ADAPTIVE GENERATIVE PATTERNS

Design and Construction of Prague Biennale Pavilion

GIORGOS ARTOPOULOS, STANISLAV ROUDAVSKI
Cambridge University Digital Studios & Moving Image Studio (CUMIS), 1 Bene't Place, Lensfield Road, Cambridge CB2 1EL, UK
george.artopoulos@gmail.com, stanislav@stanislavroudavski.net

AND

FRANÇOIS PENZ
fp12@cam.ac.uk

Abstract. This paper describes an experimental practice-based research project that considered design process, implementation and construction of a pavilion built to be part of the Performative Space section of the International Biennale of Contemporary Art, Prague 2005. The project was conceptualized as a time-bound performative situation with a parasite-like relationship to its host environment. Its design has emerged through an innovative iterative process that utilized digital simulative and procedural techniques and was formed in response to place-specific behavioral challenges. This paper presents the project as an in-depth case-study of digital methods in design, mass customization and unified methods of production. In particular, it considers the use of Voronoi patterns for production of structural elements providing detail on programming and construction techniques in relationship to design aspirations and practical constraints.

1. Project Background

The project was developed in the Cambridge University Moving Image Studio (CUMIS) and Digital Studios of the Department of Architecture for the International Biennale of Contemporary Arts (13 June – 11 September 2005) organized under the auspices of the Czech National Gallery. The

1 The first two author surnames are listed in alphabetical order.
project was initiated in November 2004; production of the building elements began in the second half of May 2005 and together with construction continued until the end of June 2005. During the last months, the team worked in the Veletržní Palace (the Museum of Modern Art) in Prague. The last two weeks of work were done after the public was admitted to the exhibition so that the construction process has become integrated into the life of the Biennale as a performance in its own right.

![Figure 1. View of the structure photographed during construction (a photograph).](image1)

_Veletržní Palace_ is a 1920s functionalist structure made of reinforced concrete. It suffered badly in a massive fire in the 1970s. The lower-level spaces are still not fully restored and have a rough unfinished look. The pavilion was designed to fit into a stairwell (Figure 2) that connects the main exhibition hall located on the ground floor, the secondary entrance to the museum that becomes the primary entrance during the evening events and the performance/exhibition spaces of the lower level. The pavilion consisted of the purposely designed structure and of the interactive audio-visual system that was able to respond to the behaviour of the visitors.

![Figure 2. Three views of the stairwell space prepared for the construction (photographs).](image2)

The design process consisted of three parts. Firstly, dynamic simulation and time-based processes were used to produce two organic surfaces fitting into the stairwell space. Secondly, the computer-driven responsive audio-visual system was laid out in relationship to these surfaces. This system incorporated image-based computer vision and was able to create real-time
audio and video compositions in reaction to people’s movement through the space. Interesting in their own right, the aforementioned parts of the design will be described in detail elsewhere. This paper focuses on the third part of the workflow describing how the two organic surfaces (Figure 3) produced during the first phase of the design process were materialized as building components. This description sets out a technical framework for the procedural production of a built structure under a variety of situational constraints and in response to the performative requirements of the audio-visual system.

![Figure 3. A perspective into the stairwell (a digital rendering).](image)

![Figure 4. Structure installed in the stairwell and a fragment showing local curvature-dependant variations (photographs).](image)

1.1 PAPER AS MATERIAL

Shigeru Ban (Sigeru Ban Architects, 2005) is perhaps the most established architect to have used paper and cardboard regularly in his practice. He uses paper tubes to replace beams and columns, or in arrays to form walls. A closer precedent is provided by Horst Kiechle (2004) who develops small ephemeral structures made of cardboard or paper at
architectural and object scales. In his work, the low-cost easy-to-manufacture paper-based materials are often used to define and reconfigure performance spaces. Kiechle uses flat, usually triangular elements to establish a curvilinear or irregular boundary between the regions of solid and void. His lamps provide a good example of relatively complex self-supporting paper structures.

Different utilization of paper can be seen in the use of honeycomb structures. Kraft- or aramid-paper honeycombs spacing two veneers are an established flat-board building material. The same principle of honeycomb sandwich is used for curvilinear and monocoque geometry in boats, airplanes and other situations where weight is an important consideration. The preliminary research has shown that paper and cardboard can be successfully used to design and build small-scale architectural installations featuring multi-parametric complex geometries under tight financial and temporal constraints.

