

Creation and Comparison of Sustainable Neighbourhood Patterns

Jennifer Fernquist*

Department of Computer Science, UBC

ABSTRACT

In this paper I describe an interactive system designed to help landscape architects easily create and compare sustainable neighbourhood patterns that meet predetermined criteria. I first obtained application design ideas after observing the group create patterns in their current manner with paper maps and buildings in a mock pattern collaboration session. Ultimately I created a brand new pattern design system that offers users familiar interactions while providing a plethora of new functionality along with infovis solutions to help them evaluate and compare various neighbourhood layouts. This version of the application was built on a desktop computer but its fundamental design considerations support its inevitable transition to a large multi-top table for collaborative work.

Index Terms: Information Visualization, CSCW, Sustainability

1 INTRODUCTION

Real-world collaboration on multi-touch tabletops is still a relatively underexplored area. This paper investigates one application and offers a group a high tech solution for their collaboration needs, providing both novel and familiar interactions for carrying out their activity as well as some data visualization components to aid them in their decision making process. While this system will eventually be ported to a large SMART multi-touch table for use by multiple participants, this first iteration offers single user interaction on a standard desktop machine.

I am working with the School of Architecture and Landscape Architecture group at UBC, lead by Dr. Ron Kellett, who specialize in designing sustainable neighbourhood patterns for communities within the Greater Vancouver Regional District [7]. The group collaboratively configure groups of elements together to create a portion of a neighbourhood, or pattern. These elements are typically buildings of various types such as residential, commercial, or mixed use. Each element has several numerical attributes associated with it, including:

- **Population density:** the number of people that reside in a building.
- **Jobs:** the number of jobs this building provides.
- **Energy consumption:** the number of GW/h the building consumes.
- **Functional land use:** the amount of residential/commercial/etc. space in a building.
- **Floor-to-area ratio:** the ratio of the total floor area of the building to the size of the parcel.
- **Surface coverage:** the amount of pervious or impervious surfaces that record water drainage.

*e-mail: adara@cs.ubc.ca

Once a pattern is created numerous quantitative outputs can be computed, including:

- **Total population density:** the total number of people will live in the neighbourhood.
- **Total jobs:** the total number of jobs that are created by the commercial units in the area.
- **Total energy consumption:** the total amount of energy the entire neighbourhood consumes.
- **Transit density:** are enough people in jobs to justify high density transit.
- **Land use diversity:** how many different kinds of buildings are there in the pattern.
- **Pedestrian connectivity:** how easy is it for people to walk around the neighbourhood.

The group currently designs patterns using pieces of paper on a large table relative to the size of the pattern being created. They do not currently have a method of dynamically computing output values for ad hoc pattern creations. In this project I aim to provide this group with a first iteration of a pattern creation system on a desktop computer. It is designed with tabletop and collaborative principles in mind for when it is ported to a multi-touch table device. The system strives to provide a familiar interaction with maps and elements while providing a great deal more functionality for selecting elements as well as creating, evaluating and comparing patterns.

1.1 Data Set

I used data provided to me by the group, including a CSV file with a list of 92 elements each with 134 attributes exported from their online database (<http://elementsdb.sala.ubc.ca/>). I was also given 29 images of elements that they printed and used during their pattern creation sessions. After discussion we decided for this iteration to take a subset of 29 elements from the CSV file, those of which they had images for. These contain 3 building types (residential, commercial, and mixed use) with three different densities (low, medium, and high). For each element we consider the 4 attributes that they are most interested in: population, jobs, energy use, and floor-to-area ration (FAR). Due to size constraints during this initial development process, I currently use a fabricated map image approximately 1/4 the size of a map typically used by the group.

2 RELATED WORK

The EMDialog system by Hinrichs et al. [4] presented a large interactive display showing two interconnected information visualizations. It was placed in a museum as an installation, augmenting an exhibit on Emily Carr and providing information about the artist's life. This system had no instructions and many users had difficulty discerning exactly how they should interact with it. Since none of the museum-goers were expert users, perhaps more care should have been taken to design the interface to be more intuitive so supported interactions are obvious.

There has been some work done developing co-located collaborative infovis solutions on tabletops devices. Isenberg and Carpendale [6] created a system for hierarchical data comparison tasks on a large table display. Their approach made use of both shared and individual views to support both individual and group work. Certain aspects of this work differ from mine, primarily that the Landscape Architecture group does not conduct work individually and that their tabletop device only supports stylus input (one per user) while ours will support multiple finger inputs from all participants.

