Computer-Aided Design 41 (2009) 159–172

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

ArchiDNA: An interactive system for creating 2D and 3D conceptual drawings in architectural design

Doo Young Kwon [a,*], Mark D. Gross [b], Ellen Yi-Luen Do [c]

[a] Computer Graphics Laboratory, ETH Zurich, Switzerland

[b] Computational Design Lab, Carnegie Mellon University, United States

[c] College of Architecture and College of Computing, Georgia Institute of Technology, United States

ARTICLE INFO

Article history:
Received 15 August 2006
Accepted 6 July 2008

Keywords:
2D and 3D conceptual drawing creation
Algorithmic shape generation
Schematic design process

ABSTRACT

We present an interactive system called ArchiDNA for creating 2D and 3D conceptual drawings in architectural design. We developed a novel principle of shape generation called match-and-attach by analyzing drawing styles of a contemporary architect, Peter Eisenman. The process consists of user interaction techniques and a set of rules that decide how one or more shapes attach to another shape. One key ingredient of our process is a unique concept for the interactive semi-automatic shape generation that uses the combination of algorithmic rules of a computer and designers' manual inputs. These techniques enable designers to use CAD software in the early stages of architectural designs to explore conceptual building forms. ArchiDNA dynamically responds to drawing inputs, configures 2D shapes, and converts them to 3D shapes in a similar style. We intend to complement existing CAD software and computational drawing pipelines for intuitive 2D and 3D conceptual drawing creation.

1. Introduction

Architectural designers draw shapes during the conceptual design process to explore possible forms for the building they have in mind. The shapes they draw derive from various sources: geometry, previous design precedents, or even visual analogies [4]. Through continued sketching and refinement, designers explore variations of these shapes and ideas and develop them further. They continue to draw experimentally with shapes and variations until they recognize that the forms they have depicted are a promising basis for further design development.

Herbert argues that *study drawing* is a medium for designers to find formal design ideas [16]. Unlike final plan drawings with details and specifications, study drawings emphasize graphic ambiguity, which potentially enables a new unintended interpretation in many different ways [35]. Thus, this kind of drawing is useful for designers to explore a building form in the conceptual design process. The ambiguity of drawing may be obtained with irregular overlays of various colors, in shapes, spaces, lines, or in images with any specified degree of irregularity [12].

Today designers use software not only to represent the final product but also to explore a conceptual form during the schematic phase of design [23]. Different from a final drawing, the main purpose of a conceptual CAD drawing is to iteratively represent and develop a visual idea with geometric ambiguity until a suitable form is found. Current CAD systems provide a variety of useful drawing functionalities ranging from simple line or curve drawing to highly specialized shape editing tools. Still, it is time-consuming and labor-intensive to produce many alternative conceptual drawings. This is why many designers use a computer only to represent a final product, but not in the early and creative phases of design.

To overcome this limitation, many rule-based implementations have been proposed to automatically generate drawings using the power of algorithmic generation. The common goal of these systems is to first describe a class of forms to be generated with a set of rules, and then generate drawings based on these rules. In this context, the use of styles has also emerged as an attractive design research. Various approaches have attempted to codify a characterized style from a given drawing with a set of rules, and produce new drawings in a similar style [5,20]. Although they provide outstanding results, there is still a difference between the computational drawing process and the conventional drawing process. Therefore, many designers forgo using software even with its potential advantages in shape exploration.

Now that sophisticated CAD systems exist with advanced display technologies, we would like to make the computer a more creative partner in the conceptual design process. We propose that two key ingredients for such drawing tools are *interactive drawing process* and *shape generation rules*. For instance, a system

* Corresponding address: Virtual Builders, 3rd Floor, 134 Shinchon-Dong, Sudaemun-Ku, Seoul 120-749, Republic of Korea. Tel.: +82 2 312 9249; fax: +82 2 312 9248.

E-mail addresses: dkwon@vbuilders.co.kr (D.Y. Kwon), mdgross@cmu.edu (M.D. Gross), ellendo@cc.gatech.edu (E. Yi-Luen Do).

0010-4485/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2008.07.007

to generate a conceptual drawing should dynamically respond to drawing inputs of designers with algorithmic rules, and also it should provide meaningful, rather than random alternatives. Thus, designers can be encouraged to use their drawing skills, and maximize a chance to generate forms that fit their ultimate design goals.

This paper investigates how a certain style can be codified into a drawing process, and how the process can be embedded into an interactive graphic system that can serve in the early stages of architectural design. As such, our work offers two main contributions.

First, we propose a unique semi-automatic shape generation process called *match-and-attach*. The process enables designers to create 2D and 3D conceptual drawings that exhibit defined stylistic properties. We began with the drawings of the well known architect Peter Eisenman. Through an analysis of Eisenman's design style, we derived a series of steps combining user interactions and shape replacement rules within a generative design framework. Using match-and-attach, designers can quickly generate Eisenman-like drawings.

Second, we explored the general use of match-and-attach in early stages of architectural design. To this end, we developed an interactive system, called ArchiDNA that provides an intuitive conceptual design environment with full control of match-and-attach. Defining their own styles or modifying other styles, designers in ArchiDNA can explore variations that may assist the designer's creativity as they search for new forms. In particular, ArchiDNA leverages the use of conventional drawing skills (e.g., making line or curve drawings using a mouse, and freehand sketch drawing using a digital pen) in an integrated graphical user interface.

The rest of the paper is structured as follows. Section 2 describes related work, introducing examples of style analysis of drawings and various shape generation systems. Sections 3 and 4 present the main contributions of the paper, shape generation process and ArchiDNA software respectively. Example drawings created using ArchiDNA can be found in Section 5. Section 6 concludes the paper with discussions and reflection.

2. Related work

2.1. Analysis of design style

Style is a kind of design knowledge that characterizes a particular design artifact or a group of design works [30]. In particular, style in drawings can be described as consisting of certain perceived shape elements and structures to compose those elements [2,13]. To describe styles formally, Stiny introduced the idea of shape grammar [33], which specifies a set of rules that each replace one shape or part of a shape in a drawing with another. This simple substitution is the basic element of a grammar; Stiny showed that a collection of carefully designed replacement rules can describe a certain style. Work on shape grammar and its computer implementations has analyzed various drawing styles and showed how to generate new instances of existing styles from simple 2D drawings to complex building designs [21].

For example, Stiny's Ice-ray grammar [32] describes and generates instances of Chinese lattice design. It captures the compositional principle of lattice design as a set of rules. The ice-ray grammar generates various patterns. Four rules were defined through an analysis of Chinese window grilles: One rule subdivides a shape by inserting a straight line across a rectangle. Another divides the rectangle into two trapezoids; a third divides the lower trapezoid further into two trapezoids. Finally a fourth rule splits the upper pentagon into a triangle and a pentagon. These subdivisions are applied recursively to generate a Chinese lattice pattern.

In an architectural example, Stiny and Mitchell defined a series of rules for buildings designed by the sixteenth-century architect Andrea Palladio [27,34]. The rules describe the generation of architectural plans that consist of walls, spaces, windows, and entrances. The rules start from defining a single point, which shows the location of the plan on a site. A rectangular grid structures the initial layout and controls all subsequent stages of plan generation. Based on the grid structure, external walls and rectangular spaces are generated to form rooms in the plan. The principal entrances and columns are then added with windows and doors inserted into the walls to complete the plan.

In a similar fashion, Koning and Eizenberg developed a set of rules for a Frank Lloyd Wright prairie house style [24]. The rules extend one 3D object by attaching another object to the right side of the existing object. The composition of a house is completed with semantically named zones such as living and service areas, and porches and bedrooms. Rules also add terraces, a basement, and a second story, and complete the generation of the prairie house by adding a roof and a chimney.

Flemming studied the architectural characteristics of Queen Anne houses and proposed a set of separate grammars for the generation of building plans [10]. Based on Flemming's style analysis, Heisserman implemented the Genesis system, which uses a boundary solid grammar to model Queen Anne style houses [15]. Flemming also defined rules that illustrate how building elements can be placed in space to help students learn architectural design [11]. Students used the rules, first, to understand existing designs, and then, to develop a new design using the grammar.

Hersey and Freedman developed the PlanMaker system [17], which, like Stiny and Mitchell's work, generates possible Palladian style building plans. They developed a unique split process that divides a rectangle horizontally or vertically and then re-splits the previously split rectangles. A split is controlled by three variables: a split direction (horizontal, vertical, or both), number of rooms and split ratio. The number of rooms determines how many rooms will be split; the ratio defines the proportions of the resulting room. This simple split system generates horizontally symmetrical and modular villa plans like Palladio's. A user of PlanMaker chooses the next split operation to be performed by clicking at a particular place in the floorplan.

2.2. Shape grammar generation systems

A shape grammar is a computational abstraction: it is simply a set of shape replacement rules. However, the replacement rules can be executed by a program that generates and displays the results graphically. For example, a program (like Heisserman's Genesis) can be written to exhaustively generate drawings according to the grammar; or to randomly generate drawings; or to generate and select drawings by some criteria. Shape replacement rules can also be executed one by one, under the user's control, as part of an interactive drawing editor. That is the approach we follow here, in the ArchiDNA program.

Tapia's GEdit [36], a two-dimensional shape interpreter, enables designers to draw shapes, and manipulate them to set up substitution rules that define how one shape replaces another. Wang's 3D shaper [37] generates 3D drawings based on the rules of shape grammar. Designers define a size, a type and labels of shapes as well as the spatial arrangement between shapes by typing numerical parameters in a dialogue. The program generates 3D forms.

Duarte proposed an interesting interactive system for "mass customizing" housing based on a programming grammar and a designing grammar [7]. The system first guides users to input design information through a series of questions. Then the system generates designs in the style of Siza's Malagueira houses using a set of pre-defined rules derived from an analysis of these buildings.

2.3. Algorithmic shape generation

Another powerful approach is a programming-based system that provides a symbolic language. Users can write (and debug) formal descriptions of geometric objects, which are then rendered visually by the program. Instead of using shape rules with visual interpretation, designers can build 2D or 3D geometric forms by writing simple scripts through symbolic interpretation. Even though the process is different from conventional drawings, this approach provides a drawing environment where designers can understand the basic ideas of programming and algorithmic shape generation within the context of drawing.