1.2 PATTERNS IN CONSTRUCTION

The spatial intervention was a significant part of the project. It intended to provide two kinds of impact. Firstly, the structure impinged on people’s behavior such as movement through space or visual access with the intention to engender new social encounters. Secondly, the structure in combination with projected images created a visual field that could inform the visitors, redirect their attention and involve their bodies in the making of the dynamic visual form. Plant-cell microscopy images, urban-texture photographs and dance movies were processed so that their pattern-based nature was made apparent. The visual field consisted of the pattern-based spatial structure, pattern-based video imagery and the participant’s bodily movement through the space. While static after construction, the structure was intended as a procedural response to the given environment. Its responsive nature called for a modular, component-based arrangement able to conform to geometrically complex surfaces and flexibly adapt to the changes in local conditions. These requirements led to the interest in honeycomb and similar structures.

Experimental work of A. Kudless (2005) explored the use of honeycomb structures for the construction of curvilinear geometry. One of the Biennale project’s goals was to see if this approach could be taken further with the use of non-periodic patterns capable of local change. The pattern adopted after preliminary research was a Voronoi diagram (Figure 10, C) defined as “the partitioning of a plane with \( n \) points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other” (Weisstein, 2005).
Voronoi patterns are in wide use in various applications in computer science. In architecture, Thomas Wingate (2005) used Voronoi diagram to determine the form for the decorative ice-blocks used in a flat vertical wall of the Ice Hotel, Sweden.

1.3 DIGITAL TOOLS

The project’s methodology is a first step towards an approach that incorporates architectural form-finding, multi-media design and fabrication of building components as parts of a unified performative process. In the foundation of this process are the protocols for the cross-platform data exchange. Currently, all of the three areas utilize digital tools that are capable of programmable data-wrangling. The unified digital-fabrication workflow offers other benefits apart from convenience. Design via parameter-readjustment allows for work with relational diagrams. During development, the designer is able to, and in fact has to, move up and down the branches of the process tree, reviewing the feedback and readjusting the inputs. Even though everything in the computer system is ultimately solvable, the designer waives the right to control results explicitly and instead guides the process with multiple indirect measures. The sacrifice of arbitrary control leads to gains in the capacity to deal with complex systems as systems, without reducing them to basic components. The bifurcations of the process with which the designer becomes intimately involved often lead to solutions that could not have been pre-specified from the start as spatial layouts or even as design goals. The feedback is often real-time or comparatively fast. The ability to tweak different components of the process allows the designer to learn about system relationships via experimentation. It becomes possible to plot alternative design paths in terms of multiple sequential versions. This process of probing and recoiling is exploratory by nature. The design process acquires a character of an investigative tool rather than that of a method with which a wilful author promotes his or her worldview. Significantly, the design suggestions derived from the procedural design process can be very different from those intuited at the beginning. Insights gained in this way can both educate the designer and lead to innovative solutions.

It is often said that specification of the problem is part of the solution. Thus, educating the designer is a significant goal. Within this project, the educated designer is providing a vital link between the solvable system and the fuzzy reality of the in-world situation. The designer is aware of many practical circumstances such as constraints on resources, materials, money, time and knowledge or multiple design goals related to various peers groups. The design process begins when on-site observations are expressed in the digital domain in terms of virtual forces impacting upon geometric systems and proceeds as interactions between these systems and the designer.
For the first stage of the design process we needed an integrated multi-
faceted programmable software environment capable of animation and
dynamic simulation. In response to these requirements, Alias Maya (Alias,
2005) was chosen for its capabilities to deal with complex geometries. With
Maya we were able to establish form-finding design-processes incorporating
a number of techniques such as layered randomization, particle simulation
and force simulation using dynamic curves and fields. These techniques
were combined into an active diagram able to support versioning and
prototype breeding. Maya was also used on-site to layout the suspension
system.

In order to develop the building components, we needed a programmable
environment able to support unfolding and construction-oriented content
organization. Rhinoceros 3D (Robert McNeel & Associates, 2005) was used
for Voronoi tile-generation, cell-structure generation, unfolding of cell-walls
and nesting of the cell-skins and cell-walls. FormZ (Auto-Des-Sys, Inc.,
2005) was used to unfold polygonal geometry of the cell-skins. Finally,
MAX/MSP/Jitter (Cycling ’74, 2005) programming environment was used
for the design of digital audio-visual responsive system.

