B. Dave, A. I. Li, N. Gu, H.-J. Park (eds.), *New Frontiers: Proceedings of the 15th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2010,* 531–540. 

# A SYSTEM FOR FORM FOSTERING

## *Parametric modeling of responsive forms in mixed reality*

FLORA DILYS SALIM
*Spatial Information Architecture Laboratory (SIAL)*
*RMIT University, Melbourne, Australia*
*flora.salim@rmit.edu.au*

HUGO MULDER
*Arup Advanced Technology + Research, London, United Kingdom*
*hugo.mulder@arup.com*

and

JANE BURRY
*SIAL, RMIT University, Melbourne, Australia*
*jane.burry@rmit.edu.au*

**Abstract.** This paper investigates the integration of mixed reality with parametric modeling. This supports the concept of Form Fostering, which goes beyond the traditional form finding. Form Fostering takes sensory input from the physical world to inform a parametric model. We will present a prototype of the system that we have developed, which includes the use of a Wii Remote, an Arduino processing board, servo actuators and a camera as haptic input and interaction devices for Form Fostering. The potential benefits of designing in mixed reality are significant since designers can get real-time feedback from both the physical context and from changing physical design constraints represented by virtual parametric relationships. In order to leverage seamless interaction and activity between the physical and the virtual world, it is invaluable to consider sensing as an input for design.

**Keywords.** Parametric modeling; mixed reality; responsive architecture; responsive form; physical interaction; form fostering.

## 1. Introduction

The adoption of ubiquitous computing and sensory technology into interactive and responsive architectural design changes the ways in which we think about architecture, how we describe architecture and how we design architecture. The advances in ubiquitous computing and growth of computational power have driven the notion of the mixed reality, which is the spectrum between physical reality and virtual reality, as illustrated in Figure 1 (Milgram and Kishino, 1994).

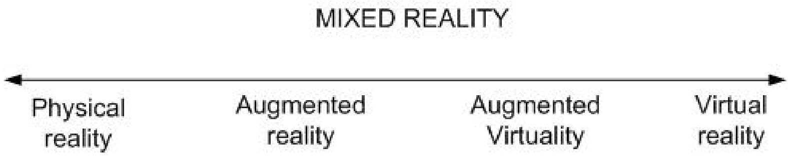

*Figure 1. Milgram's virtuality continuum (Milgram and Kishino, 1994).*

In augmented reality, users are interacting with physical objects, whereas in augmented virtuality, users are interacting with the virtual environment, yet the respective virtual and physical objects are displayed seamlessly (Hughes et al., 2005). The scope of interactivity and mixed reality modeling in this paper lies between the two augmentations, with the focus on parametric design for the purpose of Form Fostering.

3D architectural computer models in virtual space are generally highly malleable in parametric software like Digital Project (Dassault Systèmes CATIA), Generative Components (Bentley Microstation) or Grasshopper (McNeel Rhinoceros). The methods and tools for designing, prototyping and calibration of architecture in the mixed reality are missing from current practice and research. Given the flexible modeling power of existing parametric modeling software and inherent parametric thinking in design, there is a potential in extending these tools and techniques to allow for a more continuous process in generating architecture. The methods and tools for design in the mixed reality described in this paper are built upon existing parametric software capabilities. To our knowledge, the research and practice of modeling and simulating (responsive) architecture in a mixed-reality parametric design environment is largely non-existent.

This paper is organized as follows. Section 2 presents the related work in responsive architecture and design interfaces. Section 3 introduces the notion of Form Fostering. Section 4 presents Form Fostering experiments using Wii Remote and Arduino. Section 5 concludes the paper and presents proposals for further work.

## 2. Related work: responsive architecture and design interfaces

The ubiquity of bits and atoms in the mixed reality has created a great span of new frontiers for architectural design (Massachusetts Institute of Technology, 2008). Since the introduction of "kinetic architecture" in 1970 by Zuk and Clark (1970), there are various terms that have been used to connote the ability of architecture, space, structure, or building to respond to stimuli from users or the environment by means of changes in shape, organization, content or appearance. Some of these responses are more complex than others, can include feedbacks and are adaptive.

