Eye Tracking Methods and Applications @ Clemson

With Slight Emphasis on Interaction

Andrew T. Duchowski
duchowski@acm.org

School of Computing
Clemson University

July 23, 2007

Abstract

A brief exposé of eye tracking research, education, and service at Clemson University is presented. Spanning about a decade, past work involves visual inspection, eye movement analysis, training, and gaze-contingent displays. Current projects include interaction and offline analysis (e.g., scanpath comparison as a usability metric). Real-time approaches will be discussed with applications to accessibility, videoconferencing, eye movement prediction, and peripheral color degradation.

What is eye tracking?

- Method for calculating what user is looking at.
- Accomplished through computer vision techniques imaging pupil and corneal reflection of (near) infra-red light.
- Performed in real-time, hence applicable to gaze-controlled user interface control (e.g., eyes used as pointer in traditional WIMP metaphor).
- Off-line analysis of real-time recorded eye movements can lead to novel insights into human perception, behaviour, etc.
- Recent technological trends (digital cameras) have once again revitalized the field (permanently this time, it seems).

What is eye tracking?

- Method for calculating what user is looking at.
- Accomplished through computer vision techniques imaging pupil and corneal reflection of (near) infra-red light.
- Performed in real-time, hence applicable to gaze-controlled user interface control (e.g., eyes used as pointer in traditional WIMP metaphor).
- Off-line analysis of real-time recorded eye movements can lead to novel insights into human perception, behaviour, etc.
- Recent technological trends (digital cameras) have once again revitalized the field (permanently this time, it seems).

What is eye tracking?

- Method for calculating what user is looking at.
- Accomplished through computer vision techniques imaging pupil and corneal reflection of (near) infra-red light.
- Performed in real-time, hence applicable to gaze-controlled user interface control (e.g., eyes used as pointer in traditional WIMP metaphor).
- Off-line analysis of real-time recorded eye movements can lead to novel insights into human perception, behaviour, etc.
- Recent technological trends (digital cameras) have once again revitalized the field (permanently this time, it seems).

What is eye tracking?

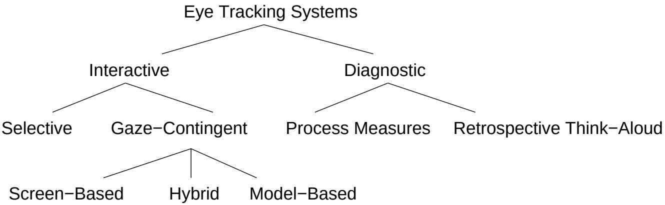
- Method for calculating what user is looking at.
- Accomplished through computer vision techniques imaging pupil and corneal reflection of (near) infra-red light.
- Performed in real-time, hence applicable to gaze-controlled user interface control (e.g., eyes used as pointer in traditional WIMP metaphor).
- Off-line analysis of real-time recorded eye movements can lead to novel insights into human perception, behaviour, etc.
- Recent technological trends (digital cameras) have once again revitalized the field (permanently this time, it seems).

What is eye tracking?

- Method for calculating what user is looking at.
- Accomplished through computer vision techniques imaging pupil and corneal reflection of (near) infra-red light.
- Performed in real-time, hence applicable to gaze-controlled user interface control (e.g., eyes used as pointer in traditional WIMP metaphor).
- Off-line analysis of real-time recorded eye movements can lead to novel insights into human perception, behaviour, etc.
- Recent technological trends (digital cameras) have once again revitalized the field (permanently this time, it seems).