In the style-capturing spirit of shape grammars, but with a different software approach, Kirsch and Kirsch analyzed the style of Richard Diebenkorn's paintings [18] and Miro's Constellation series paintings [19]. They identified the major shape components of the paintings and developed a set of small programs, each of which allows users to place and deform a shape in various ways such as horizontal and vertical scaling, and rotating. The system is implemented as an interactive editor, in which the user decides at each step how to develop the painting design.

A more general system that does not try to capture the essence of any particular style, but provides operations interactively and enables a designer to construct a design from drawing tools is Moustapha's ICE (Interactive Configuration Explorer) program [28]. ICE provides an interactive construction system that uses regulators such as symmetry operators and affine transforms as the operations in a form generation system. The designer builds a drawing or model by sequentially performing operations, and can dynamically change the parameters of any operation that has led to the current state.

An early interactive graphical programming environment was the Logo language [26]. With Logo, children create and manipulate shapes by typing a series of instructions or writing procedures. Graphics are implemented using "turtle", or differential geometry [1]. The designer creates a drawing by directing the movements of a screen "turtle" that draws a line wherever it goes. By embedding drawing instructions in programs, the designer can create families of parametric, or algorithmic, drawings.

In the wake of Logo, Design by Numbers [25] is an interactive graphical programming environment that is intended as an introduction to programming for graphic designers. To generate images, the user controls the appearance of visual elements such as dot, line, and field with simple procedural programming. The interface enables a designer to write simple code to produce a 2D graphic image. Similarly, Processing [29] provides a sandbox version of Java that enables beginning artists and designers to write code to generate interactive images. FormWriter [14] is a 3D turtle geometry system that was used to teach computational form generation to students. In one experiment, students analyzed the structure of Islamic architectural designs and then wrote code to generate buildings with similar characteristics.

2.4. ArchiDNA — shape replacement + interaction for conceptual design

We describe here an interactive graphic system that can serve in early stages in architectural design. Whereas previous approaches generate rather concrete design outputs, we are interested in supporting conceptual drawing. Our approach was driven from a particular design style characterized in the 2D and 3D conceptual drawings of Peter Eisenman, a well-known contemporary American architect. Analyzing Eisenman's drawings, we defined a unique shape generation process called *match-and-attach*, which attaches one or more shapes (an applier-shape) to another shape (a base-shape). Match-and-attach consists of affine transforms, (i.e., a combination of translation, rotation, and scaling) that are controlled by the geometrical properties of a base-shape. The process is unique in terms of semi-automatic shape generation that uses the combination of the designers' manual input and the algorithmic shape operations with defined stylistic properties.

Our work on ArchiDNA is inspired by pioneering works in shape grammar theory [31] and the computer implementations to build intuitive design tools. Unlike a more general shape grammar system, our shape generation rules are limited to affine transformations that add new shapes to the edges of shapes already in the drawing. ArchiDNA complements CAD drawing tools, by adding a specific shape replacement mechanism that seems to support stylistic conceptual 2D and 3D drawing creation and display.

In particular, ArchiDNA explores the use of algorithmic shape generation incorporated with the designer's drawing actions for making 2D and 3D conceptual drawings. In this respect, the role of a designer in ArchiDNA is different from most shape generation systems. In "classical" shape grammar systems designers specify substitution rules and select alternatives that the system generates; in programming-based systems they write and execute code and examine the results. Designers in ArchiDNA need only draw shapes to generate architectural-looking drawings with predefined shape attributes. Because of this underlying intuitiveness, ArchiDNA is easy to use and easy to learn for designers so that it can be an intermediate tool to use advanced algorithmic systems that require a skill of visual and symbolic grammar interpretation.

3. Shape generation process

3.1. Drawing style analysis

Our shape generation process was developed through the analysis of Peter Eisenman's drawings. Peter Eisenman is a famous contemporary architect who uses 2D and 3D conceptual drawings to find the relationship of the formal to the conceptual [8]. In many projects, Eisenman used drawings both to search for a form and idea, and also to explain how the form and idea can be manipulated as a motif [8]. He sets up a series of ideas, rules, or strategies and draws into those, trying to find some form in those ideas. He intentionally manipulates and utilizes the effects of drawing to explore a design problem [8].

Fig. 1 shows a set of drawings from Peter Eisenman's House of Cards project [9]. This series of drawings illustrate how he uses drawings as a generative device to explore formal design idea. He repeatedly adds lines to the previous drawings and develops a plan of the house from left to right. He transforms portions of the design and draws more details.

Eisenman's drawing style is well represented in the drawings created for the Biocentrum biology research center project. Fig. 2(a) shows a final design model of this project. His design concept was inspired by the process of DNA replication. He generated the building form by manipulating shapes that represent the four elements of DNA structure: Adenine (A), Guanine (G), Cyanine (C), and Thymine (T) [Fig. 2(b)]. Four distinct shapes are commonly used to represent these amino acids: an arch (A), a ribbon (G), a pentagon (C) and a concave (T) [Fig. 2(c)]. Eisenman used these as the building blocks for his Biocentrum design. We chose this work because Eisenman's drawings express geometry that is describable and measurable. We hypothesized that his formal idea could be represented as algorithms that manipulate these shapes.

Eisenman's 2D and 3D drawings illustrate how the form of the building came from abstract representations of DNA structure.

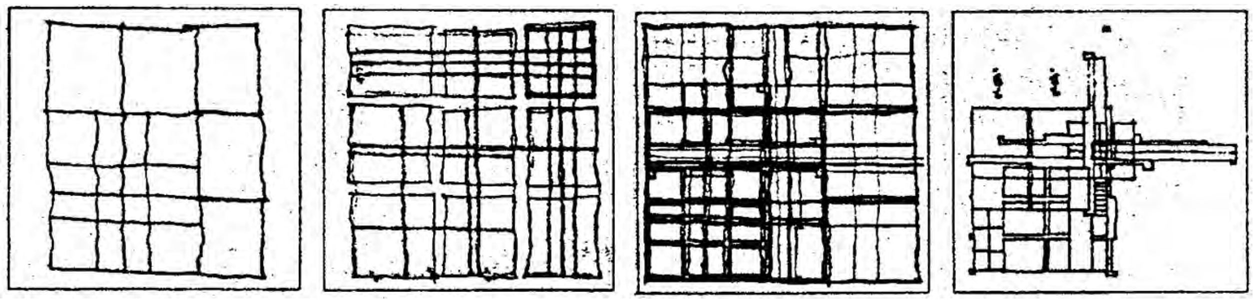

Fig. 1. A sequence of conceptual drawings of Peter Eisenman for the House of Cards project. Source: [9].

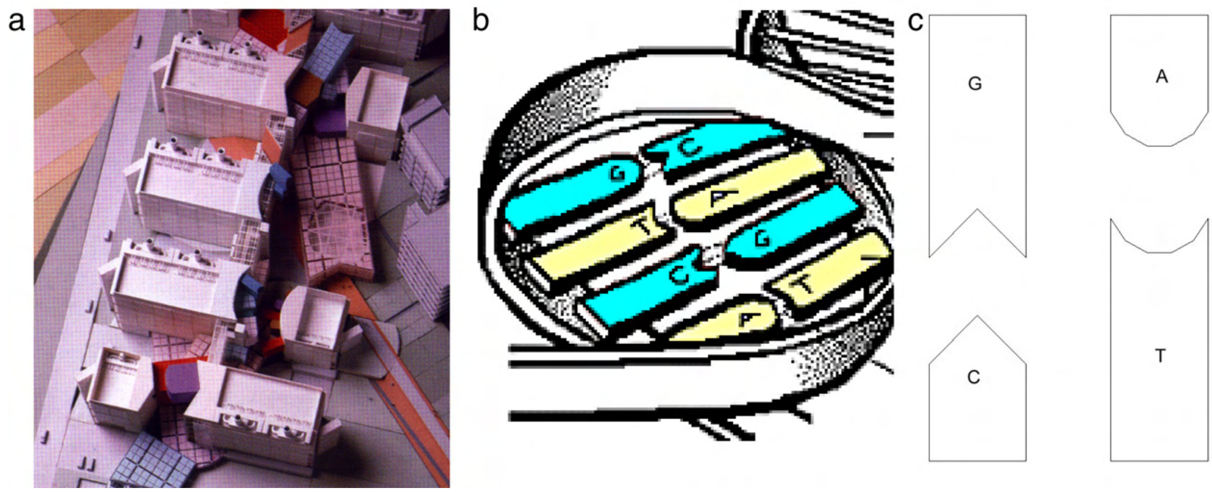

Fig. 2. (a) Final Biocentrum design model. Photo Credit: Dick Frank. (b) Diagram of DNA showing amino acids. (c) Four distinct shapes in amino acids. Source: [8].

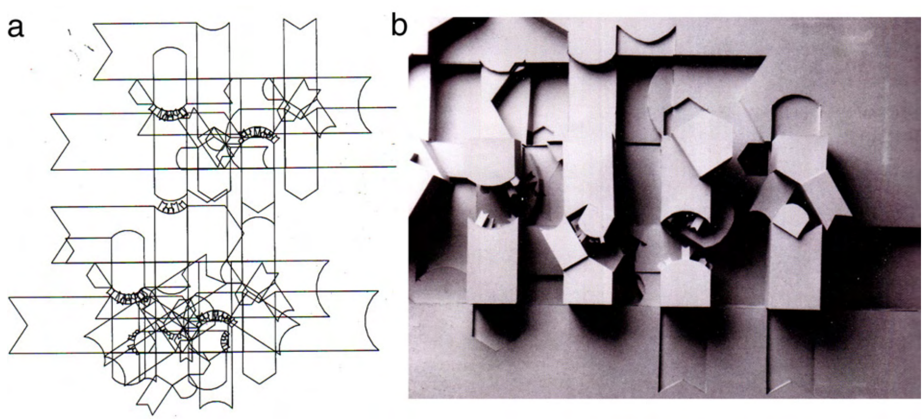

Fig. 3. (a) 2D conceptual drawings. (b) 3D conceptual physical model for Biocentrum (1996). Photo Credit: Dick Frank. Source: [8].

In the 2D drawing [Fig. 3(a)], the four shapes that represent the four elements of DNA structure (A–T–C–G) were translated and transformed to compose the final drawing. The 2D drawings were extruded into a 3D form [Fig. 3(b)].

3.2. Match-and-attach process

"Match-and-attach" is a semi-automatic shape generation process that combines automatic shape generation with the manual inputs of a designer. Using match-and-attach, designers can generate 2D and 3D conceptual drawings that express a certain drawing style. We analyzed drawings of the Biocentrum building design by identifying Eisenman's shape elements and the ways he composed those elements.

