Creating Focus:
Augmented Reality for Better Standards

Matt Aronoff
National Institute of Standards and Technology
About NIST

NIST is a national research laboratory that is part of the U.S. Department of Commerce, more specifically the Technology Administration. Our mission is to improve quality of life by working with industry to develop and apply technology, measurements, and standards.

now, about Focus...
As the complexity of the information being communicated increases, traditional methods of standards development are breaking down.

- The environment in which data is used has become more complex, as has the data itself.
- B2B interaction now includes CAD models, diagnostic data, process control data, quality data.

Possible solution: apply existing, rigorous methods for software development to standards.
An ad-hoc approach to standards development can result in standards that are:

- ambiguous
- redundant and/or missing information
- overly complex
- conflicting
- inefficient
Solution

Scope and requirements
- What information and interactions are we trying to capture?

Implementation
- How will that information be stored and moved around?
What are we trying to do?

- Improve the ability of domain experts to create underlying data models.
- Improve collaborative data modeling sessions between domain experts
  - Global economy means experts are located across the world
    - travel is often impractical
- Existing solutions
  - Audio -- picture is worth a thousand words; modeling is visual
  - Browser-based screen sharing software -- interactivity suffers
  - Video conference -- similar to audio conference, provides no additional information about the model
Focus: a distributed 3D modeling tool

- Uses augmented reality to provide an appealing modeling experience for domain experts
- Distributed, shared environment
- 3D improves visualization of complex models
Focus: a distributed 3D modeling tool

“...Applications that involve the overlay of virtual imagery on the real world.”
-- ARToolkit home page

Use specified markers (fiducials) to position rendered 3D objects within a video stream

When combined with a 3D display device (e.g., a head-mounted display), allows more of the model to be viewed and understood at once than with a 2D display
Focus: a distributed 3D modeling tool

Design

- Aim to keep cost below that of a dedicated video conferencing system
  - Off-the-shelf 640x480 webcams
  - eMagin z800 OLED HMD (less than $1k)
- Modular and cross-platform
  - Written in Ruby
- Freely available for further development
Trimurti

- dependency injection framework
- provides module management based on input requirements and output capabilities
- new modules can be swapped in VERY easily (~4 lines of code)
- minimizes application-specific code
In order for a user to interact with Focus, need to know 3 things:

1. User's position relative to the model
2. Position of the user's hand relative to the model
3. Any actions that the user is performing on the model
Positions (user and hand) are tracked using **fiducials**

- Pre-defined shapes that Focus knows to look for in the user's video feed
- Primary fiducial represents the position of the model
- Secondary fiducial attached to the user's hand

Actions are registered in one of two ways:

- Gestures
- Standard mouse clicks
GUI Design
Flat surface arrayed with available tools
- Like palette in existing graphical editors, etc.
- Tools may have attributes, and may be piled together

One user has the active tool rack at any one time; others are inactive
- Inactive tool rack shows active user, active tool, button to request control

When a user requests control, active user is notified
- When control is released, the new user’s tool rack becomes active
Object creation

- User selects the desired tool (Vertex, Edge, etc.) and performs the "Create" gesture.
- Object created at the position of the user's hand.
Selection & Moving

- Selection can be single-element, toggleable, or area-based
  - Selection cuboid

- All tools are capable of selecting objects

- All movement via drag & drop using the Drag Tool
  - Live feedback
    - Dragged items become translucent
    - When a drop operation would fail, the selection becomes red.
Changes are instant and persistent by default.

Unlimited undo functionality is handled by the system's command processor.

Alternate configuration places a green "accept" and red "reject" sphere above and below the changed object.
Searching

- Real time
- Type-ahead-find
  - If nothing is selected, the user can begin typing, and matching results will become selected
- Advanced search tool
  - Within entire model
  - Within selection
  - Within a certain number of connections to the current selection
Issues encountered
Created SWIG (Simplified Wrapper and Interface Generator) bindings

could not wrap entire API for some toolkits
Multiple languages

- Bindings broke object oriented code
  - SWIG frequently (but not always) failed to handle arguments passed by reference

- Memory management issues
  - Workarounds for pass-by-reference problem resulted in extra copies of objects in memory
  - Garbage collection (Ruby has it, C and C++ do not)
  - Coin3D scene caching

- Layered debugging
  - Any problem in one of the C or C++ libraries had to be debugged separately with a C debugger.
Existing webcams varied widely in autofocus and low light ability.

Resolution restricts maximum distance between the camera and fiducials.

Fiducial detection algorithm needs more pixels to function properly.

Upcoming cameras like the 2.5-ounce Iconix HD-RH1 (1080p resolution) may improve interaction nearly 7x the number of pixels.
Picture quality varies

MacBook Pro's internal iSight (uses USB) gave crisp image; external Apple iSight (uses Firewire) gave darker, more blurry images

In between: Logitech QuickCams (USB), Unibrain Fire-i (Firewire)

Fiducial recognition suffers

Frame speed varies

Some cameras seemed unable to keep up with the rest of the system

around 10–15 fps, dragging and new object creation become unreliable
Display devices

- Limited-resolution HMDs
  - Narrow field of view, even with recent HMDs (less than 40 degrees)
  - Blacked-out peripheral vision
- Another option is a see-through HMD
  - lower requirements for high-resolution video
  - lower processing requirements
Future steps
Future steps

Complete next iteration
- Search tool
- Networking (partially complete)
- Interface enhancements (adding feedback)

Modeling support
- additional diagram types (currently, only class models are included)
- improve UML coverage
- import & export functionality

Hardware
- accelerometer-based hand tracking
- see-through HMD
- higher resolution cameras
Also forking a second, re-imagined version of Focus

- existing modular infrastructure
- new, simplified interface (2D)
- target a specific, information-standards-oriented subset of UML
Questions?

matthew.aronoff@nist.gov

References:

