GIVE-ME: Gamification In Virtual Environments for Multimodal Evaluation Framework

WAI KHOO

JANUARY 20, 2016

ZHIGANG ZHU, TONY RO, YINGLI TIAN, ARIES ARDITI
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
Introduction

• 161 million, 2002 -> 285 million, 2014 (~77% increase)
• 39 million blind, 246 million low vision.
• One of the biggest challenges to independence is navigation [1]
• Many visual AT reviews can be found in literature [2, 3].
Introduction: Problem Statement

• Despite all this works, there is a lack of robust evaluation and scientific comparison of the effectiveness and friendliness of ATs.
  • Establish benchmark for heterogeneous systems.
  • Informed decision when shopping for an AT.

• Note that we’re focusing on ATs for navigation. Furthermore, we understand that there may not be an “one-size-fit-all” AT.

• Often times, ATs are not compared with state-of-the-art systems/algorithms.
  • Main reason: many ATs research are not reproducible because either the system is dependent on a specific hardware or the system is not fully described.
Introduction: Virtual Reality, Gamification, and Multimodality

• This proposed framework is a **unified formal** evaluation and comparison approach.

• Differ from Degara et. al. (2013) [2]
  • Allows for additional transducing methods, not just sounds.

  • Focused on cognitive mapping in unknown space.
Introduction: Virtual Reality, Gamification, and Multimodality

- Determine the best input/output devices.
- Rapid prototyping.
  - Reduce cost and time in development.
- Psychophysics evaluation.
  - Functional neuroimaging for virtual navigation.
- Early stakeholder involvement
  - Interacting with users, some of whom desire a more integral role in ATs research.
- Excellent evaluation tool before beta testing.
Introduction: Virtual Reality, Gamification, and Multimodality

• Enrich experiment by providing fun and engaging sessions.

• Encourage subjects to participate -> sustainable evaluation and data collection.

• Games can be distributed to others as a kind of crowd-sourcing data collection method.

• Can be nicely packaged into a simulation or training tool.
Introduction: Virtual Reality, Gamification, and Multimodality

• Allows for alternative perception.
• Allows for flexible framework setup.
  • Any sensors + any stimulators.
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
Framework

• Introducing GIVE-ME framework
  • **Gamification In Virtual Environments for Multimodal Evaluation**
  • We will focus the framework on blind navigation, but can be extended/applied to others (e.g.s, application for ASD).
Framework: User Interface
1) Controllers
2) Multimodal Stimulators

- Sound cue generation *(audio)*
  - Through speakers
  - Interfaced via the host’s OS (and then via Unity3D Audio class)
- Text-to-speech *(audio)*
  - Through speakers
  - Interfaced via a plugin
- Vibration motors *(haptic/tactile)*
  - Interfaced via serial port/Arduino
- Brainport *(electrode)*
  - Interfaced via TCP/IP socket (localhost)
3) Measurement Device

• Purpose: for external ("offline") data collection

• Connect to devices such as EEG
  • Custom program to interface via a parallel port plugin
  • Send a 8-bit unsigned integer (trigger)

• Other qualitative data
  • Questionnaires
  • Observations

• Offline measurement has to be synchronized with online ("VE data") measurement.
  • Timestamp
Framework: Foundation
1) Multimodal (Virtual) Sensors

• Common multimodal sensors: infrared, sonar, and RGB-D

• Infrared (IR) sensor
  • Light-based sensors with a very narrow beam angle
  • Specifications:
    • Minimum distance: 10 cm
    • Detection distance: 80 cm
    • Beam width: 12 cm -> about 8.5 degree beam angle
  • Limitations:
    • Needs to be pointed exactly at object for detection.
    • Cross interference.
    • Noisy data
1) Multimodal (Virtual) Sensors