2. Scripted Production

![Figure 6. Terminology and the structure of the cells. A) Offset point, shown circled. B) Base
point, shown circled. C) Direction of offset along the normals, shown as dashed lines. D) Cell.
cell-walls and cell-skins, shown as shading changes. J) Cell-wall insets. K) Outer shell. L) Input

The first stage of the procedural-design process, run in Maya, produced two
organic, topologically cylindrical NURBS-surfaces. The ready surfaces were
then exported as IGES files and transferred into Rhino where the generation
of the building elements was conducted.
Production of building elements was automated and the technical procedure was encapsulated in two scripts\textsuperscript{2} that were run one after the other. The first script governs the distribution of points on a given NURBS surface. The second script uses these points as input for the generation of a Voronoi pattern and building elements based on it.

\textit{Figure 7.} Side view as designed. The height of the stairwell in the supplied documentation was underestimated; the actual height was 10.000mm. A) Outer shell. B) Inner shell. C) Approximate area observed by the computer-vision system. D) Video projections. E) Disused lift. F) Computers and the sound system. G) Speakers. H) Free passage is left unobstructed all the way along the stairwell. (a digital rendering)

\textsuperscript{2} RhinoScript is a VBScript-based scripting language.
2.1. DISTRIBUTION OF POINTS

As the first step, the script asks the designer to specify a NURBS surface to be used as input. Next, the designer is given an opportunity to specify the number of points to be created. The points can be distributed according to one of three methods as described below:

**Constant.** This method attempts to distribute the given number of points on a given surface uniformly so that the resulting distances between neighboring points are close to equal. The method begins with a rough estimation of the minimum possible distance between points for a given surface and a given number of points where the points are distributed uniformly.

\[
D_{\text{min}} = \sqrt{\frac{2 \cdot S_{\text{surf}}}{N_{\text{pts}} \cdot \theta}}
\]  

(1)

Where \(D_{\text{min}}\) is the estimate of the minimum distance between points, \(S_{\text{surf}}\) is the surface area of the input surface and \(N_{\text{pts}}\) is the number of points on the surface.
The result of the estimation is suggested as the default designer-input for the minimum distance between points \( D_{\text{min}} \). Input values smaller than the estimation will produce less evenly spread distributions while the values significantly larger than the estimate will force the program into an infinite cycle as the script will struggle to find suitable point locations. This behavior is due to the nature of the procedure that attempts to add one point at a time looking to satisfy the condition \( D_{\text{cur}} \geq D_{\text{min}} \) where \( D_{\text{cur}} \) is the distance between the currently tested point and its nearest neighbor. If \( D_{\text{cur}} \geq D_{\text{min}} \) is true, a new point is created, otherwise a new point is attempted and the procedure goes into a cycle until the condition is satisfied.

**Curvature.** This method relates the point density to the amount of surface curvature so that the higher the surface curvature the higher the point density. This relationship is established in two steps. First, the given surface is sampled and the curvature data is collected into a sorted array. The number of sampling points can be explicitly specified by the designer allowing for crude-force acceleration during test runs. The designer is then given the opportunity to specify \( D_{\text{min}} \) and \( D_{\text{max}} \) with the latter being the maximum allowable distance between points. \( D_{\text{cur}} \) is then found as follows:

\[
D_{\text{cur}} = \left( \frac{1}{C_{\text{pmax}}} - \frac{1}{C_{\text{cmax}}} \right) \left( D_{\text{max}} - D_{\text{min}} \right) + D_{\text{min}}
\]

(2)

Where in addition to the aforementioned definitions \( C_{\text{pmax}} \) is the maximum principal curvature at the attempted point, \( C_{\text{cmax}} \) is the maximum sampled curvature for the given surface and \( C_{\text{cmin}} \) is the minimum sampled curvature for the given surface.

In practice, this kind of calculation means that even though \( D_{\text{max}} \) is collected as the designer input and used in calculations, it is not directly constraining the maximum distances between points so that if \( N_{\text{pts}} \) and \( D_{\text{min}} \) are both relatively low, the resulting distribution can contain areas with \( D_{\text{cur}} \) values far greater than \( D_{\text{max}} \) as input by the designer. Such gaps can produce the cells that are too large to be structurally sound.

**Inverse Curvature.** This method is similar to the Curvature method but the relationship of curvature to density is inverted with the areas of higher curvature containing lower point-densities. The use of the Inverse Curvature method has implications similar to that of the Curvature method.

\[
D_{\text{cur}} = \left( \frac{1}{C_{\text{pmax}}} - \frac{1}{C_{\text{cmax}}} \right) \left( D_{\text{max}} - D_{\text{min}} \right) + D_{\text{min}}
\]

(3)
Several such methods can be combined to produce the final point-cloud. If this technique is used, each method is run separately with the desired settings and the resulting point-clouds are added together.

