Title: **Polytrim: Collaborative Setting for Environmental Design -** Paper Ref:118

John W. Danahy, Rodney Hoinkes
Centre for Landscape Research,
School of Architecture and Landscape Architecture,
University of Toronto,
230 College Street,
Toronto, Ontario, M5S 1A1
CANADA

Link to Images

Abstract

This paper begins with a review of the structuring values and questions the Centre for Landscape Research (CLR) is interested in answering with its testbed software system Polytrim (and its derivatives; CLRview, CLRpaint, CLRmosaic available via anonomous ftp over the internet [1]). The mid section of the paper serves as a guide to Polytrim's structure and implementation issues. Some of the most enduring and significant principles learned from Polytrim's use over the past six years of use in research, teaching and professional practice are introduced. The paper will end with an overview of characteristics that we believe our next generaton of software should achieve. The CLR's digital library on the World-Wide Web provides an extensive set of illustrations and detailed descriptions of the ideas and figures presented in this paper. Endnotes provide specific internet addresses for those that wish to read, see or use the system.

Keywords: dialogue, interaction, collaboration, integration, setting

1 Introduction

Polytrim is a testbed software system intended to amplify one's capacity to undertake research, teaching and practice in the discipline of landscape architecture. To the best of our knowledge, it represents the most comprehensive attempt to integrate the various areas of knowledge and landscape representation in one computational system. The system that will be described is used to explore possibilities for computational media. It is a prototyping toolkit and it serves as a testing ground for exploring new possibilities for environmental design. It combines data types, computational tools and user methods useful to architects, urban designers, planners and environmental scientists concerned with spatial knowledge. Since landscape architecture overlaps with these disciplines, people may find Polytrim a useful model to study. Several universities have made use of Polytrim and over 1200 organizations are using CLRview. CLRview is a public-domain visualization tool made from the Polytrim toolkit [2] .

The toolkit system combines data types and computational tools found in a range of digital design and planning media (Virtual Reality, GIS, CAD, Animation, sound, hyper-links, collaborative-work via internet, image processing, text, digital library tools, data base, interactive exhibitions, customizable interfaces, user history tracking, scripting language and 'C' library support).

2 Building a Town in Grade Two: Prosthesis for Spatial Dialogue

To properly set the stage for this paper, we will review the structuring values and questions CLR is interested in answering with Polytrim.

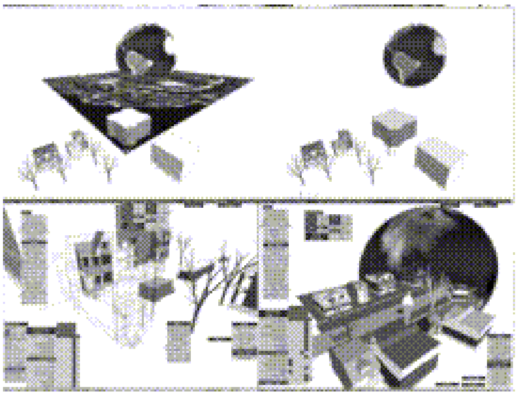


Figure # 1: Images from use in a Grade Two Class

"Wow look at that ".... "bash them together" .... "move it over there"; the sounds of a grade two class playing with my computer and the software we built for urban design. These kids are designing in perspective and animating not just the walk-through but the process of design. They are playing. A kit of parts, a Victorian row house, a nineteen forties bungalow, an apartment, trees, leaves, cars, people and some straight forward "operators" in a quick computer and presto most of the drawing skills I spent years learning to master become unnecessary. The play I experienced as a child and was what really motivated me to start into this profession in grade seven is back and in the same tool I'm using to do professional work. Mind you those years of learning to draw and cut models also helped me develop a facility with the language of design that takes time to mature in any media.

Literacy is not instant. Operating on the alphabet and vocabulary of architectural and spatial form has always made play in conventional architectural design media a difficult and elusive quality for most people. It has acted as a barrier, denying a place in legitimate discussion for the years of spatial experience every layperson has to contribute to design inquiry. Outside the artistically gifted, architecture has been inaccessible. Something we all live in but cannot discuss. By discuss, we mean engage in a dialogue - a spatial dialogue about our homes, communities and favourite places. The high priests of form and science have been the only people capable of discussing Architecture or Nature's landscape.

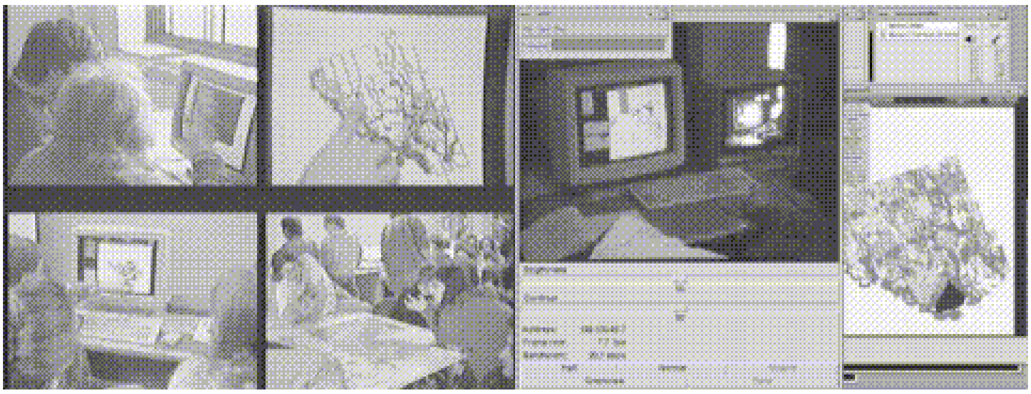


Figure #2 and #3: Seaton Workshop Images of Discussion & Internet Instruction on REX

Our aim in research has been to bridge this gap. To build a spatial media - a robust responsive media - that allows us to commit heresy the way a grade two class does when it has the chance. To make spatial ideas worth discussing. To make it possible for the less practiced to engage in architecture and use that engagement as a basis for challenging architecture to be more meaningful. To connect it to anyone that participates in a culture and its environment.

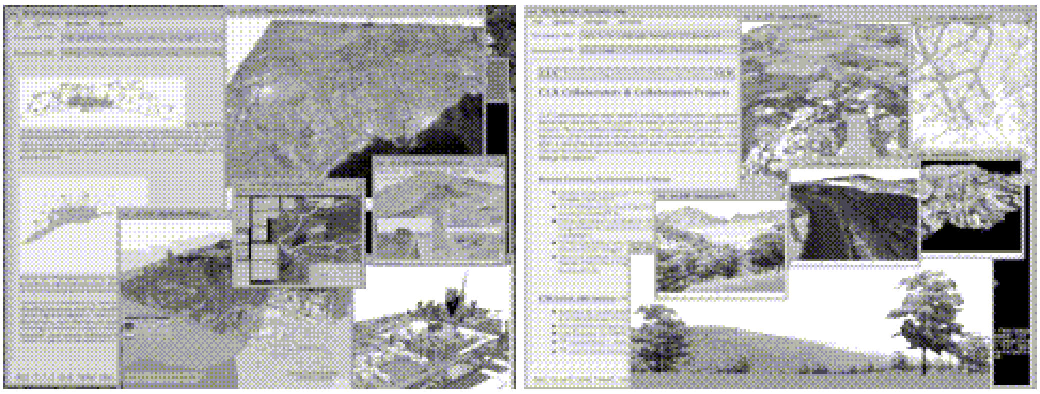
CLR tries to make information technology that provides the prosthetic ability to make more intelligent and responsible environmental decisions. Our experience is in designing, building and adapting information technologies to solve real-world problems in landscape architecture, architecture and environmental planning. A fundamental premise of CLR work is that spatial language must be supported in a computational media accessible to everyone in society if society is to make more intelligent environmental decisions.

For us, the fact that a grade two class can operate on an architectural vocabulary in a playful manner represents an important property that the design media we build should capture. We want a variety of people in society to operate directly on a spatial design vocabulary to express themselves. The expression should be useful and not require graphic giftedness and years of training to "draw". We believe that people should be able to operate (make intelligent pictures) on architectural and landscape vocabularies and engage design experts in dialogue that extends beyond words and diagrams.

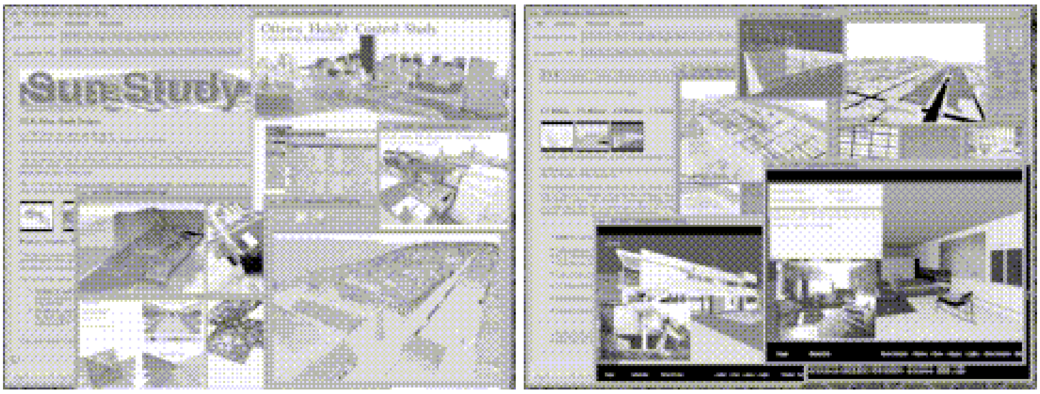
Our hypothesis is that computational spatial media offer the potential for a majority of people in society (ranging from children to professionals) to think about landscape and architecture using spatial language across a range of scales of detail and symbolism.

The Centre for Landscape Research sees its mandate as contributing to the refinement of exisiting practice and the design of new landscape architectural discipline. The language one uses to be intellectual - to think about a spatial design problem - has a tremendous influence on what is thinkable. Computing offers a potential that in the judgement of the authours will lead to a radical transformation of practice in the environmental design disciplines.