Fig. 4 shows an overview of the match-and-attach process. It consists of four steps of user interaction techniques and two operation groups for 2D and 3D shape generation. Each operation is controlled by the geometrical properties of a base-shape. These rules algorithmically amplify and complement a designer's input drawing.

In match-and-attach, designers first define one or more applier-shapes and then select a base-shape. Then the system generates a 2D shape configuration based on user inputs (i.e. selecting base-shapes) and the 2D shape generation rules. Finally the generated drawing is converted to a 3D model based on 3D generation rules. Following this simple process, even a beginning user can quickly generate drawings within a given style. The following sections describe the 2D and 3D shape operations, and demonstrate the generation of Eisenman-like drawings using match-and-attach.

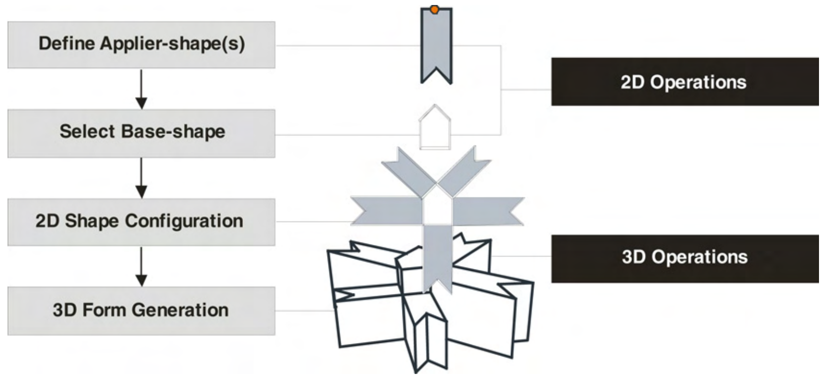

Fig. 4. Overview of match-and-attach process.

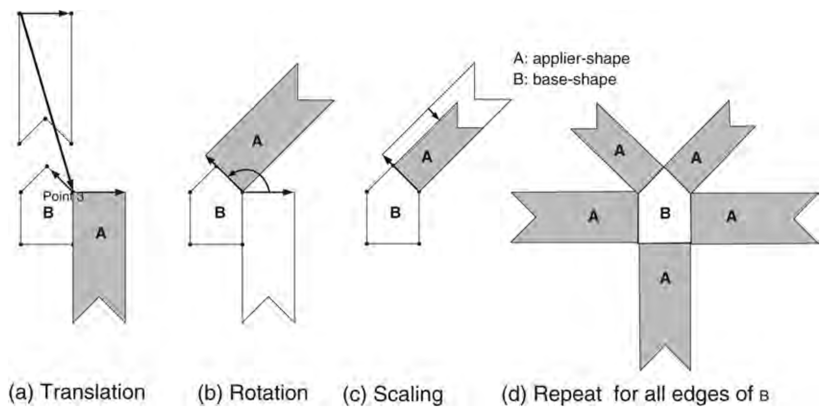

Fig. 5. Four shape operations with applier-shape to base-shape.

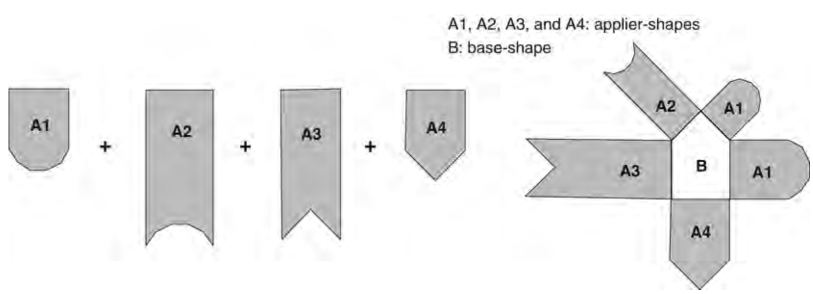

Fig. 6. Four shapes (A1–A2–A3–A4) matched and attached sequentially to edges of base-shape B. Shape A1 occurs twice.

3.2.1. Operations for 2D shape generation

We investigated Eisenman's 2D drawings to understand their geometrical composition and how they might have been produced. Then, we specified four initial shapes: arch, concave, pentagon, and ribbon that represent the four DNA elements (A–T–C–G) respectively. We derived a sequence of operations that determine how one or more shapes attach to another shape. Fig. 5 shows an example of the four-stage procedure (or sequence of operations) using a ribbon (as applier-shape) and a pentagon (as a base-shape). The four stages are (a) translation, (b) rotation, and (c) scaling of the applier shape to match and attach to an edge of the base-shape, and (d) repeating this procedure for all the edges for the base-shape.

A shape object may either play the role of an *applier-shape* (object to be transformed and generated) or a *base-shape* (a fixed shape object to which applier-shape objects attach). Each applier-shape has an *anchor-edge* that is used to match and attach to the edges of a base-shape. A base-shape consists of *attachable* and *un-attachable* edges. Once a designer sets the parameters for match-and-attach, any shape can serve as a base-shape to generate shape configurations. In Fig. 5, the bottom edge of the ribbon is an anchor edge and all edges of the pentagon are attachable.

Our drawing analysis showed that Eisenman applies multiple different shapes to a base-shape with a certain order. We derived another rule to attach multiple applier-shapes to a single base-shape in sequence when there is more than one applier-shape. Fig. 6 illustrates an example that uses the four different applier-shapes in sequence (A1–A2–A3–A4) matched and attached to the base-shape B. It starts with shape A1 and attaches other shapes in sequence, counter-clockwise.

3.2.2. Generation of 2D Eisenman-like drawings

We validated 2D shape generations of the match-and-attach process by making 2D drawings in the style of Eisenman's drawings for the Biocentrum building. We started with eight shapes (2 copies of each of the four DNA shape elements). We first matched and attached the shape G (ribbon) to the base-shape C (pentagon). The process begins by copying and attaching the applier-shape G to every edge of base-shape C [Fig. 7(a)]. Fig. 7(b) demonstrates this process again with a base-shape A (arch), which has eight edges including five short line segments that approximate the curve. Eight shapes are generated, and matched and attached to the eight

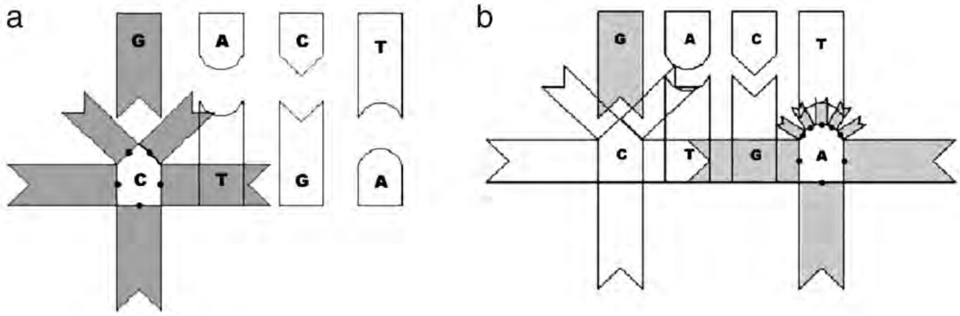

Fig. 7. Matching and attaching the applier-shape G (ribbon) to the base-shape C (pentagon) and to the base-shape A (arch) subsequently.

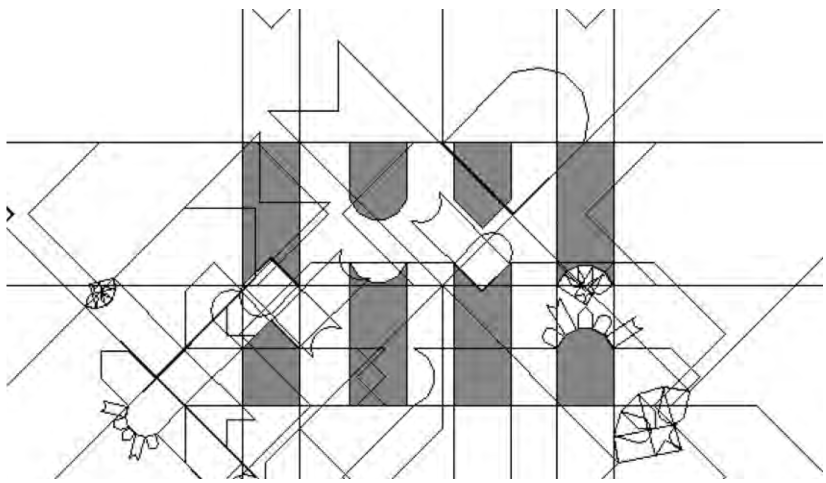

Fig. 8. 2D Eisenman-like drawing generated with four applier-shapes (A–G–C–T). Compare with original 2D drawing in Fig. 3(a).

edges of the base-shape A [Fig. 7(b)]. This example makes clear that our match-and-attach process scales the applier-shape so that its edge dimension matches that of the base-shape where it is attached.

Now we use multiple applier-shapes such as the four shapes (A–G–C–T) and select a base-shape repeatedly. In this way we explore interesting configurations. Fig. 8 shows four applier-shape objects matched and attached to several base-shape objects to quickly generate drawings similar to Eisenman's originals [Fig. 3(a)]. Each applier-shape is moved, rotated, and scaled based on the edges of the various base-shapes we selected.

Fig. 9 shows a possible process and variations of Eisenman-like drawings. Similar to Eisenman's drawing process illustrated in Fig. 1, using ArchiDNA, the designer could repeatedly add shapes and develop portions of the drawing in more detail [Fig. 9(a)]. Fig. 9(b) shows two variations of shape configuration when different initial shape layouts are used.

3.2.3. Operations for 3D shape generation

The operations for 3D shape generation are based on extrusion, a straightforward way of making a 2D shape into a 3D form. Extrusion is particularly useful for creating block-like building masses in architectural design.

We analyzed Eisenman's final design model image [Fig. 2(a)] and conceptual physical model image [Fig. 3(b)]. It seems that the height of each 3D object is proportional to the area of its base polygon, except for eight base shapes that are located in the center with the same height. So we extrude 2D shapes to a height determined by a function of its area, and we enable designers to manually assign a fixed height to a certain shape in a similar way that Eisenman did for eight base shapes.

