ABSTRACT

Beginning from a provocation in Auguste Blanqui’s *Eternity by the Stars*, this paper reports on a new methodology of digital collage for urban design. The research is situated relative to the current discourses surrounding both voxelization and point-cloud data structures in order to motivate the concept of a recombinant approach to design in existing cities. Building on these sources, and with reference to recent developments in mesh shape composition techniques, the paper presents the resulting software implementation “Point-Cloud-Paint”: a tool that enables collage-based combinatorial experimentation with urban point-cloud data.
INTRODUCTION

Combinatorial Urbanism

In his *Eternity by the Stars*, the 19th century revolutionary philosopher Auguste Blanqui develops a compelling theory of the universe. Blanqui arrives, in the text, at the following logic: assuming that there are a finite number of possible chemical elements (i.e. the periodic table), and a spatially infinite universe, it follows that at any given moment, all possible configurations must coexist somewhere in the universe. It is worth noting the striking resemblance between this description and contemporary multiverse theories, more precisely, what Max Tegmark calls a “Level 1 Multiverse” (Tegmark 2003). Since this work was composed while he was held captive on the French prison island of Taureau, it seems likely that Blanqui was particularly interested in the many universes in which he himself was not a prisoner. For designers, the most compelling aspect of the theory is introduced during a curiously practical digression in the text, where Blanqui laments the paucity of experimentation and diversity in Baron Haussmann’s contemporaneous plans for the city of Paris (Blanqui 2013). Transposing his multiverse theory to the urban scale, he asks: what if we could rethink the design of urban spaces in terms of all possible configurations of built material? In opposition to an urbanism of consistent facades planned to line the new boulevards, he suggests an urbanism of combinatorial architectural variety. This remark by Blanqui suggests an interesting stance for contemporary urban design: willing to speculate at a visionary scale, yet grounded in the built realities of the existing city, and enabling a high level of heterogeneity of urban form. The goal of the research described below is to establish a design process and software tool that support such an approach to urban design.

Collage

It could be argued that precisely such a methodology already exists in processes of digital image collage. Collage has a long tradition as a popular technique of representation in urban-scale design propositions. In part, its usefulness derives from the ease with which urban form, use, and atmosphere can be synthetically described by drawing on existing imagery. Especially at the master-planning scale, when civic and architectural details remain largely unresolved, collage acts as a gesture towards a desired outcome by combining and blending elements of different urban scenes. At the same time, the static character and single vantage point of these images limits the opportunity to evaluate the proposal in terms of analytical measurements; limits their translation into other kinds of representation, such as physical models; and even limits testing the proposal in terms of additional viewpoints. This paper describes a three-dimensional extension of the pixel manipulation techniques of image collage that seeks to

---

2 Louis Auguste Blanqui, French socialist, philosopher, political revolutionary (1805–1881).
3 Georges-Eugène Haussmann, Prefect of the Seine Department of France (1809–1891).
preserve the quick, loose, and intuitive process that designers are familiar with from tools such as Adobe Photoshop.

Voxels and Point-Clouds

While voxels are the most direct three-dimensional equivalent of pixels, they do not in fact play a role in the software implementation presented below. Instead, the software is built around another prevalent type of three-dimensional point-based data: point-clouds. It is however still worth reflecting on the data structures and the discourses associated with both in order to understand the broader context of this research. Voxels are defined as evenly distributed in a three-dimensional grid, meaning that they are describable using a list of Boolean values (indicating the presence or absence of a solid voxel cell at a given grid point) or a list of the active point positions. As an accumulation of cells, voxel-grids can describe solid geometries, albeit discretized at the dimension of the individual cell. On the other hand, point-clouds can be less regularly ordered and hence individual coordinates of each point must be stored in the cloud’s data structure. Point-clouds are often employed to delineate a surface or set of surfaces (for example through a mesh triangulation of the points). Finally, algorithms exist for converting a point-cloud surface into a voxel-grid (through a series of containment tests on a meshed point-cloud), and vice versa, through for example the marching cubes algorithm (Lorensen 1987). In spite of their basic similarity, that both are fundamentally describable as sets of points in Cartesian space, the discourses surrounding ‘voxelization’ and ‘point-cloud’ approaches to design and representation have tended to be isolated from each other.