Polytrim has been created in an application domain of carefully selected professional and academic projects designed to test these ideas about computing in the discipline of landscape architecture and its allied disciplines. The range of projects are enumerated in the figures that follow [3]

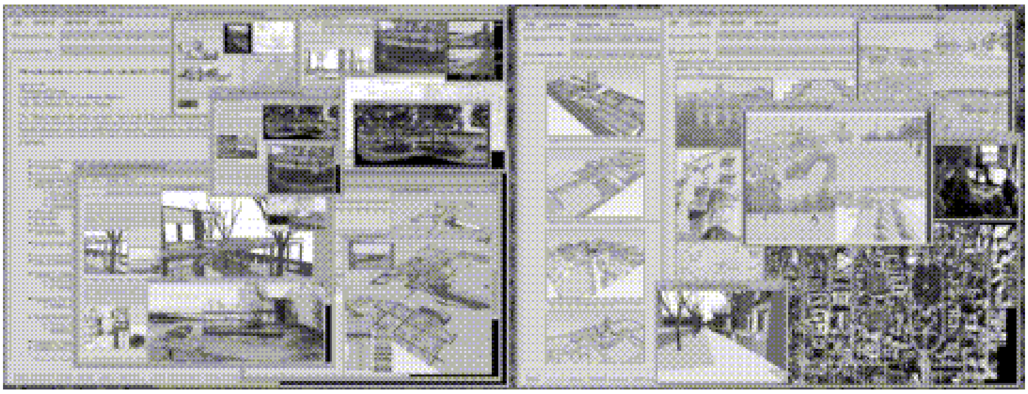


Figures #4 and #5. Research and Practice Projects in Urban and Environmental Planning and Landscape Architecture; Toronto, Burlington, Monroe County, Utah, North Carolina, Rhine Valley.

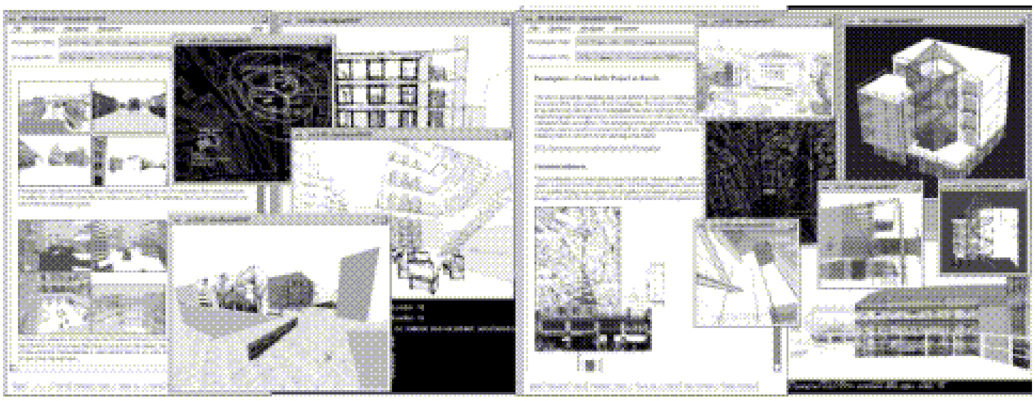


Figures #6 and #7. Research and Practice Projects in Urban Design and Architecture; Ottawa, Toronto, Montreal, Pittsburgh, France.

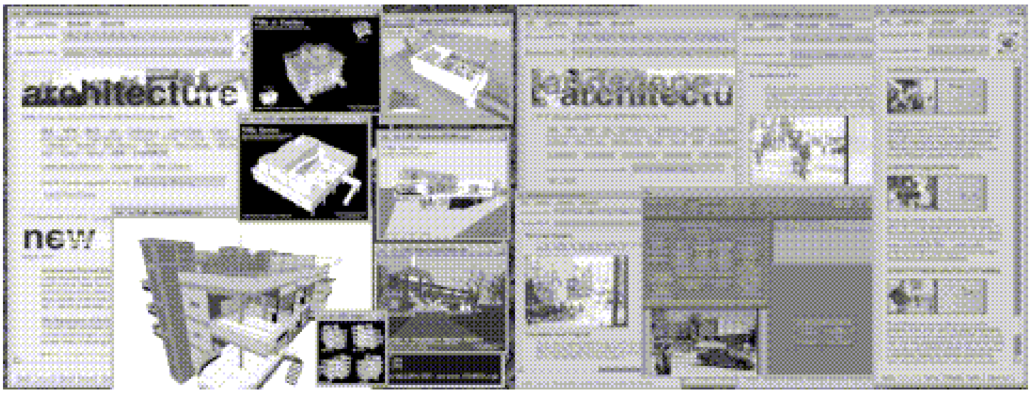
Figures #8 and #9. Collaborative Research and Teaching Projects and Workshops; Architecture Option Studio, Seaton Workshop and Internet Teaching.



Figures #10 and #11. Teaching in Landscape Architecture; Planting Design, Urban Design, Urban Precedents.



Figures #12 and #13. Teaching in Architecture; Conceptual Sketches, Urban Design, and Infill Library Buildings.



Figures #14 and #15. Virtual Libraries; Architecture and Landscape Architecture Resources, Precedent Descriptions and Models, Social Factors, Planting Databases.

---

3 Polytrim : Structure - Strengths and Weaknesses, Opportunities and Trade-offs

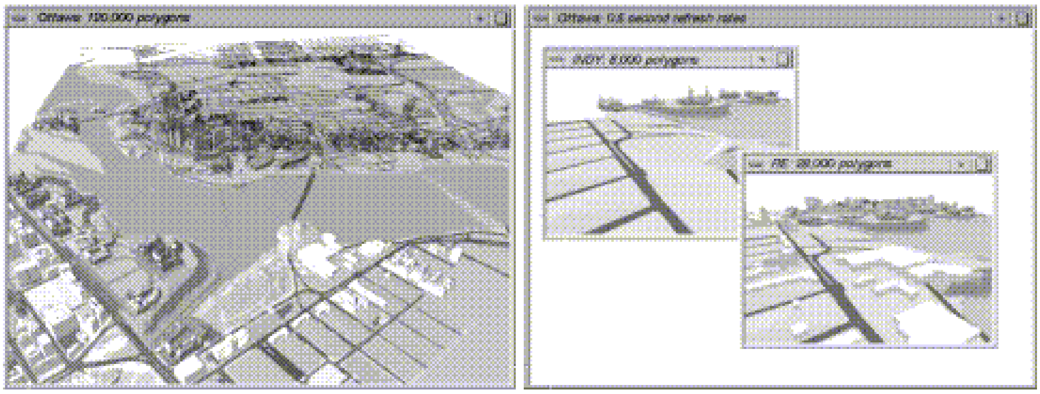
This section of the paper will discuss the technical lessons learned in the course of developing Polytrim for the applications illustrated previously. The section will trace its beginnings as a series of separate programs and discuss structure and implementation issues involved in integrating those original programs in polytrim. The section will also discuss the software engineering principles that made it possible for Polytrim to evolve and be continuously adapted as project, teaching and research demands have changed and new dimensions of computational media are integrated to the system.