Horn et al [5] describe a collaborative infovis system for a Microsoft Surface multi-touch table that is designed to explore the Encyclopedia of Life [1]. They use Voronoi treemaps to display the large hierarchical data set, labelling each Voronoi cell with an organism category or name. While the authors offer several good touch solutions in their application, they do not fully account for the design considerations of having multiple users. The primary task of the system is to read through several hierarchical layers of text labels, a task only 25% of people at the table can do easily since the Voronoi treemap does not seem to be rotatable. Also, there is a single primary task; there is only one path that can be taken to navigate through the single treemap and this is likely to be dominated by a single user.

In [9] Munzner et al. describe QuestVis which is a visualization system for an environment sustainability model. It allows individual users to explore a high-dimensional data set of 294 dimensions in an effort to increase their understanding of how making certain decisions can affect the overall sustainability of the area. This system provides helpful insights into the design of a visualization that presents sustainability data, however it handles far more dimensionality than the system described in this paper.

3 OBSERVATION SESSION

I conducted an initial observation session with all members of the group to oversee and understand their current pattern creation approach and attempt to glean insights into the design of my application. Many of the research methodologies I employed during this session were based on the observations conducted in [11] and [8]. The session went through a mock pattern creation task, utilizing real elements and corresponding data with a real map used in a previous session. They invented a task of attempting to increase the population in the area by 550 people as well as add an unspecified number of jobs. The 1.5 hour session was videotaped and I observed and took notes.

After the task was defined they wrote the goals on a whiteboard which they referenced repeatedly as they designed the pattern to see how close they were to reaching them. The materials they used to create the pattern included a large paper map and dozens of small pieces of paper that represented the buildings, or elements, which they would tape to the map. Prior to the start of the session the pieces of paper were placed in a pile to one side of the map, sorted manually by type and then by density by one of the team members. The pile of elements remained stationary at the side of the table throughout the session. The paper elements had pictures with top-down views of the buildings, coloured by type and density, along with some markings to indicate a few of their attributes. Figure 1 has an example of one of their low density residential element pieces. Only one group member was familiar with the notation used on the element pieces (e.g. ‘3p’ means that this dwelling houses 3 people) and she had to describe them repeatedly to the rest of the team. All group members were familiar with the colour scheme of the buildings and their types. During the entire session the map was stationary; it was never moved or rotated to be closer to or face any particular person.

While deciding which elements to place, the team would consider 2-3 potential elements to place. One team member would attempt to calculate in his head the approximate output values if

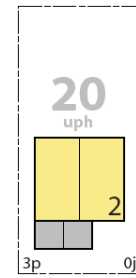


Figure 1: An example of an element used by the group during their session. This image of a low density residential building was on a small piece of paper that could be taped to a large paper map of a neighbourhood.

several of a particular element were laid on the map. In the event that he did not have the information in his memory he went to his computer to look it up. Once an element was decided upon, another team member would retrieve one or more of them from the pile of paper elements and pass them to his team if they were closer to desired portion of the map than he. After several pieces were placed and taped onto the map the team member who understood the markings on the elements would calculate the total output attributes in an Excel spreadsheet. They would then evaluate how close they were to reaching their goal. This continued for several iterations.

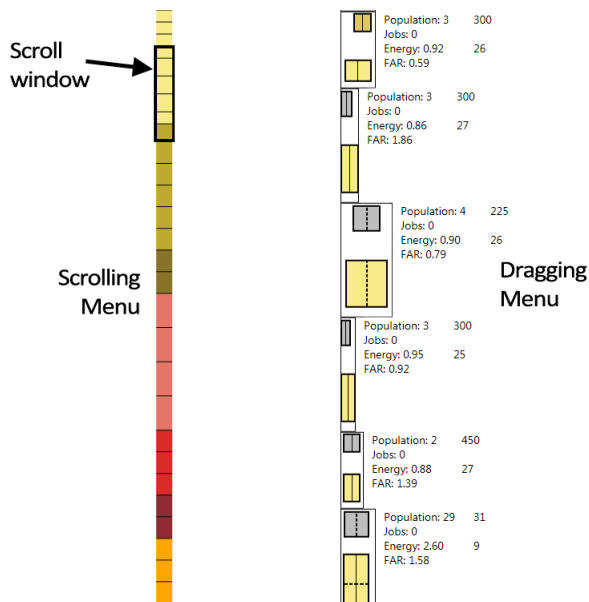
After they met all of their goals and the pattern was complete one of the group members offered an additional simulated task, stating that a city official declared the neighbourhood has too much commercial space. The group then set about meeting the housing requirements while reducing the number of commercial buildings used. They peeled off the undesirable elements and re-taped new ones with less commercial space, working in the same fashion as before until they met their updated goal.