The cybernetician Gordon Pask has suggested an architectural system that is involved in a continual conversation with the users and the environment. In 1978, Pask's key collaborators, Cedric Price and John Frazer, conceptualized the Generator project, known as the first concept for an intelligent building (Frazer, 1995). Adaptive architecture has only physically materialized in the last decade. Built examples are the Aegis Hyposurface project (SIAL, 2006) by dECOi that is an interactive metal surface that responds with shape changes to multiple inputs simultaneously. Kas Oosterhuis (ONL) developed the Muscle interactive pavilion (Oosterhuis and Biloria, 2008) that responds to environmental stimuli and theoretically to other similar pavilions. The Digital Water Pavilion, designed by the Carlo Ratti (carlorattiassociati) and the MIT Senseable City Lab, encompasses an interactive and responsive water façade in Zaragoza, Spain (Massachusetts Institute of Technology, 2008). In these examples of responsive architecture, physical changes to the building occur in real-time and are driven by physical events, physical variations in the environment, or physical interaction between the architecture and the users.

As an interface to assist product design, the Augmented Foam was created (Lee and Park, 2005) as Lee and Park pointed out the need for physical and tangible interaction with 3D CAD models. The Luminous Planning table by MIT Media Lab and School of Architecture and Planning was developed for urban simulation (Ben-Joseph et al., 2001). The integration between the physical building models and the digital drawing, plans and models allows real-time simulation of sun, wind, and shade being projected on the physical models sitting on the table. The building models can be moved or rotated to generate real-time feedback of the impact of the variations. However, the physical-digital integration is not feeding into parametric design software, therefore the modification of the model is limited to changing the location and orientation of the buildings. The Tangible User Interface (TUI) by MIT Media Lab was designed to interact with Geographic Information Systems (GIS) data and parameters (Ratti et al., 2004). The TUI table is largely used for supporting landscape design as interaction with physical objects or markers are augmented on the visualized digital terrain. This work has reported the potential for 3D free-form modeling, integration of physical and digital modeling, and the need for incorporating real-time actuated feedback on modeling in the virtual and the physical environment as future work. The work also does not mention the integration with parametric modeling software for adaptive architecture design.

## 3. Form fostering

Form Fostering challenges the traditional approach of informing a virtual model purely with virtual 'knowledge'. It seeks to measure data in the physical world and 'capture' relations and interactions that exist in the physical world (physical parameters) in a larger model that is not only virtual. The architectural design models therefore become mixed: partly virtual and partly physical.

Figure 2 depicts a scenario of associations between parameters in a model following the Form Fostering approach. A parameter can be associated with another virtual parameter and informed by virtual events. A parameter can also be linked with the physical environment and receives input from sensor or haptic devices. A parameter can also be static or constrained if it is informed for example by a static object in the built environment. This can be useful if an existing building or an already built part of the building is to be included in the model. This allows for design to take place in the mixed reality. Also, because the model will exist in various stages of the mixed reality, both the early stages of design (mainly virtual) and the final stages of construction (mainly physical) can be captured in the same model. In responsive architecture the relations between the building and the environment remain dynamic, even when the construction process has finished: the parametric model has materialized.

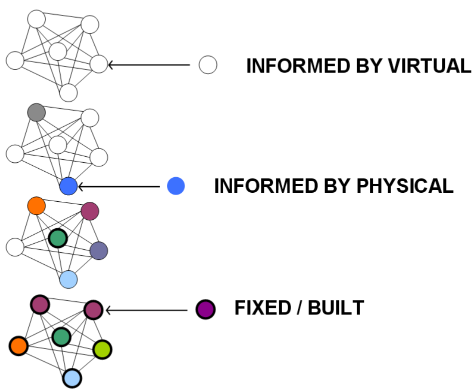


*Figure 2. Parametric associations in form fostering.*

Form Fostering requires interfaces for the mixed reality, both to interact with users as with the environment. The dynamic feedback loop is facilitated through informing the virtual models with changes in the physical world as well as informing the physical objects with changes in the virtual world. Real-time updates in the model can be managed in Form Fostering since designers can now use sensors, which stream data from the physical environment, as input to drive the parametric variations in the model.

## 4. Form fostering system prototyping experiments

The experiments described in this paper use Bentley Generative Components (GC) to build the parametric model. GC software is based on Bentley Microstation and allows users to define objects in Microstation through a different interface. Users can add additional features and functions by coding, compiling and adding Dynamic Link Library (DLL) as plug-ins to GC. Through GC, parametric relations can be defined between objects in Microstation. Complex models can be produced in this way as the relations can also be based on script or code.

We present two types of Form Fostering experiments. The first one is using a Nintendo Wii Remote (Wiimote). And the second one is using an Arduino processing board, a servo actuator and a camera. Both experiments are developed only in four days using low-cost materials. In comparison to the high-cost and complex TUI and ClayTools system, the tools developed in our experiment are much more economical.