The primary ideas of a quick playful environment were initially embodied in the proof-of-concept CLR software programs Display, TUUMS, TRIM, and Polyed by the late 1980's [Danahy, 1990]. Their basic premises of a landscape architectural language and intuitive interactive speeds still hold today but their lack of integration posed a disjunction in the fluid synthesis of ideas. Polytrim was developed to explore the bridging of the gaps left by these separate programs, integrating the disparate modeling and visualization languages of its predecessors. Each of these previous systems shared common interactive characteristics but used vastly different data structures. For polytrim to embody the capabilities of these programs, it initially including all of the different structures, and eventually generalized many of their concepts into a more manageable and extensible system that works across domains of inquiry.

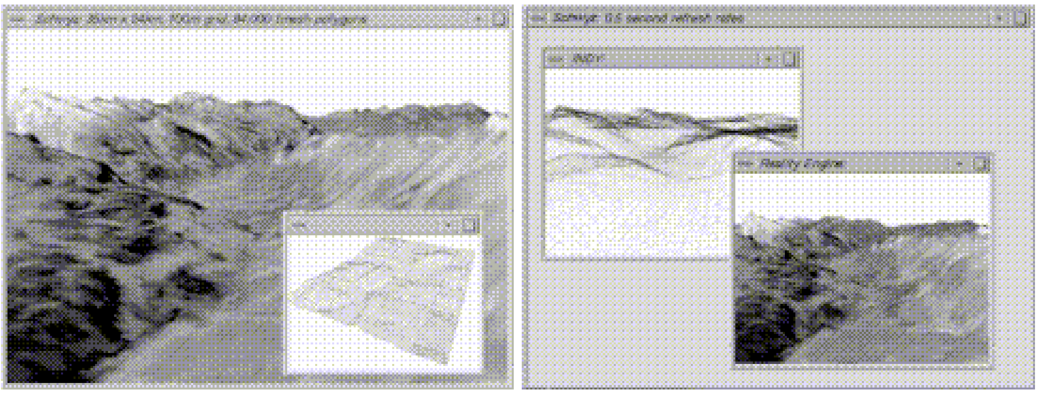
3.1 Quickness: A Function of RAM, Graphics, and Data Structures

At a technical level, the fundamental premise that all tools should operate in a quick manner is accomplished through two primary software development strategies; limiting the data it can work on to that which will fit in RAM, and tightly coupling the representational language with structures for high-performance rendering. Fitting data into RAM does not sound so odd or revolutionary and indeed it is not, but we still find that most CAD and GIS environments do not use this technique. They rely upon virtual RAM strategies which permit them to work with very large datasets, often born out of mapping or construction needs with extreme levels of detail or quantities of data.

Software tends to be designed around its database needs - its language, and then tries to match that to whatever graphics rendering library (Windows GDI, X, OpenGL, etc.) the system will run upon. The opposite approach is a design for maximizing the graphics capabilities while enforcing a new structure upon any input data, often resulting in two structures - a language structure and a rendering structure (Open Inventor, PEX, most VR systems). Polytrim balances on the edge of these, obtaining near-peak performance of graphics while retaining more logical language-based model structures. When all of the data can be found in RAM, no additional delays of transferring from disk are required and a maximum speed transfer of data to the graphics engine can occur. In the case of Polytrim, the graphics engine we have used is IRIS GL(tm) supplied by Silicon Graphics on all of their workstations. For the duration of the development of Polytrim and its predecessors, this graphics library has remained relatively stable and backwards-compatible, allowing development to continue without adapting to new standards every few years, while maintaining peak performance levels on new generations of hardware. Where there are trade-offs between structures for rendering speed versus modeling language, polytrim offers type conversions at the touch of a menu to gain faster response in graphics or modeling.



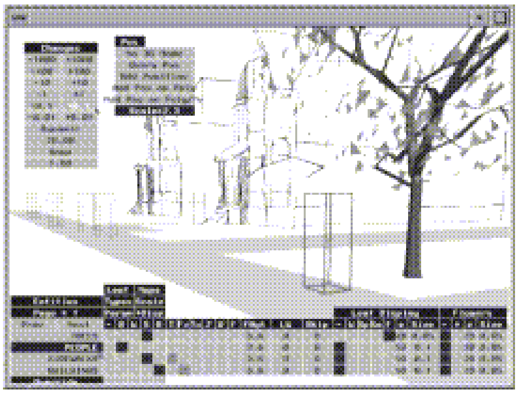
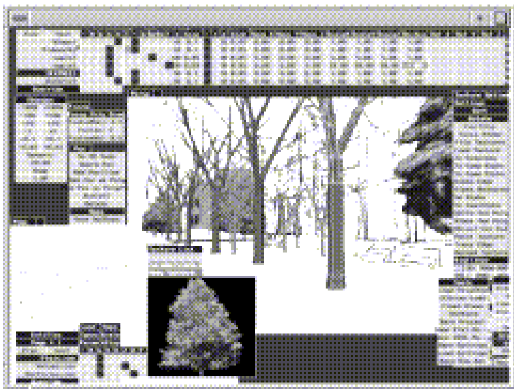
Figures #16 and #17. A large urban dataset of Ottawa (120,000 polygons) and what can be rendered in 0.5sec on an entry-level Indy (8,000 polygons) and a high-end Reality-Engine (28,000 polygons), with normal models, not marketing polygons.