4 SOLUTION

The pattern creation system I developed consists of three primary components: menus for scrolling through and selecting from lists of elements; a static map image to place the elements on; and a series of dynamic charts that display the output values for the current and any saved patterns. An overview of the system can be seen in Figure 8. Many of the fundamental tabletop design considerations specific to placement of and interaction with the components were inspired by [10].

4.1 Scrolling and Dragging Menus

There are two ways in which users can navigate through the list of elements; the scrolling menu or the dragging menu. The scrolling menu displays an overview of the entire list of elements (Figure 2(a)); each block in the scrolling menu represents an element. The height of the block is relative to the height of the element while the widths of all blocks are consistent. The blocks/elements are sorted first by type (residential, commercial, mixed use) and then by density (low, medium, high). This is consistent with the method employed by the group during the collaboration session that I observed. The colouring of the blocks is also consistent with the group’s current usage, with yellow, red, and orange indicating residential, commercial, and mixed use, respectively. Density is shown by darkening the building type colour, where darker is more dense. The scrolling menu also contains a black rectangle that acts as a window whose size is relative to the visible portion of the dragging menu. Users can interact with the scrolling menu either by dragging the black window or by clicking on a block, the latter of which causes the black window and subsequently the dragging menu to center on that element.



(a) The scrolling menu. (b) The dragging menu. Users can navigate by the properties of the various elements dragging the window or by and navigate by dragging the elements clicking on a block, which up or down. represents an element, and the window will center to that block.

Figure 2: The two element selection menus. The scrolling menu provides an overview of all of the elements; each block represents a building and the black scroll window can be dragged for quick navigation. The dragging menu display detailed information about the elements within the scroll window. This menu can also be scrolled by dragging the elements for more precise navigation.

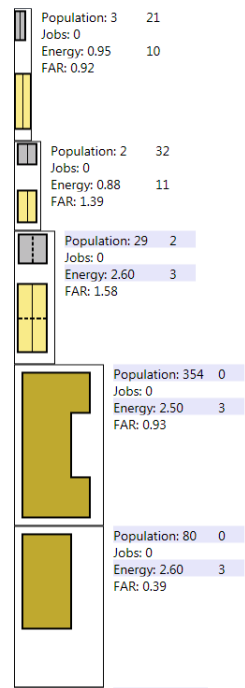


Figure 3: A portion of the dragging menu. The elements with lavender highlighting indicate that 5 or less of those elements can be placed in the map before a target value is reached. For example, if 4 of the third element in this list are placed in the map then the total energy usage will be more than the target value for this pattern.

The dragging menu shows the full sizes of elements plus information about them (Figure 2(b)). It displays the top-down view of the elements in sizes relative to each other and the scale of the map. Along with these images the dragging menu shows the four input attributes for each element. In this design the input attributes have full named labels rather than small, cryptic symbols written directly on the images. This is not only more comprehensible in general but more visible due to the resolution of the screen. In addition, next to the Population, Jobs, and Energy attributes there is a digit that indicates the number of this type of element that can be placed in the map before the target value for that attribute is reached. When this digit is 5 or less, the attribute line for the element is highlighted in a pastel lavender colour to indicate to users that this element is potentially not a desirable candidate (see Figure 3). This light, low saturated colour minimizes interference with the black text [12].

As a user manipulates the black window of the scrolling menu the dragging menu is updated to show the full element size of those element blocks that appear within the scroll window. The dragging menu can also be manipulated by touching an element and dragging it up or down for fine-grained navigation.