Note that if shapes are created within bigger shapes, the inner ones will be hidden. It often prohibits a designer from observing interesting building objects. Therefore, another rule controls the direction of extrusion by comparing the area against a threshold. If the area is larger than the threshold, the height of the shape is assigned a negative value, and the 3D object extrudes downwards from the ground of the building. Otherwise, the shape will compose a building mass projecting upward.

A designer can see both small and big objects at the same time. During design, a certain downward-extruded-object can be switched to be an upward-building mass by simply changing its height value from negative to positive. Fig. 10 shows that the small pentagon A1 extrudes upwards whereas the larger pentagon A2 extrudes downwards because its area is larger than the threshold [Fig. 10(b), (c)]. On the other hand, the height of shape B (ribbon) is fixed with a user-defined value.

3.2.4. Generation of 3D Eisenman-like drawings

Following these rules for 3D shape generation, the system automatically generates 3D objects by extruding the 2D drawings. Fig. 11 shows a generated 3D Eisenman-like model. The eight base-shapes (two pairs of A–T–G–C shapes) in the center are extruded to a user-defined height. The heights of the other shapes are determined automatically.

4. ArchiDNA software

The operations explained in the previous section are powerful enough to generate stylistic conceptual drawings. The important remaining question is how to use them. Based on the match-and-attach process described in Section 3.2, we designed a novel interactive system called ArchiDNA. ArchiDNA provides a

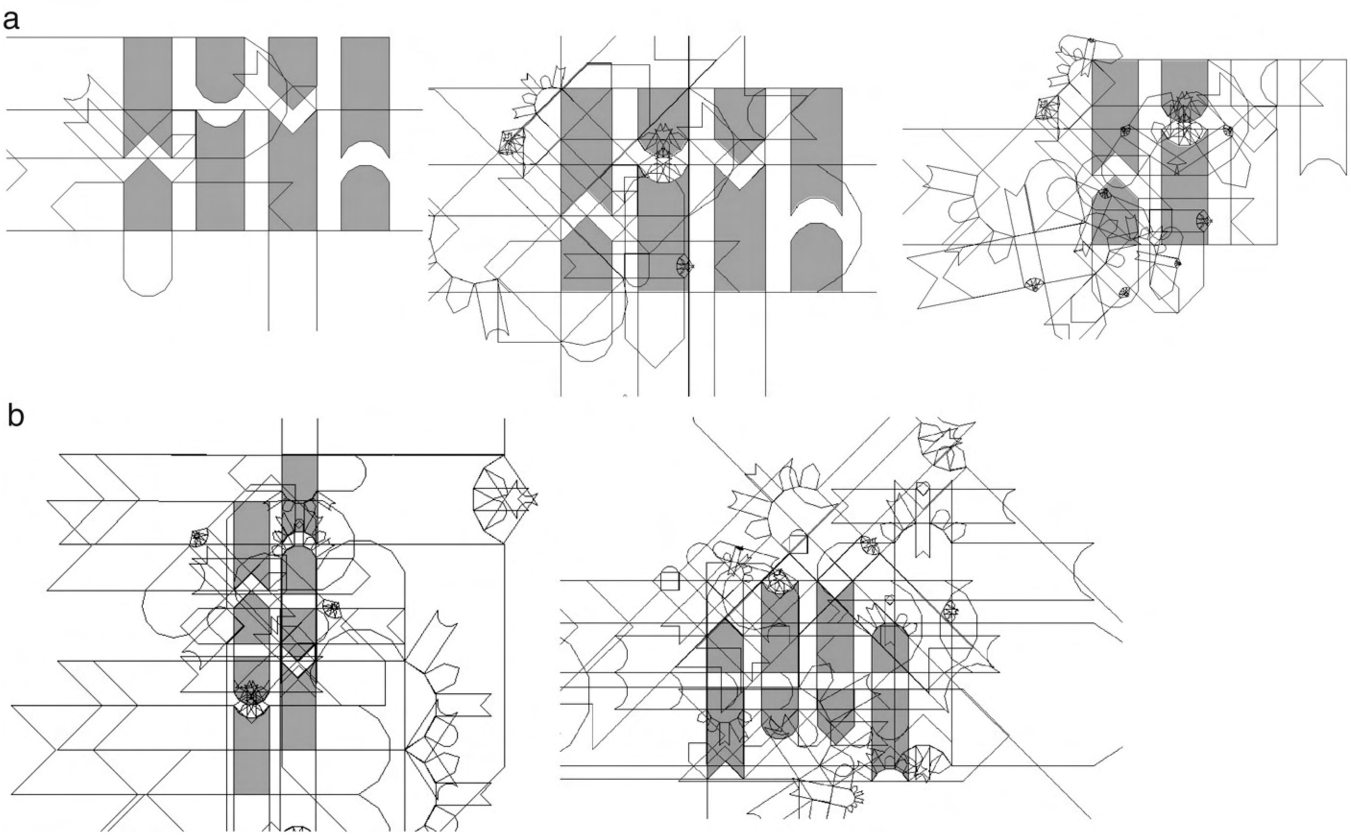

Fig. 9. (a) Interactive shape generation with match-and-attach following Eisenman's Biocentrum design. Each step from left to right shows similar design details as the sequence of Eisenman's conceptual drawings [Fig. 1]. (b) Design variations using different layouts of initial base-shapes.

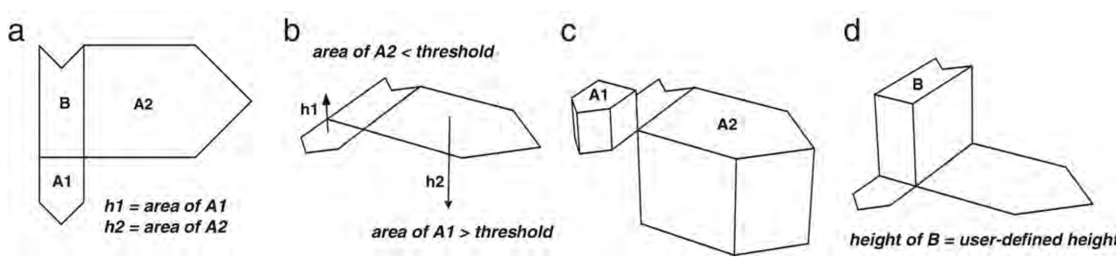

Fig. 10. 3D building form generation in ArchiDNA. (a) Calculating the area of two applier-shapes (A1&A2) and assigning heights. (b) Comparing areas with a threshold and deciding up and down. (c) Extrusion of small shape A1 upward and extrusion of large shape A2 downward. (d) Extrusion of shape B with a user-defined height.

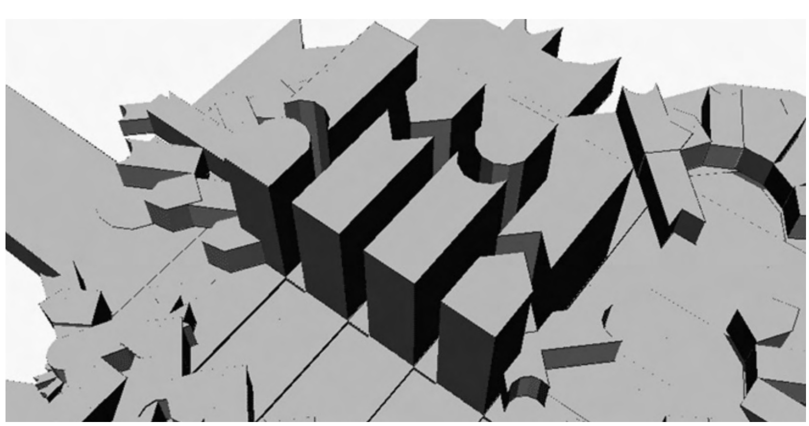

Fig. 11. 3D Eisenman-like drawing generated from the 2D drawing in Fig. 8. Compare with the original conceptual 3D physical model image in Fig. 3(b).

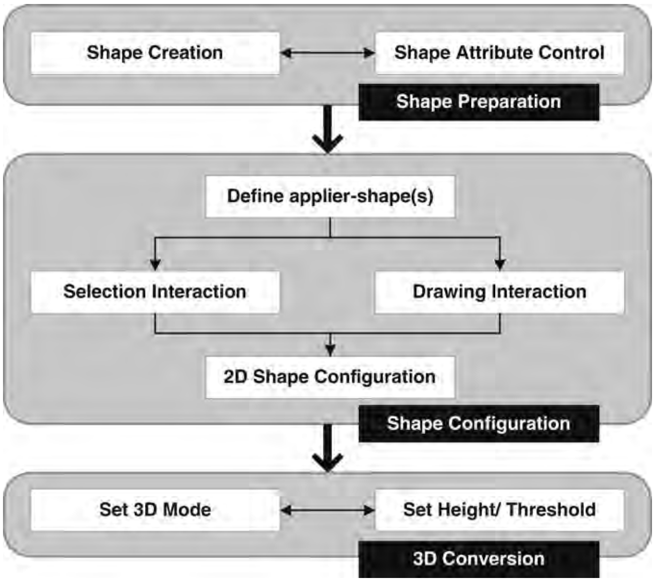

Fig. 12. Overview of the drawing procedure for ArchiDNA.

unified conceptual framework to generate 2D and 3D conceptual drawings. Our goal is to support an easy and powerful process to generate not only Eisenman-like drawings but also drawings in other styles.

The current ArchiDNA software is implemented as a web application using JDK (Java Development Kit) 1.4. The system runs on a variety of operating systems with a pre-installed JDK library, and can be embedded in a web browser. We use the Java 2D API to display shapes with different line styles and shape hatching patterns that are useful to visually distinguish particular shapes from others. Using the Java3D API we implemented a 3D viewer where the designer can navigate and edit the 3D drawings. The final 2D and 3D drawings can be saved in different file formats for sharing the design with others and for further development or analysis in other software.

In the remainder of this section, we will first give a system overview and drawing environment of ArchiDNA, and then explain how to use match-and-attach in detail.

4.1. System overview and ArchiDNA drawing environment

The ArchiDNA drawing procedure has three main phases (Fig. 12): In *shape preparation* (Section 4.2), the designer creates shapes and edits them as in other CAD programs. Then, the designer controls shape attributes and defines applier-shape(s). In *shape configuration* (Section 4.3), after defining a set of applier-shapes with an appropriate order, the designer generates configurations by either clicking on existing shape objects or drawing new base-shape(s). In the final *3D conversion* (Section 4.4), the designer generates different 3D drawings by changing the options of extrusion.