In his recent writings on voxelization, Mario Carpo celebrates the vast potentials it offers in terms of configurability: all manner of geometry are describable as different subsets of a voxel grid (Carpo 2016). He suggests that voxelization—as both data structure and design methodology—is a paradigmatic break from the smooth calculus of earlier experiments in digital form making. No longer do forms need to be expressible in terms of mathematical formula, instead, the full range of spatial possibilities can be explored through a binary three-dimensional grid, where each grid-cube is either solid or void (Carpo 2016). On the other hand, the discourse surrounding point-clouds has often focused on the role they can play as an outcome from LiDAR scanning of natural geographies and urban sites. These discussions tend to emphasize the technology’s capacity for a heretofore-unachievable fidelity of representation. Christophe Girot, for example, argues that “a point-cloud model offers more holistic truth than a layered plan” (Girot 2013). While voxels are celebrated as opening an endless set of design possibilities, point-clouds are championed for exactly the opposite capacity: delivering the highest-precision synchronic description of an existing condition.

What would it mean to try and connect these two methods and their respective discourses? In what sense could the combinatorial speculations that Carpo sees as enabled by voxel approaches be applied to the high-fidelity description of the world contained in point-clouds? What new design agency could arise from a capacity for quick voxel-like experiments on urban sites? In the remainder of this paper, these questions will be addressed through the description of a new urban design software tool (“Point-Cloud-Paint” or PCP). This tool supports the sampling
and collaging of point-cloud data and the free reconfiguration of elements from the existing city. After presenting the software, a typical workflow case study is described, and the paper concludes with consideration of further opportunities to extend this research.

**METHODS**

**The Question of Elements**

In Blanqui’s analogy between the universe and Paris, what remains unclear is the question of elements: what is the urban equivalent to the periodic table? While considering how best to answer this question, any number of conceptual frameworks recommend themselves as models for urban elements (for example, different approaches to typology, urban systems, etc.). Since the scale and specificity of the elements determines the range of potential variations to be explored, the contention in this research is that an appropriate answer must strike a balance between two extremes: on the one hand, an approach that is too fine grained, such as the set of chemical elements, and on the other hand, one that is too coarse grained, such as some concepts of building typology. While the former would yield an infinite set of combinations, the clear majority of which would be spatially meaningless, the latter is likely to preclude all kinds of interesting experimentation. The approach we have pursued is to allow the designer to define their own elements by cropping or cloning them when required from existing urban point-clouds. These elements can be drawn either from the city for which the design is intended, or from entirely different cities. In this way, the designer is able to avoid the sheer expanse of combinatorial possibilities, but at the same time, the scale and variety of elements are not prescribed in advance. It is hoped that this openness of the tool can facilitate a range of different applications, from the design of hybridized buildings using multiple architectural sources up to large-scale master-planning exercises.

**Point-Cloud-Paint**

PCP is a software tool programmed by the author in the Java programming language. For the sake of visualization, it relies on features of the Processing library (Reas and Fry 2014). PCP is made up of a user interface that provides collaging controls, a database composed of ortho-corrected imagery and point-cloud data (spatially coordinated to one another), and a multi-threaded computational process responding to new brush motions by the user within the workspace.

As shown in Figure 5, the PCP interface is split into two halves: on the left is the source portion of the process, where the user can select content from an existing city that they want to sample. For the sake of speed and simplicity, this is presented using ortho-corrected aerial imagery that is spatially-coordinated to the point-cloud data. On the right side – the target area – a three-dimensional axonometric view is provided of the site the user is working on. This can be set initially to an empty space or to duplicate some existing site in the city. Working between the left and right, the user can, for example, define a source or stamp point within the aerial image, and begin painting out a portion of the corresponding point-cloud. As the brush moves in the workspace, this movement is used to calculate a translation relative to the brush’s initial position and to identify the equivalent movement from the stamp in the source material. Points
in the source material are then copied over to the workspace, completing the process. The dimensions of the site available to the user are 500 x 500 m by default. This assumption provides sufficient working space for many kinds of experiment while also simplifying the navigational requirements; at this size, the whole site is reasonably perceptible at once with only basic rotation and zoom controls.