4.2 Map Area

The center of the screen displays a map of the area the pattern is being designed for (Figure 4). For the purposes of this project the map is approximately 1/4 of the size as those that are typically used by the group. The grey lines denote roads and the light yellow blocks are empty lots that buildings are placed into. Users can select elements to place in the map by dragging their image from the dragging menu. They are placed in the map simply by dragging them to the desired location and releasing the mouse. The elements can be rotated or translated with and they employ a 'flicking' behaviour so they can be tossed across the screen, perhaps in the future to other users. Once an element is in the map area its attribute information

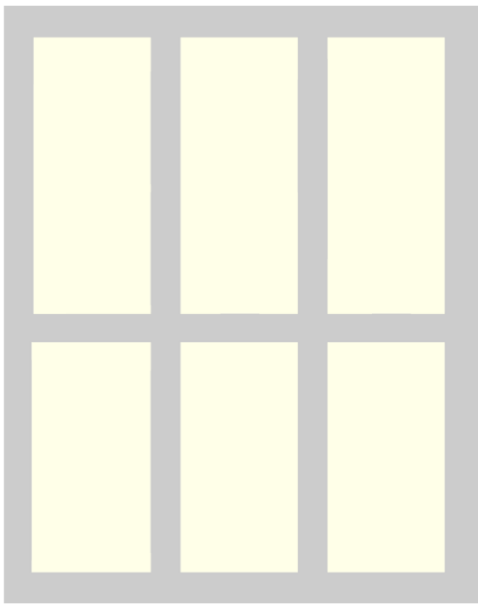


Figure 4: The map displayed in the center of the screen. For the purposes of this project, this fabricated map is approximately 1/4 of the size of those that are typically used.

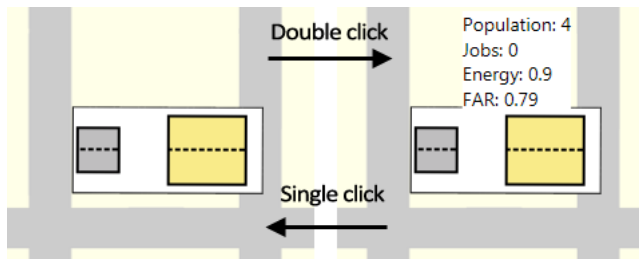


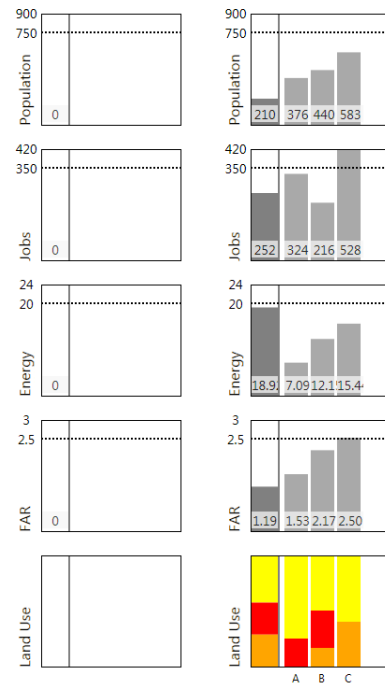
Figure 5: When an element in the map is double-clicked, its attribute information is shown in a text box. The information can be hidden again by single clicking the text box.

can be obtained by double clicking on its image to bring up a text block with the data (see Figure 5). The text block is removed with a single click. A single element can be deleted from the map by dragging and releasing it off the map and the entire map can be cleared by clicking the 'Clear Pattern' button.

4.3 Output Charts

There are five charts displayed on the screen, one for each of the four attributes and one for land usage (Figure 6(a)). Design of this component was partially inspired by [9]. Separate charts for the attributes are used since they all have different scales. Next to the first 4 charts are numerical labels indicating the target value for that attribute, along with a dashed line, and a maximum value. The target value must be selected before the creation process and the maximum value is 20% greater than the target. The motivation behind distinguishing these two values is due to the complexity in trying to meet four separate target values for each of the attributes. Since it is highly unlikely that all targets will be met exactly and one or more values may end up being above their target value, the system permits a visualization of precisely how far above the target an output value really is. Figure 6(b) shows the chart component after three patterns have been saved.

Each element that enters the map causes the charts to be up-



(a) The initial display (b) After three patterns have been saved.

Figure 6: The five charts represent the five output attributes that the Landscape Architecture group is interested in evaluating. The numerical labels on the top four charts indicate the target value, next to the dotted line, and a maximum value.

dated. The population, jobs, and energy values are simple summations while the FAR is an average for the whole pattern. The land use chart displays a stacked bar that is always 100% full to show the percentage of the types of buildings that make up the pattern. The colours used in the land use chart are consistent with the colours used for the element types (yellow, red, and orange). The top four bar charts display their bars as grey. Since the map itself is very colourful due to the collage of elements I did not want to use any of those colours as the bar colour. I also chose not to use colours closer to the blue end of the spectrum since it is much more difficult to perceive detailed information in that colour and thus would be difficult to distinguish height differences between blue or purple bar heights.