Eye Tracking Applications

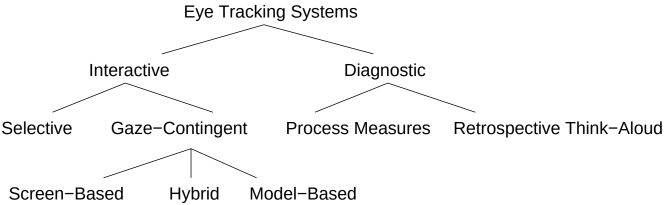
- Research context:



- On-line vs. off-line
  - On-line apps $\sim$ interactive, real-time
  - Off-line apps $\sim$ diagnostic
- Both forms rely on eye movement analysis

Eye Tracking Applications

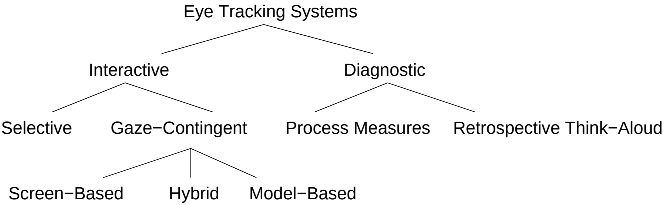
- Research context:



- On-line vs. off-line
  - On-line apps $\sim$ interactive, real-time
  - Off-line apps $\sim$ diagnostic
- Both forms rely on eye movement analysis

Eye Tracking Applications

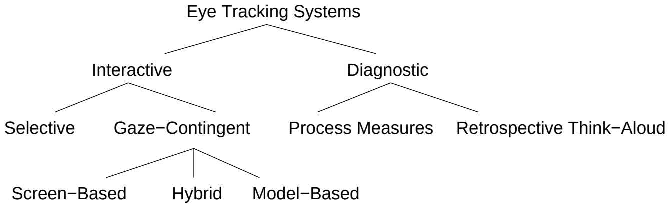
- Research context:



- On-line vs. off-line
  - On-line apps $\sim$ interactive, real-time
  - Off-line apps $\sim$ diagnostic
- Both forms rely on eye movement analysis

Eye Tracking Applications

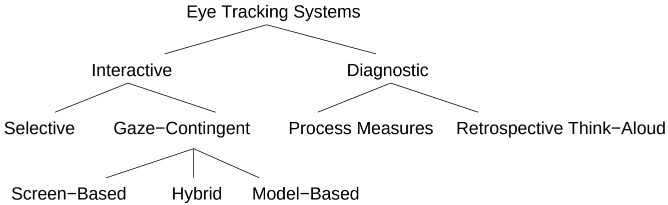
- Research context:



- On-line vs. off-line
  - On-line apps $\sim$ interactive, real-time
  - Off-line apps $\sim$ diagnostic
- Both forms rely on eye movement analysis

Eye Tracking Applications

- Research context:



- On-line vs. off-line
  - On-line apps $\sim$ interactive, real-time
  - Off-line apps $\sim$ diagnostic
- Both forms rely on eye movement analysis

Infrastructure at Clemson

Clemson's Eye Tracking Laboratory.

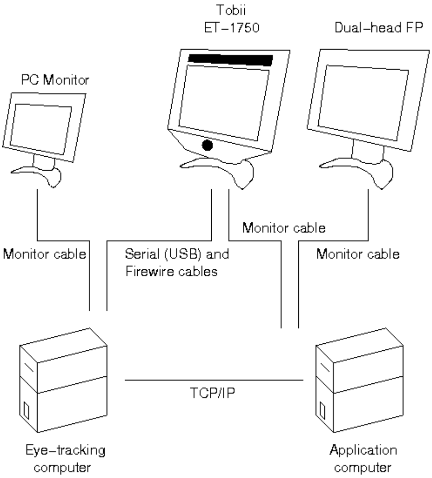
- Older equipment as well as modern 4th generation eye trackers (installed Fall '05).
- Installation supports eye tracking class and experiments.

Infrastructure at Clemson

Clemson's Eye Tracking Laboratory.

- Older equipment as well as modern 4th generation eye trackers (installed Fall '05).
- Installation supports eye tracking class and experiments.