### 4.1. THE WII REMOTE EXPERIMENT

A Wii Remote (Wiimote) is the revolutionary hand held motion detection sensor and controller for Nintendo Wii. The main components of a Wiimote are a 3-axis-accelerometer, an infrared emitter, and a Bluetooth interface. A Wiimote can be used to communicate with any computing device that has Bluetooth interface. Given its internal features, low cost, and the release of the open source for Wiimote development, in C#.NET (Peek, 2007) and Java (WiiRemoteJ, 2008), Wiimote has become a state-of-the-art wireless sensor device. Infrared tracking and augmented reality applications with Wiimote have been developed (Lee, 2008). However, none has actually incorporated the use of Wiimote to interact with parametric modeling for design. We first developed a Rhino plug-in that utilizes Wiimote to draw curves in 3D space. The successful experiment with Rhino 3D motivated the Form Fostering experiment of the Wiimote interacting with GC.

In the first experiment, the Wiimote is staged as an input for the virtual model. We have developed a custom C# function in GC that allows a Wiimote to interact with the GC environment using Wiimote's sensory information such as acceleration, up/down/left/right and button press/actions. The approach chosen was to input directly to the core of GC, thus various functions such as modifying nodes and changing the camera position in a virtual space were possible.

In the experiment depicted in Figure 3a, the parametric model set up in GC includes a random number of points scattered across random positions in space and a cube that acts as a selector of points that are located within its boundaries. As the Wiimote is moved, the cube's position updates and the cube moves and resizes accordingly. The points selected by the cube are updated whenever the cube moves.

In the next experiment (Figure 3b), a Wiimote is used to control the dynamic parametric surface. Whenever the Wiimote moves, the surface reforms itself dynamically based on the position of the Wiimote controller point located under the surface. This experiment has an unlimited number of real-world analogies to the creation of a responsive surface or façade. If other sensors such as light, temperature or wind sensors can be linked to such a surface, it would be useful to simulate a surface that can respond, adapt, and reform itself to suit the changing environment (e.g. a roof that extends its eaves on a hot sunny day, or a wall that curves itself when someone sits in a windy non-enclosed space).

In the experiment depicted in Figure 3c, a Wiimote is used to draw a BsplineCurve. Although the mouse is the most usable tool to interact with any 3D CAD programs, it is essentially a 2D interaction tool and is a difficult tool to use for 3D sketching in CAD (Lawson, 2004). Given the unrestricted movement of the Wiimote in the physical space and its ability not only to feed x, y, z positions in space (using infra-red emitter and tracker), but also to stream gravity acceleration data as the Wiimote is moved, it has the potential to be a tool complementary to the mouse for 3D sketching.

The experiment shown in Figure 4a demonstrates the use of the Wiimote as a camera view controller. The parametric model in GC contains a sphere object. The Wiimote is connected to a GC camera object that is linked with a controller point traveling on a sphere and gives a snapshot of the model from the camera point of view. The camera views the surface from a certain direction and angle from the sphere and can take snapshots of the view.

The parametric model in the following experiment (Figure 4b) has a direction vector as the parameter interfacing with the Wiimote's x, y and z gravity acceleration movement. The movement of the Wiimote in a certain direction updates the position of the axis (direction vector) of the object and causes the object to move along the same direction as the Wiimote.

The last experiment (Figure 4c) demonstrates the use of the Wiimote buttons (up/down/left/right/A/B) as a "remote control" to dynamically move components across the grid points in the model or to magnify or shrink the selected object in the model 'on-the-fly'. For example, pressing 'B' button enlarges the GC parameterized component associated with the Wiimote.

### 4.2. THE EYE

The second experiment uses a different protocol. It allows for both input into GC and output from GC. The experiment setup was a simple relation between a physical model and a virtual representation of an eye. The virtual model mimics the physical model and vice versa, therefore allowing for calibration and for trans-reality feedback loops.

The physical model of the eye was a little robot that consists of a (web) camera and two servo motors. The orientation of one servo motor allows the camera to rotate along a vertical axis, and the other allows for movement along a horizontal axis. The servos rotate 180 degrees. The view of the camera could therefore cover roughly half a sphere like the eye of a chameleon (figure 5).

