Get Real[1] Towards Performance-Driven Computational Geometry

Neri Oxman

[1]"GetReal" is one of the many retrieval methods within the VB scripting language which prompts the user for numerical or textual input required by the program in order to execute the script. The word "real" designates a real number as opposed to an integer only. By way of metaphor, this implies the translation of performance-driven data into geometry.

Get Real Towards Performance-Driven Computational Geometry

Neri Oxman

In historic design conventions geometry has traditionally promoted *descriptive* manifestations of form. Beyond the realm of geometry, the concept of *performance* which may inform such manifestations also carries important potential for design generation. This work explores the relation between geometry and performance from a computational-geometry perspective. It does so by revisiting certain analytical tools offered in most of today's 3-D modelers which support the evaluation of any generated surface geometry specifically curvature and draft angle analysis. It is demonstrated that these tools can be reconstructed with added functionality assigning 3-D geometrical features informed by structural and environmental performance respectively. In the examples illustrated surface thickness (as a function of structural performance) is assigned to curvature values, and transparency (as a function of light penetration performance) is assigned to light analysis values. In a broader scope this work promotes a methodology of performance-informed form generation by means of computational geometry. Vector and tensor math was exploited to reconstruct existing analytical tools adapted to function as design generators.

1. INTRODUCTION

Computational geometry is the study of algorithms generated to solve problems *in terms* of geometry [1]. Unsurprisingly many problems in computational geometry are classical in nature; however the discipline was developed parallel to advances in computer graphics and computer-aided design and manufacturing for the purpose of visualization and materialization processes respectively. Recent developments in computational geometry along with the expansion of CAD software packages supporting geometrical modeling and design exploration have brought about the advance of analytical tools. Traditional CAD applications have long allowed for the straightforward calculation of absolute and relative location of features in Cartesian space. Such tools have now been expanded to include complex computational methods for non-Euclidian geometries such as B-Spline surfaces and NURBS curves.

Much has been written about the role of computational geometry in the description, representation and illustration of form; however the assumption that certain attributes which lie outside the realm of geometry such as spatial, structural, and/or environmental performance may be examined and acted upon by the very methods of computational geometry have only recently been proposed [1].

This work assumes an inherent, and potentially an instrumental relation between geometry and performance in devising advanced analytical functions (some of which already exist today as built-in user features) to support generative design explorations.

2. BACKGROUND

2.1. Working Definitions

The accessibility of both computer software and hardware since the early Nineties has motivated a renaissance in architecture celebrating formal expression. Such formal orientation, strongly driven by a new level of interest in geometric complexity also referred to as "non-standard" and "free form", is recently shifting the discourse from issues of form representations to issues of form generation. The notion of *performance* plays a significant role in this light and has entered the discipline at a time where an abundance of computational tools for design simulation, evaluation and optimization reside [2]. Performance based design utilizes digital technologies that support the generation of form resulting from design performance, such as structural and/or environmental attributes. It is considered by many to serve as a methodology for design generation where the incorporation of performance criteria informs processes of formal manipulation [3]. However, the integration of performance simulation and geometrical manipulation still remains a significant challenge.

Geometrical manipulation has been enhanced by the ease of formal manipulation enabled by Non-Uniform Rational B-Splines, commonly known as NURBS [4]. NURBS modeling has for some time supported the design of complex forms based on surface geometries.

NURBS are mathematical representations of 3-D geometry that can accurately describe any shape from the simple to the most complex 3-D organic free-form surface or solid. Because of their flexibility and interactivity in use as well as their accuracy, NURBS models can be used in various processes from illustration and animation to manufacturing.

The work illustrated here was developed in the Rhinoceros software package using the Visual Basic scripting environment. It assumes a constructed 3-D surface with no thickness and evaluates the geometrical characteristics of such surface. On the basis of the evaluations, certain functions are applied to the existing geometry to further develop its performance related attributes based on specific structural and light performance criteria. In operation it employs a NURBS surface as input and generates a 3-D solid geometry as output.

2.2. State of the Art

The Predominance of Geometry

Assuming a symbiotic relationship with geometry, design incorporates many issues that are independent of any specific formal configuration. These issues may be defined as the "parameter space" for a given design problem. Such "spaces" may be regarded as "pre-geometric" in nature; having arrived at a particular configuration, there exist potentially various alternative material interpretations of that particular configuration which may be regarded as "post-geometric" issues [5]. This work attempts to eliminate such characteristic procedural hierarchies which may potentially exist between "pre" and "post" geometrical design operations (i.e. form-generation first, material and/or performance evaluation later) and offer a new methodology for the incorporation of material performance directly and explicitly into the geometric representation. Some innovative work along these lines has been carried out which argues that models for design exploration promoting different forms of design representation should be bridged to support the discovery of novel designs [6].

Performance-Based Material Distribution through Computational Geometry

Computational geometry, beyond serving as a form of description, offers opportunities for the representation and interpretation of such description informed by statements which lie beyond geometry but may be defined by the very same methods. Traditionally, when formal expression becomes geometrically complex, we apply conventional means to simplify the design; in many cases we reformulate it using componentization. This process may be defined as parsing the irregularity of form into elementary components [7]. It becomes even more effective when such components can be modeled to include all associative relationships between geometric units so as to allow for efficient reconfiguration using global variables. In such applications the object being designed is often modeled as an assembly of geometrically defined components. Even if the building is not actually to be fabricated from such components, it is usually conceptualized and modeled in these terms [8].

The design logic of such an approach assumes the appropriateness of the idea of assembly, not necessarily from a construction viewpoint, but as an implicit design perception. What if we were to replace the notion of material assembly with that of material distribution? When we consider "distribution" as a method which allows for localized formal traits to be expressed while at the same time retains its qualities as a consistent global system, we liberate ourselves from the need to break things down into units.