The project utilized the total of 1560 points and an equal number of corresponding cells. 700 points and Curvature method were used to generate the point-cloud for the inner shell. The point-cloud for the outer shell was made with 860 points using a combination of Curvature and Constant methods. The combination of the two methods made it possible to enforce the desired $D_{\text{max}}$ threshold on the distances between points. This was achieved by running the script with Constant method using 250 points and the default $D_{\text{min}}$ estimate provided by the script. This caused the points to be near-uniformly distributed across the surface during the first run. The second run of the script used 610 points distributed by Curvature method with desired $D_{\text{min}}$ and $D_{\text{max}}$ values estimated given the geometry of the shell surfaces and the structural properties of the cardboard. The numbers of points in all cases were found via multiple readjustments of parameters in the generation chain.

Figure 9. Variations in shell structure, inner shell. 1) Fragment showing the structural consequences after a point-cloud is added to another point-cloud. 1_A) The minimal distances between points were controlled during the distribution of points for each point-cloud.
However, when one cloud was placed over the other, the distances between some point-pairs could be well within these thresholds. 1_B) An extra cell-wall inserted between the two points. 1_C) A point in a cloud. 2_B) An image showing structural variations. Settings: two point-clouds used, 1st cloud – Constant method, 150 points; 2nd cloud – Curvature method, 700 points; curvature-dependant cell-wall height, minimum cell-wall height – 50mm, maximum cell-wall height – 250mm. 2_A) “Rivers” of high density at high-curvature areas. 2_B) Low-curvature areas. 2_C) A point. 2_D) High cell-wall. 2_E) Low cell-wall. 3_B) One point-cloud used – Constant method, 20 points; Constant method for cell-wall height, 150mm. 4_B) One point-cloud used – Constant method, 200 points; Constant method for cell-wall height, 150mm. (digital renderings).

The results had to satisfy a number of structural and experiential criteria. The most fundamental of these are cell sizes and proportions. A related negotiable criterion was the degree with which the cell-walls and cell-skins were able to conform to the curvilinearity of the input surfaces. As cell-walls and cell-skins are defined by straight edges, tighter tessellations result in better conformity. Another significant criterion is cell geometry. All cell-walls and cell-skins are non-planar surfaces (Figure 6, I). This non-planarity adds to structural rigidity but can also make it more difficult to match the cells in the assembly and even cause breakage in the material. Therefore, high non-planarity had to be avoided. Cell quantity, together with other parameters, would have a direct impact on the weight of the structure and the number of operations required for its production.

Experientially, the iterative process looked to uncover the effects on visual density and variety of surface texture, visual permeability along varying view lines and production of shadow patterns.
The relationship between curvature and point density as established by Curvature and Inverse Curvature methods can be adversely affected by the surfaces that include areas with extremely high or extremely low curvature. If the whole array of sampled curvature values was used for the test calculations, the script tended to produce confined areas with high density while distributing the points on the rest of the surface almost uniformly. Thus, it was necessary to introduce the facility that would allow the designer to visualize the sampled curvature values and to clamp the value range if necessary. To achieve this, the given array of curvature values was first normalized to fit the 0-to-100 range and then displayed as a graph (Figure 10, A). At this stage, the designer was offered an option to clip the lower and/or higher portions of the range discarding part of the data. This detail of the process might appear to be a trivial problem intrinsic to the specific implementation. However, the experience of designing the structures for this project demonstrated that such “problems” should rather be seen as opportunities fundamental to the process because their impact is far from “trivial” in terms of methodological formalisms and practical choices. While
the in-depth theoretical discussion is beyond the boundaries of this paper, this clamping procedure is mentioned here as a typical example of a feedback-loop relationship between the designer and the model. Instead of the much-criticized rigidity that is seen as a typical property of digital tools, this relationship is not dissimilar to that between a designer and a paper sketch. Through multiple iterations and refinement, the voluntary input is calibrated against the meaning already invested into the system. Thus, calculable numeric data acquires the exploratory suggestive capacity that sets up a dialogical relationship between multiple versions generated in response to modified input and sets the ground for the discovery of new form.