Estimating theoretical upper bound of sensors coverage

\[ \alpha = 180 - \tan^{-1}\left(\frac{w}{2d}\right) \]  

\[ x = \sqrt{d^2 + \left(\frac{w}{2}\right)^2} \]  

\[ y = \sqrt{x^2 + r_h^2 - 2r_h x \cos \alpha} \]  

\[ \theta' = \frac{\theta}{2} \]  

\[ \sin \theta' = \frac{\sin \alpha}{x} \]  

\[ \theta' = \sin^{-1}\left(\frac{x \sin \alpha}{y}\right) \]  

\[ \theta = 2 \sin^{-1}\left(\frac{x \sin \alpha}{y}\right) \]
1) Multimodal (Virtual) Sensors

- Statistical data from CDC [12]
  - Average male
    - Height: 175.9 cm
    - Girth: 100.9 cm (32.12 cm in diameter)
    - totalSensor: 732
  - Average female
    - Height: 162.1 cm
    - Girth: 95.2 cm (30.3 cm in diameter)
    - totalSensor: 661

\[ totalSensor = \left( \frac{360^\circ}{\theta} \right) \times \left( \frac{H}{w} \right) \] (5)

1,000 sensors? NO PROBLEM!
1) Multimodal (Virtual) Sensors

```
Algorithm 1 Virtual Infrared Sensor

Require: maxRange > 0.0, direction ← Vector3()
Ensure: 0 ≤ distance ≤ maxRange

1: procedure IR
2:      hitInfo ← RaycastHit() ▷ Initialize hitInfo
3:      distance ← 0.0
4:   loop
5:      startPt ← transform.position ▷ Changes when avatar is walking
6:      if Raycast(startPt, direction, hitInfo, maxRange) then
7:         distance ← hitInfo.distance
8:      else
9:         distance ← 0.0
10:   end if
11: end loop
12: end procedure
```
1) Multimodal (Virtual) Sensors

- **Sonar**
- 8 meters
- 150 degree

Algorithm 2 Virtual Sonar Sensor

```
Require: maxRange > 0.0, radius > 0.0, norm ← Vector3()
Ensure: 0 ≤ distance ≤ maxRange
1: procedure SONAR
2:   hitInfo ← RaycastHit()                      ▷ Initialize hitInfo
3:   distance ← 0.0
4:   loop
5:     startPt ← transform.position  ▷ Changes when avatar is walking
6:     minDist ← maxRange
7:     for i,j ← 0, radius do
8:       if radius^2 ≥ i^2 + j^2 then
9:         direction ← vector from startPt to (i,j,maxRange), offset from norm
10:        if Raycast(startPt, direction, hitInfo, maxRange) then
11:           if hitInfo.distance < minDist then
12:             minDist ← hitInfo.distance
13:           end if
14:         end if
15:       end if
16:     end for
17:     distance ← minDist
18: end loop
19: end procedure
```
1) Multimodal (Virtual) Sensors

- **RGB-D** (i.e., with image size $width \times height$ pixels)
  - Optical
    - Use the camera (game) view
  - Depth
    - Use $width \times height$ raycasts at each pixel location.
    - Raycasts parallel to the avatar's positive z-axis (right-hand rule).
    - Maximum range of 4 meters.
1) Multimodal (Virtual) Sensors: Transducing

- **Processing**
  - Convert raw data to **meaningful information**
  - E.g., quantize/threshold range data into 3 intervals (close, near, and far).

- **Transmission**
  - Send meaningful info to stimulators of **another modality**
  - E.g., encode the interval with appropriate vibration levels
  - **Communication protocols:**
    - USB/serial port
    - Bluetooth Low Energy
    - TCP/IP
2) Game Mechanics

- **Provides** a set of goals to achieve and **define** how the user can interact with the game.

- **Task definition**
  - What needs to be done in order to finish.
  - *Simple* task: Navigate from point A to B, as fast and as few errors as possible.
  - *Complex* task: Exploratory in nature with termination conditions (e.g., time-out).