There is space in each chart for up to five bars and the information for up to 4 saved patterns to be displayed simultaneously. Since the group typically creates and compares 3-4 potential patterns, this is sufficient. The leftmost bar in the charts, separated by a vertical line from the rest of the chart, always represents the output values for the current pattern on the map. At any point the current pattern can be saved by clicking the 'Save Pattern' button. When that button is pressed two things occur: first, a small image of the map appears at the bottom of the screen and it is automatically assigned an alphabetical character label (Figure 7); second, the bars and output values for the pattern are stored in the larger portion of the bar chart (Figure 6(b)). In the chart view, the bars for the same pattern are in the same vertical line with their corresponding alphabetical character at the bottom. The left-to-right ordering of the small map images is the same as the left-to-right ordering of the saved bars in the chart component. A pattern can be restored to the main map area by clicking on its small map image representation.

5 IMPLEMENTATION



Figure 7: The small multiple representation of a saved pattern that appears on the bottom of the screen. The pattern is automatically labeled with a letter.

The entire application was built with Visual Studios 2008 in C# and utilized the SMART Table SDK libraries. I did not make use of any other libraries or existing infovis technologies. I chose to develop with this SDK rather than develop in another language that would support the use of infovis components so as to more easily port the application to a multi-touch table. The SMART libraries provide built-in interaction for the element objects on the map, including the rotate, translate, and flicking behaviour; however, the touch input handling portion of the libraries prevent the use of even basic C# inputs and controls, such as menus and scroll bars, since they do not respond to emulated touch inputs. Nor was I able to re-emulate the touch inputs as mouse clicks. As a result I developed all components from scratch as well as touch input handlers for all interactions except for those that perform basic element manipulation, such as rotation and translation.

Touch input handlers were added to both of the menus. The scrolling menu separately manages touches to the scroll window or to the individual blocks, calling the dragging menu when it is being navigated. The dragging menu handles touches to its elements, calling the scrolling menu handler when they are translated in the y-direction or creating new elements when dragging in the x-direction and pulled off of the menu. Custom touch input handlers were also added to the Save and Clear buttons, to the saved pattern images to manage pattern restoration, to the elements placed on the map so their attribute data can be popped up, and to the popped up text blocks.

The entire application was built with Visual Studios 2008 in C# and utilized the SMART Table SDK libraries. I did not make use of any other libraries or existing infovis technologies. I chose to develop with this SDK rather than develop in another language that would support the use of infovis components so as to more easily port the application to a multi-touch table. The SMART libraries provide built-in interaction for the element objects on the map, including the rotate, translate, and flicking behaviour; however, the touch input handling

The scrolling menu consists of 29 rectangles, one for each element, with 28 black line separators. The entire height is the length of the screen while the heights of the blocks are proportional to the original heights of the elements. The widths are kept consistent for clarity. Each element is checked for its type and density while determining its corresponding block colour. The black window is a hollow rectangle, the height of which is relative to the visible portion of the dragging menu. The displacement of the window is calculated and then used to move itself and the dragging menu accordingly, and vice versa while using the dragging menu. The window cannot leave the bounds of the screen and the top and bottom of the dragging menu do not go below the top or above the bottom of the screen.

I heavily modified the original image files for the elements to first remove the symbols and then to increase the quality. Information specific to the elements, including the location to their modified top-down view image and attribute data, are read into the program from a CSV file. The elements objects in the dragging menu, as well as those in the map area, are derived from a class in a SMART Table library and contain information pertaining to their location and rotation angle. When a user pulls an element from the dragging menu the original object is maintained under the user's cursor while another element is created to replace the one in the menu. When a pattern is saved a copy is made of all of the elements in the map as well as the map itself and the copies are shrunk and translated to the lower portion of the screen, with their translation and rotation abilities removed. Clicking on the small map image first deletes all existing objects on the main map and then restores the saved objects to their original locations and orientations on the map.

The charts were created manually using no chart creation libraries. All of the locations, sizes, and contents of the chart outlines, bars, labels, target lines, target, current and max values are individually calculated and rendered at the correct positions. The chart information is updated each time a calculation is performed. This occurs when an element is added or deleted, when the map is cleared, or when a saved pattern is restored. Each new calculation causes the rectangles of the bar charts to be erased and redrawn to their new heights. The labels with the current bar values are also erased and redrawn. In addition, after a new calculation the information in the dragging map is updated to reflect the new candidacy information for the elements and highlighting is added or removed where appropriate.