Tobii Eye Tracking Workstation



Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 17″display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280$\times$1024 17$''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 17″display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 $17''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 $17''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 $17''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 $17''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Infrastructure Specs

- Older technology is, by today's standards, truly cumbersome.
- Current eye tracking stations consist of:
  1. Tobii ET-1750 eye tracker (50 Hz sampling rate, 1280×1024 $17''$display, 0.5° accuracy),
  2. Sun W2100z server running Windows XP (dual 2.0 GHz AMD Opteron 246 CPUs, 2GB RAM),
  3. Sun Ultra 20 client running Linux (2.2 GHz AMD Opteron 148 CPU, 1GB RAM, NVidia GeForce 7800 GTX).
- All machines are connected to 1Gb/s Ethernet subnet.
- Tobii's SDK and TCP/IP client/server model makes Linux development much simpler than before.
- Already obsolete!

Thanks to our sponsors ($\sim$1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

+ . . . supplemental awards

Thanks to our sponsors ($\sim$1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

\+ . . . supplemental awards

Thanks to our sponsors ($\sim$1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

+ . . . supplemental awards

Thanks to our sponsors (∼1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

\+ . . . supplemental awards

Thanks to our sponsors (∼1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

\+ . . . supplemental awards

Thanks to our sponsors (~1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

\+ . . . supplemental awards

Thanks to our sponsors (∼1.3M)

- NASA ($265,742, with IE): visual inspection training
- NSF CAREER ($212,022): graphics, education, service
- NSF ITR ($313,361, with IE): eye tracked avatars in VR
- NSF ATE ($389,655, with IE): aircraft inspection training
- DoD/Navy/SPAWAR Charleston ($115,528): usability engineering

\+ . . . supplemental awards

Publications

- Developed techniques for binocular eye movement measurement in VR for cargo bay visual inspection (VRST 2001 (A. Duchowski et al., 2001)).
- Developed velocity-based eye movement analysis in 3D (ETRA 2002 (A. Duchowski et al., 2002)).
- Adapted string editing approach for scanpath comparison in $2\frac{1}{2}$D VE (EuroGraphics 2002 (Marmitt & Duchowski, 2002)).
- Extended scanpath comparison to 3D (VSS 2003 (A. T. Duchowski et al., 2003)).
- Demonstrated effectiveness of feedforward training with scanpaths (SIGCHI 2005[1] (Sadasivan et al., 2005)).

[1] Nominated for best paper, top 5% of 371 submissions.

Publications

- Developed techniques for binocular eye movement measurement in VR for cargo bay visual inspection (VRST 2001 (A. Duchowski et al., 2001)).
- Developed velocity-based eye movement analysis in 3D (ETRA 2002 (A. Duchowski et al., 2002)).
- Adapted string editing approach for scanpath comparison in $2\frac{1}{2}$D VE (EuroGraphics 2002 (Marmitt & Duchowski, 2002)).
- Extended scanpath comparison to 3D (VSS 2003 (A. T. Duchowski et al., 2003)).
- Demonstrated effectiveness of feedforward training with scanpaths (SIGCHI 2005[1] (Sadasivan et al., 2005)).

[1] Nominated for best paper, top 5% of 371 submissions.

Publications

- Developed techniques for binocular eye movement measurement in VR for cargo bay visual inspection (VRST 2001 (A. Duchowski et al., 2001)).
- Developed velocity-based eye movement analysis in 3D (ETRA 2002 (A. Duchowski et al., 2002)).
- Adapted string editing approach for scanpath comparison in $2\frac{1}{2}$D VE (EuroGraphics 2002 (Marmitt & Duchowski, 2002)).
- Extended scanpath comparison to 3D (VSS 2003 (A. T. Duchowski et al., 2003)).
- Demonstrated effectiveness of feedforward training with scanpaths (SIGCHI 2005[1] (Sadasivan et al., 2005)).

[1] Nominated for best paper, top 5% of 371 submissions.

Publications

- Developed techniques for binocular eye movement measurement in VR for cargo bay visual inspection (VRST 2001 (A. Duchowski et al., 2001)).
- Developed velocity-based eye movement analysis in 3D (ETRA 2002 (A. Duchowski et al., 2002)).
- Adapted string editing approach for scanpath comparison in $2\frac{1}{2}$D VE (EuroGraphics 2002 (Marmitt & Duchowski, 2002)).
- Extended scanpath comparison to 3D (VSS 2003 (A. T. Duchowski et al., 2003)).
- Demonstrated effectiveness of feedforward training with scanpaths (SIGCHI 2005[1] (Sadasivan et al., 2005)).

