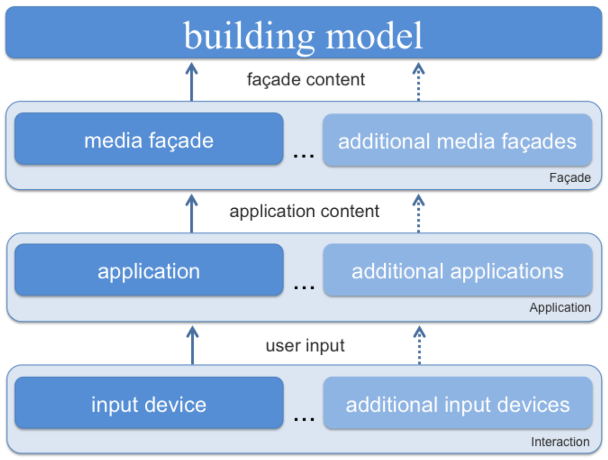


Figure 3. The structure of the toolkit. The building model, media façades, applications and input devices are organized in separate modules, retaining a flexible structure. Communication takes place only between pairs of modules. Input devices send user input to the applications, the applications communicate their visual content to media façades and they communicate the façade content to the building model.

4. **Content development**: The content has to (1) fit the size, resolution and form factor of the display and (2) it needs to be appropriate in a highly public context. Furthermore, due to the size and visibility, content can be perceived differently on a media façade than on a miniature model.

5. **Interactivity**: Due to their size and therefore required minimal viewing distance, user interaction by direct touch input is not possible. New forms of interaction-at-a-distance need to be integrated.

6. **Portability**: Existing development tools are highly tailored to a specific façade and setting. Hence, applications are also tailored to one setting, inhibiting their portability to other façades or displays.

The aforementioned challenges can be addressed by creating special purpose prototyping toolkits, addressing some of the challenges, but not covering all of them. However, existing toolkits come with limitations. With the media façade toolkit, we address the aforementioned limitations. Furthermore, in his classification of media façades, Haeusler [15] lists a number of distinguishing factors, including their size, shape, form factors and display technology, which also need to be considered when creating prototyping toolkits. Our simulator needs to capture these properties as accurately as possible. The building itself into which the media façade is embedded needs to be captured in the simulator. The physical properties of the building, as well as the surrounding architectural space, dictate possible viewing angles for a user while interacting with content of the façade. They further influence the general visibility of the media façade. Furthermore, the scale, size and form factor of the media façade are important factors for correctly mapping and displaying content on the façade. As pointed out by Haeusler [15], media façades can be categorized by their technical specifications. The utilized display technology influences how an application performs un-

```
<SIMULATOR Input type="Keyboard">
   <input>
      <signal>LEFT</signal>
      <code>LEFT_ARROW</code>
   </input>

   <input>
      ( . . . )
   </input>
   (...)
</SIMULATOR Input>
```

Figure 4. Specification of user input. The supported input signals of the simulator (e.g., LEFT) are mapped onto input codes of external input devices (e.g., LEFT_ARROW). An input mapping needs to be created for every supported input device.

der different lighting conditions. For example, colors might be perceived differently and the general visibility of the content might change under different lighting conditions. Furthermore, in order to develop interactive applications for media façades, user input and interaction need to be adequately modeled in the simulation.

A general simulation toolkit for interactive media façade installations needs to support (1) any number of users using (2) any class of input device to (3) control an application (4) running in real-time on a media façade of (5) any technology, shape and size. The media façade can be (6) embedded in any building with (7) surrounding architecture. Due to the characteristics of media façades and the dynamic nature of the environment they are deployed into, addressing all these issues to a full extent in a simulation is a challenging task. However, we can provide a toolkit that allows simulating interactive applications on arbitrary media façades by designing the toolkit in a modularized design approach, providing a high level of flexibility.

Architecture

The media façade toolkit follows an extended *model, view, controller* pattern, which was introduced by Krasner and Pope [18]. As can be seen in Figure 3, it is designed with a modular approach, strictly separating building models, media façades, applications and interaction.