Adding the product of the two methods together produced structurally interesting variations in point distribution. When two point-clouds were combined into one, the $D_{\text{min}}$ threshold was not enforced and some of the points in the resulting point-cloud were positioned very close to each other (Fig. 1, 1_A; Fig. 10, D). Instances of such tight positioning occurred across the whole surface introducing another layer of complexity into the structure with the visual impact being more prominent in the areas with low curvatures where $D_{\text{min}}$ thresholds was at their largest.

The type of a method, the weighting of a method (i.e. what percentage of the total number of points is controlled by the given method) and the number of methods used constitute input gateways that accept data specified by the designer or supplied by another procedure (refer to Fig. 9 for examples). The designer input can be established by running multiple iterations of the routine and the certain versions can be judged as more suitable based on the number of criteria to do with human behavior or construction logic.

The curvature/density relationship as implemented in the script exemplifies the adaptive capacity of the approach. Other procedural relationships of this kind can be established and the controlling input can be provided via on-site observations or computational methods. The examples of such data might include isovists/viewsheds, light-level and body-movement measurements or simulative AI routines. The parameters of the structure that might be driven by such data include fenestration; cell-skin and cell-wall transparency, color, light reflectance, light transmittance and other material properties; cell-wall widths, cell-wall orientation, cell density, cell uniformity and the like. For example, the project featured experiments linking the orientation of the cell-walls to the positions of the video projectors. Such linking enables the structure to guide the moving-image formation by opening or blocking light cones, controlling shadow distributions, framing views and articulating the sculptural properties of the structure. Further experimentation along these lines will be conducted in the future. At the present stage, the project’s approach to form-finding and the production of building elements seeks to establish the methodological
backbone for the open and flexible process able to respond to multiple complex data inputs.

2.2 GENERATION OF CELLS

Figure 11. A perspective along the direction of the inter-shell canyon (a digital rendering).

The cell-generation script used Constant, Curvature and Inverse Curvature methods that are similar to those implemented in the point-distribution script (see the description earlier in the text). In the cell-generation script, the curvature data was used to determine cell-wall widths.

The second script began with the request for the designer to pick a NURBS surface and a point-cloud (as produced by the point-distribution script described above). The designer was also asked to specify the name prefix for the cells to be created. After these steps, the designer was given a number of options that controlled the building elements about to be created and specified their attributes.

- Should the cell-skins be created? If the answer was “yes”, should they be created at inner edges of cell-walls, outer edges or both?
- Should the insets to compensate for material thickness be created? If the answer is “yes”, the designer had to specify the width by which the insets should be offset from the original cell-walls.
- Should the cell-walls be unfolded?
- Which method should be used to establish the width of cell-walls? Constant, Curvature or Inverse Curvature? If Constant was chosen, the designer was requested to specify the cell-wall width. If Curvature or Inverse Curvature was chosen, the designer was requested to specify the minimum and maximum cell-wall widths.
• For point-clouds containing more than 20 points the designer was requested to specify an optimization factor constraining the number of points taken into consideration during calculations.

• If the unfolding procedure was activated earlier in option selection, the designer was requested to specify the location where the log file was to be saved. The log file contained a record of the process with error reports and precision warnings for each cell.

This section describes the flow of the script used to generate the cells and the skins for the outer shell. The script began by unwrapping the NURBS surface and the point-cloud from the UV space into the XYZ model-space such that the resulting rectangle of the NURBS surface was resting on the horizontal XY plane. The points in the 2D point-cloud were then sorted in the ascending order. When the points are sorted in this way, it becomes easier to locate them and their associated cells in digital models, during manufacturing and during construction. The sorted point-cloud was then transferred back onto the 3D surface and each point of this point-cloud was later used as an anchor for textual cell-labels. Next, the surface was sampled at each point of the point-cloud and the curvature data was stored, ready to be used by the Curvature method that was chosen to establish cell-wall widths. The script then drew a curvature graph to allow the designer to clip the areas with extreme curvatures.

After that, the script ran a looped routine generating Voronoi tiles within the unwrapped NURBS rectangle (Fig. 10, B) using a method the first version of which was written by David Rutten (2005) and further developed by Thomas Wingate (2005) and Andrew Kudless (2005).

This method uses Rhino’s functions and commands to perform the necessary calculations. The method creates one tile at a time by first establishing the positions for and creating the vertex points, then drawing border lines through the points and finally trimming the tile curve with the surface boundary using Boolean curve subtraction. This method worked well in our case. However, it was comparatively slow because the process created new geometry at every step - a purely numerical approach would work much faster. Additionally, large point-clouds force the script to consider too many points when drawing tiles thus slowing down the performance dramatically. To avoid this problem an aggressive acceleration threshold was implemented via an introduction of a designer-specified optimization factor constraining the number of points taken into consideration to an arbitrary number. This measure was highly effective but can potentially be error-prone if the designer-specified threshold forces the algorithm to miss significant points.