Generative Computational Geometry vs. Optimization

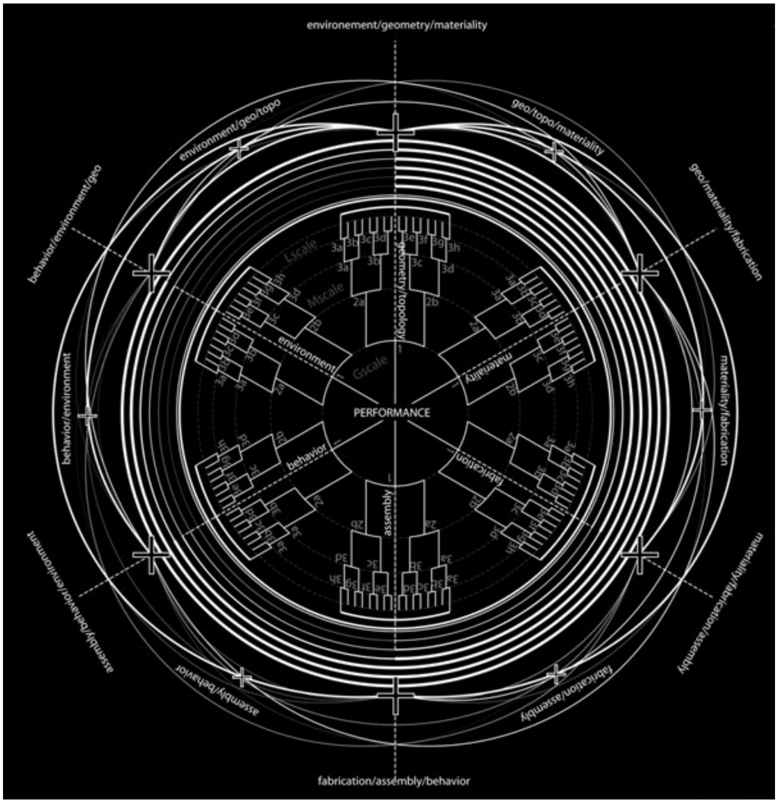
Recent initiatives in computational geometry capitalize upon optimization-based routines using ESO (Evolutionary Structural Optimization). Such routines are based on finite element analysis methods (FEA) capable of optimizing the formal geometry of an object to obtain minimum volume through an iterative design processes under even stress-distribution [9]. However, such analytical methods have rarely been used as generative tools allowing the designer to shift freely between representations and allow, by means of optimization, for difference in kind to occur in addition to difference in degree.

2.3. Problem Definition

Computational geometry has customarily been used as a means for description and/or analysis of form. To a lesser extent it has been made instrumental for purposes of design generation. Given the significance of such tools to explorations of shape and form, the limitation remains the partitioning between methodological models of description and models of, and for, generation. The integration of analytical tools and techniques as propositional rather than descriptive may provide the user the capability to exploit work with computational geometry as a driver for the design process possessing built-in performance considerations.

2.4. Aims and Objectives

Multi-objective representation where geometrical entities (or forms of description) promote speculations regarding the structural and/or environmental performance of the model endorses a design process that is generative in nature. The main objective of this work, as a central prerequisite to a potential paradigm shift in generative design, is to promote a novel methodology which supports the seamless integration of geometry and performance (Figure 1). Other design drivers may include material properties [3], fabrication methods and assembly strategies. However, these drivers exist merely as equipotent tools enabling the designer to establish his/her own path or hierarchy as it may be informed by such an approach.



◀ Figure 1. Theoretical design space: the diagram illustrates different potential design drivers (i.e. geometry, materiality, fabrication etc) and the way in which they may be evaluated and informed by performance.

Recent work along those lines has appeared in a former publication which examines the interface between *material performance and geometry* by discussing the association between elasticity (as a material property) and curvature. Such association was achieved by means of homogenizing protocols which translate physical material properties into geometrical functions [3]. Building upon previous exploration this paper looks at the interface between *structural performance and geometry*, by incorporating structural performance data as geometrical features.

Given any free-form NURBS surface geometry the aim was to generate additional geometrical and material entities (such as thickness and transparency respectively) which transform the zero-thickness NURBS surface to a 3-dimensional entity based on the visual analysis tools (such as curvature analysis or draft-angle analysis respectively) applied to that case-study surface.

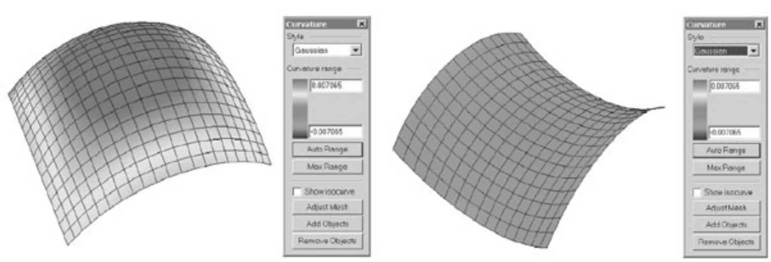
2.5. Organization

Following the introductory sections, Chapter 3 includes a short overview of the exiting analytical tools, their functionality and applications as they exist today. Two analysis tools are described in particular: the curvature analysis tool, and the draft analysis tool. Following the introduction, the theoretical and mathematical foundations for the reconstruction of such tools is described. Particularly vector and tensor math are introduced as foundation knowledge for the demonstration of the reconstructed tools that follow. Chapters 4 and 5 are identical in structure, and describe the reconstruction of the curvature analysis and the draft-angle analysis tools respectively. Each of these chapters explores the existing functionality and the added functionality to these tools. Chapter 6 introduces some design implications which are related to this exploration, followed by conclusions and discussion of potential contributions.