[1] Nominated for best paper, top 5% of 371 submissions.

Publications

- Developed techniques for binocular eye movement measurement in VR for cargo bay visual inspection (VRST 2001 (A. Duchowski et al., 2001)).
- Developed velocity-based eye movement analysis in 3D (ETRA 2002 (A. Duchowski et al., 2002)).
- Adapted string editing approach for scanpath comparison in $2\frac{1}{2}$D VE (EuroGraphics 2002 (Marmitt & Duchowski, 2002)).
- Extended scanpath comparison to 3D (VSS 2003 (A. T. Duchowski et al., 2003)).
- Demonstrated effectiveness of feedforward training with scanpaths (SIGCHI 2005[1] (Sadasivan et al., 2005)).

[1] Nominated for best paper, top 5% of 371 submissions.

Determine gaze point (x_g, y_g, z_g)

- Given instantaneous eye tracked coordinates, (x_l, y_l) and (x_r, y_r), and head-tracked head position coordinates, (x_h, y_h, z_h), the coordinates of the gaze point, (x_g, y_g, z_g), are:

$$
\begin{aligned}
x_g &= (1-s)x_h + s(x_l + x_r)/2 \\
y_g &= (1-s)y_h + s(y_l + y_r)/2 \\
z_g &= (1-s)z_h + sf,
\end{aligned}
$$

where $s = b/(x_l - x_r + b)$, b is the baseline distance between the left and right eye centers, and f is the distance to the near viewing plane along the head-centric z-axis.

Incorporate head position

- Given raw gaze intersection points in 3D, eye movement analysis requires that:
  1. the head position, $\mathbf{h}$, is recorded to calculate visual angle,
  2. with two successive intersection points in three-space, $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and the head position at each instance, $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$, the visual angle is obtained:

$$\theta_i = \cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|}, \quad i \in [0, n),$$

     where n is the sample size and $\mathbf{v}_i = \mathbf{p}_i - \overline{\mathbf{h}}$ and $\overline{\mathbf{h}}$ is the averaged head position over the sample time period.

Incorporate head position

- Given raw gaze intersection points in 3D, eye movement analysis requires that:
  1. the head position, **h**, is recorded to calculate visual angle,
  2. with two successive intersection points in three-space, $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and the head position at each instance, $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$, the visual angle is obtained:

$$\theta_i = \cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|}, \quad i \in [0, n),$$

     where n is the sample size and $\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{h}}$ and $\bar{\mathbf{h}}$ is the averaged head position over the sample time period.

Incorporate head position

- Given raw gaze intersection points in 3D, eye movement analysis requires that:
    1. the head position, $\mathbf{h}$, is recorded to calculate visual angle,
    2. with two successive intersection points in three-space, $\mathbf{p}_i = (x_i, y_i, z_i)$ and $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and the head position at each instance, $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$, the visual angle is obtained:

    $$\theta_i = \cos^{-1} \frac{\mathbf{v}_i \cdot \mathbf{v}_{i+1}}{\| \mathbf{v}_i \| \| \mathbf{v}_{i+1} \|}, \quad i \in [0, n),$$

    where n is the sample size and $\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{h}}$ and $\bar{\mathbf{h}}$ is the averaged head position over the sample time period.

Real-time classification

- An FIR filter, $\mathbf{h}_k$, is used for velocity-based saccade detection,

$$\dot{\theta}_i = \frac{1}{\Delta t}\sum_{j=0}^{k}\theta_{i+j}\mathbf{h}_j, \quad i \in [0, n-k),$$

expressed in deg/s, where k is the filter length, $\Delta t = k - i$.