- **Avatar Behavior**
  - What controller commands are valid.
  - E.g., constant walking speed or turn/rotate in place.
  - Define what audio cues are constantly audible and which are in response to an action or proximity.
3) Virtual Environment Toolbox

• **Third party game engine**
  • *Unity3D*: popular, excellent documentation, & tutorials
  • C#, Javascript.
  • Multi-platform.

• **Controller setup (optional)**
  • Standard input devices are compatible with Unity3D. Capture specific key action events.
  • Other controllers (e.g., Microsoft Kinect) not natively supported by Unity3D
    • Need an external/separate program capable of communicating and exchanging data with Unity3D.
3) Virtual Environment Toolbox

• Environment Design
  • Static/dynamic objects; objects interaction; placing sound cues and collectibles.
4) Data Collection

- Type of data:
  - Multimodal sensory data
    - E.g., range data generated by virtual sensors and sounds
  - Brain/behavioral measurement, as obtained from the Measurement Device
    - E.g., EEG readings and observations
  - Control/action data
    - E.g., User inputs, events, and game state.

- These data contribute to task evaluation and ground truth establishment.
4) Data Collection

- **Recommend** the following metrics for analysis:
  1. **Acceptability**: Design that is useful, reliable, robust, aesthetic, and has positive impact on quality of life of user.
  2. **Compatibility**: Design that is compatible with lifestyle of user and other technologies.
  3. **Adaptability**: Design that can be easily adjusted (i.e., function, location).
  4. **Friendly**: Low learning curve for the system; easy to use.
  5. **Performance**: Overall performance

- While these metrics can be posed as **open-ended questions**, it can also be presented as **rating surveys**. Data to assess **friendly** and **performance** features can also be collected and **extrapolated** from the VE. Data includes but not limited to:
  1. Time to completion
  2. Number of errors (e.g., bumping into obstacles, incorrect response)
  3. Game score
  4. User’s trajectory
  5. User's brain data (e.g., EEG, fMRI)
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
GIVE-ME Workflow

1. Identify the application and tasks (games) to virtualize.
   • Generally related to navigation, but can be adapted to other objectives.
   • Tasks can be as simple as navigation from origin to destination.

2. Identify multimodal sensor(s) and stimulator(s) needed.

3. Virtualize the sensors and establish connection from VE to stimulators.
   • Simulate sensor based on real spec (can be a creative task).
   • Send sensor readings to stimulators via a communication channel.
Case Study 1: **BrainportNav**

**Controller:**
Joystick

**Stimulator:**
Electrode array

**Virtual sensor:**
Camera in VE

Unity3D pathfinding

BrainportNav Results

**Run 1**
- Time = 151 seconds
- Accuracy = 0.95

**Run 2**
- Time = 91 seconds
- Accuracy = 0.95

**Run 3**
- Time = 78 seconds
- Accuracy = 0.89
**BrainportNav Results**

<table>
<thead>
<tr>
<th>Subj_Run</th>
<th>L_Correct</th>
<th>Total_L</th>
<th>R_Correct</th>
<th>Total_R</th>
<th>S_Correct</th>
<th>Total_S</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub1_R1</td>
<td>55</td>
<td>68</td>
<td>69</td>
<td>82</td>
<td>3</td>
<td>3</td>
<td>0.83</td>
</tr>
<tr>
<td>Sub1_R2</td>
<td>6</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>11</td>
<td>14</td>
<td>0.78</td>
</tr>
<tr>
<td>Sub1_R3</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>0.89</td>
</tr>
<tr>
<td>Sub2_R1</td>
<td>49</td>
<td>63</td>
<td>56</td>
<td>79</td>
<td>5</td>
<td>7</td>
<td>0.74</td>
</tr>
<tr>
<td>Sub2_R2</td>
<td>179</td>
<td>184</td>
<td>172</td>
<td>173</td>
<td>4</td>
<td>4</td>
<td>0.98</td>
</tr>
<tr>
<td>Sub2_R3</td>
<td>20</td>
<td>23</td>
<td>13</td>
<td>22</td>
<td>17</td>
<td>21</td>
<td>0.76</td>
</tr>
<tr>
<td>Sub3_R1</td>
<td>10</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>0.95</td>
</tr>
<tr>
<td>Sub3_R2</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>0.95</td>
</tr>
<tr>
<td>Sub3_R3</td>
<td>12</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>14</td>
<td>16</td>
<td>0.89</td>
</tr>
<tr>
<td>Sub4_R1</td>
<td>12</td>
<td>23</td>
<td>18</td>
<td>25</td>
<td>15</td>
<td>24</td>
<td>0.63</td>
</tr>
<tr>
<td>Sub4_R2</td>
<td>8</td>
<td>12</td>
<td>11</td>
<td>14</td>
<td>15</td>
<td>17</td>
<td>0.79</td>
</tr>
<tr>
<td>Sub4_R3</td>
<td>6</td>
<td>17</td>
<td>19</td>
<td>19</td>
<td>12</td>
<td>19</td>
<td>0.67</td>
</tr>
</tbody>
</table>