The two servos are controlled using an Arduino controller board. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software (Arduino, 2009). Arduino controller boards connect to other hardware and have a chip that can be programmed and then

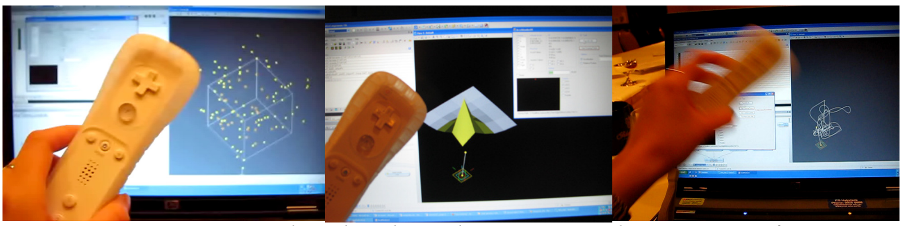

*Figure 3. a: A Wiimote and a cube selector. b: A Wiimote and parametric surfaces. c: A Wiimote for 3D sketching.*

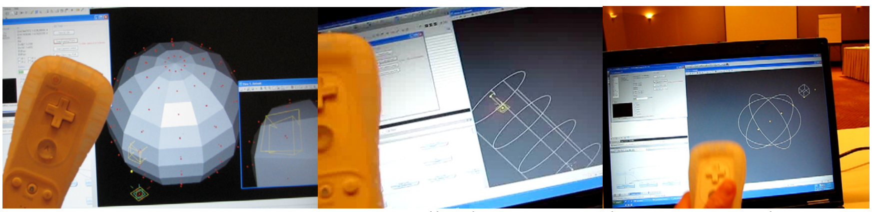

*Figure 4. a: A Wiimote as camera view controller. b: Dancing pole. c: Dynamic object mover, magnifier or shrinkage controller.*

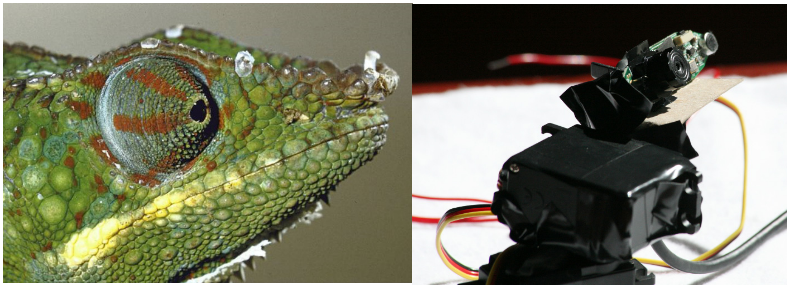
Figure 5. The Physical Sensor and Actuator of the Eye

run in a stand-alone mode. The board can also stay connected to a USB port of a computer and communicate through a serial protocol. The controller board was programmed to wait for a signal on the serial port. If the signal comes in the right format, the signal is decoded and the two servos are instructed to rotate to a defined angle.

The webcam was also connected to a computer. Open-source software (openFrameworks, 2009) was used and adapted to control the webcam and compare consecutive frames. Areas in the image that are different from the previous are marked. Therefore the software can tell that firstly there is motion in the image, and secondly where in the image that motion takes place. The software was adapted to send data to the serial port in a format that Arduino could interpret. The largest area of motion in the picture indicated the point of interest and the software would make the camera point towards that point of interest. When the camera moves however, consecutive images look completely different. Therefore a delay was added between instructions, and it was found that motion tracking of a single object along an evenly coloured background worked best. With the integration of openCV (OpenCV, 2009) for face recognition and tracking, the background image is ignored and the performance of the motion tracking is improved.

The virtual model is a sphere in GC with a dish-like solid to represent the iris and the pupil (figure 6). The size of the eyeball was kept fixed, but the rotation of the eye was based on two graph variables for a rotation along a vertical axis and rotation along a horizontal axis. The size of the pupil was controllable through another graph variable.

There were two directions of communication: the GC output mode and the GC input mode.

In the GC output mode the eyeball was controlled in GC using the sliders alongside the graph variables. A function was written in C# that would create a communication channel with the serial port. On a change of one of the variables in GC, this would send data over that channel. The data in turn would be

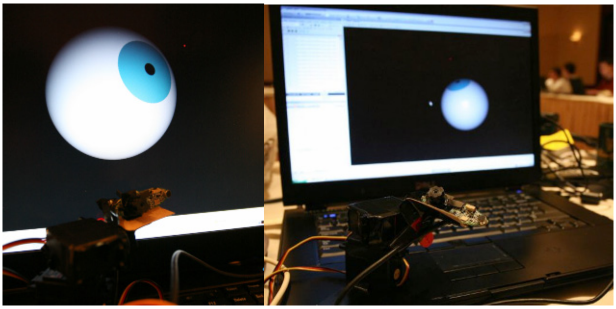

*Figure 6. The Virtual Eye Interacting with the Physical Eye*

interpreted by Arduino and rotate the camera. A window on the screen would display the live video stream of the camera.