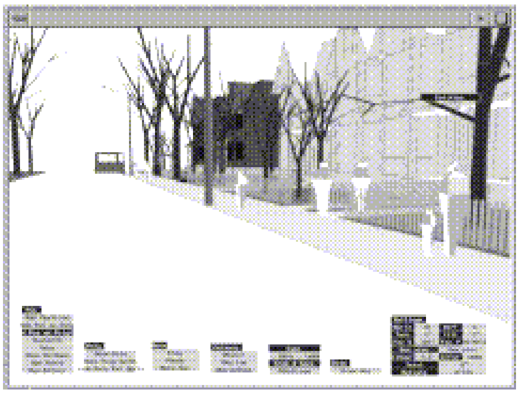
Figures #18 and #19. A large landscape model of Central Switzerland (84,000 meshed polygons), with parametric subdivision techniques, and what can be rendered in 0.5sec on an entry-level Indy (wireframe only) versus high-end Reality-Engine (solid model).

3.2 Interface Strategies to Managing Complexity and Quick Inquiry

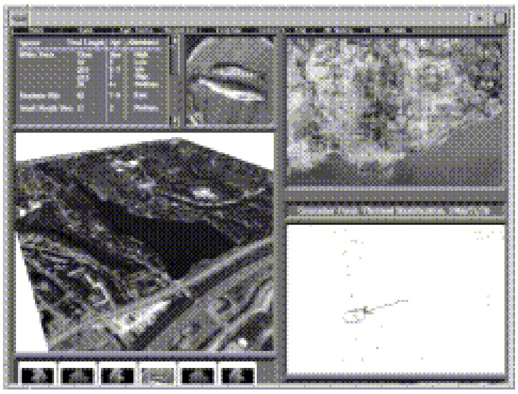
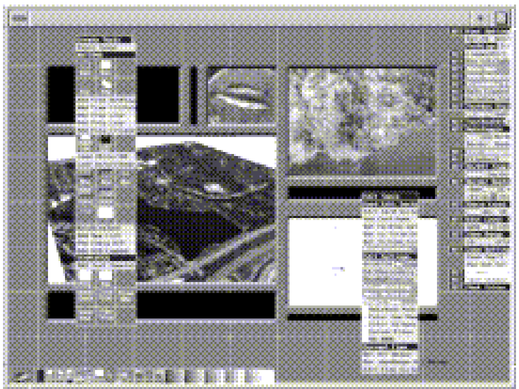
Polytrim allows for many continual trade-offs; faster modeling or faster rendering, better graphics quality or faster interaction, higher precision or quick checks. The user interface is the primary access point to such decisions. Many software programs assume or force you to work in one set of modes due to state changes that can only accessed by 3-level deep dialog boxes or drop-down menus to get at options, making them multi-step processes (Mac/OS, Windows, Motif, etc.). Polytrim follows a different approach through the use of its own interface library. Its interface allows any number of critical variables to remain open to direct manipulation by the user at any moment. Many such concepts are starting to be recognized and utilized in conventional software (Toolbars, Floating Palettes) but still do not maximize the potential benefits of such approaches. For example, attached to Polytrim's UI is a set of dynamic manipulators through a 'Changes Box' which allows for continual and rapid changes of dynamic parameters, including control of any number of parameters at one time. At the level of controlling computational emphasis in a highly-interactive environment, Polytrim relies upon its 'Entities Box'. The Entity Box provides representation control over all entities loaded into the program, allowing for rapid changing of display style (wireframe, solid, outlined) or abstraction (polygons or outer bounding boxes) on each category of information as needed. This allows more detail or clarity in areas under current scrutiny while retaining simpler contextual expressions, or simplifies the whole database to navigate around more robustly until another critical focus is found. Further from the simple display control, the Entity Box controls more detailed display options on a per-entity basis; fuzzy lines, vertex highlights, Z coordinate display, leafs and texture draping; as well as parametric manipulations of the entities; translation, scale, rotation, foliage density and polygon skipping. Finally, as a user finds they need particular capabilities readily exposed for utilization, they can customize the toolkit environment with their own menu setups for effectively controlling the screen real-estate and mental map needed to make them accessible.

Figures #20 and #21. Entity Box control over display mode, fuzzy lines, and parametric leafs, textures, and tree scaling.

Figures #22 and #23. A complex work setting with the tools needed for a streetscaping study as given, and as re-defined by the user for the process at hand.

Figures #24 and #25. An interactive multiple media interface crafted in trimcard as a new work interface, and the interface (trimcard) used to build it.

3.3 Behaviours and Constraints: Managing Landscape Language

From its beginnings, polytrim has supported a number of common modeling behaviours related to landscape architecture. One of the first was the notion that things sit on the terrain - a basic form of gravity constraint. People navigate a landscape, staying at eye height above the terrain, building masses sit into the terrain at their lowest interpolated point, or walls follow the terrain, changing their elevation at each interpolated point. As the tools started to handle more complex building massing, it was necessary to allow tower masses to 'sit on' the podium mass, stacking up and automatically updating their base 'Z' coordinates as the lower built form changed its height.