3. TOOLS DESCRIPTION AND RECONSTRUCTION

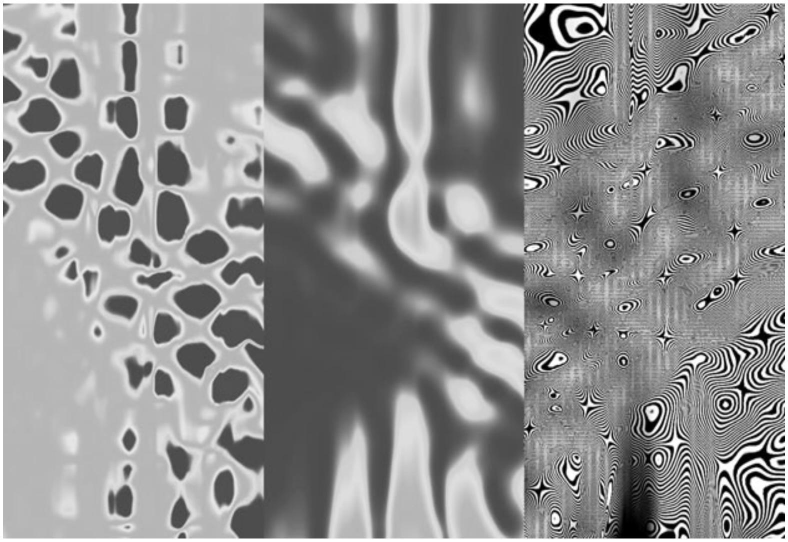
3.1. Tools Description

Traditional analysis tools inherent in most CAD applications are based on straight-forward computations of arithmetic objects with magnitude (scalars) and/or direction (vectors). Some of such tools include: point position, length, distance, angle, radius, bounding box, normal direction, area, area cancroids, area moments, volume, volume cancroids, volume moments etc. The development of NURBS (Non-Uniform Rationale B-Splines) based software had brought about a new generation of tools targeted towards the analysis of surface features. Most NURBS modeling environments known and used today include a suite of tools categorized as *NURBS analysis tools*. Such tools offer a set of functions which visually analyze surface features such as types of curvature which are geometrically calculated. The analysis algorithm is hidden from the user and the results are in most cases (i.e. Rhinoceros, Digital Project etc.) displayed as a gradient color map to which are assigned attributes of a particular analysis tool. The most known or used of such tools is the *curvature analysis tool* which determines the degree of curvature across any given surface as a color map indicating the type and degree of curvature at any given point on the surface (Figure 2).



► Figure 2. Left: Surface with positive Gaussian curvature (synclastic). The surface is bowl-like. Right: Surface with negative Gaussian curvature (anticlastic). The surface is saddle-like.

The color map includes an array of colors ranging from red to blue, red indicating synclastic curvature (bowl-like surface features) and blue indicating anticlastic curvature (saddle-like surface features). Other NURBS-based analysis tools (Figure 3) include geometric continuity, deviation, curvature graph on curves and surfaces, naked edges, and working



◀ Figure 3. Visual NURBS surface analysis tools: Left: Curvature analysis (used to evaluate curvature). Middle: Draft-Angle analysis (used to evaluate curvature in relation to viewing point). Right: Zebra analysis (used to evaluate surface smoothness).

surface analysis view-port modes (draft angle, zebra stripe, environment map with surface color blend, Gaussian curvature, mean curvature, and minimum or maximum radius of curvature).

3.2. Tools Reconstruction: Theoretical Foundations

The aim of this project as previously stated above, was to evolve a 3-D geometry which is based upon, and corresponds to, the geometrical features, as have been analyzed by the software, of any given zero-thickness surface geometry such that this surface may potentially become a 3-D design artifact. In order to simplify this aim two of the most common analysis tools were remodeled and recomputed with additional functions supporting the generation of 3-dimensional objects. Each tool was computed and coupled with a method statement regarding the generation of additional attributes. Such attributes included thickness and transparency which were computed for every sampled point across the surface. In this project, the thickness of the surface was attributed to two types of analysis: curvature analysis and draft angle analysis.

In this re-conceptualization the thickness of the surface was a function of its curvature and its location (relatively to a given light source) as was measured and reported at any given point. This thickness, which will now be referred to as the *informed geometrical feature* may potentially add spatial, structural and environmental data in the following stages of design materialization. From this follows that the "thickness" attribute which correspond to the curvature-analysis protocol may suggest structural stability or spatial enclosure, whereas the "thickness" attribute which is applied relatively to the position of a light source may indicate degrees of translucency which display a range of light effects from opaque to transparent depending on the surface thickness. So depending on the material which would later be assigned for the design, and assuming such material may changes its transparency as a function of its thickness (such as foam or plastics for example), thick profiles will be read as opaque and thin profiles will give the effect of being almost transparent.

Such processes, by the very nature of assigning geometrical attributes *informed* by means of computational analysis, demonstrate the very many forms of translation which may well remain hidden from the user or, alternatively, may exist explicitly at her/his disposal. The translation path in this case is comprised of operations ranging from curvature analysis to the assignment of *informed geometrical features* based on such analyses. Such process is cyclic by nature as it may correspond to different geometrical parameters (representing physical performances attributes) and may be applied and re-adjusted iteratively in the generation of the final form.

The question of assigning performative interpreters to geometrical data requires the build-up of some translational functions to parse the math and transform it into performance data.

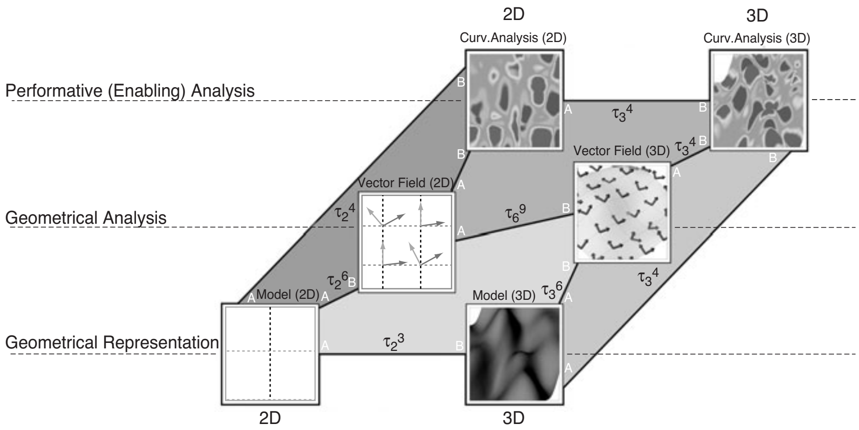
3.3. Tools Reconstruction: Mathematical Foundations

When considering how tools of analysis may be modeled in order to incorporate 'propositional interpreters' of performance criteria (i.e. the relation between the degrees of curvature to structural performance, or the relation between thickness and degrees of transparency) we assume that geometric quantities and descriptions may be made to incorporate and represent the physical properties of matter. Thus we seek to incorporate performative material knowledge within the three-dimensional (thickness) representation of geometric form. Geometric form thus becomes "materialized" from the point of view of performance analysis. This potentially renders analysis and generation of computational geometry: iterative in both directions; interactive; and inherently performative.

The fields of physics, structural engineering and material science contain many cases in which formulas are specified to include and solve relations between physical properties of matter through geometry. Such for example are Maxwell's "method of drawing lines of force and equi-potential surfaces" from the late 70's [10].

Granted the existence and knowledge descriptive capacity of such representations, the aim now becomes to rewrite such notational correspondence in terms of geometry. The field of computational geometry contains such quantities which can become instrumental in this process. These geometrical and/or physical quantities may be categorized by considering the degrees of freedom inherent in their description (Figure 4).

The scalar quantities are those that can be represented by a single number (i.e. speed, mass, and temperature). There are also vector-like quantities such as force that require a list of numbers for their description (so that direction can be accounted for). Finally, quantities such as quadratic



◀ Figure 4. Computational geometry reciprocal transformations diagram: the diagram illustrates 2D (left) and 3D (right) representations from bottom to top and increasing in complexity, from the basic descriptive geometrical representations (bottom), to forms of analysis (middle) and performative-enabling analysis representations (top).

forms naturally require a multiply-indexed array for their representation. These latter quantities can only be conceived of as tensors. Actually, the tensor notion is quite general and applies to all of the above examples; scalars and vectors are special kinds of tensors. The feature that distinguishes a scalar from a vector, and distinguishes both of those from a more general tensor quantity is the number of indices in the representing array. This number is called the rank (or the order) of a tensor. Thus, scalars are rank zero tensors (with no indices at all) and vectors are rank-one tensors.

When relating two types of vectors (such as displacement and gravity) in some mathematical notation, we are in essence generating a tensor object. Tensors which relate two vectors of the same type are known as polar tensors, whereas tensors which relate two vectors of different types are known as axial tensors. The different vector types may include for instance velocity, displacement, acceleration, gravity or torque. Vectors and tensors together make up a space of an arbitrary dimension, *n*. In most cases this description implicitly denotes some space (no need for an explicit space of position). In order to break away from the primacy of numbers over matter (or performance), and to allow the user to do math on spatial entities (and not only numbers), we may begin to look at vectors as spatial and mathematical things, rather then purely numerical entities.

We may now take advantage of the usual vector algebra operations available in 3D space (R3) to study the curvature (departing from linearity) and torsion (departing from planarity) of curves in space. Since we are interested in curves with non-zero speed everywhere, we can always re-parameterize to achieve unit speed. Let us now move briefly from curves to surfaces through the description of the manifold. Manifolds are important objects in mathematics and physics because they allow more complicated structures to be expressed and understood in terms of the relatively well-understood properties of simpler spaces.

A *manifold* is a representation of a mathematical space in which every point has a neighborhood which resembles Euclidean space, but in which the overall geometry may be more complicated. Manifolds may be classified according to dimension. For example, lines are one-dimensional manifolds, and planes two-dimensional manifolds. In a one-dimensional manifold (or one-manifold), every point has a neighborhood that looks like a segment of a line. Examples of one-manifolds include a line, a circle, and two separate circles. In a two-manifold, every point has a neighborhood that looks like a disk. Examples include a plane, the surface of a sphere, and the surface of a torus. A Riemannian manifold is a manifold possessing a metric tensor. Simply put, the metric tensor is a function which tells how to compute the distance between any two points on a given space. The metric tensor is defined abstractly as an inner product of every tangent space of a manifold such that the inner product is a symmetric, non-degenerate, bilinear form on a vector space. This means that it takes two vectors as arguments and produces a real number. It should be noted that the array-of-numbers representation of a tensor *is not the same thing* as the tensor. An image and the object represented by the image are not the same thing. The mass of a stone is not a number. Rather, the mass can be described by a number relative to some specified unit mass. Similarly, a given numerical representation of a tensor only makes sense in a particular coordinate system. Some well known examples of tensors in geometry are quadratic forms, and the curvature tensor.

In the framework of this work, some fundamental vector math was used to regenerate the analysis tools along with their added functionality. Two scripts were developed corresponding to the two surface-based analytical tools (curvature analysis and draft angle analysis). The following sections describe the sequence of operations that were executed to reconstruct and reconfigure these tools for the purpose of performance-based design generation.

4. STRUCTURAL PERFORMANCE BASED COMPUTATIONAL GEOMETRY: CURVATURE ANALYSIS AND RECONSTRUCTION

4.1. Curvature Analysis Tool

The Curvature Analysis command in modeling software packages such as Rhinoceros and Digital Project is one of a series of visual surface analysis commands. These commands use NURBS surface evaluation and rendering techniques to visually analyze and display surface smoothness, curvature, and other geometrical properties. Such commands may potentially inform or guide the design process in that geometrical attributes may be translated or interpreted as performance manifestations. This section describes the regeneration of the Surface Curvature Analysis command and the process in which spatial and structural information are the outcome of manipulating a free-form surface to give it structural integrity using computational geometry tools and algorithms. The aim is to employ an existing geometry-based tool of analysis in order to foresee the structural properties of the input geometry.

Firstly, let us describe the surface analysis command as it exists in the software[2]. Following the basic definitions, a detailed description of the script which was used to regenerate the command in a design context will be given.

Some basic definitions:

(1) Gaussian and Mean Curvature: At any point on a given curve in the plane, the tangent line is the line best approximating the curve passing through this point. In addition, it is possible to represent the best approximating circle that passes through this point and is tangent to the curve. The reciprocal of the radius of such a circle is the curvature of the curve at this point.

(2) Principal curvatures: The principal curvatures of a surface at any given point are the minimum and maximum of the normal curvatures at that point. Normal curvatures are the curvatures of curves on the surface lying in planes including the tangent vector at that given point. The principal curvatures are used to compute the Gaussian and Mean curvatures of the surface.

(3) Gaussian curvature: The Gaussian curvature of a surface at a point is the product of the principal curvatures at that point. The tangent plane of any point with positive Gaussian curvature touches the surface at a single point, whereas the tangent plane of any point with negative Gaussian curvature cuts the surface. Any point with zero mean curvature has negative or zero Gaussian curvature.

(4) Mean curvature: The Mean curvature of a surface at a point is one half the sum of the principal curvatures at that point. Any point with zero mean curvature has negative or zero Gaussian curvature. Surfaces with zero mean curvature everywhere are minimal surfaces. Surfaces with constant mean curvature everywhere are often referred to as CMC (Constant Mean Curvature) surfaces. CMC surfaces have the same mean curvature everywhere on the surface. Physical processes which can be modeled by CMC surfaces include the formation of soap bubbles, both free and attached to objects. A soap bubble, unlike a simple soap film, encloses a volume and exists in equilibrium where slightly greater pressure inside the bubble is balanced by the area-minimizing forces of the bubble itself. Minimal surfaces are the subset of CMC surfaces where the curvature is zero everywhere. Physical processes which can be modeled by minimal surfaces include the formation of soap films spanning fixed objects, such as wire loops. A soap film is not distorted by air pressure (which is equal on both sides) and is free to minimize its area. This

[2] The software referred to in this work is the Rhinoceros and Digital Projects platforms in particular.