In other aspects of interface and controls, particularly the brushes, the features of PCP are three-dimensional extensions of familiar tools of Adobe Photoshop. The set of brushes presently implemented can be described as ‘three-by-two’: two brush shapes, two brush modes, and two brush types. The brush shapes include, first, an extruded circular brush that affects points at any height within a horizontal radius of the mouse’s position in the local coordinate system (see Figure 6), and second, a “spherical” brush that affects all points within radius-distance from the brush center. With this brush, the vertical position of the center point is controlled through keys, while the mouse controls its position on the horizontal plane (see Figure 7). While the computer graphics literature referred to above has typically implemented 3D brushes as a cylinder oriented to the target mesh’s face normals, this is not a particularly meaningful orientation for urban point-clouds. Instead, PCP supports working with a plan-oriented brush or freely collaging into 3D space with the spherical brush. Given the option of defining a fixed rotation and translation between source and workspace, these two brush orientations give a great deal of flexibility. Both of these brushes are usable in two different modes: first, in replacement mode, any existing points on the target site that are within the brush’s radius are removed before the new points are placed. In Figure 8, for example, a substantial block-building is duplicated and dropped in to replace part of the existing urban context. And second, the additive mode, where the new points are simply combined with the existing, such as in Figure 9, where numerous church spires are added to a perimeter block building. Finally, two brush types are available. The first is a direct sampling clone brush: by locating a reference point somewhere in the source ortho-image, the user is able to apply the surrounding point-cloud to the new site. Pressing the mouse sets up a geometric translation between the reference point and current mouse point. A vertical translation can be manually added to accommodate for differences in ground elevation between two data sources. This can be seen in Figure 10, where a series of strips are generated through repeated sampling and a consistent starting point. The second brush supports crop-and-paste: with this brush, the user can isolate a desired element (by erasing the surrounding points). The element is then stored to a catalog which can be dropped into the new site, such as in Figure 11 where a set of typical urban elements have been isolated on the site. When these elements are deployed amidst an existing cloud of points, current points within the bounding area of the new element are erased (just as in the replacement brush).

Between these shapes, modes, and types, Point-Cloud-Paint provides the fundamental capabilities for collaging together new urban design speculations from existing point-clouds. Since it is impractical to store the point-cloud of an entire city in computer memory, PCP only holds the collage itself and a small portion of the source material in memory at one time. Instead, PCP reacts to any pan-motions in the source panel by loading up any unloaded portion of a local area (500 x 500 m) around the new point of focus and releasing any points that fall out of this range. PCP utilizes a double-precision vector class to store the
location of cloud points, avoiding the possibility of floating-point
imprecisions that can effect geometry far from the datum of its
geographic coordinate system. While loading new points into
memory, PCP also uses each point’s location to look up a corre-
sponding colour value in the ortho-imagery. This value is stored
together as a single integer alongside the location so that each
point will have a colour assigned when the user begins collaging
into the workspace. As a means of assigning colour to a point-
cloud, this method is convenient but rather limited; any points
located immediately above and below each other are assigned
the same colour value since they are both vertically projected to
the same location of the plan-oriented imagery. This is especially
limiting in the case of tall buildings, where entire facades end up
acquiring the colour of the roof’s edge.

Each time the user activates a brush, a subset of the local area
of the point-cloud must undergo some action (for example, be
copied or be erased). Determining this subset means evaluating
which points are within the brush’s radius distance from the
center-point of the brush. Even though PCP is only concerned
with a subset of the city point-cloud at one time, iterating
through this set of points to evaluate distance is still unman-
ageable if the objective is smooth interaction. PCP resolves this
computing problem by organizing the point-cloud into a recur-
sive octree data structure (Meagher 1980). Points are arranged
into nested cells so that, given the brush’s location and radius,
PCP can calculate which points might be affected by recursively
testing for overlap between the brush and cells of increasingly
fine resolution. This data structure significantly improves interac-
tion speeds, since it means that the vast majority of points can
be ignored for the vast majority of user interactions.