As a generalized relational database was added to the repertoire of polytrim geometry to handle GIS data and flexible multimedia connections, the ability to attach language (attributes) to geometry in a generalized manner facilitated a number of new possible behaviours in polytrim. By combining the scripting language of polytrim with a flexible encoding language (attributes), users can devise new behaviours to which polytrim responds. To date in the generalized environment, only navigation behaviours are facilitated dynamically, with sensors identifying the geometry below the navigator and ahead of them (at a user-specified distance on the ground as well as vertical surfaces in the distance). From the point of social factors, particularly in the urban landscape, one can simulate how a user's view may change as they pay attention to the icy surface they are walking or slow down if too many other people and objects are around them.

3.4 Rich and Accessible Datasets

As the tools and techniques of Polytrim were explored in a variety of project scales and types, several issues of data richness and accessibility became prominent. While the three and four-dimensional environment was a logical vehicle for exploring design and planning issues, it was still extremely time-consuming to not only model a single project but to adequately represent it's context, both in terms of a three-dimensional model and a wider variety of multiple media representations.

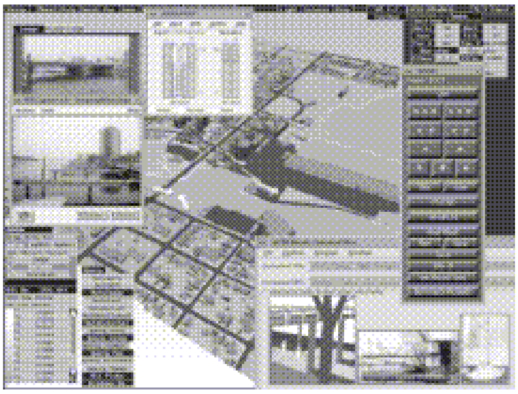
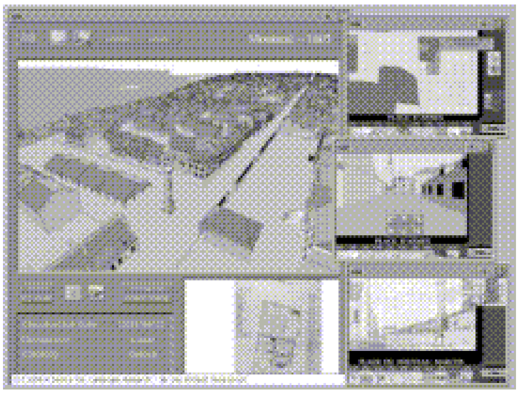
While there have existed any number of 3d modeling tools for some years which have to varying degrees facilitated the crafting of small projects or extrapolating that over large areas, there has been little more significant support for rapidly developing larger three-dimensional contexts, urban environments and larger landscapes. As a starting point, these do not often need the vast continuous detail levels until particular projects or parts are explored more fully. They do however need to provide a useful contextual shell for more detailed work, and while this can be accomplished through existing brute-force techniques (manually crafting everything in the same manner as the detailed area), it is a significant time drain and rarely done for quick smaller projects, only in large-scale work of long-term significance [Grant, 1993]. To facilitate the discussing and representation of larger contextual issues in a rapid manner, a series of techniques for 3d inference have been developed in polytrim. Whereas comprehensive three-dimensional models are still a rarity to obtain at the start of most projects, two-dimensional material is often far more accessible. By applying a small number of attributes (as can and is often done in municipal GIS) to such 2d mapping, one can make inferences about a three-dimensional form:

Element Geometry Attributes

Built Massing polygon FLOORS, FLOORHT
Built Zoning polygon OBJNUM, OBJNAME, OBJDIST
Infrastructure line OBJNAME, OBJDIST
Forest polygon OBJNUM, OBJNAME, OBJSCALE, OBJDIST

This has been successfully integrated with more advanced remote sensing capabilities (for roof profiles, orchard tree identification, etc.) to comprehensively automate the generation of urban and land forms [Hoinkes, R. and Lange, E. 1995]. The further advantage of this approach is that the useful attributes that one can find in many GIS systems are carried through this process into the 3D form.

To expand the range of material polytrim can connect to, each element (polygon, line, point) can have an attribute database. A generic environment evaluates the results of an activation (click, walk-over, script access) of an object and performs the necessary moves based upon its attributes (IMAGE attribute loads a digital image and displays it in an available image viewport, VCRSTART starts an analog video clip playing from a particular time code, etc.). Further from the multiple media types directly supported in trimcard/polytrim (Digital Images, Analog Video from WORM disc and VISCA controllable video tape, and ASCII text), external programs can be used to fill gaps in polytrim's capabilities, by matching particular attribute naming to outside programs; AUDIO connecting to a sound player program, DVIDEO connecting to an MPEG/QuickTime digital movie player, MIDI sending signals to MIDI devices, EXTURL to use polytrim models or trimcard interfaces to communicate over the internet World-Wide Web (WWW) browser (Mosaic, Netscape). While this type of connection may simply appear to be the same as the technique used in hypertext systems (such as those found on the WWW), the difference here is that any number of these sources of information can be accessed at one time, a given building or button element may have images, audio, text, and more information attached to it, and filters can be applied to isolate particular subsets if necessary for particular purposes.