After all the tiles were generated, the points that define the tile vertices were transferred from the 2D plane onto the 3D input surface using their UV coordinates (Figure 10, E). For each on-surface point, a lofted point was found along a vector normal to the surface and at a distance determined by
the Curvature method. Polyline boundaries of the tiles were then drawn through the base set of points and through the newly created set of points offset along the direction of the surface normal (Figure 6, C).

A polygonal cell-skin (Figure 6, G) was generated at base using the base polyline and then a cell-wall (Figure 6, E) was generated as a loft between the base and the offset polylines. A unique textual label was then positioned at the centre of the cell.

Cells were generated so that the vertices of the adjacent sides shared the same coordinates. This geometrically precise solution did not take into account the thickness of the cell-wall material, the thickness of the cell-skin flap and the thickness of the glue. This introduced an error that accumulated to a significant amount. There were 76 touching cell-walls along the top edge of the outer shell. Assuming that each doubled cell-wall had a thickness of 6mm it is possible to approximate that the length of the top edge of the shell as constructed would be by 456mm longer than of that modeled digitally. This is a significant amount and the script attempted to compensate for it by creating copies of cell-walls that were inset towards the centre of the cell, along the cell-wall normals, by a designer-specified amount (Figure 6, J).

Cell-skin was then named and exported as a .DXF file into a specifically created directory. Rhino does not support unfolding for polygonal geometry and this task was accomplished in FormZ. FormZ 3.9 that was available for the project did not support scripting and the unfolding of the cell-skins was accomplished manually. This process could be automated in a later version of the software. In addition to unfolding, flaps for gluing (Figure 6, F) were manually added to each unfolded cell-skin even though this procedure could potentially also be encapsulated into a custom script.

Next, the script unfolded the cell-wall and laid it out on the horizontal plane, positioning it alongside the other unfolded strips and assigning to it a textual label. The textual output of the UnrollSrf command used for unfolding was written into a log file. This was a necessary quality-control measure. Each side of a cell-wall produced by this process is a non-planar surface and this non-planarity can reach high degrees if the cell-wall sides are long and the local surface-curvature is high. In some cases, the UnrollSrf command can fail completely or generate a warning if it registers that the area of the resulting unfolded surface varies significantly from the area of the original surface. The log file provided the facility to catch and treat these problem cases.

All types of objects created by the script were automatically sorted into the relevant layers and named where appropriate to facilitate manual inspection and manipulation. Being able to quickly identify and/or track each detail is crucial when dealing with many uniquely parameterized but topologically identical elements. The content, size, position and orientation
of the labels were all taken into account when managing components. Several project-specific auxiliary scripts were written to aid the tasks of selection and manipulation.

Writing an entry for the unfolded cell-wall into the log file and assigning a random color to the newly created elements for easy visual identification completed one iteration of the loop. The same sequence was repeated for each cell. After all of the cells were created, the script closed the log file, set up the viewports as appropriate, turned off auxiliary layers and exited.

We failed to find the software that would be able to successfully automate the nesting of the unfolded components. Therefore, the unfolded cell-walls were manually nested to fit the dimensions of the cardboard sheets (see Table 2) and the size of the laser-cutter table (1850mm x 920mm). Some longer cell-wall strips had to be split in two and in rare cases three parts, which again emphasized the importance of consistent content management.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Inner shell</th>
<th>Outer shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cells</td>
<td>No.</td>
<td>700</td>
<td>860</td>
</tr>
<tr>
<td>Point-distribution method</td>
<td>name</td>
<td>Curvature</td>
<td>Curvature</td>
</tr>
<tr>
<td>$D_{\text{min}}$</td>
<td>mm</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$D_{\text{max}}$</td>
<td>mm</td>
<td>600</td>
<td>800</td>
</tr>
<tr>
<td>Cell-width method</td>
<td>name</td>
<td>Constant</td>
<td>Curvature</td>
</tr>
<tr>
<td>Cell-width range</td>
<td>mm</td>
<td>70</td>
<td>70 - 160</td>
</tr>
<tr>
<td>Total area of cell-walls</td>
<td>m$^2$</td>
<td>48</td>
<td>101</td>
</tr>
<tr>
<td>Total area of cell-skins</td>
<td>m$^2$</td>
<td>37</td>
<td>59</td>
</tr>
<tr>
<td>Average number of sides per cell</td>
<td>No.</td>
<td>5.8</td>
<td>5.8</td>
</tr>
<tr>
<td>Cell-wall cardboard caliper</td>
<td>μm</td>
<td>3000</td>
<td>2000</td>
</tr>
<tr>
<td>Total weight of cell-walls</td>
<td>kg</td>
<td>94</td>
<td>138</td>
</tr>
</tbody>
</table>

