Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

GENERAL-PURPOSE 3D ANIMATION WITH VITASCOPE

Vineet R. Kamat

Department of Civil and Environmental Engineering
University of Michigan
Ann Arbor, MI 48109-2125, U.S.A.

Julio C. Martinez

Department of Civil and Environmental Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0105, U.S.A.

ABSTRACT

This paper presents VITASCOPE, a general-purpose, user-extensible 3D visualization system for animating processes that are modeled using Discrete-Event Simulation tools. VITASCOPE is an ASCII stream driven 3D animation tool that runs readily on a wide variety of hardware platforms ranging from commonly used desktops and laptops to high-end, room-sized, immersive virtual environments such as the CAVE. VITASCOPE parses and interprets an ASCII stream, and imports and manipulates existing 3D CAD models of the simulated system entities to visually depict modeled processes. The used CAD models may be created in any CAD modeling package capable of exporting models in the VRML format. VITASCOPE's architecture is open and loosely-coupled, thus allowing it to serve as a post-processed or concurrent 3D visualization engine for simulation models created in a wide variety of languages and tools. VITASCOPE is built on the industry standard OpenGL graphics library and runs on Windows and Irix based workstations.

1 INTRODUCTION

In the world of operations analysis and design using Discrete-Event Simulation (DES), the ability to see a 3D animation of an operation that has been simulated allows for three very important things: 1) The developer of the simulation model can make sure that there are no errors in the coding (Verification); 2) The domain experts and decision makers can discover differences between the way they understand the operation and the way the model developer understands it (Validation); and 3) The model can be communicated effectively which, coupled with verification and validation, makes it credible and thus used in making decisions.

This paper presents VITASCOPE, a general-purpose, user-extensible 3D animation system for visualizing simulated processes in smooth, continuous, 3D virtual worlds. Using an ASCII animation trace and CAD models of simulated system entities, VITASCOPE recreates simulated processes with chronological and spatial accuracy in 3D. VITASCOPE can animate processes modeled in any general or special-purpose simulation tool that is capable of writing out formatted text output during a simulation run. VITASCOPE is thus independent of any particular simulation tool. In addition, the system is independent of any particular CAD modeling tool and can use 3D models created in any CAD package to visualize simulated processes.

2 THE VITASCOPE VISUALIZATION SYSTEM

VITASCOPE describes animated 3D worlds that show how simulated operations are carried out, using simple text statements and references to existing 3D CAD models. VITASCOPE's simple text animation description language is meant to be written out by end-user programmable software such as DES systems, and allows a computer to create a world that is accurate in time and space; and which shows people, machines, and materials interacting as they perform the simulated processes.

Modeled operations are visualized by processing sequential, time-ordered animation commands written in the VITASCOPE language. The animation commands are contained in an ASCII text file hereinafter referred to as the animation trace file or simply the trace file. Any simulation software capable of writing custom text output during a simulation run can generate trace files automatically. These include most of the programmable generic and special-purpose simulation languages and systems as well as high-level programming languages such as C and C++. Non-language based simulation tools may also be adapted to generate trace files during a simulation (Henriksen 1998).

VITASCOPE uses 3D models of all pertinent resources and system entities to depict the simulated operations and evolving products in 3D. VITASCOPE does not possess any built-in 3D model building capability. Instead, required 3D models of system entities can be imported from a wide variety of 3D CAD modeling software tools. VITASCOPE provides direct support for the VRML CAD file format. Geometry files from practically any 3D CAD modeling program can be easily exported or converted into VRML format.

VITASCOPE can be used to animate modeled operations either in post-processor mode (where the animation runs after a simulation run is complete) or in concurrent mode (where the animation runs as the simulation is running). The VITASCOPE visualization engine has the following features:

- Maintains an independent animation clock, the speed of which can be controlled as desired by a user or a concurrently running simulation model.
- Allows a user to navigate freely in 3D virtual space and place himself/herself at any desired vantage point by controlling the camera using a keyboard or mouse.
- Allows a user to jump ahead (fast-forward) or back (rewind) to any desired simulation time.
- Allows a user to start and pause an animation at any time to make static observations of the system.