The modules are organized hierarchically. By placing abstraction layers; specifying the communication format between the modules and allowing communication in only one direction between two particular connected modules, we prevent dependencies while providing means to communicate and send data. The toolkit's modules communicate as follows (see Figure 3): Input devices send data to applications. Applications send their content to façades, and only façades communicate with the model directly. By doing this, we further provide the possibility of replacing instances of certain modules while retaining all other components. E.g., the building hosting a media façade can be replaced easily on the fly and without modifying any other component, such as the application or the media façade itself. Exploiting this architecture is the key for designing a flexible framework. By limiting the communication to the next element in the hierarchy, we also limit the number of abstraction layers needed.

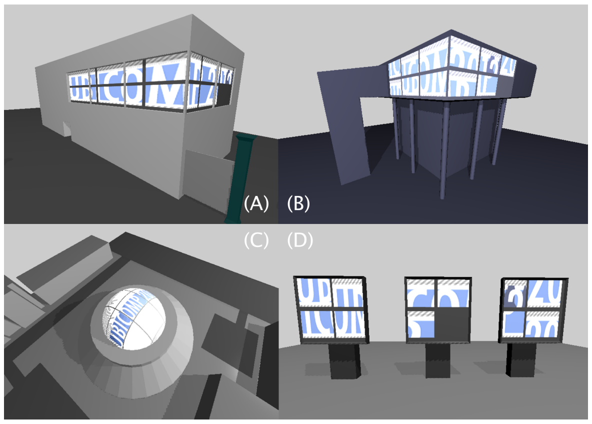


Figure 5. A puzzle game mapped to four media façades with different form factors. (A),(B): Rectangular façades. (C) A spherical façade and (D) a rectangular façade which consists of three non-coherent parts.

We now give a detailed description of the modules contained in the toolkit.

Interaction

The interaction module of the simulator defines the supported interaction techniques and specifies the mapping of user input to simulator commands. As the interaction and application modules are strictly separated, a developer can develop and specify the input and interaction modalities as so-called input senders, independent from the actual application. An application only needs to specify which internal commands, signals or gestures are supported within the application. To be independent from any input device, applications listen for input signals according to the *Observer Pattern*, introduced by Gamma et al. [13]. The developer can specify in XML (http://www.w3.org/xml) notation how the signals sent from an arbitrary input device such as a keyboard, mouse, smartphone, etc. map to the supported input commands of the application. An example of a mapping from external to internal signals is depicted in Figure 4. Input senders are not necessarily concrete physical devices held by the user. All possible interaction techniques and devices can serve as input senders, as long their output can be encoded in the described mapping. Dalsgaard and Halskov [8] give examples of installations that also support whole body and gestural interaction without dedicated input devices.

Application

An application can be any program producing visual application content that can be displayed on a digital screen. An application can support interaction by listening for and reacting to user input from a specified interaction module. The number of parallel users in this case is not restricted by the simulator, but it may very well be restricted by the application itself, if desired. Due to the modularized design of the simulator, an interactive application supports all interaction techniques and modalities specified in the simulator at the same time, without forcing the developer of an application to choose one. The input commands only need to be specified once as described in Figure 4. However, support for interactivity is optional, in order to also support visualizing videos, animations or general

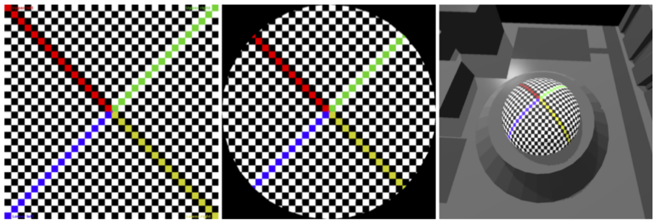


Figure 6. Mapping an application onto a media façade model as a texture. To get a better impression of the mapping and the introduced distortion, the texture shows a test image rather than an actual application