For the GC input mode use was made of a web server that runs as a plug-in in GC. The web server when started waits for http requests of a specified form. This way the values of graph variables can be changed for example. The camera position was changed by moving objects that the camera would track. Based on the instructions sent to the camera, the software also sends http requests to the local web server. The graph variables would be updated and the graphic display in GC updates accordingly. The size of the pupil was driven by the size of the movement blob in the video stream.

## 5. Conclusion and future work

The system prototypes for experiments with form fostering validate the hypothesis of the potential of parametric modeling software as a simulation bench for responsive architecture design. When integrated with sensors, actuators, or any haptic devices, an early-stage design of responsive architecture can now be simulated. Design variations in the virtual model can be generated by parametric changes from the physical environment and vice versa. Although responsive architecture may drive this research, it is envisaged that the outcomes are more broadly usable in parametric design.

The experiments showed practical issues that need resolving such as delays in the system that complicate direct feedback. Also, with increasing complexity of interactions, the response of the parametric software would decrease. For a more generic approach and development of this field, a unified method for communication between various parts of software and hardware would be beneficial. This would allow flexibility so that "input," "output," and "process" can be openly defined by the designers themselves. This work is in progress and envisaged to be an open-source platform for educational and research purposes.

Other work will focus on the continuity of the responsive design through virtual and physical stages and includes experiments on capturing social and environmental parameters. Experiments in the near future are envisaged with the integration of thermal, light, humidity sensors and weather stations as well as data from social networking websites to influence variations in the parametric model. An investigation of tools and techniques is also taking place to inform the system with building performance and energy analysis.

## References

Arduino: 2009, *Arduino, HomePage,* available from: http://www.arduino.cc/ (accessed 9 November 2009).

Ben-Joseph, E., Ishii, H., Underkoffler, J., Piper, B. and Yeung, L.: 2001, Urban simulation and the luminous planning table: bridging the gap between the digital and the tangible, *Journal of planning education and research,* **21**, 195–202.

Frazer, J.: 1995, *An evolutionary architecture,* Architectural Association Publications, London.

Hughes, C. E., Stapleton, C. B., Hughes, D. E. and Smith, E.: 2005, Mixed reality in education, entertainment and training: an interdisciplinary approach, *IEEE computer graphics and applications,* **26**(6), 24–30.

Lawson, B.: 2004, *What designers know,* Architectural Press, Elsevier, Oxford.

Lee, J. C.: 2008, Hacking the Nintendo Wii remote, *IEEE pervasive computing,* **7**(3), 39–45.

Lee, W. and Park, J.: 2005, Augmented foam: a tangible augmented reality for product design, *Proceedings of the 4th IEEE / ACM international symposium on mixed and augmented reality,* IEEE Computer Society, Vienna.

Massachusetts Institute of Technology: 2008, Digital water pavilion makes a splash in Spain, *ScienceDaily,* http://www.sciencedaily.com/releases/2008/06/080617115237.htm.

Milgram, P. and Kishino, F.: 1994, Taxonomy of mixed reality visual displays, *IEICE transactions on information systems,* E77-D, 12, 1321–1329.

Oosterhuis, K. and Biloria, N.: 2008, Interactions with proactive architectural spaces: the muscle projects, *Communications of the ACM,* **51**(6), 70–78.

OpenCV: 2009, *OpenCV wiki,* Available from: http://opencv.willowgarage.com/wiki/ (accessed 9 November 2009).

openFrameworks: 2009, *openFrameworks*, Available from: http://www.openframeworks.cc/ (accessed 9 November 2009).

Peek, B.: 2007, *Managed library for Nintendo's Wiimote,* Available from: http://blogs.msdn.com/coding4fun/archive/2007/03-/14/1879033.aspx (accessed).

Ratti, C., Wang, Y., Ishii, H., Piper, B. and Frenchman, D.: 2004, Tangible user interfaces (TUIs): a novel paradigm for GIS, *Transactions in GIS,* **8**(4), 407–421.

SIAL: 2006, *Aegis hyposurface*, Available from: http://www.sial.rmit.edu.au/Projects/Aegis_Hyposurface.php (accessed 4 November 2009).

Zuk, W. and Clark, R. H.: 1970, *Kinetic architecture,* Van Nostrand Reinhold, New York.