3 THE GENERAL-PURPOSE APPROACH

VITASCOPE is a straight-line language. In other words, VITASCOPE language statements that describe a dynamic visualization are only processed sequentially. The language needs no flow control (i.e., while or for loops) because that is achieved by the intended generating processes (e.g., a simulation model while it runs).

The designed 3D animation language is implemented in a corresponding Virtual Environment (VE) application. The VITASCOPE application sequentially 1) Parses and interprets individual animation language statements, 2) Executes appropriate computer graphics algorithms in response to the interpreted statements, and 3) Manages the dynamically evolving CAD database to represent the communicated operation dynamically in a 3D virtual world. To communicate a dynamic visualization, external processes use VITASCOPE's parametric language statements to 1) Control the simulation time and speed, 2) Define multiple motion trajectories that constitute resource movement paths, 3) Create scene objects (terrain, equipment, materials, machines etc.) by instantiating pre-created CAD models, 4) Define behavioral properties for instantiated scene objects, and 5) Manipulate scene objects to describe their dynamic behavior.

3.1 Loosely-Coupled Interface

Figure 1 presents the schema that describes the relationships between the VITASCOPE language, the VE application, computer graphics facilities (algorithms, data structures etc.), and external software authoring processes (i.e. simulation models). Simulation models generate information that is required for and sufficient to describe the modeled processes in a dynamic 3D virtual world. The conceptual mechanics of how VITASCOPE converts quantitative information communicated by external processes into dynamic smooth motion are relatively straightforward.

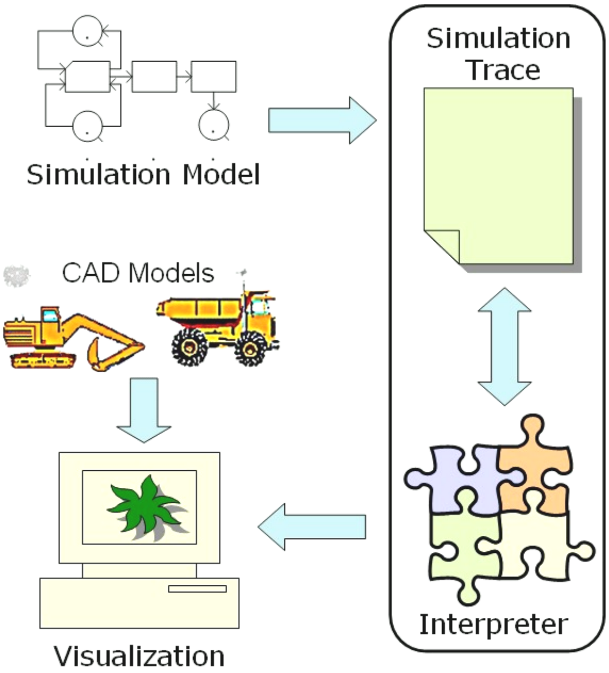

Figure 1: VITASCOPE Animation Schema

3.1.1 ASCII Input Streams

The VITASCOPE VE application reads and interprets animation statements (in the syntax of the VITASCOPE language) that describe, chronologically (and sequentially), the things that happen in a modeled operation. The animation statements representing static and dynamic events are processed sequentially by the visualization engine. In the post-processor mode, all the animation statement are prerecorded in an animation trace file. In concurrent mode, the statements are generated and processed concurrently.

The animation events can be communicated by any external software process capable of generating formatted text output as it runs. Such an external process (e.g. a DES model) can be instrumented (i.e. programmed) to dynamically generate and/or write the animation statements while it runs. The communicated VITASCOPE animation statements can refer (by filename) to existing 3D CAD objects created with other tools for this purpose.

3.1.2 References to 3D CAD Models

VITASCOPE can read CAD files in many different file formats including VRML. By using the existing CAD models and interpreting the animation instructions communicated by external processes, VITASCOPE recreates a virtual world representation of the modeled operations. This is achieved by invoking appropriate computer graphics algorithms and routines and applying them to manipulate appropriate CAD models in a 3D virtual world.