3. Manufacturing and Construction

The files containing the data for plotting and cutting of the components were prepared in an appropriate format to be sent to the manufacturer. This was a labour-intensive and mostly manual process.

Cell-walls were pre-manufactured on a laser-cutting table (Figure 12, A). The cutting machine was calibrated to create two types of cuts: a near-full cut to separate the detail from the rest of the material and a partial cut to establish fold-lines. The details were never fully cut out because it was expected that the whole sheets would be easier to transport and the details would be easy to punch out on the site of assembly. However, this introduced badly-cut cell-wall boundaries in places where the card was not fully flat on the surface of the cutting table and the separation of details
proved a time-consuming process. Cell-skins were first plotted on transparent film and then hand-cut to shape and the gluing flaps were pre-folded. The logistics were a major challenge. Large numbers of similar-looking but uniquely-shaped elements required stringent procedures in naming, sorting, transporting and assembling. This part of work was conducted in Cambridge and the ready components were then transported to Prague.

In Prague, the first step in element preparation included the removal of the pre-cut cell-wall strips from the cardboard sheets (Figure 12, B), folding end-to-end and joining with reinforced adhesive tape. Then, the cell-skin was attached to the corresponding cell-wall with hot-melt glue making up a complete cell (Figure 12, C). Assignment, positioning and orientation of labels were crucial at this stage. Thereafter, the assembly of patches (Figure 12, F) consisting of a group of adjacent cells was undertaken in the areas that provided easy access and sufficient flat-floor area. The adjacent cell-walls were also attached with hot-melt glue. The assembly was considered to be an engaging performance in its own right worthy of access by the volunteers and the visiting public. The assembly of components directly in the stairwell would also be a practical impossibility because that space was poorly lit, confined, did not have large enough areas of flat floor and was in constant use during the days. The cell patches had to be made to specific dimensions so that it would be possible to move and mount them without specialised machinery and so that they would fit into doors and freight lifts of the museum. For this purpose the outer shell was arbitrarily split into 14 vertical patches of approximately the same size, each with an individual suspension point. The practice has shown the selected method to be far from perfect as it was difficult to assemble curvilinear patches on the flat floor and even more difficult to suspend and match their seams in the dangerous heights of the stairwell space.

The next stage included the installation of the suspension structure in the stairwell and mounting of the assembled cell patches. As anticipated, problems were occurring at the fold lines of the cells-walls where the cardboard was weakened by score-lines. The structural load had to extend through the cells and wire ropes towards the ceiling. The wires were fixed to the touching cell-walls of the upper rim. The locations of the suspension point had a significant impact on the deformation of the shells. The cell walls below the suspension points were in tension and that could lead to bending while the parts of the structure between the suspension points would attempt to sag. The structure was very light but given the thicknesses of the cell-walls and the dimensions of the cells the stresses were significant. Structural continuity around the corner-areas of the cells was crucial. The cells adjacent to suspension points were reinforced with extra tape and the nearby cells were joined with metal bolts in addition to glue (Figure 12, D).

The final stage of the construction was the suspension of the cell patches in the stairwell. Each shell could only assume the final form when fully assembled. Before the patches were joined together, nothing was forcing them into their ultimate from and they would distort under the influence of gravity and pull/push forces applied during assembly. Forcing the patches into the designated shapes in an eight-meter-high confined space was a considerable challenge. A full scaffolding system had to be constructed and subsequently gradually lowered and disassembled as the patches were put into place (Figure 1; Figure 2, right). Given this technique it would likely to be
beneficial to sub-split and assemble the shells in smaller horizontal rather than vertical sections.

Several possible layouts of the attachment to the existing stairwell structure were considered. Because of the bad condition of the ceiling and the difficulty in reaching it, it was decided to position several wire ropes horizontally, fastening them to the walls. These ropes meant to provide a grid-like basis for the secondary ropes extending to the suspension points. Temporary synthetic ropes were used to hoist the patches into place and orient them correctly. Two adjacent patches suspended and positioned as appropriate were then connected along the shared seam with bolts and nuts.