To support full control of the drawing procedure, the drawing environment of ArchiDNA enables designers to freely draw various shapes, adjust and control all operation parameters. Furthermore, ArchiDNA provides a unified interface so that the designer need not switch among multiple windows. The interface has three main panels [Fig. 13]: a drawing panel, a configuration panel, and a control panel. The control panel embeds four different control palettes alternatively on demand: a drawing control, a shape attribute control, an applier-shape list control, and a 3D control.

We wanted to provide an intuitive way to create shapes and use them as applier-shape and base-shape in different panels during the drawing process. We use a *drag and drop* technique. Using this technique, the designer can register shapes to the list by dragging a shape over the list palette located in the control panel. The designer can also use drag and drop to add initial base-shapes to the shape configuration panel.

4.2. Shape preparation phase

Here the designer makes shapes that will serve as applier- and base-shapes. If a designer already has previously prepared a shape, this phase can be skipped. This phase has two sub-processes: Create Shapes and Control Shapes.

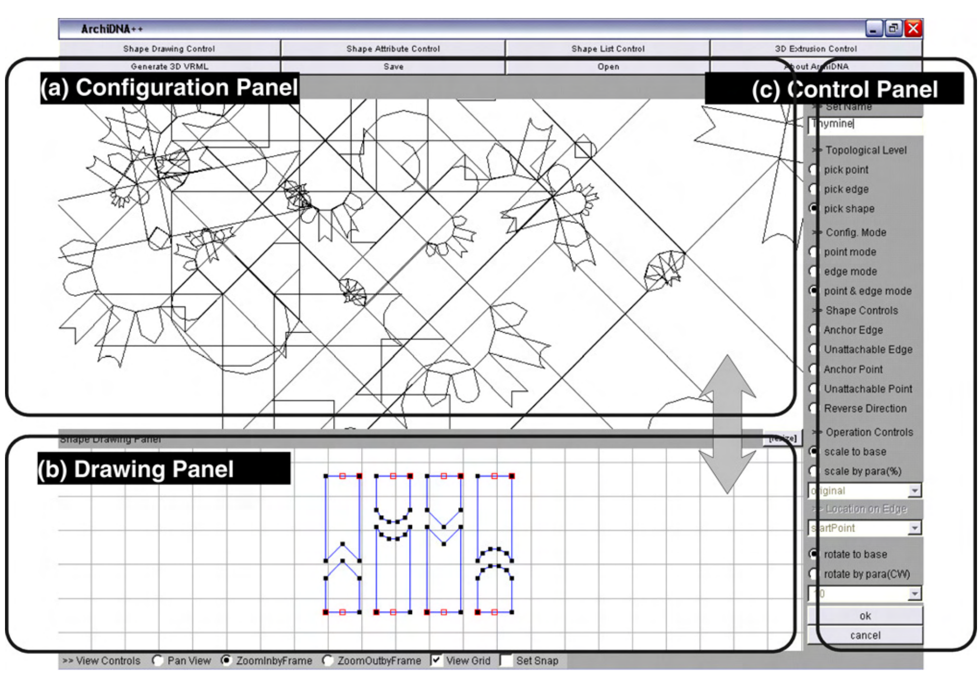

Fig. 13. Snapshot of ArchiDNA drawing environment.

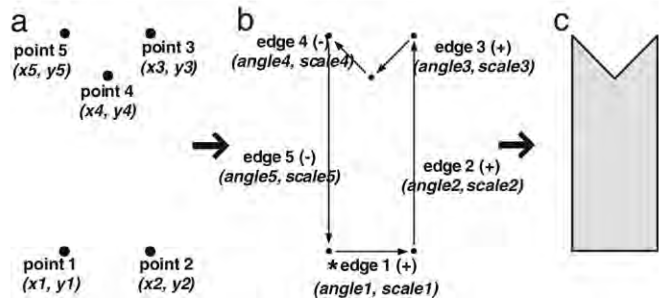

Fig. 14. Representation of a ribbon-like shape with points and edges: (a) Five points with 2D coordinates [x, y]. (b) Five edges with an anchor-edge mark [$*$], an attachable-edge mark [+], a un-attachable-edge mark [−], and operation values [angle, scale]. (c) The final display on the screen.

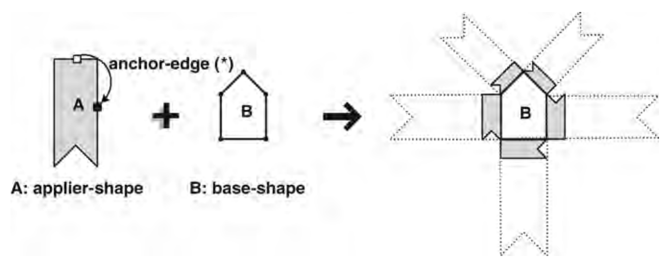

Fig. 15. Changing the anchor-edge of applier-shape (ribbon) and the result of applying it to a base-shape (pentagon); dotted lines illustrate the default configuration with the anchor-edge at the top of the ribbon.

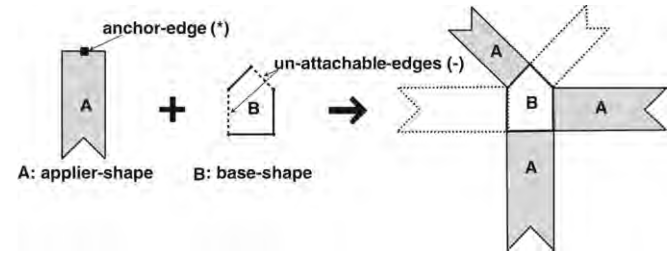

Fig. 16. Changing two attachable-edges of a base-shape (pentagon) to un-attachable edges and its result with an applier-shape (ribbon).

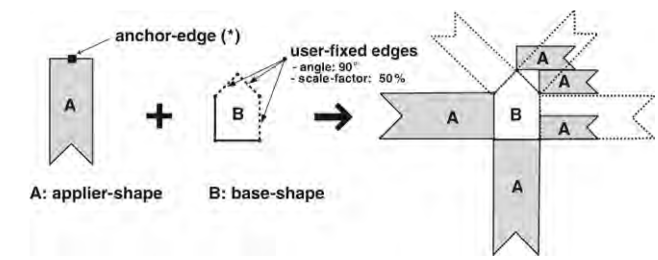

Fig. 17. Fixing three edges with angle (90°) and scale-factor (50%); the result shows three applier-shape objects not matched to the base-shape object.

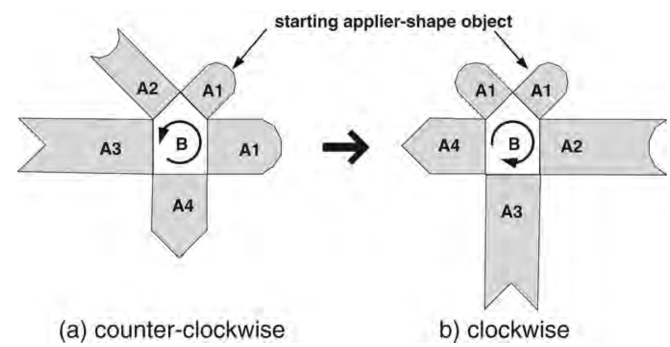

Fig. 18. (a) Counter-clockwise base-shape. (b) Clockwise base-shape with four applier-shapes (A1–A2–A3–A4).

4.2.1. Shape creation

ArchiDNA is a vector-based system that uses a hierarchical and flexible object-oriented data structure to represent various 2D shapes. A shape object consists of one or multiple straight edge objects, each referencing a start point and an end point.

Fig. 14 shows the representation of a ribbon-like shape in two-dimensional space. An edge is determined by two distinct points and stores attributes, including attachability, numerical attributes (angle and scale), and direction, appearance attributes such as line width and color. One edge of a shape is marked with a star (*) as an anchor-edge; the first edge is set as a default anchor-edge. All edges are marked with a plus (+) for attachable-edge or a minus (−) for un-attachable-edge. The display of shape attributes can be turned on/off depending on user preference.

Like other CAD software, ArchiDNA provides drawing primitives such as line, triangle, rectangle, poly-line, segmented curve line and sketch line. It also supports simple editing (select, move, rotate, and delete) to modify the created shapes. Designers can make their own shapes and save them as a shape vocabulary which they can reuse for other designs.

4.2.2. Shape attribute control

In this section, we explain how users can control match-and-attach by changing shape attributes. There are four attributes: one for applier-shape (anchor edge) and three for base-shape (attachability, rotation/scaling, and direction).

Anchor edge control: A designer can control an anchor edge that defines which part of an applier-shape meets a base-shape. For instance, Fig. 15 shows a change of the anchor-edge from the top edge (default) to the right edge in the applier-shape. The system generates a different configuration matching and attaching the right edge to all edges of a base-shape.

Attachability control: By default the applier-shape can attach to all edges of a base-shape. The designer can change attachable-edges to un-attachable edges. Fig. 16 shows how when two edges of the base-shape object are labeled un-attachable, those two edges remain free of any instances of the applier-shape.

Rotation and scaling control: The applier-shape object matches and attaches to an edge of a base-shape object [Fig. 5]. ArchiDNA enables a designer to control the rotation and scaling of an applier-shape. The designer sets the operation values (angle and scale-factor) to an edge of a base-shape object. Fig. 17 shows a different result of fixing the base-shape's three edges with a certain angle (90°) and a scale factor (50%). The three applier-shape objects were translated to each edge of the base-shape object and rotated and scaled with user-defined values (not matching the base-shape object).

Direction control: The order of a base-shape's edges is that in which its points were originally created. The order determines the direction in which applier-shapes attach. ArchiDNA enables a designer to change the direction of a base-shape.

Fig. 18 shows examples of changing the direction of a base-shape between counter-clockwise (default) and different configuration outputs. In Fig. 18(a) four applier-shape objects (A1–A2–A3–A4) attach to the default base-shape object in sequence counter-clockwise [also see Fig. 6]. If the designer changes the direction to clockwise, the four applier-shape objects attach clockwise in sequence, generating a different shape configuration [Fig. 18(b)].