- Acceleration is obtained via a subsequent convolution of velocity, $\dot{\theta}_i$, with an acceleration filter, $\mathbf{g}_j$,

$$\ddot{\theta}_i = \frac{1}{\Delta t}\sum_{j=0}^{k}\dot{\theta}_{i+j}\mathbf{g}_j, \quad i \in [0, n-k),$$

expressed in deg/s^2, where k is the filter length, $\Delta t = k - i$.

Real-time classification

- An FIR filter, $\mathbf{h}_k$, is used for velocity-based saccade detection,

$$\dot{\theta}_i = \frac{1}{\Delta t}\sum_{j=0}^{k}\theta_{i+j}\mathbf{h}_j, \quad i \in [0, n-k),$$

expressed in deg/s, where k is the filter length, $\Delta t = k - i$.

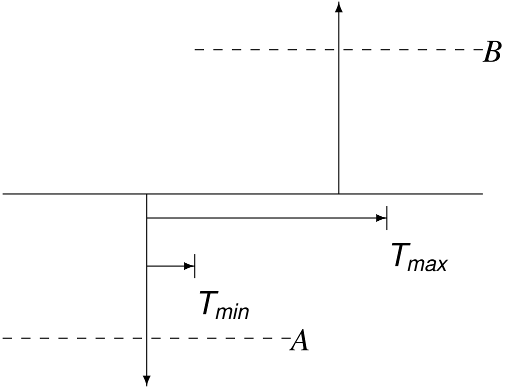
- Acceleration is obtained via a subsequent convolution of velocity, $\dot{\theta}_i$, with an acceleration filter, $\mathbf{g}_j$,

$$\ddot{\theta}_i = \frac{1}{\Delta t}\sum_{j=0}^{k}\dot{\theta}_{i+j}\mathbf{g}_j, \quad i \in [0, n-k),$$

expressed in deg/s^2, where k is the filter length, $\Delta t = k - i$.

Acceleration-based filtering

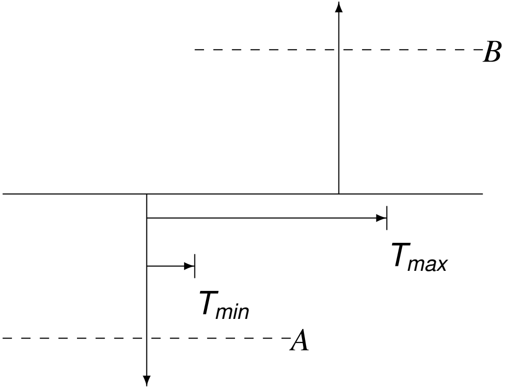
- If $|\ \ddot{\theta}_i\ | > A$, the corresponding gaze intersection point $\mathbf{p}_i$ is treated as the beginning of a saccade.
- Three additional conditions are evaluated to characterize a saccade, as illustrated below.



1. $|\ \ddot{\theta}_i\ | > A$
2. $|\ \ddot{\theta}_{i+j}\ | > B$
3. $sgn(\ddot{\theta}_{i+j}) \neq sgn(\ddot{\theta}_i)$
4. $T_{min} < j - i < T_{max}$

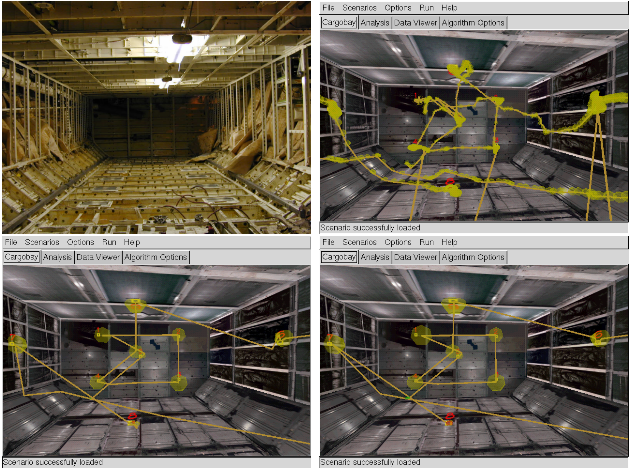
Acceleration-based filtering

- If $|\ddot{\theta}_i| > A$, the corresponding gaze intersection point $\mathbf{p}_i$ is treated as the beginning of a saccade.
- Three additional conditions are evaluated to characterize a saccade, as illustrated below.