6 RESULTS

In this section I outline a typical scenario of use while designing a neighbourhood pattern.

6.1 Scenario of Use

A user wishes to create a pattern for a pre-defined street layout. They load the system and are presented with the screen in Figure 8. The scrolling and dragging menus are visible to the left, the street map is in the center, and the output charts are to the left of the screen. The user has predefined targets they wish to meet for each of the output values and these are indicated by the labels next to the dotted lines on the charts.

The user begins designing the pattern, pulling elements from the dragging menu and placing them on the map. They can scroll faster by dragging the scroll window or clicking on a block in the left scrolling menu, or navigate more slowly by using the dragging menu. As they place each element in the map the charts are automatically updated with their corresponding values. Once they have placed several elements in the map their screen begins to look like Figure 9.

As a user designs their neighbourhood pattern, they must keep in mind several factors: the amount of space available and the size of the elements; the attributes of the elements and their impact on

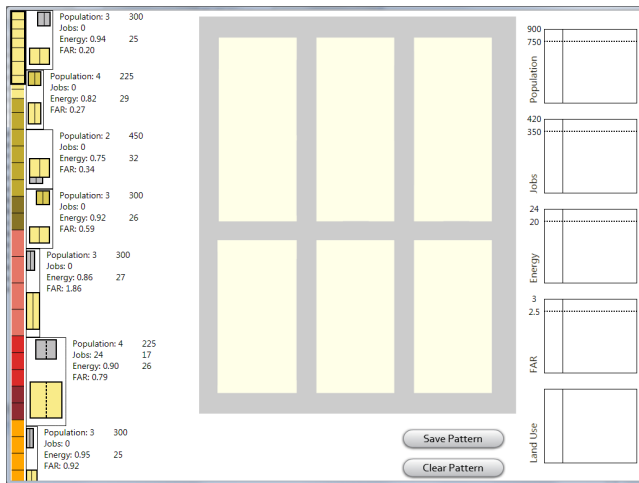


Figure 8: The entire system display before interaction. The scrolling and dragging menus are on the right, the map area is in the center, and the five charts are on the right of the screen.

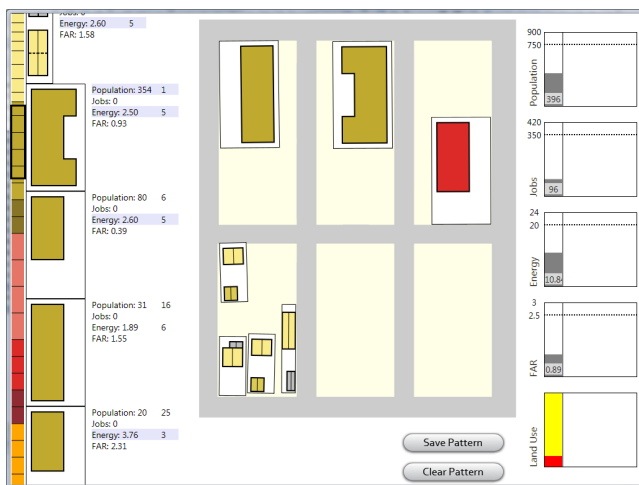


Figure 9: A user has placed several elements in the map. The charts are automatically updated to reflect the current output values.



Figure 10: A user has finished creating a pattern and saved it.

the output; and the value of all of the outputs relative to each other. Each of these factors is addressed in this system. First, the sizes of the elements are relative to each other and the size of the map so as a user navigates the dragging menu they have spatial reference to determine which elements are currently of an appropriate size. Second, the elements in the dragging menu are clearly labelled with their attributes. Once an element is in the map its attributes can also be prompted. In addition, the user has a visual cue in the highlighting as to which elements may or may not be ideal candidates for their next selection. Finally, the output values are dynamically calculated and displayed in the same vertical line for quick reference.

Once the user has finished their pattern and met the target values to the best of their ability, they click on the 'Save Pattern' button and the system stores its information (Figure 10). This pattern becomes denoted by the letter 'A' and is represented by the small map image at the bottom of the screen. This image is a smaller version of the main map display of the pattern. Its output attributes and bars are also saved in the chart component. They are aligned in the same vertical column which is also labelled 'A'. Now the user can select the 'Clear Pattern' button to design another neighbourhood since their initial attempt is safely stored.

