Feedback: Senses

- Visual
- Auditory
- Tactile/Cutaneous
  - Touch
  - Pressure
  - Vibration
  - Temperature
  - Pain
- Kinesthetic
- Vestibular
- Olfactory
- Gustatory
- Nociceptive (pain receptors in skin, cornea, gut,…)

**Feedback** is information conveyed back to the user through one or more senses when performing a task.

Proprioceptive = Kinesthetic OR Kinesthetic + Vestibular
**Feedback: Types**

- **Reactive**
  - Self-generated sensory feedback
    - E.g., Proprioceptive sense of user’s movements
- **Instrumental**
  - Generated directly by interface when operated by user
    - E.g., Force feedback from control user is holding
- **Operational**
  - Generated by system as a result of user’s interaction with interface
    - E.g., Visual feedback of world moving relative to vehicle user is driving

**Feedback: Consistency**

- **Cross-sensory**
  - Consistency across senses
- **Spatial**
  - Stimulus-response (S-R) compatibility
    - Whether mapping of user stimulus to system response is compatible (e.g., in direction, orientation, or magnitude)
  - **Nulling**
    - Whether returning device to initial position/orientation → returning controlled objects to initial position/orientation
Feedback: Consistency

- Temporal
  - Latency
    - External sensory feedback vs. internal sensory feedback
      - E.g., visual vs. proprioceptive
    - Virtual sensory feedback vs. real sensory feedback
      - E.g., synthesized visual vs. real visual
    - Mean vs. variance

Feedback: Sensory Substitution

- Useful when some feedback channel is not available
  - Replace tactile feedback with visual or auditory feedback
  - Replace pressure with vibration
  - “Virtual synesthesia”
Passive Haptic Feedback

- Take advantage of shape/position/feel of real objects co-located with virtual ones (AKA static haptics)
  - Physical floor coplanar with virtual ground
  - Physical tabletop coplanar with virtual tabletop

Passive haptic feedback can feel “right” especially when combined with visual/audio feedback

Passive Haptic Feedback

- Passive interface props
  - User interacts with physical correlates to virtual visual objects
  - Example: Tracked ball (or doll's head) and tracked plastic strip can control models of brain and cutting plane in neurosurgical planning
    - Simplified props avoid false expectations about details
  - You can create these using Vuforia
    - polygons, polyhedra, cylinders

Feiner, COMS W4172, Spring 2014
Passive Haptic Feedback

- Passive interface props
  - User interacts with physical correlates to virtual visual objects
  - Example: Styrofoam blocks serve as both projection surfaces and physical obstacles in virtual environment

K.-L. Low, G. Welch, A. Lastra, and H. Fuchs, 2001

Passive Haptic Feedback

- Opportunistic controls
  - User interface is harvested from existing objects in task domain with appropriate shape, feel, response
  - Each opportunistic control combines
    - Portion of real world
    - 3D widget
    - One or more gestures
    - Simple vision-based tracking

S. Henderson and S. Feiner, VRST 2008, TVCG 2010
Tangible User Interfaces (TUI)

- “Employ physical objects, surfaces, and spaces as tangible embodiments of digital information.” — H. Ishii
- Use tracked physical blocks (“phicons”) to represent buildings, tools
  - B. Ullmer and H. Ishii, metaDesk, UIST 97
- Tracked objects control projected graphics
  - J. Underkoffler and H. Ishii, Luminous Room, SIGGRAPH 99

Tangible User Interfaces
Example: Illuminating Clay

- B. Piper, C. Ratti, Y. Wang, B. Zhu, S. Getzoyan, & H. Ishii, CHI 2002
- Allow users to interact with real freeform surfaces
- Users
  - Molds surface of clay (or sand)
- System
  - Determines height of surface
  - Projects imagery corresponding to task domain
Simplification

- Map a potentially complex interaction to a simpler, but more useful, one
  - Often using a more sophisticated sounding metaphor

Constraints

- Relations between variables that are automatically maintained/satisfied by the system
  - Example: Geometric constraints
    - Distance between points
    - Angle between lines
    - Ratio of line lengths, angles

Sutherland’s constraints operate on vertices. In the example on the right, the two shorter lines are constrained to be parallel and equal in length.
Constraints