3.2 Advantages

VITASCOPE specifies an open description to describe dynamic processes and implements an interpreter that invokes appropriate computer graphics facilities to represent the described processes in a smooth, continuous 3D virtual world. By doing so, VITASCOPE introduces a tangible interface or layer of abstraction that effectively separates dynamic visualizations from the simulation models that can generate them. This has a very distinct advantage.

Tight coupling between a simulation tool and a visualization engine requires that the simulation model be created only using the native system's simulation engine, which in many cases might not be suitable for modeling processes in the modeler's domain. Furthermore, this also compels model developers desirous of animating their models in 3D to learn and use a different simulation tool than the one they are proficient with. For instance, a GPSS user intending to visualize simulated processes in 3D would have to entirely recreate (i.e. recode) her GPSS simulation models in the language of a different, 3D animation enabled simulation tool. This is a big disadvantage since the time and effort invested by typical modelers in achieving proficiency in a particular simulation tool of choice is phenomenal.

Tight coupling between the VITASCOPE visualization engine and a particular simulation tool would have compelled modelers who desired to visualize their models (created in their system of choice) to learn and use the simulation tool with which the visualization engine was coupled. Loose coupling between simulation systems and the VITASCOPE visualization system thus allows 3D animations of modeled processes to be generated from any end-user programmable simulation tool.

4 CREATING 3D ANIMATIONS IN VITASCOPE

VITASCOPE graphically illustrates modeled operations based on logged simulation model runtime data. Simulation models generate VITASCOPE animation trace files as they run. The syntax of the VITASCOPE language had to be simple enough to allow simulation models to be easily instrumented to generate the trace files. At the same time, the language command set had to be sufficiently expressive to enable trace file statements to describe the dynamics of arbitrarily complex modeled operations.

Table 1 lists a few key animation commands in the VITASCOPE language and provides a concise explanation of their functionality. VITASCOPE trace file statements can span multiple lines with arguments separated by white space. Arguments that include white space must be enclosed in single quotations. A statement ends with a semicolon. Comments can be placed in trace files by making the first non white space character after a statement a "/". The comment continues until the end of the line.

Table 1: Usage of Selected VITASCOPE Statements

Statement	Functionality
TIME	Indicates the simulation time at which all subsequent commands take place.
CLASS	Associates a class of simulation entities with their geometric description contained in a CAD file.
CREATE	Creates specific simulation objects by instantiating predefined classes.
PLACE	Places simulation objects at particular locations or at the beginning of resource movement paths.
SET CLASS… FORECLEARANCE	Specifies the minimum distance to be maintained between two following objects of the same class.
MOVE	Simulation objects begin moving on resource movement paths at the time specified by a preceding TIME statement.
ROTATE	Simulation objects begin rotating along specified planes at the time specified by a preceding TIME statement.

VITASCOPE language commands can be broadly classified into four categories, i.e. System, Scene-Building, Property-Setting, and Motion-Depicting Commands. TIME is the primary VITASCOPE system command. The TIME statement keeps track of the simulation time during visualizations by indicating the instant at which all subsequent commands until the next TIME statement take place. Every discrete event that is recorded by a simulation model in a trace file will have a preceding TIME statement to indicate the simulation time at which the event took place.

The Scene-Building commands set up the visualization and manage the initial and dynamic creation and destruction of simulation entities. The CLASS commands identify the CAD files that contain the geometric representation of simulation entities. CREATE commands subsequently create specific simulation objects at various times in the simulation by instantiating predefined classes.

The Property-Setting commands allow the specification and manipulation of certain physical properties for simulation entities at both a class and object level. For instance, the SET CLASS FORECLEARANCE command enables the authoring simulation model to specify the minimum distance to be maintained between two following objects of the same class during visualization (e.g. trucks traveling on the same path). The SET OBJECT FORECLEARANCE similarly permits this property to be specified on a per-object basis if necessary.