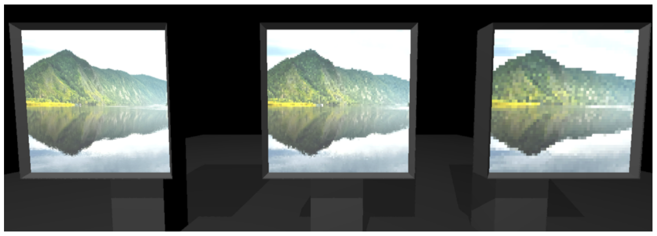


Figure 7. An application displayed on three façades with three different resolutions. From left to right: full, half and one tenth of the resolution of the original application

visual content on a media façade. Figure 5 depicts an application which was mapped onto four different media façades.

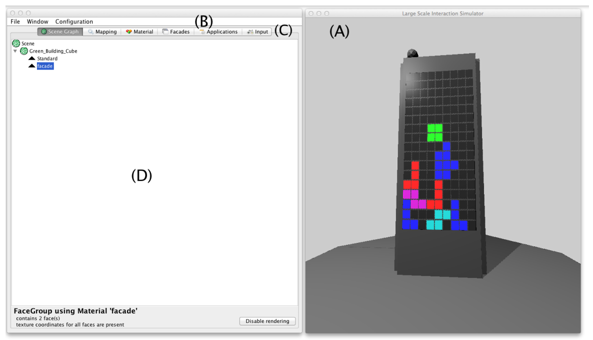
Media Façade
Within the simulator, a module which transforms the visual output of an application into content that can be displayed on the building model is denoted as a media façade. The visual appearance of the application is rendered into a separate frame buffer. For the rendering, the properties of the façade model such as its resolution, the supported color model or lighting conditions of the environment are taken into account. A model can include multiple media façades, but one media façade can show at most one application. In a case where multiple applications are intended to run on one media façade, the media façade can be divided into individual media façades, each holding one application.

Displaying the content of an application on a media façade is realized by mapping the application's content frame buffer onto the media façade in the 3D model as a texture. First introduced by Catmull [7], texturing has become a widely used and common technique. The term texturing denotes the process of covering the surface of a 3D object with 2D images, which are called *textures*. Heckbert defines a texture as *a detailed pattern that is repeated many times to tile the plane, or more generally, a multidimensional image that is mapped to a multidimensional space* [16]. This allows 2D content to be displayed on a model with a 3D form factor. Applications developed for a regular, rectangular digital screen, such as a common desktop monitor or a situated public display, usually have a well-defined application frame with a quadratic or rectangular shape. Hence, this results in a quadratic or rectangular texture when mapping such an application onto the surface of a 3D object. Within the media façade toolkit, the application's content frame buffer in general has a well-defined rectangular shape as well. By supporting transparency, the toolkit also allows for mapping irregular shapes onto a façade. This can be used to map applications onto media façades having a non-cohesive shape (see Figure 2 B). In this case, the application frame buffer can be considered as a rectangular bounding box for an arbitrary and not necessarily coherent form.
A user can mark every surface in the 3D model as a possible media façade by naming its texture as such. Stahl and Haupert illustrate a way to capture an application's visuals, mirroring it into a virtual display in real-time [30]. However, the virtual displays in this case have a rectangular shape and a 2D form factor. To compute the texture mapping, we adapt the idea of Gehring and Krüger [14] to apply cartographic map projections. Since they apply map projections to create 2D maps from a 3D surface, we invert the projections to map a 2D image onto a 3D surface (see Figure 6). For any point on the face in 3D space, a matching point in 2D space can be interpolated. The texture is calculated out of a finite number of pixels but the interpolation result is continuous; therefore the color value of this point on the texture often composes multiple pixels, e.g. bi-linearly filtered in the x- and y-directions. This value is incorporated during the shading process. The final color of a fragment is made up out of the result of the Phong shading computation [24] and the sampled value. The mapping of the application onto the façade is a critical issue and choosing the right mapping depends on the shape of the underlying building. Gehring and Krüger provided mappings for the most common form factors [14], which we use as a foundation. There is no canonical way to wrap a texture around a 3D surface, and reducing the dimension of the space often comes at the cost of not all edges between vertices being preserved, resulting in distortions. When mapping an application onto a façade, as depicted in Figure 6, an initial test image is rendered onto the façade, visualizing distortions. The mapping can then be adjusted manually with controls provided in the toolkit.