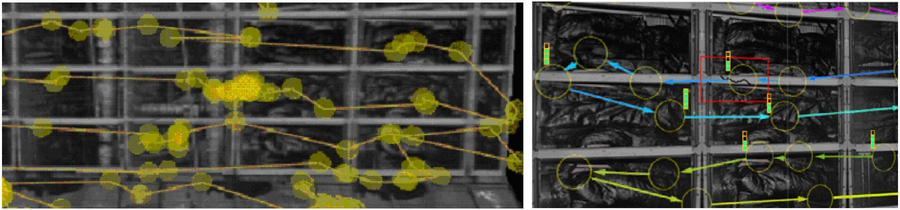
1. $|\ddot{\theta}_i| > A$
2. $|\ddot{\theta}_{i+j}| > B$
3. $sgn(\ddot{\theta}_{i+j}) \neq sgn(\ddot{\theta}_i)$
4. $T_{min} < j - i < T_{max}$

Results



Results: physical cargo bay (upper left), raw data (upper right), 2-tap (lower left), 5-tap (lower right) velocity filter.

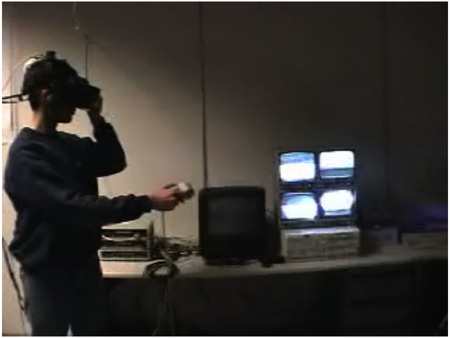
Training application



Expert inspector scanpath (left), feedforward training display (right).

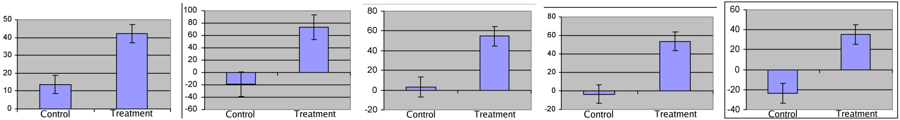
- Following analysis, eye movements abstracted further to display expert's search strategy.
- Eye movements constitute *process measures* and augment traditional *performance measures* (speed and accuracy).

Stimulus



Sadasivan's CHI 05 video.

Outcomes

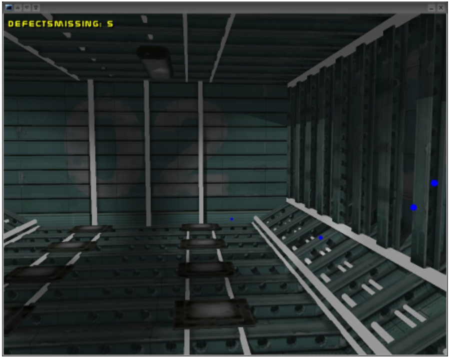


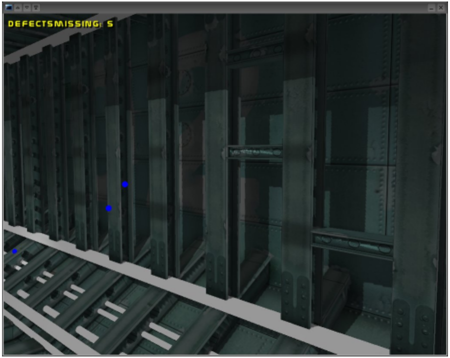
(a) True positives. (b) Time to complete. (c) Fixation points. (d) Fixation groups. (e) Fixation duration.

Performance and process measures: relative difference (%).

- Speed-accuracy tradeoff was observed: performance improved while completion time increased
- Eye movement analysis suggests that training made novice inspectors slower but more deliberate

Moving to 3D immersive environment

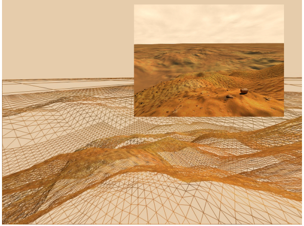




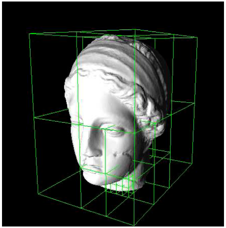
3D cargo bay.

- Visual inspection in 3D CVE.
- Problem: artificial defect and gaze point registration.

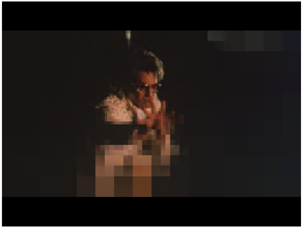
Pubs and pics



(Bob Danforth)



(Hunter Murphy)

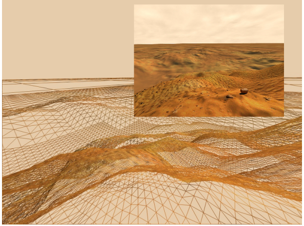


(Nathan Cournia)

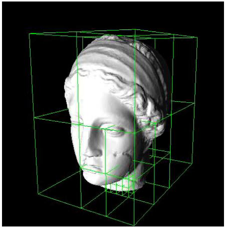
GCD examples.

- Showed feasibility of gaze-contingent terrain generation (Smart Graphics 2000 (Danforth et al., 2000)).
- Reported 10-fold frame rate speed up (EuroGraphics 2001 (Murphy & Duchowski, 2001)).
- Described GPU-based video-based gaze-contingent display (TOMCCAP 2007 (A. T. Duchowski & Çöltekin, 2007)).

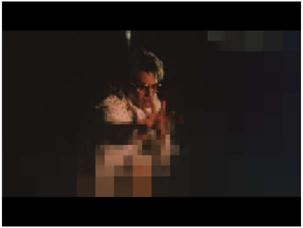
Pubs and pics



(Bob Danforth)



(Hunter Murphy)

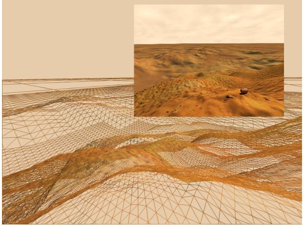


(Nathan Cournia)

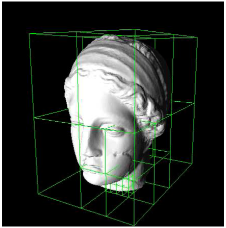
GCD examples.

- Showed feasibility of gaze-contingent terrain generation (Smart Graphics 2000 (Danforth et al., 2000)).
- Reported 10-fold frame rate speed up (EuroGraphics 2001 (Murphy & Duchowski, 2001)).
- Described GPU-based video-based gaze-contingent display (TOMCCAP 2007 (A. T. Duchowski & Çöltekin, 2007)).

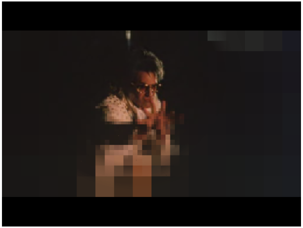
Pubs and pics



(Bob Danforth)



(Hunter Murphy)



(Nathan Cournia)

GCD examples.

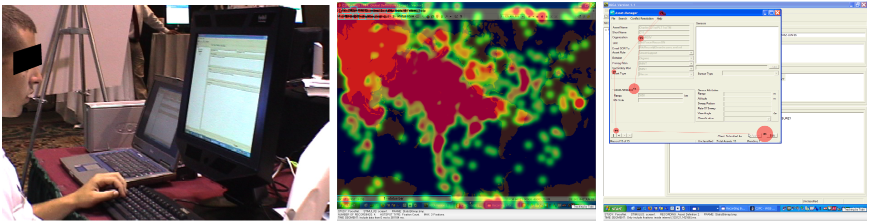
- Showed feasibility of gaze-contingent terrain generation (Smart Graphics 2000 (Danforth et al., 2000)).
- Reported 10-fold frame rate speed up (EuroGraphics 2001 (Murphy & Duchowski, 2001)).
- Described GPU-based video-based gaze-contingent display (TOMCCAP 2007 (A. T. Duchowski & Çöltekin, 2007)).