Figures #26 and #27. A multiple media interface to Port Credit, an Urban Design studio, integrating web, video, audio, and digital image resources, and 18th Century Montreal exposing filtered ethnicity, and landuse attributes with three-dimensional reconstructions and historic artist sketches and paintings.

As these rich environments have been developed in a number of projects (as in the CLR/CCA Opening the Gates of 18th Century Montreal and ETH Architecture and CAAD, Eileen Gray projects), there has been a strong desire to make the material accessible to people. The investment in such extensive and expensive datasets often comes from some public endeavour (public projects, funding, or accountability), pressuring the users/developers to make the resultant material available. As this material is then made accessible, the average person must be able to interact with it robustly, to built experience and come to their own understanding of it, not just trust the expert. This has been enabled in our environment through the development of the trimcard interface to polytrim. Trimcard is an over-arching interface to the whole polytrim toolkit. Designers can combine the requisite views (Perspective, Orthographic, Image, Textual, etc.) with their own interface to necessary tools or inquiry modes. Through this approach, one can create a series of filters over the data and operators of polytrim for particular user groups. Attribute filters can show only those types of information that are desired in the interface, and providing as much or little control as is desirable. This interface can be provided in exhibitions with a variety of interfaces (touch screen, mouse, or spacemouse/ball).

3.5 Interaction in 3D

So far we have discussed the technical fundamentals that have enabled polytrim to behave in a robust manner at a graphics and interface level, and introduced the first of the foundation modeling concepts it includes - behaviours. With this environment, we have achieved the capability to model rapidly and somewhat logically. To carry the concepts of spatial design further, these techniques are not only good enough for modeling in two and visualizing subsequently in three dimensions, we need to be able to then interact in 3D, to adjust aspects of the model so they work correctly in a closer approximation to our real senses, and provide a more nature vehicle for dialogue (as explored in the grade two class). Polytrim has a number of direct three-dimensional interactors, allowing for the placement or displacement of design forms, application of facades, and the sketching of ideas on terrain meshes. These tools take as a basis the contextual models developed through the 3D inference tools, combined with some logical behaviours (how things sit on the terrain or each other), and a quick 3D rendering system. These tools are relatively simple at this time; move, scale, and rotate a kit of parts (trees, houses, cars, people), add or remove floors from building massing, or begin to explore sketching on meshed terrain surfaces. The hope is that we can facilitate new forms of dialogue directly in three dimensions for a wider audience, allowing them to interact with a design vocabulary and providing a series of common behaviours (sits on terrain or building) to simulate burdensome relationships (i.e.. interpolating Z values).

3.6 Collaborative Polytrim

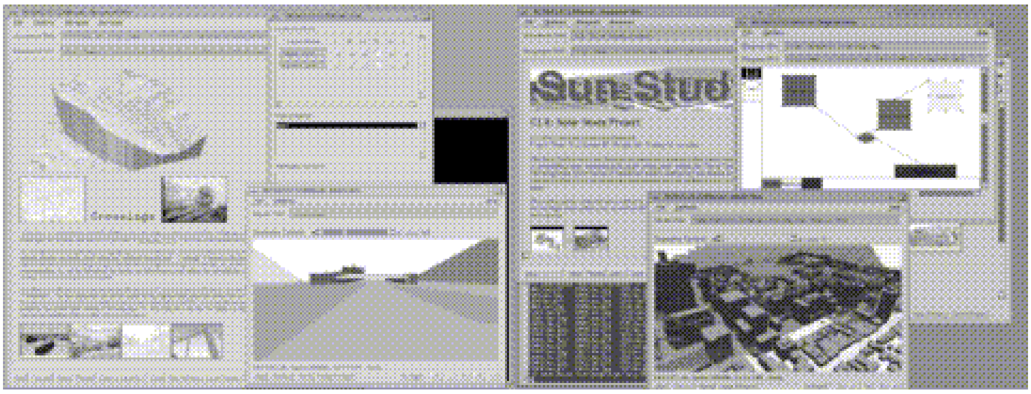
As the projects undertaken with polytrim have expanded in collaborative scope, there has been a strong impetus to continue the active types of discussions produced while groups surround a machine to groups in front of other machines throughout the world. In the spring of 1991, a group of students in Computer-Supported Cooperative Work (CSCW) at the University of Toronto did a study of the CLR. In this study they examined the ways in which we collaborated with colleagues who were using polytrim around the world to work on projects. The result of this study was that collaborative design efforts suffered due to the indirect references typically provided on the internet (email, ftp of images) and could be assisted by direct synchronous reference to datasets. An initial implementation of techniques by which multiple participants could share in the navigation of three-dimensional models over the internet was developed, building on top of the scripting language structure in polytrim. This has been explored and expanded in the meantime, providing not only shared views but also individual control (with the option to reference someone else's view at any moment) and combined with videoconferencing software as a basis for student presentation and proposed methods for distance-based professional work. While the ability to navigate collaboratively is interesting, it is not compelling enough on its own to support dialogue. These network techniques are being expanded to engage in dialogue over changing designs, particularly activating the 3D tools for object manipulation through the script server, with the hope that a richer medium for testing these ideas will lead to clearer ideas of the needs of synchronous design collaboration, not mearly presentation.