Model
The Model describes a 3D representation of the physical properties of the building hosting the media façade, as well as its surrounding architecture. It contains objects that are present in the scenery, as well as lighting and weather conditions and any other external factors that are relevant for the scenery. This information can be included at an arbitrary level of detail. Furthermore, the Model describes all areas that could possibly serve as media façade locations. Any surface in the 3D model can be turned into a possible location for hosting a media façade by simply naming the surface as such. When visualizing the content of an application on a media façade, the Model data is used to calculate occlusion, visibility and the mapping of the application's content onto the media façade's surface.

The media façade toolkit supports 3D scenes in the common *Wavefront*[3] **.obj** format. The basic components of the **.obj** format are vertices, texture coordinates and normals. These components can be combined into faces. Faces that are not triangles are broken apart into triangles during the import of

[3]http://goo.gl/quq8h



FaceGroup using Material 'facade'
contains 2 face(s)
texture coordinates for all faces are present

Figure 8. The interface of the media façade toolkit: (A) the rendering view, (B) the control view, (C) control groups and (D) the detailed settings.

the scene. The toolkit represents the entire scene as a list of model objects in the rendering module, which we describe the follow. The Model comprises a single object in the scene, containing all objects forming the scene.

Rendering

The rendering module is the core module of the toolkit. It holds the main program loop, which is responsible for controlling the entire simulation process. Only one rendering module can exist within the media façade toolkit at all times. Hence, the rendering module is implemented as a *Singleton* as defined by Gamma et al. [13]. This means that only one instance of the renderer exists at any time. The rendering module contains and manages all models, applications, façades, buffers and input senders needed for the simulation. The three main responsibilities of the renderer are: (1) When starting the simulation, the rendering module initially starts all applications that will be included in the current simulation, regardless of whether they are currently mapped onto media façades. Applications that are associated to a façade are automatically rendered on the particular façade. (2) The renderer is responsible for the general rendering of the 3D scene displayed on the screen. Within this task, all buffers holding information on the scenery are rendered using appropriate shaders. The application frame buffers holding the visual content of the applications are added as textures onto the particular faces that are marked as media façades. The original resolution of the application is mapped onto the actual resolution of the particular media façade (see Figure 7). Finally, the rendering module invokes all input listeners for handling user input. (3) While running, the rendering module continuously updates the scenery.

To allow for hybrid prototyping similar to the City Compiler [22], the toolkit allows the integration of physical miniature models into the simulation as follows: Additional virtual cameras can be placed inside the scene. The view rendered from the perspective of the additional cameras can be sent to external screens. We can build a physical miniature model of a building (e.g., by 3D printing) and integrate digital display elements or a pico-projector into the physical model to simulate the media façade. By sending the output of an additional camera looking at the content of one simulated media façade to this embedded display, we create a physical model of the simulated media façade. With this, we allow for hybrid prototyping similar to the City Compiler [22], but with all the additional features and the flexibility provided by our toolkit.

Implementation

We developed the reference implementation of the media façade toolkit using Java. Java uses a virtual machine, which allows the toolkit to run on different computer architectures and operating systems without further ado. To render the 3D scenery, we utilize the "Lightweight Java Game Library"[4] (LWJGL). Through this library, we can access native OpenGL[5] functionalities to render the 3D scenery directly on the GPU of the graphics card, in order to improve the performance. The applications displayed on the façade are also written in Java, implementing an interface defining methods to provide the visual content of the application to an application frame buffer and to listen for user input. The toolkit currently supports only Java application that implement the provided interface. Applications written in other programming languages can easily be supported by providing the corresponding wrappers for the particular languages. The current implementation of the simulator only supports a single application per façade. However, multiple applications can be supported by dividing the façade into several parts, assigning a different — or instances of the same — application to every part. In the current implementation of the simulator, we exploit the dynamic class loading capabilities of Java to allow a developer to easily integrate new functionalities and features into the simulator on-the-fly. The developer only needs to provide the particular **.class** files in a specific sub-directory of the simulator.