As the user collages points from the source material into
the workspace, octree cells that the brush interacts with are
flagged as requiring update. Meanwhile, taking advantage of
the eminently parallel character of point-cloud calculations,
one or more in a set of thread-independent processes go to
work. These threads look for flagged cells to update by adding,
replacing, or removing points depending on which brush is active.
As the sophistication of the brush increases, the computing
time involved in updating an individual cell increases, but the
multi-threaded implementation ensures that interactivity is never
compromised.
RESULTS AND REFLECTIONS

The result of the research presented in the paper is the software tool PCP itself. During the course of development, the capacities for rapidly testing different urban configurations have become clear. While the most provocative outcomes might await the collection of additional point-cloud datasets from other cities, already PCP has been demonstrated to support quick sketch experiments of hybridized architectures (Figures 13 and 14), as well as larger urban experimentation (Figure 1).

At the same time, certain limitations of the collaging technique and areas for future improvement have become apparent during testing. In particular, the absence of any automatic process for managing the vertical translation of points sampled from sites with different elevations can be quite tedious and renders any sloped portions of the city all but unusable. The default circular and spherical brushes implemented present difficulties in the collaging of orthogonal features such as buildings and intersections; depending on the brush size, it is often either difficult or time consuming to accurately manage such corners.

The goal of this research has been to develop a method and software tool that respond to the urban design provocation found in Blanqui’s text. Conceptually, pursuing this goal has meant hybridizing the discourse surrounding voxelization with the discourse and technology of point-clouds in order to expand the latter’s perceived capacities beyond the representation of current conditions. Technically, developing a tool (PCP) for a collage-like combinatorial interaction with point-clouds has led to an engagement with the literature on interactive mesh modeling. From this work, PCP adapts the concept of a three-dimensional brush interface. At the same time, the specific demands of dense urban point-clouds has resulted in a different set of interface and technical priorities.

CONCLUSION

Based on the findings of this paper, a combinatorial approach to point clouds offers promising opportunities as a methodology for urban design. In concluding, a couple of particular directions for future research must be mentioned. In the course of development and testing, the link between point-clouds and corresponding ortho-imagery was very productive—it helped significantly with understanding shape and urban features within the point-cloud. Yet this imagery is only one of many kinds of dataset which could be coordinated with the point-cloud. For example if hidden vector representations of building footprints were linked to the LiDAR data, and these vectors were clipped, moved, and modified along with the points, it would be possible to immediately provide analytical feedback during collage (for example parcel coverage, density, etc.). Similarly, linking topographic data (possibly as a separate mesh) would enable collaged points to replace existing content but still respond to the
features of the site (by displacing each point vertically according to the difference of elevation between source and destination).

Collecting and coordinating these data sets can also contribute to a second future direction of research—there are significant opportunities to be explored in the integration of machine learning algorithms into the PCP workflow. Notwithstanding the assertion earlier in this paper that what constitutes an urban element should be left open, a classifying algorithm could be productively trained using other datasets to distinguish features such as roads, green spaces, and buildings. While a literature exists on machine learning based analysis of urban point-clouds, authors tend—just as in the point-cloud discourse discussed earlier in this paper—to focus on the analysis of existing conditions rather than the role that analysis and classification can play in recombinatory and collage approaches to design.

REFERENCES


IMAGE CREDITS
Figure 2: Portrait of L.A. Blanqui by Amelie-Suzanne Serre, 1835.

Figure 3: Portrait of Baron G.E. Haussmann by Pierre Petit, 1862.

All other drawings and images by the author.

**Ultan Byrne** is a researcher with previous degrees in architecture and philosophy. In a combination of teaching, writing, and programming, Ultan considers the relationships between technologies of digital networking, new design software, and enduring questions of architectural/urban design. Ultan is a lecturer at the John H. Daniels Faculty at the University of Toronto.