The Motion-Depicting commands form the core components of the VITASCOPE animation language. These commands depict the dynamic state of the simulation models. For instance, the MOVE command moves instantiated simulation objects on movement paths to describe the motion of resources and other simulation entities. ROTATE is another primary Motion-Depicting command that allows the depiction of rotation of simulation entities during visualization.

A portion of a sample trace file written in the VITASCOPE language is presented below. The results of processing these statements can be easily explained by looking at Table 1 and the trace file segment.

```
PATH LoadToDump (5,2,0) (2,2,3) (-1,2,4);
CLASS Truck A30C.wrl;
SET CLASS Truck FORECLEARANCE 5;
TIME 0.00;
CREATE Truck1 Truck;
PLACE Truck1 ON LoadToDump;
TIME 12.00;
MOVE Truck1 LoadToDump 24.00;
```

A path, "LoadToDump", is defined by specifying the beginning, ending, and all intermediate 3D coordinates of the points constituting the path. The example defines a path consisting of two segments. Next, a class, "Truck" is defined by the CAD file A30C.wrl. The file A30C.wrl contains the 3D model of a truck in VRML format. The next statement then specifies the fore clearance for objects of class "Truck" to be five units (meters in this case). This ensures that all objects of that class maintain a minimum distance from the leading object when following or accumulating behind other objects on the same movement paths.

As soon as the animation commences (i.e. at time 0.00), one truck conforming to the geometry defined by class "Truck" is created and placed at the beginning of path "LoadToDump". Further processing of the trace file stream is suspended until simulation time 12.00 is reached. At this time "Truck1" will start moving on the path "LoadToDump" and will require 24.00 simulation time units to complete the journey. During the animation, the ratio of simulated time to viewing time, known as the viewing ratio, is maintained at a user-specified constant value. For example, with a viewing ratio of 6, the animation will show "Truck1" at the beginning of path "LoadToDump" for 2 seconds and then traversing the path for 4 seconds.

In general, VITASCOPE is a CPU and graphics intensive application. It is built on top of the Open GL graphics library. The animation performance attained on any machine (i.e. the frame rate) depends on several factors such as the complexity of the scene and the used CAD objects, the number of textures used and visible in the scene at any given time, the number of simulation objects in motion at any given time, and the processing power and 3D graphics acceleration capabilities of the host machine. VITASCOPE was designed, implemented, and tested on a machine that had a 1.8 GHz CPU and a NVIDIA graphics card with 32 MB of video RAM; a configuration that is commonplace if not the bare minimum on most computers available today. In all our animations on this machine, a performance of at least 30 fps was achieved. The same animations, when run on a workstation class PC with a 3.2 GHz processor and 256 MB of video RAM achieved a frame rate equal to the native screen refresh frequency of the machine (60-70 fps).

5 EXTENSIBILITY

VITASCOPE's animation methods are implemented in an extensible framework that allows the 3D animation language to be seamlessly extended by others. New animation statements can be designed and implemented into the framework without having to understand or modify the implementation of the existing animation language set. This is possible through VITASCOPE's Application Programming Interface (API). As depicted earlier in Figure 1, each animation statement communicated by a simulation model is interpreted by VITASCOPE's interpreter and appropriate computer graphics algorithms are invoked to visually depict the modeled processes. The invoked algorithms and routines manipulate appropriate CAD models that represent simulation entities after instantiating them in a 3D virtual world. For each statement that comprises the 3D animation language, the implementation's interpreter provides code to decipher the statement's parameters and convert them to a format that can be communicated to the underlying computer graphics algorithms and routines.

In order to make this schema extensible, we designed methods that allow others to use the provided API and seamlessly augment the interpreter's vocabulary without having to modify or fully understand the mechanics of the existing interpreter components. Moreover, while core VITASCOPE statements directly invoke computer graphics algorithms, extension statements can choose to invoke concatenated sequences of the core statements to describe modeled processes in a vocabulary that is at a higher level than elementary geometric transformations. This approach is described schematically in Figure 2.