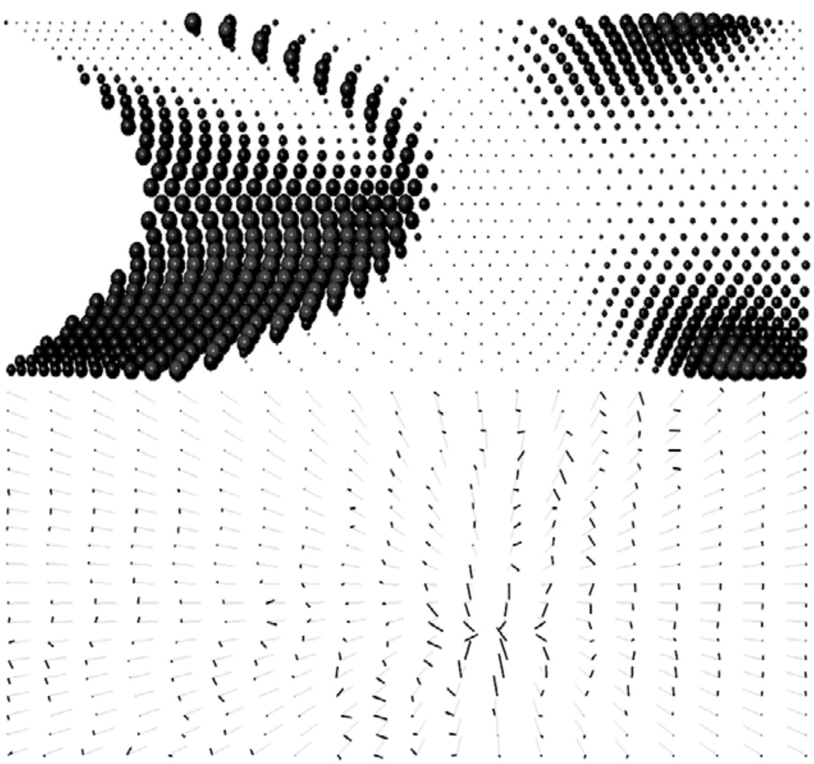
contrasts with a soap bubble, which encloses a fixed quantity of air and has unequal pressures on its inside and outside.

To summarize, a smooth surface has two principal curvatures. The Gaussian curvature is the product of the principal curvatures. The Mean curvature is the average of the two principal curvatures. By convention, most of the software packages which incorporate such analysis tools do so by assigning a color-coded pattern on top of the surface which assists the user to determine the type and degree of curvature for any given surface. In the framework of surface curvature analysis, the color red is usually assigned to a positive value of the Gaussian curvature, green is assigned to zero Gaussian curvature, and blue to negative value of Gaussian curvature. Any points on the surface with curvature values between the values which have been specified by the user will be displayed using the corresponding color. For example, points with a curvature value half way between the specified values will be green. Points on the surface that have curvature values beyond the red end of the range will be red and points with curvature values beyond the blue end of the range will be blue. A positive Gaussian curvature value means the surface is bowl-like and is also called: *synclastic* curvature. A negative value means the surface is saddle-like and is also called: *anticlastic* curvature. A zero value means the surface is flat in at least one direction (i.e. planes, cylinders, and cones). The Mean curvature displays the absolute value of the mean curvature and is useful for finding areas of abrupt change in the surface curvature. The Max radius option is useful for flat spot detection. By default, red areas in the model indicate flat spots where the curvature is practically zero. The Min radius option determines whether the surface includes areas where it may bend tightly (so as to generate an intersection) when it is offset beyond a certain threshold limit determined by the user. In this case, the Red color will be set as the radius of offset distance, and the blue will indicate this dimension, multiplied by a factor of 1.5. The red areas indicate regions in the surface which will self-intersect upon offset. Blue areas are geometrically sound in this respect. Areas from green towards red should be viewed with suspicion.

4.2. Tool Reconstruction

The aim of reconstructing the curvature analysis tool was to use the analysis as a 3-D from-generator driven by structural performance considerations. In this case surface thickness is created by offsetting the original surface in a non-homogeneous manner, corresponding to the surface curvature. By convention, highly curved areas across the surface have been assigned minimal thickness; while smooth regions have been assigned maximum thickness. This method allows for the application of curvature-dependant differentiated thickness to the original zero-thickness surface and acts as a "smoothing" function across its entire surface. This method also has structural implications with regards to the self-stabilization of the surface upon orientation considerations: the smoothness function allows treating surface thickness in the context of the global structural performance. In this project specifically, the aim was to design a wall-mounted element which would be structurally sound and self-supportive while still remaining light-weight and economical to fabricate.

The script was generated based on an exiting NURBS surface model. This surface has zero-thickness prior to the application of the script. Initially, the script runs an automatic surface re-parameterization method. The parameter values of the surface object are recalculated so that the parameter space of the surface object is roughly the same size as the 3-D geometry of the object (surface generated by user). This function may be executed automatically (by default) to allow for a quick calibration of the parameter space. Proceeding re-parameterization, the script asks the user to enter the number of rows and columns across the surface to establish its underlying geometry and define a grid of registration nodes. Every surface is roughly rectangular. Surfaces have three directions: u ("rows"), v ("columns"), and normal. The u and v directions are like the weave of cloth or screen. The u-direction is indicated by the red arrow and the v-direction is indicated by the green arrow. The normal direction is indicated by the white arrow. The u, v, and normal directions may be thought of as corresponding to the x, y and z axes of the surface. The "rows" and "columns" entered by the user