The collaborative system enabled by the script server is not the only fundamental collaborative capability developed with polytrim, in fact the development itself has fostered a level of research code-collaboration. In the late 1980's and early 1990's, two code-level collaborations where undertaken within polytrim (and its immediate predecessors), one with Steve Kleinman under Mark Lindhult at the University of Massachusetts, exploring additional site modeling tools, and the CSCW project described above. Each of these proved a daunting task at each end, explaining a complex code made up of many disparate dedicated structures and trying to figure out where to integrate new ideas. In the years that have passed the methods of integrating code and the generalization of many code structures in polytrim has enabled a new level of code collaboration. This is now routinely used internally as a way of isolating new code under development (so conflicts with core code are minimized), and to test their robustness. These techniques are now starting to be explored by others and have resulted in the recent rapid integration of new terrain 'bulldozing' code developed by Caroline Westort (University of Zurich), requiring only a few hours to convert and integrate. It is our hope that such efforts will help expand the capabilities of the system, and assist in clarifying the general structures such a system needs to support over a wider collaboration notion of design research.

4 Developing Properties: Summary and Future Directions

In the process of the evaluation of polytrim over the past six years, we have observed a number of things more clearly. It should however be evident that there has been a progression from a series of more dedicated and symbolic computational tools, through those that are dedicated but start to behave on real form and language, finally toward generalized real tools. Polytrim has not nearly reached the end of this evolution but we have continued to look forward at what may be generalizable and as it stretches there, a look back is continually made to evaluate its quickness - robustness against the initial proof-of-concepts that begot it. We have observed that it is rare to find the robustness we initially explored within dedicated systems in the more generalized system and further research on this must continue to evaluate the balance and be cognizant of the mental overhead we introduce.

On a very positive note, we have found that we do not need to give up interactive characteristics that facilitate design when we enter other areas of computational application, i.e. GIS, multimedia, and hypermedia. GIS data queries can be performed on most design and planning projects we deal with at interactive speeds [Hoinkes, R. and Mitchell, R. 1994], and multimedia and hypermedia should not only be thought of as presentations of pre-generated material, they have a potentially larger role in filtering and associating material and tools needed in collaborative work environments.



Figures #28 & #29. "Crossings" interactive hypermodel of a landscape and design interventions contributed by artists, architects, and landscape architects around the world with connections back to net-based resources. CLRMosaic development of a net-capable asynchronous hypertext, hypermodel, and process diagramming environment for the communication of design concepts in a more comprehensive manner.

At a more fundamental scale, we have come to see that as one is able to rapidly and dynamically explore alternatives, by one's self and increasingly with others over networks, being able to track ones' processes is essential, who made what decisions, when, based upon which authorities, and with what representations. We see that a spatial diagrammatic interface to the design process is required, both to follow and organize what one is doing based on fundamental ideas and relationships. Increasingly, this must also work, and can provide the greatest benefits to collaborative work across networks, where product representations are the primary communication form right now, not the questions and issues that developed that representation.

Acknowledgments

We would like to acknowledge the contributions of the numerous colleagues that have contributed directly to the development of polytrim or explored its potential in their research and teaching projects. At the Centre for Landscape Research, Shannon McKenzie, Stephen Bohus, Ed Broeders, Prof. Robert Wright, Prof. Ed Fife, and Prof. Stacey Spiegel; at the Canadian Centre for Architecture Montreal Research Group, Russ Mitchell, Alan Stewart, and Jennifer Waywell, at the Harvard University Graduate School of Design, Prof. Stephen Ervin, Prof. Carl Steinitz, and Prof. Malcolm McCullough, at the Swiss Federal Technical Institute Zurich (ETH) ORL Institute Prof. Willy Schmid, Sigrid Hehl-Lange, and Eckart Lange, Architecture and CAAD Prof. Gerhard Schmitt, Bharat Dave, Stefan Hecker Kitsios, and Eric van der Marc, at TU Delft, Christian Mueller, at the University of Toronto CSCW group, Prof. Ron Baeker, R. Hauser, E. Karagiorgos, D. Nastos, and M. Stede, at the University of Zurich, Geography Department, Caroline Westort, and at the University of Massachusetts, Prof. Mark Lindhult and Steven Kleinman.

Endnotes

1. CLRview is a subset of the visualization tools and techniques of polytrim, used to integrate CAD and GIS sources. CLRpaint is the image processing component of polytrim, also available as a separate environment. CLRMosaic is an integration of 3D visualization with hypertext to form an interactive hypermedia environment. More information is available via WWW at: http://www.clr.toronto.edu/software.html and they can be obtained by FTP at: ftp.clr.toronto.edu in /pub/sgi.
2. Between June 1992 and June 1994, over 1200 FTP transfers of clrview occured from our anonymous FTP server. Current statistics indicate ~50 new FTP's of clrview occur each month.
3. CLR professional and research projects can be found through the CLRnet Home Page: http://www.clr.toronto.edu/clr.html by looking in 'CLR Projects', 'Research', and 'Projects with Associates'. Teaching efforts utilizing polytrim and other CLR software can be found at http://www.clr.toronto.edu/teaching.html

Bibliography

Danahy, John W. "Irises in a Landscape: An Experiment in Dynamic Interaction and Teaching Design Studio", The Electronic Design Studio, (Cambridge, MA: MIT Press, 1990); 363-375.

Grant, Mike. "ISSUE: Interactive Software Systems for the Urban Environment", CAAD Futures 1993, (New York: North-Holland, 1993); 557- 564.

Hoinkes, R. and Lange, E. 3D for Free - Toolkit Expands Visual Dimensions of GIS, (GIS World Inc., July 1995); 54-56.

Hoinkes, R. and Mitchell, R. "Playing with Time: Continuous Temporal Mapping Strategies for Interactive Environments," 6th Canadian GIS Conference, (Ottawa: Natural Resources Canada, 1994); 318-329.