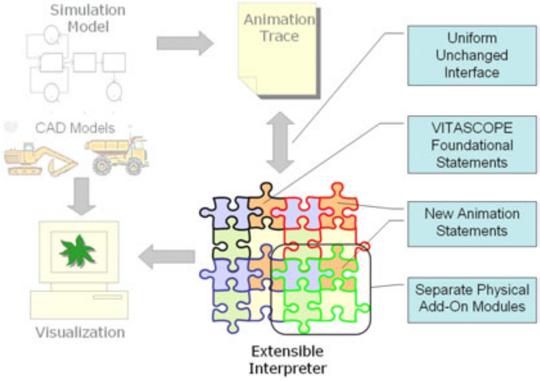

Figure 2: VITASCOPE's Extensible Animation Scheme

Others can design new animation statements and provide code to interpret those statements (represented as additional jigsaw puzzle pieces in Figure 2) without having to modify or understand the code that comprises existing interpreter components. When simulation models communicate events using the new designed statements, the implementation's interpreter automatically invokes the correct segment of code in the correct physical software module.

5.1 Validation of Extensibility

VITASCOPE's add-on interface has been used to design and implement several extensions to the core animation language. The extensions have been implemented as software tools that allow modelers to visualize several simulated processes that could not be animated using only the core language set. We provide here only a brief description of selected add-ons (extensions) designed using VITASCOPE's add-on interface. The purpose of the discussion is to highlight the nature of the useful higher-level 3D animation capabilities that can now be implemented incrementally by capitalizing on the work already in place.

5.1.1 Example 1 - ParticleWorks

VITASCOPE's geometric transformation-based core animation language does not contain statements that can describe modeled processes that involve use of unstructured, fluid materials. Numerous unstructured materials generally capable of flowing (e.g. concrete, dirt, mortar, sand, slurry, and water) are however together central to many industrial and construction processes.

In construction for instance, common processes such as dumping dirt, distributing water, dewatering caissons, placing concrete, sandblasting, and slurry-wall construction cannot be realistically visualized in smooth, continuous 3D worlds unless methods to accurately represent dynamic volumes of the involved unstructured materials are designed.

The ParticleWorks add-on for VITASCOPE implements efficient methods that modelers can use to visualize such simulated processes involving fluid, unstructured, materials that are generally capable of flowing. The work capitalizes on a classical computer graphics concept called particle systems and VITASCOPE's add-on interface to design simple, parametric text methods to represent arbitrary dynamic volumes of fluid materials in 3D virtual environments.

Modelers can use these add-on defined statements to instrument simulation models to automatically generate dynamic visualizations of any modeled operations that commonly handle and process fluid materials. An animation snapshot of a dewatering process described using ParticleWorks statements is presented in Figure 3.

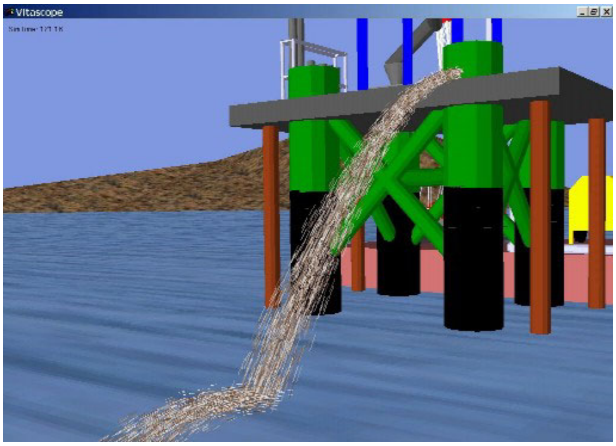

Figure 3: Animation Snapshot of a Dewatering Process

5.1.2 Example 2 - C-Collide

C-Collide is an add-on designed for VITASCOPE that modelers can use to identify and report any and all undesirable conflicts that can occur among static (e.g. structure in-place, idle equipment), dynamic (e.g. active machines and workers), and abstract (e.g. hazard or protected spaces) simulation entities in dynamic 3D process visualizations. Common types of clashes that can occur in real systems and that C-COLLIDE can identify beforehand in process visualizations include 1) intersection among physical in-place components (i.e. design interferences), 2) intersection among in-place components, components in transit, and/or pieces of moving equipment (i.e. process interferences), 3) craft interferences and accidents e.g. collision between two pieces of equipment operating in the same area, and 4) space intrusions e.g. any resource (worker or equipment) encroaching arbitrarily shaped hazard or protected areas of the jobsite or factory.