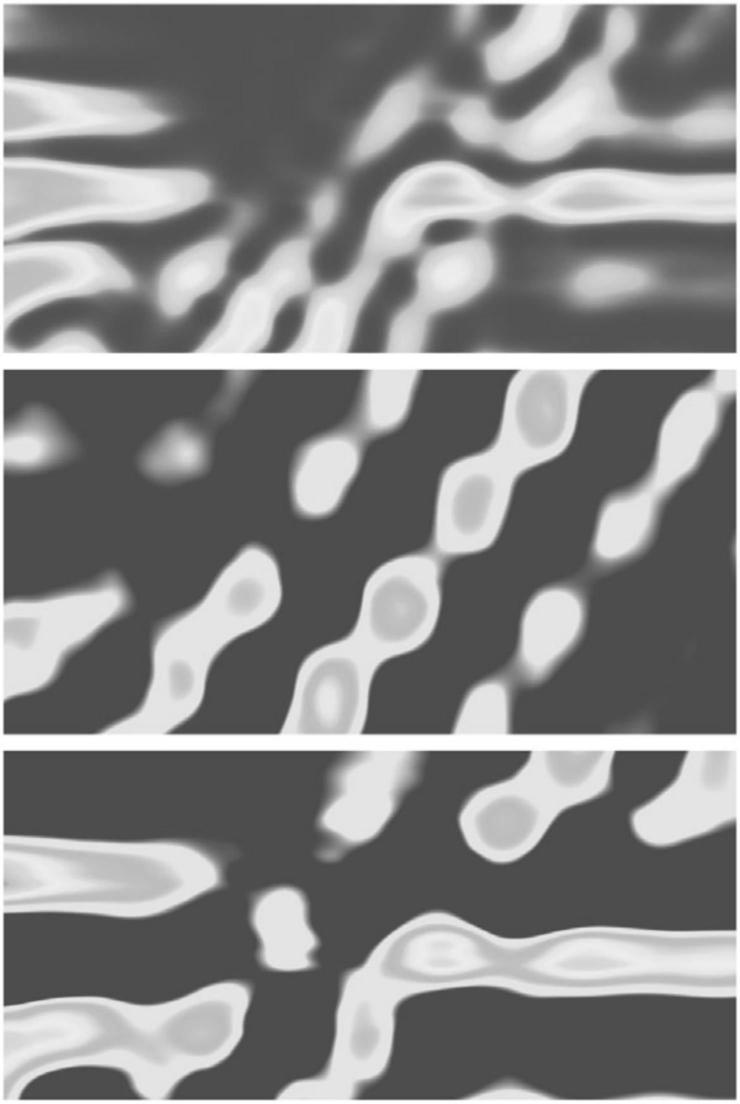
◀ Figure 5. Curvature analysis reconstruction and tectonic generation: the modeled surface is reconstructed as an array of spheres the size and color of which correspond to the degree of curvature mapped by the function.

establish the granularity of u and v intersection points: the higher the values, the more points distributed across the surface for the purpose of sampling or attribute assignment which will take place at a later stage. In the next step the script computes surface normals and plots them in the modeling environment. This function returns two 3-D points that define the normal to a surface at a parameter. It takes two parameters as input: the object's identifier (the user generated surface) and an array containing the UV parameter to evaluate. The array elements which are returned include a point on the surface at the specified parameter (given at each u and v intersection) and a point normal to the surface at the specified parameter. The normal registration allows computing the curvature registered in each U and V point as defined by the user and assign a color to those points. Finally, based on the sampled curvature an offsetting function assigns surface thickness matching the curvature analysis mapping. Maximum and minimum thicknesses are defined by the user and scaled automatically to generate thickness range according to sampled surface curvature values (Figure 5).

5. LIGHT PERFORMANCE BASED COMPUTATIONAL GEOMETRY: DRAFT ANGLE ANALYSIS AND RECONSTRUCTION

5.1. Draft Angle Analysis Tool

The draft angle analysis maps out the projection pattern on a given surface from the point of view of a predefined construction plane. The projection is the transformation of a surface defined by points in one plane (the "construction plane", which is by default the active view port) onto another plane (the original generated surface) by connecting corresponding points on the two planes with parallel lines. The draft angle depends on the construction plane orientation. When the surface is vertical/perpendicular to the construction plane, the draft angle is zero. When the surface is parallel to the construction plane, the draft angle is 90 degrees. These angles are assigned a color map to allow for a gradient color representation of the draft angle. The Draft Angle dialog box allows the user to set the angle for the color display. The density of the mesh can also be adjusted if the level of detail is not fine enough. The "pull direction" (the direction from which the surface is being viewed, defined by the location of the construction plane) for the Draft Angle Analysis is the z-axis of the construction plane in the active view port when the command starts. The normal direction of the surface points toward the "pull direction" of the model. Changing the construction plane before using the Draft Angle Analysis command allows the user to define any direction as the pull direction. Recent CAD packages include the function for a dynamic draft angle analysis which allows moving and rotating the model in real time while analyzing the dynamic draft angle of the model (Figure 6).

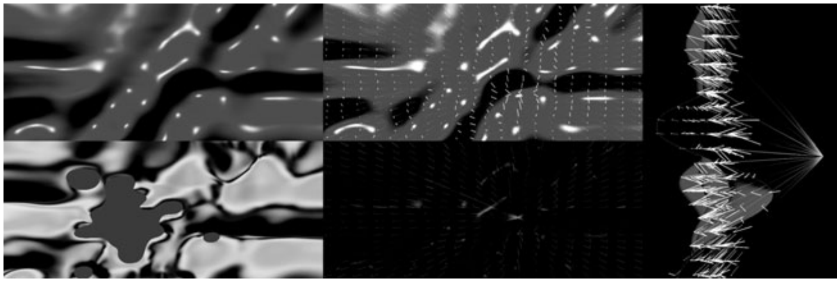


◀ Figure 6. The Draft-Angle analysis command allows the user to evaluate surface curvature relative to angle of viewing point. The image is comprised of four analyses taken from four different views (top view, isometric view, and elevation).

5.2. Tool Reconstruction

Similarly to the surface-curvature analysis tool reconstruction (see section 4.2), the draft-angle analysis tool is applied to a zero-thickness surface generated by the user. The surface re-parameterization method is applied automatically by the script, followed by U and V registration grid definition and surface vector computation. In addition to the normalized surface vectors, plotted as well are the vectors which extend from the light source (or alternatively the view-port orientation point) to the U and V surface registration points. These two sets of vectors (surface vectors and vectors connecting between the surface and the light source) are used to calculate the draft angle, which is the angle between the surface and the light source

► Figure 7. Composite image illustrating the phases of the Draft Angle Analysis tool reconstruction and application: Left-top: Initial surface generated by user. Left-bottom: final result of tool application. Right images illustrate the process of vector re-parameterization and computation of light-source angle relative to the surface.



as sampled in e very points across the surface. This set of angles may be conceptually regarded as a tensor field linking the geometrical properties of the surface to a localized agent outside it, which determines its light effects. A threshold value is entered by the user which determines the smallest angle from which the light source does not "see" the surface, an angle in which an opening is applied to the surface to allow for more light in. Finally, the range of angles is normalized to fit a range of thickness (minimum and maximum thickness is defined by the user) which allows for varied surface sections to be generated. The result of this script is a set of planar sections which modify their thickness according to their relative distance from the any user defined "light source" location (Figure 7).