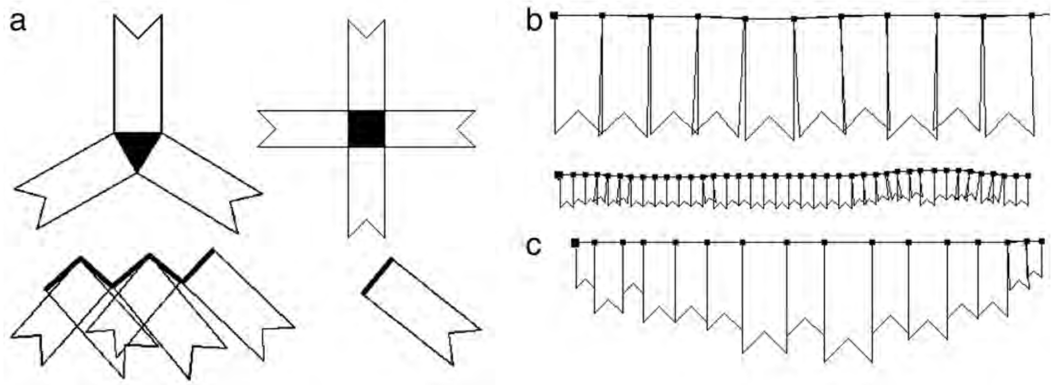

Fig. 19. Generated shape configuration using drawing interaction with a ribbon-like shape as an applier-shape. (a) Triangle, rectangle, poly-line, and single line. (b) Sketch-line using distance triggering at two different distance thresholds: 40 for the top and 10 for the bottom. (c) Sketch-line using time triggering with different speed. Note that the drawing speed was highest at the middle of the line drawing.

4.3. Shape configuration phase

Once shapes are prepared in the drawing panel using a dedicated drawing tool, the designer can generate shape configurations. The designer chooses between two interaction schemes to apply applier-shapes to base-shapes and generate shape configurations: *selection* for selecting existing shapes as a base-shape, and *drawing* for drawing new base-shapes.

4.3.1. Selection

As described in Section 3.2.2, a designer generates a shape configuration by selecting a base-shape. First, a designer must prepare (e.g. draw or select) applier-shape(s), and create an applier-shape list that contains the shape objects and the sequence to be attached. When the designer selects a base-shape, the instances of applier-shapes are instantiated with their attributes, and positioned at all attachable edges of the selected base-shape. ArchiDNA also enables the designer to select individual edges of a base-shape instead of using all edges. Please note that this technique can be an alternative to the attachability control.

4.3.2. Drawing

This interaction scheme integrates the match-and-attach process into the user's drawing actions. ArchiDNA automatically augments the shapes designers draw with the shape operations. A designer must first select a drawing tool. Then the designer draws various shapes as a base-shape. ArchiDNA immediately applies the previously determined applier-shapes to the edges of the base-shape, in the usual match-and-attach fashion. Fig. 19(a) shows examples generated when triangle, rectangle, poly-line, line, and sketch-line are drawn with a ribbon-like shape as an applier-shape.

When the sketch-line is drawn, the system continuously triggers application events while a designer moves the cursor over the configuration panel either using a mouse or a digital pen. We explore the use of drawing speed to control the match-and-attach process. Drawing speed is an important characteristic that controls the appearance of the line drawing along with pressure and tilt/rotation angles.

We implemented two different algorithms for event triggering: a *distance triggering* and a *time triggering*. In distance triggering, the system only creates a line when the distance between the previous point and the current point exceeds a user-defined threshold. The result consists of approximately the same length lines depending on a distance threshold value as shown in Fig. 19(b). Time triggering generates points at the current cursor position at regular intervals of time. Fig. 19(c) shows the result shape which has lines with different lengths controlled by the speed of drawing. In this mode, when the cursor stays at the same position, many points can be generated. To prevent this we check the distance of the current point position from the previous point and allow the generation of a point only when the distance exceeds a certain threshold.

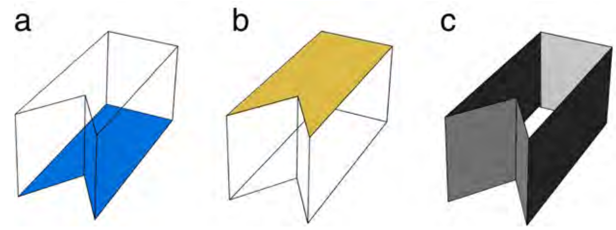

Fig. 20. 3D conversion with three types of objects: (a) floor, (b) roof, and (c) wall.

4.4. 3D conversion

In the last phase of the drawing procedure, ArchiDNA enables a designer to create different 3D drawings that can be useful for building mass design. As described in Section 3.2.3, 2D shapes are extruded by the assigned building height by computing the area. In principle, a 3D mass model is generated by grouping three types of objects namely a *floor*, a *roof*, and a *wall* based on the architectural design scenario. The floor object [Fig. 20(a)] is the same as the original 2D shape and the roof object [Fig. 20(b)] is generated simply by adding a Z-coordinate to the 2D coordinates of the 2D shape. Rectangular wall objects [Fig. 20(c)] are generated reconstructing the edges of the ground and the roof object as a bottom and a top respectively. When converting a 2D open shape to 3D, only wall objects are generated without a ground and roof object.

4.4.1. Set a height and a direction threshold

In Section 3.2.3, we explained that the building height is determined either automatically or manually. In automatic mode, ArchiDNA assigns the height based on the area value for a closed-shape or the boundary length for an open-shape. In manual mode, a designer can fix a height of a shape. 3D conversion in ArchiDNA operates on a threshold to determine whether the shape is extruded upward or downward. A designer can specify a threshold value by selecting a standard shape of which area (closed shape) or length (open-shape) to be used as a threshold.

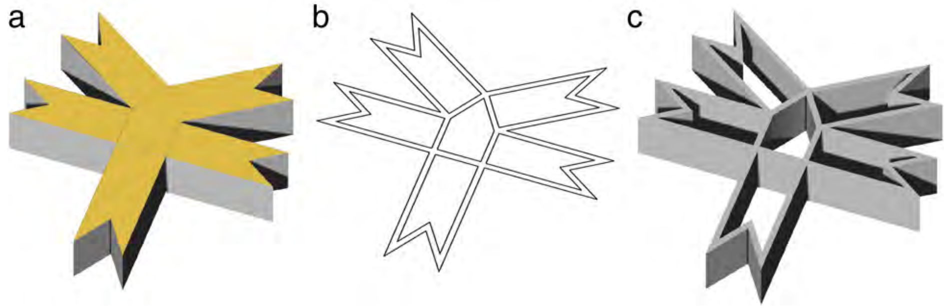

Fig. 21. (a) 3D model in the block mode. (b) 2D enclosure and its final 3D enclosure model in the wall mode.

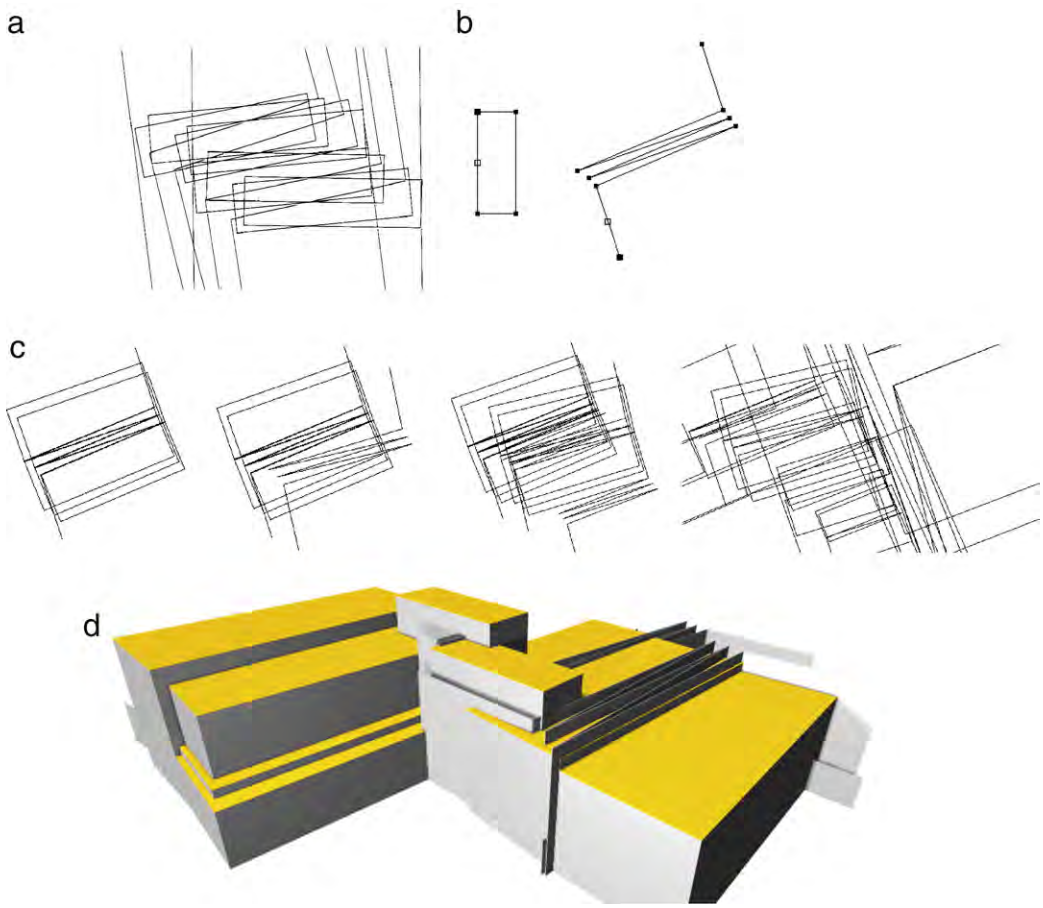

Fig. 22. (a) Peter Eisenman's conceptual drawing for Groningen Video Pavilion Project (Eisenman, 1990). Source: [8], (b) Shape vocabulary (rectangle and zigzag) for the generation of Eisenman-like drawings. (c) Interactive shape generation in ArchiDNA by matching and attaching a rectangle to a zigzag, and vice versa. (d) One possible interesting example of 3D conversion for architectural design. Note that the building heights differ slightly because they have different areas.

4.4.2. Set 3D mode: Block mode and wall mode

ArchiDNA supports two different 3D modeling schemes: *block mode* [Fig. 21(a)] for exterior design and *wall mode* for interior design [Fig. 21(b)]. In block mode, ArchiDNA generates a 3D mass model as in Section 3.2.3. In wall-mode, ArchiDNA creates 3D enclosure models with a preset wall thickness. The basic generation principle is to generate a 2D enclosure [Fig. 21(c)] from the 2D source shape using the user-defined enclosure thickness first, and extrude to generate a 3D enclosure model. The 3D enclosure model is also useful when creating an architectural-looking 3D model that contains wall objects with 2D open shapes.