- Physically realistic
  - Gravity
  - Collision detection/avoidance
- Nonrealistic
  - Turn off gravity/inertia in CAD modeler
  - Slow down time to explore simulation
- Reduce DOFs for user input
- One-way vs. multi-way

Multi-way constraints are much trickier to implement, can be harder to understand, and can be less efficient

Intelligent Constraints

- Impose constraints based on
  - object semantics
    - class
    - front vs. back
  - geometric relationships
  - visibility
  - user interaction

- E.g.,
  - if A is on top of B, then
    - if B is moved, move A
    - If A is moved, don’t move B

Intelligent Constraints

- R. Bukowski and C. Sequin, "Object Associations," *SI3D 95*
  - Relocation procedure moves selected object along relocation manifold offset from horizontal or vertical surface
  - Association procedure (triggered by visibility/contact changes) finds associated objects and modifies position/orientation of selected object and cursor (e.g., pseudo-gravity, on-wall)

1. Selected cup is dragged under a table according to relocation procedure.
2. Visibility information determines when cup "rises" to the tabletop.
3. Association procedure modifies object position and mouse cursor position.

Intelligent Constraints

- R. Bukowski and C. Sequin, "Object Associations," *SI3D 95*
  - Relocation procedure moves selected object along relocation manifold offset from horizontal or vertical surface
  - Association procedure (triggered by visibility/contact changes) finds associated objects and modifies position/orientation of selected object and cursor (e.g., pseudo-gravity, on-wall)

1. Selected cup is dragged off table's supporting surface.
2. Cup "falls" onto lower surface.
Intelligent Constraints


- Objects arranged in hierarchy based on one-way constraints
  - Transforming a parent transforms its descendants

- Objects can have
  - Binding areas: Specify where this object can bind to other objects
  - Offer areas: Specify where other objects can bind to this object

- Binding/offer areas
  - Each defined by polygon and orientation vector
  - Can be independent from object geometry
Intelligent Constraints


- When moving *unconstrained* object $A$
  - Pick closest offer area on other objects to binding area of $A$ within limited range
  - Move/orient $A$ to enforce constraint iff binding area can fit within offer area
    - Make object $A$ a child of object $B$ to whose offer area object $A$ was bound
- When moving *constrained* object $A$
  - *Preserve* constraint if binding area of $A$ remains in offer area of $B$
  - *Break* constraint if binding area of $A$ pulled away from offer area of $B$
    - Remove object $A$ from children of object $B$ whose offer area object $A$ left
Intelligent Constraints


**Dual constraints**
- Support limited two-way constraints for nonhierarchical groups (e.g., siblings)
- Specified by geometry (point, line, or polygon) and *normal vector* (pointing away from object)

Intelligent Constraints


**Dual constraints**
- Build group: When object $A$ is moved, if one of its dual constraints is sufficiently close to an appropriately shaped dual constraint of another object $B$
  - $A$ and $B$ are *snapped* together such that geometries of the dual constraints are coincident and normal vectors are opposite
Intelligent Constraints


Dual constraints

- **Build group:** When object $A$ is moved, if one of its dual constraints is sufficiently close to an appropriately shaped dual constraint of another object $B$
  - $A$ and $B$ are snapped together such that geometries of the dual constraints are coincident and normal vectors are opposite
- **Maintain group:** When object $A$ is moved
  - All grouped objects in contact in direction of motion are moved with $A$
- **Break group:** When object $A$ is moved
  - All grouped objects not in contact in direction of motion do not move, are broken off from $A$, and form new group(s)

Dual constraint graph

- Specifies direction of connectivity of grouped objects based on normal vectors
- Supports intuitive splitting of dual groups by generalizing "objects in contact in direction of motion"
- When selected object $A$ is moved, of those objects currently grouped with $A$,
  - The only objects that move with $A$ and remain grouped with $A$ are those connected to $A$ in a direction within $90^\circ + \text{tolerance}$ (e.g., $10^\circ$) of movement direction, and those that recursively meet that requirement
  - Remaining objects are broken off from $A$ and form new fully connected group(s)
Intelligent Constraints


http://www.cse.yorku.ca/~wolfgang/videos/mive_duals.mpg