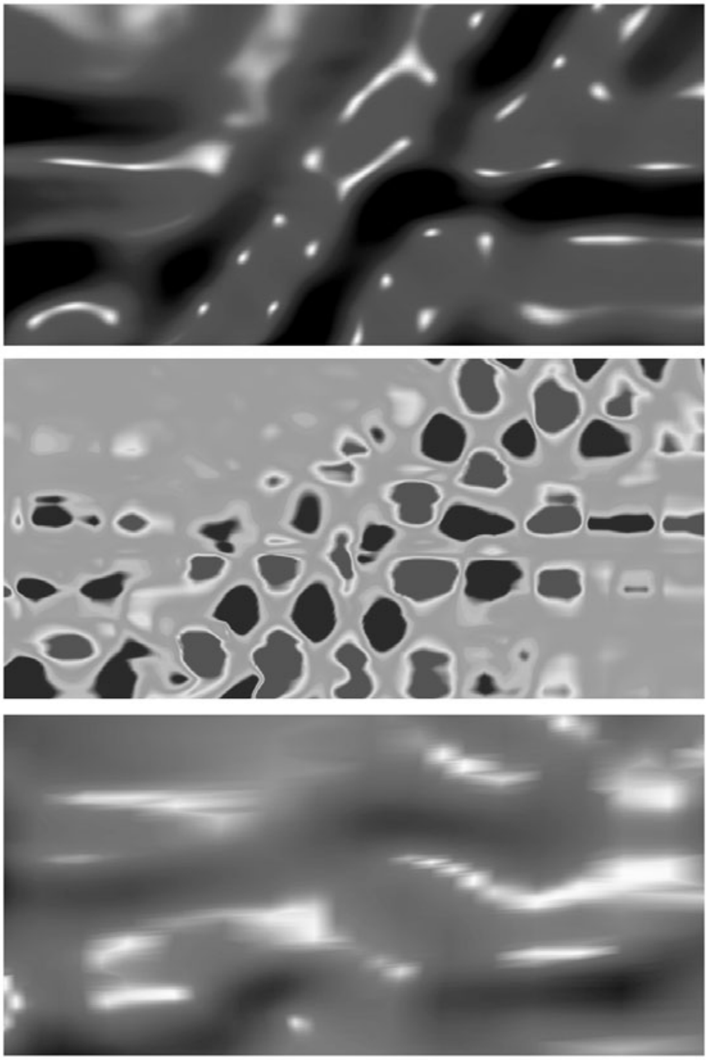
6. RESULTS AND IMPLICATIONS FOR MODELS OF COMPUTATIONAL DESIGN

6.1. Surface Curvature Analysis Script Reconstruction Results

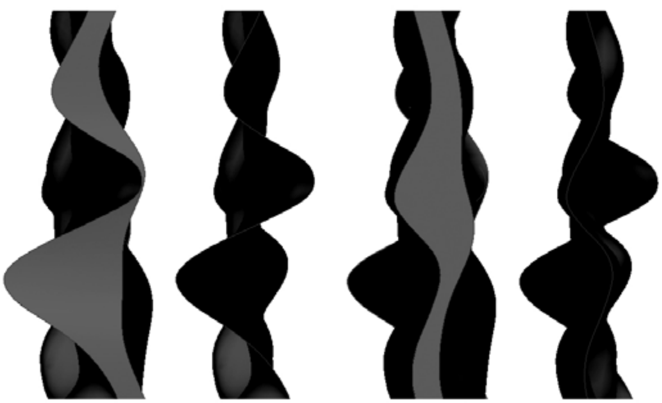
The aim was to link surface curvature data as sampled by the program, to structural performance considerations. Varied thickness was applied to each point across the surface which matched its curvature in direct relation: the smoother the curvature – the thinner the surface. This added functionality allows the user to directly associate the mapped curvature to surface thickness and to generate 3D geometries out of zero-thickness NURBS surfaces (Figures 8, 9).

6.2. Draft Angle Analysis Script Reconstruction Results

The results, as illustrated in Figure 10, demonstrate the variations of behavior with regards to light performance. The user defines the location of a "light source" relative to the existing surface and a threshold value which defines the minimum angle at which a hole is formed in the surface. In this example, the holes are formed where minimum light rays hits the surface (below an angle of 20 degrees). As a result, the hole-pattern formation is informed by the light source location and the threshold value under (or over) which the surface is fenestrated. Figure 11 demonstrates a design scenario for the Wiesner Art Gallery at MIT, as it may be informed by specific light parameters.

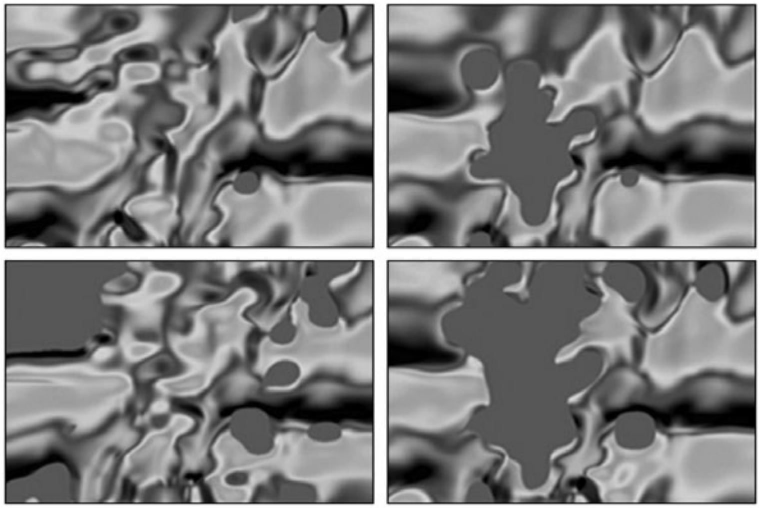


◀ Figure 8. Curvature analysis tool reconstruction: Top: initial surface. Middle: Curvature analysis. Bottom: additional surface corresponding to distributed thickness function for structural support.