C-Collide capitalizes on the VITASCOPE add-on interface and advanced documented algorithms for efficient collision detection between arbitrarily moving 3D geometric objects to design mechanisms for interference detection, control, and response in 3D process visualizations. The add-on's interference detection capabilities dynamically check each motion of VITASCOPE scene objects to determine if any pairs of scene objects interfere undesirably. This provides modelers with a lucid understanding of all interferences in any activity of an animated system.

5.1.3 Example 3 - KineMach

KineMach implements "smart" pieces of articulated construction equipment that can be instantiated and manipulated using simple text statements in a higher-level, contextual, construction terminology. Currently implemented generic pieces of equipment include a tower crane, a crawler mounted lattice boom crane, a crawler mounted backhoe, and a highway dump truck. Modelers can use KineMach statements to instantiate multiple pieces of this equipment and instruct them to perform construction processes using a high-level construction work terminology.

KineMach is implemented as an add-on to VITASCOPE. It capitalizes on robust forward and inverse kinematics algorithms from robotics literature and VITASCOPE's flexible add-on interface to design high-level statements for virtual equipment instantiation and manipulation. KineMach's statements provide engineers with a construction terminology to visually describe construction processes. In addition, KineMach greatly simplifies the process of creating the simulation models intended to automatically author visualizations, making 3D simulation-driven process animation more practical. Figure 4 presents two KineMach described cranes at work on a virtual bridge construction site generated by a DES model.

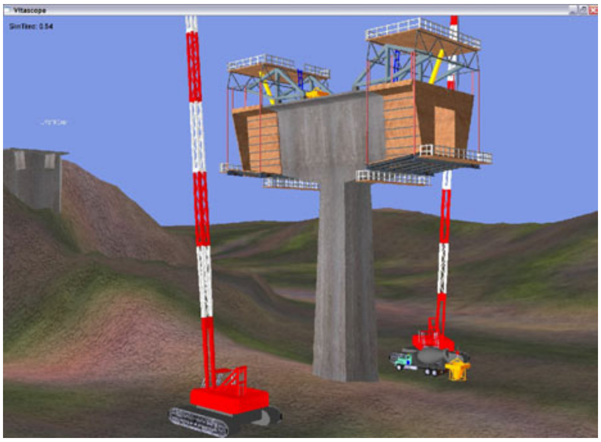

Figure 4: Animated Bridge Construction Processes

5.1.4 Example 4 - ExcelWorks

The ExcelWorks add-on for VITASCOPE is a tool designed to allow modelers to juxtapose dynamic displays of quantitative, numerical simulated operation data alongside 3D view ports during visualization. Within a 3D virtual world that presents a smooth, continuous operation, the dynamic display of relevant quantitative information can be useful and, at times, critical in conveying critical statistics about the animated operation. The dynamic quantitative information that can be displayed during animation includes the numerous simulation run-time statistics that are maintained by a running process model.

ExcelWorks capitalizes on VITASCOPE's add-on interface and the OLE automation features of MS Excel to design a text statement-controlled dynamic charting tool. Modelers can use ExcelWorks-implemented statements to define and update a dynamic graphical display of charts within an MS Excel window. Excel's automation server allows VITASCOPE to take control of its sheets and display any user-defined information graphically during visualization. Figure 5 presents a snapshot of an animation with ExcelWorks in action.