User interface

The media façade toolkit provides a visual representation of the rendered scene. It further provides a set of controls to create and modify the scene, as well as to add applications and input devices. The toolkit contains the following main components, which are depicted in Figure 8: (A) The *rendering view* is the central visualization component of the the toolkit. It visualizes the rendered scene and provides controls to navigate through it and to move the light source around. (B) The *control view* provides access to the settings and properties of the simulated environment. It holds the controls to create and modify the simulation environment. (C) *Control groups* organize the provided controls as collections of related controls. Controls are grouped into *Scene Graph, Mapping, Material, Façades, Applications* and *Input*. (D) The *detailed settings* provide the particular controls for the different control groups, as well as additional information about the current settings.

The user interface of the toolkit is separated into a rendering view and a control view purposely. The rendering can be displayed on a separate screen. By utilizing a screen larger than a common desktop screen, such as a projection wall, we can create a more holistic impression of the simulated environment and provide means for a more immersive and realistic interaction.

[4]http://www.lwjgl.org

[5]http://www.opengl.org

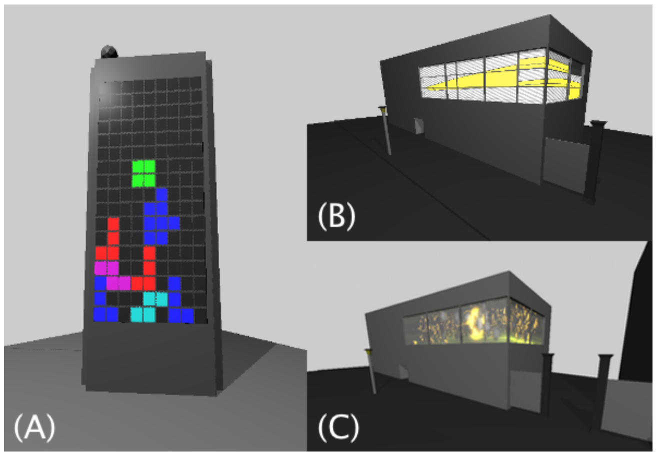


Figure 9. Applications developed using our toolkit. (a) Tetris: This game was developed as a standalone application and simulated on differently shaped façades. (b) Move-the-tile: A multi-user puzzle. The application was developed using the façade toolkit throughout the design process. (c) A video player to display videos and animation on various media façades

Initial feedback

In order to gather first feedback on the practicability of the simulator, we distributed the toolkit to five media façade application developers to utilize it for prototyping and simulating interactive applications for media façades (see Figure 9). Developers *A* and *B* are media informatics students, developers *C* and *D* are media façade application developers and developer *E* is a media façade professional, involved in building large-scale media façades. In this initial study, we did not yet conduct a qualitative or quantitative user study. Our interest was in getting first feedback on how the media façade toolkit performs in real settings and how it can enhance the process of designing transferable, interactive applications for media façades and overcome limitations of current approaches. We repeatedly conducted informal interviews with the users during the particular stages of the design process. The developers were not involved in the development process of the media façade toolkit. Developers C and D implemented interactive applications that were deployed on an actual media façade. Developers A, B and E developed standalone applications, adapted them to run within the simulator and visualized them on several virtual media façades with different form factors. All of them already had experience in developing interactive applications for media façades and they are thus familiar with current design approaches and their pitfalls. They all stated that using our simulator indeed simplified the development process tremendously. In particular, to demonstrate the benefits of the media façade toolkit, we want to outline one example of how the toolkit was used:

Within the the scope of conducting a user study for a project investigating multi-user interaction with a shared media façade, developer D utilized the media façade toolkit to develop a multi-user puzzle game intended to run on differently shaped displays in different settings and locations. As depicted in Figure 10, the game was displayed on a miniature model of a media façade using the toolkit's hybrid prototyping feature, as well as two different media façades of different size, shape and technology. In current practice, such a scenario would require extensive modifications of the application (e.g., re-writing large parts of the code) to run in all three settings. Using our toolkit, developer D implemented the application as a standard Java application, without addressing any particular façade. Running the developed application with the media façade toolkit involved the following five steps: (1) First, the particular 3D model of the façade must be loaded into the toolkit. In general, if it's not available, it can be easily created with any common modeling software. (2) In the second step, he selects the surface representing the media façade in the 3D model as well as an appropriate mapping from the set of available mappings. (3) This is followed by assigning an application to the façade and (4) selecting an input mapping. (5) Finally, the application can be started and it is automatically mapped onto the media façade in the 3D model and the graphical output is additionally sent to the connected display or façade. Once running in the toolkit, transferring the application to a different façade only requires repeating steps (1) and (2). After finishing the deployment of his application in the three different settings, we asked developer D about his experiences with our toolkit in an informal interview. He stated that he liked being able to transfer an application to another façade without modifying the application itself. He mentioned this feature as the main benefit for his project. Furthermore, he liked the possibility to reuse the interaction client he developed in his project with arbitrary applications running in the toolkit.

From developer E, we received valuable feedback from a professional perspective. He mentioned the re-usability of existing applications as a positive aspect while pointing out the possibility of prototyping complex installations simulating various form factors and technologies as the main benefit, since this was the most time-consuming and costly part of the development process. He further stated that in current practice, complex settings need to be rebuilt with real hardware as abstracted models for prototyping and testing and only a fraction of the work spent on this can be incorporated in the final result. In addition, he complained that this approach of current prototyping does not provide a good impression of the overall installations, since it only models certain features. He also liked the possibility of modeling the surroundings of the building to simulate different scenarios.

In general, all users were able to develop their applications as standalone applications without immediately tailoring them to a specific façade. None of the developers needed more than two hours to adapt the completed application to run on a particular façade. Once it was running on a façade within the toolkit, it took them on average less than five minutes to load the 3D model of a different façade and to transfer the application from the current façade to another one with a different form factor. A request made by two users was to provide general purpose client applications to allow for controlling a pointer on the façade in order to take the burden of developing them away from the designers. This is an issue we want to address in future work.

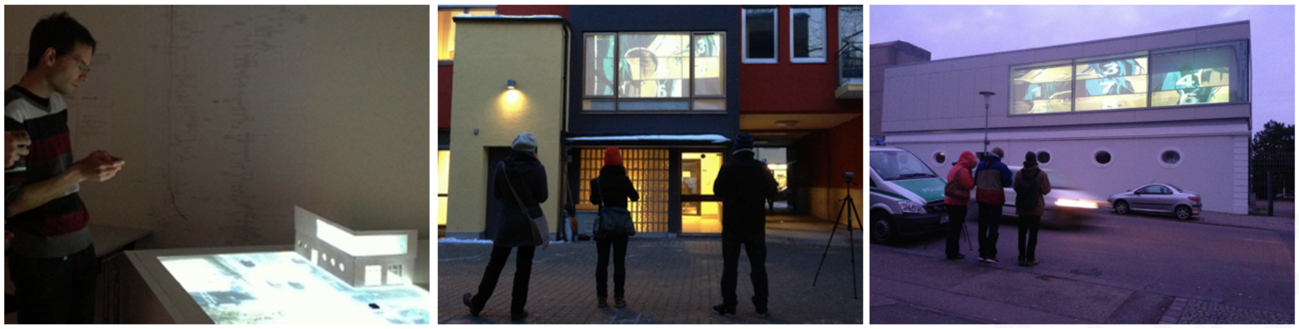


Figure 10. A puzzle created by a user running on a miniature model of the façade (left), a projected façade with different form factor and size (middle), and the media façade istelf (right). The game was ported to the three scenarios using the media façade toolkit without any modifications to the original application.