The user creates and saves two more patterns (Figure 11). Now they can easily compare each of the patterns' output attributes in the chart component to determine which will best suit the needs of the community. As is clear in Figure 11, no one pattern perfectly satisfies all of the target values so an objective decision must be made. To aid in this process the saved patterns can be restored to the main map for further viewing or manipulation, but any changes made cannot be stored with the same pattern label. In this scenario the user may wish to opt for pattern C since it offers more than enough housing, nearly meets the required number of jobs, and requires less energy than the other two patterns.

7 DISCUSSION & FUTURE WORK

In this section I discuss both the strengths and weaknesses of my system and then outline a portion of the work that will be conducted in the future of this project.

7.1 Strengths

This system provides many benefits to the Landscape Architecture group over their current paper-based pattern creation approach. The scrolling and dragging menus provide both an overview of the entire set of available elements as well as detailed information about specific elements. The scrolling menu's window provides location

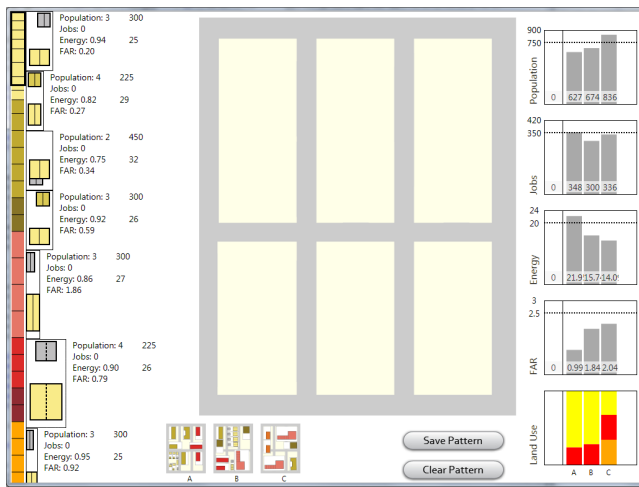


Figure 11: A user has finished creating and saving 3 patterns and can now compare them.

context for the dragging menu. Users are also provided with insights into which elements they may or may not want to select to place in the map; the lavender highlighting of element attributes in the dragging map indicates the viability of a particular element.

The saving pattern functionality provides users with the ability to quickly and easily compare patterns against each other, a fundamental activity for this group. Previously they had no method of saving iterations of patterns during collaboration since only a single paper map was used. With this system up to four iterations can be saved and restored at any time. The output values for the saved patterns can also be viewed and compared.

To help facilitate their transition from using paper to using this electronic system I incorporated a number of familiar items and interactions. I use their top-down view images for elements without the cryptic text and I maintain their colour scheme used for the building types and densities. In addition, the menu is consistently in one place and the map does not rotate or translate, consistent with their behaviour during the observation session.

Interaction with the elements once they are dragged from the menu is very much like manipulating pieces of paper. The objects in the SMART libraries offer realistic interactions with virtual objects that simulate interactions with objects in the real world. The elements can be rotated and translated with one or two fingers and they have a friction component so that they can be tossed across the screen. This attribute will allow users to pass elements to each other much like they did in the observed session.

This application frees up users' memory by displaying the attributes associated with each element and indicating how many of each can be placed in the map until the target values are reached. Users no longer have to perform mental or other calculations to determine the current status of the pattern in relation to the targets; they are automatically updated and displayed in the chart component.

7.2 Weaknesses

As a result of maintaining the colour scheme used by the group, this system does not favour the colourblind. Running screenshots of the application through Vischeck [2] gave poor results, seen in Figure 12. While the groups of blocks in the scrolling menu can still somewhat be distinguished and the dragging menu provides additional context for the elements via their attributes, once elements are placed in the map area their distinction becomes difficult. This is not currently a serious issue since, luckily, no one in the group

is colourblind; however, this may prove problematic if in the future they collaborate with a city official who is.

Another limitation of the system is its inability to provide insight to users about an element that is a good, rather than bad, candidate for inclusion in the pattern. At the start of the pattern creation process all elements are good candidates since, provided the targets are reasonable, no single one will cause any target to be overreached. However, as more and more elements are added to the map and the output values rise to the targets, the highlighting may in fact indicate good candidates; space is limited on the map and if there is only room for a couple more elements, those that have highlighting are more desirable since fewer of them are needed before the target is reached. Perhaps, then, the highlighting could serve both functions depending on the current fullness of the map.