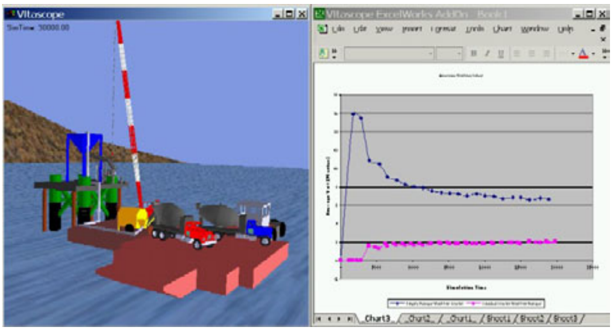

Figure 5: Animating Quantitative Data in Charts with the ExcelWorks Add-On for VITASCOPE

The VITASCOPE add-ons described above have all been implemented as software tools on both Windows and Irix (except ExcelWorks) platforms. Their implementation required no knowledge or modification of any VITASCOPE internal design features or its implementation code. The work required only an understanding of VITASCOPE's add-on interface and competence in the C++ programming language in addition to relevant domain knowledge. It is a mere coincidence that we, having designed and implemented VITASCOPE and its add-on interface ourselves, were completely familiar with its internal design and implementation.

6 SUMMARY

VITASCOPE is a general-purpose, user-extensible 3D animation system for visualizing simulated processes in smooth, continuous, 3D virtual worlds. VITASCOPE is not tied to any particular simulation language or application nor does it depend on any specific CAD modeling tool. VITASCOPE's open and loosely coupled architecture makes it possible to be used in animating processes modeled in any general or special-purpose simulation tool capable of writing formatted text output during a simulation run.

Such interaction can, at the discretion of the simulation modeler, be in post processed or concurrent mode. In addition to a powerful, expressive 3D animation language set, VITASCOPE provides an API to the visualization engine that others can use to seamlessly extend the animation language. These unmatched features make VITASCOPE a perfect tool for animating processes in any domain during all phases (verification, validation, and communication) of a simulation study.

The VITASCOPE visualization system is described in complete detail in the first author's doctoral dissertation (Kamat 2003). Additional information about VITASCOPE and its add-ons is also available at the first author's website `<http://www.engin.umich.edu/~vkamat>`.

ACKNOWLEDGMENTS

The authors thank the National Science Foundation (NSF) CAREER and ITR programs for supporting the presented work. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Henriksen, J. O. 1998. Windows-Based Animation with PROOF. In *Proceedings of the 1998 Winter Simulation Conference*, 241-247. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kamat, V. R. 2003. VITASCOPE: Extensible and Scalable 3D Visualization of Simulated Construction Operations. Doctoral dissertation, Department of Civil and Environmental Engineering, Virginia Tech, Blacksburg, VA. Available online via `<http://scholar.lib.vt.edu/theses/available/etd-04142003-222138/>` [accessed April 9, 2004].

AUTHOR BIOGRAPHIES

VINEET R. KAMAT is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of Michigan. He received a PhD in Civil Engineering at Virginia Tech in 2003; a MS in Civil Engineering at Virginia Tech in 2000; and a BE degree in Civil Engineering from Goa University (Goa, India) in 1998. He designed and implemented the VITASCOPE visualization system with J. Martinez as part of his doctoral research. In addition to visualization, his research interests include discrete event simulation, information technology, and decision support systems for construction engineering. His email and web addresses are `<vkamat@umich.edu>` and `<http://www.engin.umich.edu/~vkamat>`.

JULIO C. MARTINEZ is an Associate Professor in the Via Department of Civil and Environmental Engineering at Virginia Tech. He received his PhD in Civil Engineering at the University of Michigan in 1996; a MSE in Construction Engineering and Management from the University of Michigan in 1993; a M.S. in Civil Engineering from the University of Nebraska in 1987; and a Civil Engineer's degree from Universidad Catolica Madre y Maestra (Santiago, Dominican Republic) in 1986. He designed and implemented the STROBOSCOPE simulation language with P. Ioannou and was V. Kamat's research advisor. In addition to discrete event simulation, his research interests include construction process modeling and decision support systems for construction engineering. His email address is `<julio@vt.edu>` and his website may be accessed at `<http://strobos.ce.vt.edu>`.