5. Broader application

We designed the match-and-attach process (and the ArchiDNA software) by analyzing Eisenman's Biocentrum drawings, and we have shown, naturally, that a designer can use ArchiDNA to produce Biocentrum-like drawings. We now ask: Does this match-and-attach process have broader use? To answer this question, we used ArchiDNA to generate several other classes of design drawings. The first example is based on another building by Eisenman; the second, a building by architect Louis Kahn; the third, a design of our own; and the last is for a drawing by M.C. Escher. Although these demonstrations are anecdotal, they show that the ArchiDNA system can be used for the analysis and generation of conceptual drawings in architectural design, beyond the specific case of Eisenman's Biocentrum building.

As the first example, we applied ArchiDNA to generate other Eisenman designs. Fig. 22 illustrates the original conceptual drawings of Eisenman for the Groningen Video Pavilion project. Based on analysis of the drawings, we used two shapes (a rectangle

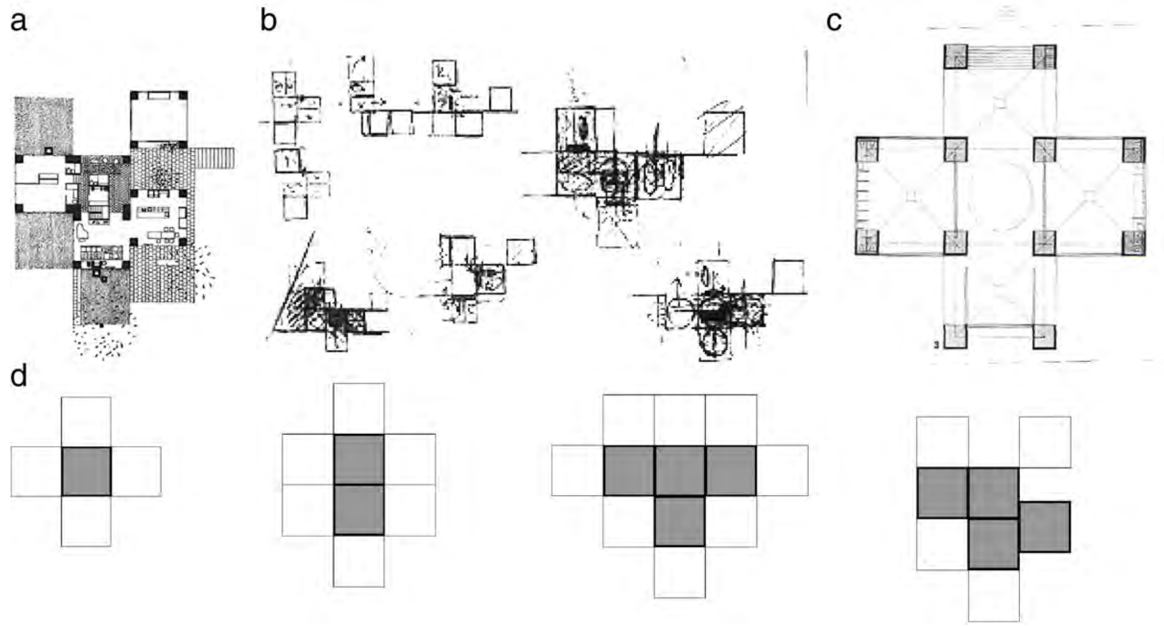

Fig. 23. Louis Kahn's original drawings: (a) Final plan drawing of Adler House (1955). Source: Drawing no. 415.9. Copyright 1977, The Louis I. Kahn Collection. (b) Original conceptual drawings for the Adler House (1955). Source: Drawing no. 415.1, copyright 1977, Kahn Collection. (c) Plan for Bath House, Trenton Jewish Community Center, Trenton, NU, 1954-59, Louis I Kahn, Source: [3]. (d) Shape generation in ArchiDNA for the Adler house using a rectangle as a base-shape and applier-shape. Note that the first result is similar to the plan for the Bath house.

and a zigzag) and followed a set of procedures to generate a 2D drawing in a similar style as depicted in Fig. 22(b). Fig. 22(d) shows one of the interesting architectural forms by converting the resulting 2D drawing in ArchiDNA.

We also demonstrated how our approach can be used to generate drawings in the style of another architect, Louis Kahn. Kahn's work is often considered an example of a bottom-up approach. He places the first component, and then adds others successively by selecting components from a limited vocabulary [3]. A series of his conceptual drawings for the Adler house [Fig. 23(a, b)] shows how multiple rectangles are manipulated during the study of the spatial relationships. Similar to the Adler house design, the Trenton bath house [Fig. 23(c)] shows an assembly of four rectangles about a central rectangle. Fig. 23(d) shows a series of configurations generated in ArchiDNA with repeated match-and-attach processes and a few manipulations. The first application generated a drawing similar to the Trenton bath house plan.

Next we show how designers can use ArchiDNA with their own design concept. The process starts with a simple freehand sketch. Fig. 24(a) shows the two sketches for abstracting a sunflower and codifying the generation of a sunflower in ArchiDNA. Using three shapes (triangle, hexagon, and ribbon) [Fig. 24(b)], we could generate 2D drawings including two intermediate drawings the same as the initial sketches [Fig. 24(c)], and interesting architectural looking 3D drawings [Fig. 24(d)].

Fig. 25 depicts examples inspired by a drawing of M.C. Escher [Fig. 25(a)]. We explored the use of drawing interactions described in Section 4.3.2. To generate drawings similar to Escher's [Fig. 25(c)], we drew four single lines in two opposite directions after selecting the applier-shape (human). Fig. 25(d) and (e) present two variations that use a rectangle and a sketch line as a base-shape respectively. In Fig. 25(e), the anchor-edge of human was changed from a back head to a foot, and the sizes of the matched and attached human were controlled by varying sketch speed.

6. Conclusion and discussion

In this paper, we presented an interactive graphical system, ArchiDNA to support architectural designers in creating 2D and 3D conceptual drawings. The key ingredients of our editor are simple but powerful shape replacement rules with dynamic user interaction schemes. Although match-and-attach is limited to affine transformations, the interactive technique is useful in exploring a formal design idea with defined styles during the conceptual design process.

We demonstrated the creation of various 2D and 3D conceptual drawings that have different styles as well as a similar style to a particular drawing. Various interesting designs can be created quickly in a limited time that might be difficult for designers using existing CAD software. ArchiDNA serves as a creative design tool that encourages the use of visual thinking and drawing techniques. We believe our work is a powerful adoption of shape replacement rules for CAD software and computational drawing pipelines.

We conclude this paper with a discussion about potentials and open problems that are of interest for future research as follows:

Intuitiveness and usability: The learning curve to use ArchiDNA is similar to that of a simple CAD application. We make a conscious effort to simplify the drawing process, as demonstrated above. Therefore even inexperienced designers will be able to quickly generate satisfactory results by using default shape operations. We expect that drawing augmented by match-and-attach will be most naturally understood by people who understand basic CAD functionalities, but in any case most designers are familiar with digital drawing and will be able to use the system.

These days, design schools teach basic operations of CAD software as well as conventional drawing skills. As ArchiDNA combines both drawing techniques, one may use it as a generative design tool for inspiration in education and practice. ArchiDNA can function as a learning tool for demonstrating the range and power of computational shape generation and for illustrating the style and process of design.

Furthermore, match-and-attach can be incorporated in a commercial CAD system, for example, implemented in the CAD

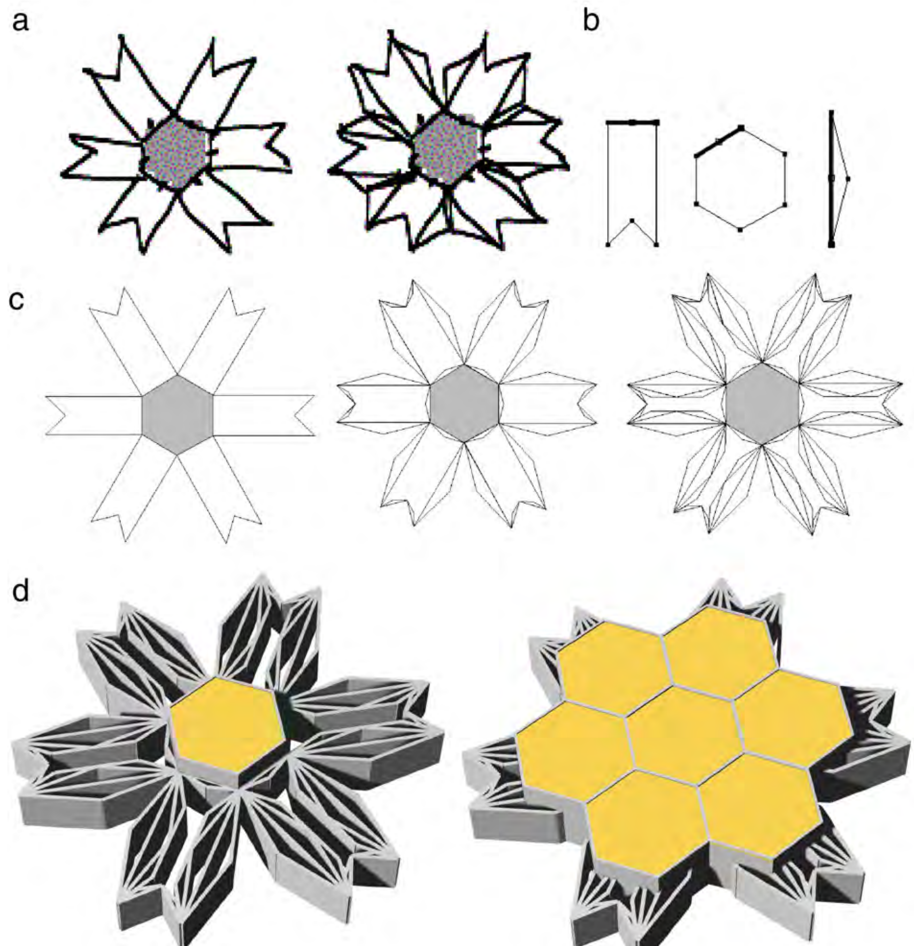

Fig. 24. (a) Conceptual freehand sketch: the first sketch drawing for generating a sunflower shape, and the second sketch drawing for elaborating the first sunflower shape. (b) Prepared shape vocabulary. (c) A series of shape configurations: the first configuration applying the ribbon to the hexagon; the second configuration applying the triangle to the ribbons; and final configuration applying the triangle to existing shapes. (d) 3D conversion of the 2D drawings. The central part is modeled as a main building mass in the block mode and the surrounding part in the wall-mode.

system's scripting language. This would make the CAD system a more interactive design tool wherein a designer generates conceptual drawings and investigates a formal design idea with defined styles and then further develops the created drawings using advanced functionalities of existing CAD software.