A considerable limitation is the requirement of specifying a hard target value for each output attribute. It may be the case that collaborators wish to simply minimize or maximize a particular output rather than reach a specific target value. For instance, during the observation session the group had a soft constraint of simply adding jobs. It could very well be the case that the energy use for a pattern should be minimized rather than be a specific value. In my system it is currently not possible to specify such vague constraints. If they were permitted it would be very difficult derive calculations that could determine which elements were good or bad candidates.

While 4 element attributes and 5 output values are currently sufficient for this group, the system may not scale well if many more values are to be considered. There is a finite amount of data that can be shown next to the elements in the dragging menu and the number of output charts that can be shown is limited by the height of the screen. Perhaps parallel coordinates such as those used in [3] could be facilitated for higher dimensions of output values.

7.3 Future work

Future work on this project will provide overview and context for the map location. I will offer the ability to zoom the map out to see the configuration of the surrounding area and where in the city the neighbourhood lies. I will also provide functionality for overlaying the neighbourhood with the Google Maps satellite view for the area so collaborators can compare their current pattern design with the current layout. Oftentimes the group designs patterns for neighbourhoods with existing buildings that will be torn down as well as those with empty lots.

I would also like to add the ability to scale the elements once they are placed on the map. The scaling would result in the attributes being automatically updated to reflect the new square footage size. The group has existing rules that determine attributes based on the square footage of a building. This entails many complications. First, that element would no longer be associated with any other element in the menus. In order to duplicate the scaled element it would either have to be added to the menu as a new element so that can be selected, be cloned from the map element, or the original element must be selected from the menu and scaled the precise new size. The former two approaches are most desirable.

It would be desirable to support both clearly and vaguely defined goals in the event that users simply want to minimize or maximize an output value. I would also like to add the ability to dynamically change the target values or optimizations while the application is running.

Future versions of this system will have a snapping feature for elements placed on the map so that they line up with streets and the boundaries of nearby buildings. Currently element manipulation feels quite 'loose' and it is difficult to orient them with a great deal of precision. Also, I will add functionality to either prevent buildings from occluding each other or indicate when they are, such as by outlining the intersecting elements with a thick black outline so they stand out.



(a) Deuteranope.



(b) Protanope.



(c) Tritanope.

Figure 12: Results of running Vischeck on a screenshot of the application.

In order to address the colourblindness problem, I will either work with the group to design a new colour scheme they are comfortable with, or keep the existing colour scheme but include markings on the building images that indicate their type. For instance, the images could have two large capital letters on them to denote their type and density, such as 'R' and 'M' for medium density residential buildings.

Finally, I will evaluate the system with the Landscape Architecture group. I will get feedback on this application and then iterate on the design. An initial user study will be carried out with undergraduate Architecture students, followed by another design iteration. Finally a user study with the senior group will be conducted to evaluate the benefits that this system provides over their existing paper-based pattern creation approach.

REFERENCES

- [1] Encyclopedia of life. <http://www.eol.org>.
- [2] R. Dougherty and A. Wade. Vischeck: Simulation of colourblind vision and images correction for colorblind viewers. <http://www.vischeck.com>.
- [3] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *IEEE Visualization*, 1999.
- [4] U. Hinrichs, H. Schmidt, and S. Carpendale. Emdialog: Bringing information visualization into the museum. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2008)*, 14(6):1181–1188, Nov. 2008.
- [5] M. S. Horn, M. Tobiasz, and C. Shen. Visualizing biodiversity with voronoi treemaps. In *Climate Change and Urban Design: Science, Policy, Education and Best Practice. Third International Council for European Urbanism Congress, Oslo, Norway*, June 2009.
- [6] P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2007)*, 12(5):1232–1238, Nov. 2007.
- [7] R. Kellett, D. Cavens, N. Miller, and J. Salter. Decision support tools for energy and sustainability in charrette-based urban design. In *Intl. Symp. Voronoi Diagrams in Science and Engineering*, Copenhagen, Denmark, Sept. 2008.
- [8] Y. Lanir. *A Paradigm for Classroom Presentations on Large, High-Resolution Displays*. PhD thesis, University of British Columbia, 2009.
- [9] T. Munzner, A. Barsky, and M. Williams. Reflections on questvis: A visualization system for an environmental sustainability model. *UBC Computer Science Technical Report TR-2009-24*, Nov. 2009.
- [10] S. D. Scott, K. D. Grant, and R. L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *European Conference on Computer-Supported Cooperative*, 2003.
- [11] J. C. Tang. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2):143–160, Feb. 1991.
- [12] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, second edition, 2004.