DISCUSSION

We introduced a rapid prototyping toolkit which allows for simulating interactive applications on arbitrary media façades. Due to its modular design, it provides a high degree of flexibility. The toolkit allows for the smooth combination of various applications, interaction techniques, media façades and building models where every module can be exchanged with ease. This offers the possibility to quickly adapt to new settings. Hence, the media façade toolkit shows a high level of potential to become a universal design tool for the designers and developers of digital content for media façades and other urban screens. In contrast to existing toolkits, a designer can easily try out given content on different media façades, in various locations, without much effort. An application can be developed without tailoring it to a specific façade or to a specific technical setup. It can be developed as a standalone application which can be mapped onto arbitrary media façades with the media façade toolkit. Furthermore, the toolkit can also be used as a standalone visualization tool. General visual content can be simulated with the toolkit as well, including pre-produced videos and animations.

Despite the potential to ease the development of interactive applications for media façades and to develop applications that can be immediately ported to different façade types, the media façade toolkit also comes with certain limitations. Due to the technical and physical properties of media façades and their highly dynamic, public deployment context, it is hardly possible to cover all settings to a full extent. Our work instead aims at providing a flexible prototyping and simulation framework, into which arbitrary building models, media façades, applications and interaction techniques can be integrated, in order to easily adapt existing setups and to ensure the portability of applications from one media façade to another. For the sake of simplicity, we currently only address pixel-based media façades, since they represent the vast majority of media façades. Mechanical media façades — media façades that are created by physically movable mechanical elements of a building — and voxel façades are currently not covered. Their implementation requires dynamic geometry, a feature that is currently not supported by the toolkit. Furthermore, automatically choosing an appropriate texture mapping is not supported by the toolkit. Since existing media façades have various different form factors, this remains an open problem. As mentioned before, the media façade toolkit currently limits the number of applications per façade to one. This limitation can be easily circumvented by dividing the façade into multiple façades which can display one application each.

CONCLUSION & FUTURE WORK

In this paper, we presented a generalized media façade toolkit capable of mimicking arbitrary media façade installations of any size, form factor, technology and hardware. By strictly separating building model, media façade, application and user interaction, the toolkit is capable of running arbitrary interactive applications on media façades with different form factors, sizes and technical capabilities. We described how, in contrast to existing approaches, the modular design of the simulator ensures the portability of applications between different media façades. The toolkit further provides the possibility of supporting interactivity.

In future work, we will address the limitations of the current toolkit. We plan to include mechanical façades, which is a challenging task since their implementation requires dynamic geometry. Furthermore, we want to investigate the problem of automatically choosing an appropriate texture mapping for given form factors. We want to further explore different mappings and form factors in order to improve the automatic mapping of applications onto media façades. Moreover, the quality of images produced by the toolkit could be improved by adding advanced techniques to render 3D scenes in a more realistic fashion. Multiple dynamic light sources, environment textures, reflections and transparency are currently not supported. In addition, we plan to support specifying hardware interfaces of media façades with the intention of being able to export the simulated applications for immediate deployment on media façades. Besides extending the implementation and its underlying conceptual architecture, we will make the toolkit publicly available for both the research community and designers of interactive installations for media façades.

REFERENCES

1. Böhmer, M., Gehring, S., Löchtefeld, M., Ostkamp. M. and Bauer, G. The mighty un-touchables: creating playful engagement on media façades. In *Proceedings of MobileHCI'11*. ACM, 2011.