Suspending patches one at a time temporarily overloaded some cells as the patches deformed under gravity. This exposed problems in cell attachments introduced by poor quality matching and gluing. Again, nuts, bolts and washers were used to reinforce the problematic connections.

Three dedicated computers running custom-written MAX/MSP/Jitter code were installed in the disused lift and connected to the camera, speakers and projectors (Fig. 7; Fig. 8). This ensured that the audio-visual system was in operation during construction and could be readjusted in response to the emergent situations.

3.1 MATERIALS

Grey-centered cardboard with two-side white lamination and transparent film coated on one side and suitable for use in A0 ink-jet plotters were used to make cell-walls and cell-skins respectively. The project utilized 3000μm cardboard for the outer shell cell-walls and 2000μm cardboard for the inner shell cell-walls. A number of tests were carried out with honeycombs and other cellular structures and the details of the implementation were discussed with engineers. The inverse relationship between the cell-wall width and the size of the cells was intended to provide near-uniform structural strength at different curvature levels while saving material, minimizing weight and avoiding highly non-planar component surfaces. As the cells were joined side-by-side, all the walls apart from upper and lower rims had double thickness. The rims were additionally reinforced with an extra layer of cardboard.

Being multi-vertex polygons with flexible joints, the cells did not automatically assume their intended form. This made them structurally weak and difficult to identify and fit. Cell-skins, made from thin and light but rigid and strong plastic film, locked each cell into shape and greatly contributed to the structural strength of the structure.
3.2 TIME AND COST

The project was under development for more than 6 months with 6 people involved in the design and fabrication processes on a regular basis and more helping in the later stages. The on-site assembly was particularly labor-intensive with more than 30 people involved in various kinds of work for not less than 20 days. Cardboard is a relatively inexpensive material. However, the production costs are not insignificant. Altogether, there were 1560 cells, each cell had an average of 5.8 sides, and each side required a series of discrete operations, the total amounting to tens of thousands of individual operations. The demands on time and cost would be significantly reduced through automation of the manufacturing stage and rationalization of the logistics but the development of a cost-effective building process would take its own time and further investment.

As a very rough estimate, it is possible to predict that with the described workflow the full structure as designed could be completed in 70 working days by a team of 10 people at a cost of £70.000 if full commercial rates were used.

4. Conclusion

This paper described a series of experimental techniques for procedural design using as a case-study a practical project developed and constructed for a large public event. The primary research aspiration behind this work was a desire to develop the backbone process that could illuminate the significant theoretical and practical issues and serve as a basis for future work. The experimentation has confirmed that adaptive non-periodic patterns such as Voronoi can be implemented into a procedural design workflow. This integration provided the first steps towards a unified design approach that considered place-specific form-finding together with custom-built audio-visual design. It was shown how structure uniformity and
density, cell orientation, cell depth, and parameters of cell-skin can be procedurally fine-tuned as interrelated system components.

Future work might look at two areas: the use of patterns other then Voronoi and more intelligent generation of patterns. For example, Cambridge University’s Department of Plant Sciences (2005) works on computational simulative models that strive to uncover the way plant-cells specialize, grow, adapt and make up complex structures via cell-to-cell interactions under the influence of local conditions. While this kind of research seeks to understand and modify the growth of plants it can also provide techniques and insights to inform the architectural design of adaptive structures able to host responsive performative situations. Morphogenetic models of growth and cell division can be adopted to and around a unified system able to satisfy multiple spatio-mechanical, functional and performative factors of a complex design situation.

Acknowledgements

Giorgos Artopoulos and Stanislav Roudavski were responsible for the production, direction and the bulk of work on the project. However, the project would not have been possible without the generous help of more than fifty people. In particular, we would like to mention content contributions from Andrew Kudless (programming), Chris Rogers (interactive system development and programming), Panos Demopoulos (sound), Iannis Artopoulos, Popi Iakovou, Nikon Microscopy (USA) (source images). We are also grateful to the following organizations for financial support: James Cropper (UK), Kappa Attica (UK), Automated Cutting Services, Ltd. (UK), Buro Happold Engineers (UK) and several Cambridge University bodies (CUMIS, Kettle’s Yard, King’s College, Queens College and Worts Fund Committee).

References

[Accessed 9 November 2005].
Department of Plant Sciences Home Page [online].