**Conclusion:** Brainport can be better used to give simple directions to people for navigation.
Case Study 2: CrowdSourceNav


Controller: Joystick
Stimulator: Text-to-speech
Virtual sensor: Camera in VE
Online crowd members

Can be used for testing algo
CrowdSourceNav Results

Maze 1
Time = 514 seconds
NumBump = 7

Maze 2
Time = 345 seconds
NumBump = 0

Maze 3
Time = 325 seconds
NumBump = 2
CrowdSourceNav Results

16 subjects

Crowd completion time is significantly different from the ground truth time (Two-sample t-test, p=0.001 at 5% significance level, df=8).

11 subjects

Crowd completion time for either aggregation method is not significantly different (two-sample t-test, p=0.432 at 5% significance level, df=6)
CrowdSourceNav Results

### Table 5: Survey Results

<table>
<thead>
<tr>
<th>Statement #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>5.27</td>
<td>6.13</td>
<td>5.93</td>
<td>6.47</td>
<td>5.20</td>
</tr>
</tbody>
</table>

**Sample size of 15**

Rating of 1 – 7 to the following statements

1. It is useful
2. It is easy to use
3. It is user friendly
4. I learned to use it quickly
5. I am satisfied with it

**Conclusion:** The system is viable and large scale experiments are needed.
Case Study 3: VibrotactileNav


VibrotactileNav Results
VibrotactileNav Results

<table>
<thead>
<tr>
<th></th>
<th>(a) For Easy Hallway</th>
<th>(b) For Complex Hallway</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>Bumping</td>
<td>Result</td>
</tr>
<tr>
<td>S1</td>
<td>257.02</td>
<td>13</td>
</tr>
<tr>
<td>S2</td>
<td>246.12</td>
<td>18</td>
</tr>
<tr>
<td>S3</td>
<td>252.54</td>
<td>12</td>
</tr>
<tr>
<td>S4</td>
<td>339.16</td>
<td>26</td>
</tr>
<tr>
<td>S5</td>
<td>316.76</td>
<td>5</td>
</tr>
<tr>
<td>S6</td>
<td>286.54</td>
<td>17</td>
</tr>
<tr>
<td>S7</td>
<td>266.70</td>
<td>32</td>
</tr>
<tr>
<td>S8</td>
<td>145.34</td>
<td>21</td>
</tr>
<tr>
<td>S9</td>
<td>185.62</td>
<td>16</td>
</tr>
<tr>
<td>S10</td>
<td>150.56</td>
<td>4</td>
</tr>
<tr>
<td>S11</td>
<td>292.30</td>
<td>26</td>
</tr>
<tr>
<td>S12</td>
<td>325.18</td>
<td>65</td>
</tr>
<tr>
<td>S13</td>
<td>210.34</td>
<td>20</td>
</tr>
<tr>
<td>S14</td>
<td>305.74</td>
<td>6</td>
</tr>
<tr>
<td>S15</td>
<td>230.38</td>
<td>15</td>
</tr>
<tr>
<td>S16</td>
<td>527.30</td>
<td>17</td>
</tr>
<tr>
<td>S17</td>
<td>389.52</td>
<td>9</td>
</tr>
<tr>
<td>S18</td>
<td>383.08</td>
<td>28</td>
</tr>
</tbody>
</table>