2. Boring, S., Baur, D., Butz, A., Gustafson, S. and Baudisch, P. Touch projector: mobile interaction through video. In *Proceedings of CHI'10*. ACM, 2010.
3. Boring, S., Gehring, S., Wiethoff, A., Blöckner, M., Schöning, J. and Butz, A. Multi-user interaction on media façades through live video on mobile devices. In *Proceedings of CHI'11*. ACM, 2011.
4. D. Bouchard. *Emboddied Emergence: Distributed Computing Manipulatives*. Sciences, 2007.
5. L. Bullivant. *Responsive Environments: Architecture, Art and Design*. V&A, London, 2006.
6. W. Buxton. *User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2007.
7. E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. University of Utah, 1974.
8. Dalsgaard, P. and Halskov, K. Designing urban media façades: cases and challenges. In *Proceedings of CHI'10*. ACM, 2010.
9. Dalsgaard, P., Halskov, K. and Nielsen, R. Towards a design space explorer for media façades. In *Proceedings of OZCHI'08*. ACM, 2008.
10. Fischer, T. and Hornecker, E. Urban HCI: spatial aspects in the design of shared encounters for media façades. In *Proceedings of CHI'12*. ACM, 2012.
11. Fischer, T., Zöllner, C. and Hornecker, E. VR/Urban: Spread.gun - design process and challenges in developing a shared encounter for media façades. In *Proceedings of BCS'10*. British Computer Society, 2010.
12. D. Frenchman and F. Rojas. *Zaragoza's Digital Mile: Place-Making in a New Public Realm.* City, 2006.
13. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns*. Addison-Wesley, 1995.
14. S. Gehring and A. Krüger. Façade map: continuous interaction with media façades using cartographic map projections. In *Proceedings of Ubicomp'12*. ACM, 2012.
15. H. Haeusler. *Media Facades: History, Technology, Content*. Avedition, 2009.
16. Heckbert, P.S. A survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.
17. P. Jacobs. *Rapid Prototyping and Manufacturing: Fundamentals of Stereo Lithography*. McGraw-Hill, 1993.
18. Krasner, G.E. and Pope, S.T. A cookbook for using the model-view-controller interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3):26–49, 1988.
19. Lim, Y.K., Stolterman, E. and Tenenberg, J. The anatomy of prototypes. *ACM Transactions on Computer-Human Interaction*, Vol. 15, ACM, 2008.
20. Marquardt, N., Diaz-Marino, R., Boring, S. and Greenberg, S. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of UIST'11*. ACM, 2011.
21. Nakanishi, Y. Virtual prototyping using miniature model and visualization for interactive public displays. In *Proeedings of DIS'12*. ACM, 2012.
22. Nakanishi, Y., Sekiguchi, K., Ohmori, T. Kitahara, S. and Akatsuka, D. Hybrid prototyping by using virtual and miniature simulation for designing spatial interactive information systems. In *Adjunct Proceedings of Pervasive'11*. Springer, 2011.
23. O'Hara, K., Perry, M., Churchill, E. and Russell, D. *Public and Situated Displays - Social and Interactional Aspects of Shared Display Technologies*. Kluwer Academic Publishers, 2003.
24. Phong, B.T. Illumination for computer generated pictures. *Commmunications of the ACM*, 18(6):311–317, 1975.
25. Rogers, Y., Sharp, H. and Preece, J. *Interaction Design - Beyond Human-Computer Interaction*. John Wiley & Sons, 2011.
26. Rudd, J. Stern, K. and Isensee, S. Low vs. high-fidelity prototyping debate. *ACM Interactions*, Vol. 3, ACM, 1996.
27. Scheible, J. and Ojala, T. MobiSpray: mobile phone as virtual spray can for painting BIG anytime anywhere on anything. In *Proceedings of SIGGRAPH'09*. ACM, 2009.
28. O. Schoch. My building is my display. In *Proceedings of eCAADe'06*, 2006.
29. Seitinger, S., Perry, D.S. and Mitchel, W.J. Urban pixels: painting the city with light. In *Proceedings of CHI'09*. ACM, 2009.
30. Stahl, C. and Haupert, J. Simulating and evaluating public situated displays in virtual environment models. In *Proceedings of MODIE'06*. ACM, 2006.
31. Wiethoff, A. and Blöckner, M. Lightbox: exploring interaction modalities with colored light. In *Proceedings of TEI'11*. ACM, 2011.
32. Wiethoff, A. and Gehring, S. Designing interaction with media façades: a case study. In *Proceedings of DIS'12*. ACM, 2012.