Shape grammar rules: Shape grammar has been demonstrated to study complex historical architecture [34] and codify a well-defined generation problem with a long sequence of rules [22].

ArchiDNA can be extended to a shape grammar system within our semi-automatic shape generation framework. ArchiDNA's match-and-attach process is a subset of the classical replacement rules of shape grammar. When the program sees a single edge of a base-shape, it can replace it with an applier-shape. Labeling in shape grammar systems can be used to indicate a matching line of the applier-shape. If there is more than one applier-shape, replacement rules can be defined for each corresponding applier-shape.

It is promising to extend the match-and-attach process with shape grammar rules for more sophisticated shape generation methods. However, the use of large number of shape rules in the match-and-attach process should be carefully motivated and planned such that designers interactively experiment and play conceptual drawing creation. In this work, we take a simple and modest approach to encourage designers to use intuitive drawings skills with the generative power of a computer.

Semantics and evaluation: In this paper, we have been concerned only with the generated structure. The promise of being able to provide more meaningful shape generation requires a way of evaluating how well a particular result meets specified design criteria.

For this purpose, we built ArchiDNA to be flexible for efficient incorporation of architectural knowledge such as space, wall and column. Thus the system can be efficiently extended to provide semantic representations for architectural plans and models. This extension would allow us to build a structurally well-defined floor plan that has information about spaces within the floor plan and their relational information [6], and also to evaluate various building performance using individual evaluation modules.

Our ultimate goal for this future work is to manage a conceptual drawing, a floor plan, and a 3D model in a single editing framework. For instance, when a change is initiated in a conceptual drawing, another corresponding operation is transparently activated to adjust the floor plan and the 3D model.

Acknowledgements

This research was supported in part by the US National Science Foundation under Grant CCLI DUE-0127579. The views and findings contained in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. An early version of this work was presented as a poster in the CAAD Futures conference at National Cheng Kung University, based on Kwon's Master Thesis at the University of Washington, Seattle.

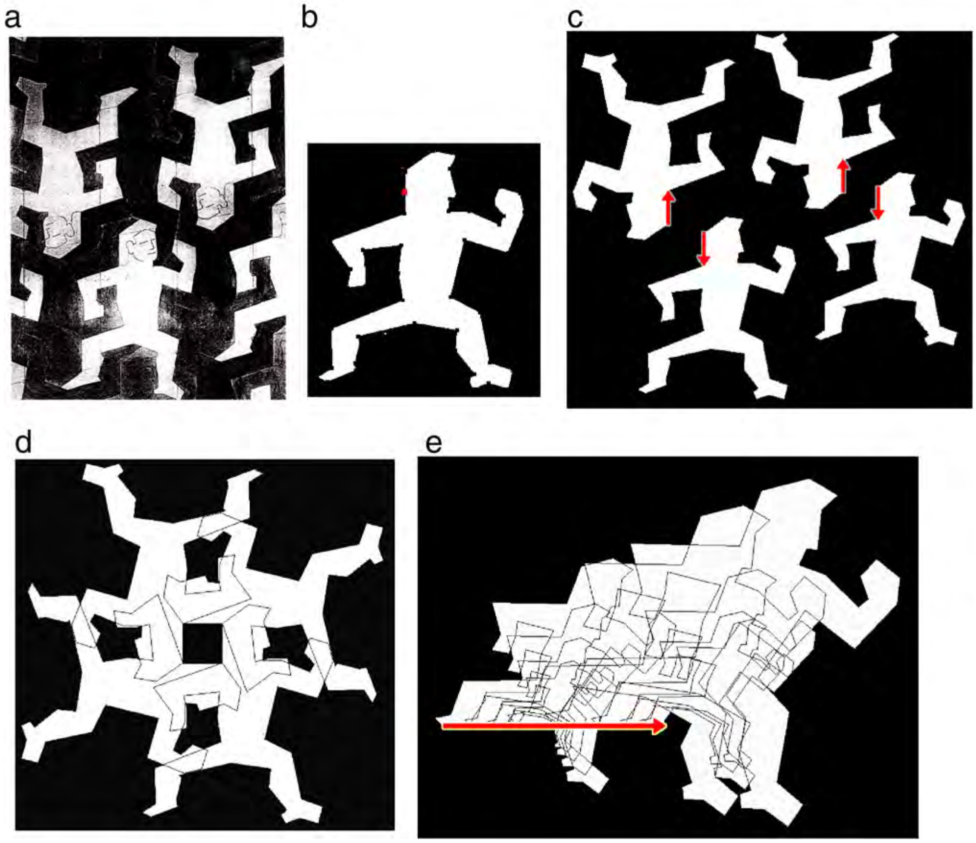

Fig. 25. (a) M.C. Escher's original drawing for Human. (b) Prepared shape vocabulary (human). (c) Generation of Escher-like drawing using drawing interaction with a single line. Two different line drawings (from down to up and from up to down) change the direction of the human similar to the original drawing. (d) The first variation uses a rectangle for drawing interaction. (e) The second variation uses a sketch-line with a time-triggering mode. The drawing speed is changed from slow to fast and the anchor-edge is changed from the back-head edge to the left-foot edge.

References

[1] Abelson H, diSessa A. Turtle geometry. MIT Press; 1980.
[2] Arnheim R. Art and visual perception: A psychology of the creative eye. Berkeley (CA): University of California Press; 1974.
[3] Brawne M. Kimbell Art Museum: Louis I. Kahn. London: Phaidon Press; 1992.
[4] Casakin H, Goldschmidt G. Expertise and the use of visual analogy: implications for design education. Design Studies 1999;20(2):153–75.
[5] Cha MY, Gero JS. Style Learning: Inductive generalization of architectural shape patterns. In: Proceedings of eCAADe conference. 1999 p. 629–44.
[6] Choi JW, Kwon DW, Hwang JE, Lertlakkhanakul J. Real-time management of spatial information of design: A space-based floor plan representation of buildings. Automation in Construction 2006;16(4):449–59.
[7] Duarte JP. Customizing mass housing: A discursive grammar for Siza's malagueira houses. Ph.D. thesis. Cambridge (MA): Massachusetts Institute of Technology; 2001.
[8] Eisenman P. Peter Eisenman diagram diaries. New York: Universe Publishing; 1999. 27–43.
[9] Eisenman P. House of cards. Oxford: Oxford University Press; 1987.
[10] Flemming U. More than the sum of parts: The grammar of Queen Anne houses. Environment and Planning 1987;B 14:323–50.
[11] Flemming U. Syntactic structures in architecture: Teaching composition with computer assistance. In: The electronic design studio. Cambridge (MA): MIT Press; 1990. p. 31–47.
[12] Fraser I, Henmi R. Envisioning architecture: An analysis of drawing. New York: Van Nostrand Reinhold; 1993.
[13] Gombrich EH. Art and illusion: A study in the psychology of pictorial representation. London: Pantheon Books; 1960.
[14] Gross MD. FormWriter: A little programming language for generating three-dimensional form algorithmically. In: Proceedings of computer aided architectural design futures 2001. Kluwer Academic Publishers; 2001. p. 577–88.
[15] Heisserman J. Generative geometric design. IEEE Computer Graphics and Applications 1994;14(2):37–45.
[16] Herbert DM. Architectural study drawings. New York: Van Nostrand Reinhold; 1993.
[17] Hersey G, Freedman R. Possible palladian villas. Cambridge (MA): MIT Press; 1992.
[18] Kirsch R, Kirsch J. The structure of paintings: Formal grammar and design. Environment and Planning B: Planning and Design 1986;13(2):163–76.
[19] Kirsch R, Kirsch J. The anatomy of painting style: Description with computer rules. Leonardo 1988;21(4):437–44.
[20] Kirsch R. Using computers to describe style. In: American Indian rock art, vol. 22. American Rock Art Research Association; 1998. p. 153–60.
[21] Knight TW. Shape grammars in education and practice: History and prospects, http://www.mit.edu/~tknight/IJDC/ (online paper), Department of Architecture, MIT; 2000.
[22] Knight TW. Computing with emergence. Environment and Planning B: Planning and Design 2003;30:125–55.
[23] Kolarevic B. Digital architectures. In: Proceedings of the ACADIA 2000 conference. 2000.
[24] Koning H, Eizenberg J. The language of the prairie: Frank Lloyd Wright's Prairie Houses. Environment and Planning B 1981;8:295–323.
[25] Maeda J. Design By numbers. Cambridge (MA): The MIT Press; 1999.
[26] MIT Logo project. http://el.media.mit.edu/logo-foundation/.
[27] Mitchell WJ. The logic of architecture. Cambridge (MA): The MIT Press; 1994.
[28] Moustapha H. Architectural explorations: A formal representation for generation and transformation of design geometry, Ph.D. dissertation. School of Architecture, Carnegie Mellon University; 2005.
[29] Processing. http://proce55ing.net/.
[30] Schapiro M. Theory and philosophy of art: Style, artist, and society. New York: George Braziller, Inc; 1994.
[31] Stiny G, Gips J. Shape grammars and the generative specification of painting and sculpture. In: Freiman CV, editor. Information processing, vol. 71. Amsterdam: North Holland; 1972. p. 1460–5.
[32] Stiny G. Ice-ray: A note on Chinese lattice designs. Environment and Planning B 1977;4:89–98.
[33] Stiny G. Computing with form and meaning in architecture. Journal of Architectural Education 1985;39(1):7–19.
[34] Stiny G, Mitchell JW. The Palladian grammar. Environment and Planning B 1978;5:5–18.
[35] Suwa M, Gero J, Purcell T. Unexpected discoveries and S-invention of design requirements: Important vehicles for a design process. Design Studies 2000;21(6):539–67.
[36] Tapia MA. GEdit: A visual implementation of a shape grammar system. Environment and Planning B: Planning and Design 1999;26:59–73.
[37] Wang Y. 3D shaper: 3D architecture form synthesizer, SMArchS paper, Cambridge (MA): Department of Architecture, Massachusetts Institute of Technology; 1999.