9 out of 18 succeeded
Succeeded:
Avg time: 280.10 sec
Avg bumps: 17.3

Failed:
Avg time: 288.65 sec
Avg bumps: 22.1

3 out of 18 succeeded
Succeeded:
Avg time: 120.25 sec
Avg bumps: 12.7

Failed:
Avg time: 353.67 sec
Avg bumps: 42.7

Conclusion: The vibrotactile array is informative about objects and wall nearby.
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
Proposed Work

• 3 tasks

1. **Path deviation analysis** among BrainportNav, CrowdSourceNav, and VibrotactileNav
   • Compute mean-squared error between subject and ground-truth trajectories.

2. **GIVE-ME software implementation**
   • Unity package

3. **VistaNav**
   • Similar to VibrotactileNav, but with newer devices and applying the GIVE-ME software package.
GIVE-ME Software Implementation

- **Reduce** game development time from months to weeks
- **Export** it as Unity package, & publish it in Unity’s store and GitHub.

- 7 components in the conceptual framework can be grouped into the following 4 questions and answers to these questions are the implementation plan.

1. How to specify the **sensors**? (Multimodal virtual sensors excluding transducing)
2. How to specify the **stimulators**? (Multimodal stimulators)
3. How to specify the **data** to be collected? (Measurement device and data collection)
4. How to specify the **environment** to be navigated in? (Controllers, game mechanics, and VE toolbox)
GIVE-ME Software Implementation

**GIVE-ME PACKAGE**

- **Sensors**
  - XML config
  - Or
  - Click-n-Drag

- **Stimulators**
  - *Comm Protocol:*
    - USB/TCP/IP
  - *Data Format:*
    - JSON/DLL

- **Data**
  - Which variables to “monitor”
  - Sampling Interval
  - File location

- **Environment**
  - Pre-built environment as a starting point

**GIVE-ME PACKAGE**:

- **Raw sensory data**
- **Data to be transduced**
- **Log file**
- **VE**
GIVE-ME Software Implementation

Framework Development Cycle
VistaNav

Controllers:
- Game pad
- Virtuix Omni

Stimulator:
- Hapi Node

Virtual sensor:
- Infrared

Objective/experiment:
- # of devices.
- Placement of it.
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
Timeline

Task 1
VistaNav

Task 2
GIVE-ME software implementation

Task 3
Path deviation analysis

Task 4
Dissertation rite-up
Outline

• Introduction
• GIVE-ME Framework
• GIVE-ME Case Studies
• Proposed Work
• Timeline
• Publications List
Publications

• Peer-reviewed Journals


• Conference Proceedings

1. Z. Zhu, **W. L. Khoo**, C. Santistevan, Y. Gosser, E. Molina, H. Tang, T. Ro, and Y. Tian. EFRI-REM at CCNY: Research Experience and Mentoring in Multimodal and Alternative Perception for Visually Impaired People. 6th IEEE Integrated STEM Education Conference (ISEC '16), March 5, 2016, Princeton, NJ. (Submitted)


• Technical Report

1. **W. Khoo** and Z. Zhu. 3D Measurements and Visualization of Gamma-Ray Cargo Inspection, City College Grove School of Engineering Journal of Student Research, May 2009
Acknowledgements

This work is supported by the U.S. National Science Foundation (NSF) through Award # EFRI-1137172 and the CUNY Graduate Center Science Fellowship (2009-2014). The work has also been supported in part by other NSF grants (Awards # CBET-1160046, IIP-1416396 and IIS-1327236), and VentureWell (formerly NCIIA, through Award # 10087-12).
References


References