◀ Figure 9. Initial and final elevation views of the surface with and without thickness. Left and right images illustrate the two elevations before and after the application of varied thickness corresponding to curvature mapping.

► Figure 10. Composite image showing four results of four corresponding iterations of the Draft-Angle script. The holes in the surface are generated when the angle between the light source and the surface approaches a minimal threshold value defined by the user. Following this, the surface is thickened locally corresponding to curvature values to allow for the differentiated thickness of the surface in its entirety to correspond to both intensity and directionality of light source.



► Figure 11. Final design for the Wiesner Art Gallery at MIT, corresponding to specific light-condition parameters and structural requirements.

6.3. Limitations

Having focused on the customized reconstruction of analytical tools and their ability to contribute to explorations driven by performance parameters, the main limitation of this work remains its dependency on the initial form to be analyzed. In other words, this application assumes the initial generation of preliminary geometry constructed by the user, and applies the evaluation and modification on top of it. The question relating to the origin(s) of form remains to be discussed and engaged with in another context. And indeed, the notion of where generation ends and evaluation begins will forever remain a design challenge.

In this light, the approach presented here relates to multiple stages within the design that occur once the initial formal has been laid out. However, the range of settings at which such an approach could become productive is vast combining generative procedures with evaluations that are predominantly associated with the end-product itself.

7. SUMMARY AND CONCLUSIONS

This work emphasized the generative potential that exists within analytical tools for geometrical evaluation. The assumption at the core of this work remains that such tools may inform the designer in her/his search for formal expression and that they contain opportunities to transcend the geometric-centered description of form by linking it to performance criteria (spatial, structural, environmental, etc).

To conclude, analytical tools are computed as geometrical statements. These statements may serve as bridging (or "multi-objective") representations between geometry and performance, geometry and construction and geometry and manufacturing. The paper sought to demonstrate such an approach by reconstructing two analyses tools for structural and environmental performance with additional functionality.

In presenting the prospects for an emerging professional profile of *informed tool-making* this paper seeks to promote a new model for contemplating form and practicing design. If such a prospect is legitimate, then it is the knowledge of computational geometry that is becoming one of the significant forms of disciplinary knowledge of the new computational design professional.

8. CONTRIBUTION

Design incorporates multiple manifestations of form from the point of view of geometry, material selection, performance, and construction. Each manifestation promotes its very own method of process and media of representation. However, some representations may be generated which support *multiple* manifestations simultaneously while reciprocally informing each other.

This paper attempted to define certain analytical forms of representation based in Computational Geometry as *enabling representations* which associate geometry to performance by means of generative computation processes. Such enabling representations are in most cases analytical in nature and offer multiple translations to occur based on a unifying code of interpretation. Such tools are also intermediate in nature, a property which renders them generative.

Beyond the notion of performance-driven interpretations based on computational geometry methods, this work has also engaged with the notion of computational analysis as a source for strategizing material distribution. Rather than breaking down the design into a series of componentized elements aiming at straightforward and simplified assemblies, this exploration demonstrates an alternative approach favoring material distribution over strategies of composition. This method promotes design manifestations which are not concerned with the notion of "buildability" to begin with, but rather let formal statements be informed by behavior and performance which result in the gradient distribution of material qualities and effects.

Computational tools associated with structural performance evaluation, simulation and optimization are traditionally utilized outside the design process or in parallel to it. The approach demonstrated in this paper seeks to integrate processes of performance evaluation with form generation by means of computational geometry.

The field of computational geometry is vast in its mathematical content however particular its application may be. In encouraging tool-making protocols which incorporate material and performance manifestations beyond their descriptive nature we are approaching what may truly be considered *as performance-based computational geometry* in the field of design.

Acknowledgements

I wish to thank Dennis Shelden, Chief Technology Officer of Gehry Technologies, for his active involvement and inspiration. I would also like to thank my friend and colleague Alexandros Tsamis, for his dedicated contribution and thoughtful insights to this work and for introducing some of the background examples presented in this paper.

References

1. Shelden, D., R., *Digital Surface Representation and the Constructability of Gehry's Architecture*, PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2002.
2. Kolarevic, B., and Malkawi, A., M., *Performative Architecture: Beyond Instrumentality*, Spon Press, New York, 2005.
3. Oxman, N., and Rosenberg, J., L., Material Based Design Computation, *International Journal of Architectural Computing*, 2007, 5(1), 25-44.
4. Cabrinha, M., From Bézier to NURBS: Integrating Material and Digital Techniques through a Plywood Shell, in: Osman, A. (Eds.), *Smart Architecture: Integration of Digital and Building Technologies*, Association for Computer-Aided Design in Architecture, Savannah (Georgia), 2005, 156-169.
5. Aish, R., From Intuition to Precision, in: Duarte, J., P., Ducla-Soares, G., Zita, S., A. (Eds.), *Digital Design: The Quest for New Paradigms*, Instituto Superior Técnico, Technical University of Lisbon, Lisbon (Portugal), 2005, 21-24.
6. Kilian, A., *Design Exploration through Bidirectional Modeling of Constraints*, PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2006.
7. Koutamanis, A., Modeling Irregular and Complex Forms, in: Nys, K., Provoost, T., Verbeke, J., Verleye, J. (Eds.), *Added Value of Computer Aided Architectural Design*, Hogeschool voor Wetenschap en Kunst - Departement Architectuur Sint-Lucas, Brussels (Belgium), 2001, 156-169.
8. Aish, R., Three-dimensional Input and Visualization, in: Pipes, A. (Eds.), *Computer-Aided Architectural Design Futures*, Butterworths Publication, Delft, 1986, 68-84.
9. Holzer D., Tang J., Xie M., and Burry M., Design Using Evolutionary Optimization and Associative Geometry, in: Martens, B., and Brown, A. (Eds.), *Learning from the Past: a Foundation for the Future*, Österreichischer Kunst und Kulturverlag, Vienna (Austria), 2005, 243-254.
10. Maxwell, J., C., *An Elementary Treatise on Electricity*, Dabor Science Publications, MA, 1977.

Neri Oxman, PhD Candidate
Computation Group, Department of Architecture
Massachusetts Institute of Technology
77 Massachusetts Avenue, room 9-265
Cambridge, MA
02139-4307, USA

neri@mit.edu