In Virutal Reality

Deictic reference in a CVE.

- Demonstrated advantage of visual deictic reference in VR for training (ETRA 2004 (A. T. Duchowski et al., 2004)).

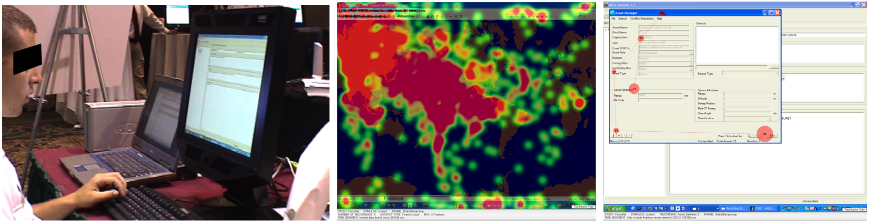
Usability engineering



ForceNet usability study and results.

- Collaboration between the military (Navy), industry (EMA), and academia.
- Used eye movements to evaluate user interfaces.
- Navy expects to apply to design of Command & Control (C2) applications.

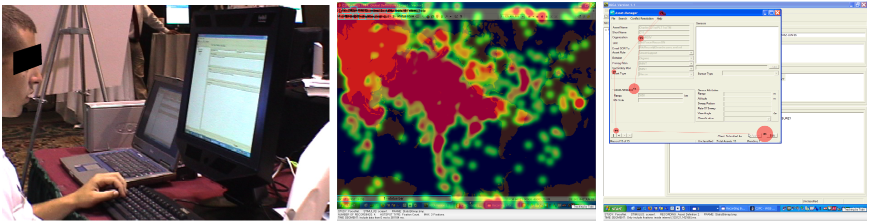
Usability engineering



ForceNet usability study and results.

- Collaboration between the military (Navy), industry (EMA), and academia.
- Used eye movements to evaluate user interfaces.
- Navy expects to apply to design of Command & Control (C2) applications.

Usability engineering

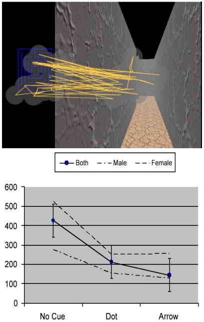


ForceNet usability study and results.

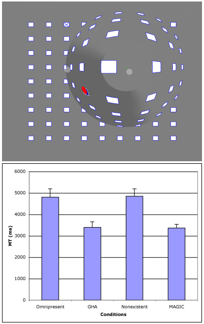
- Collaboration between the military (Navy), industry (EMA), and academia.
- Used eye movements to evaluate user interfaces.
- Navy expects to apply to design of Command & Control (C2) applications.

Eye Tracking Methodology class

- Quality of students' coursework is steadily improving.
- Scanpaths showed user attention switching to exocentric map during 3D navigation in VR (EGVE 2004 (Vembar et al., 2004)).
- Gaze-contingent fisheye lens was shown effective for Fitts' Law-type gaze selection task (GI 2005 (Ashmore et al., 2005)).



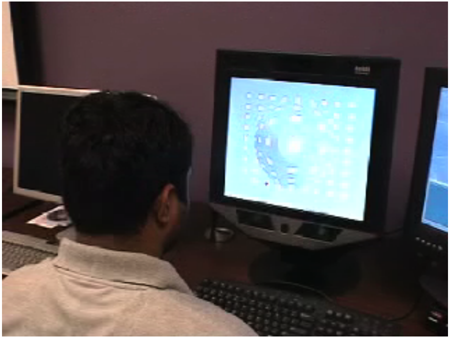
(Deepak Vembar)



(Mike Ashmore)

Students' examples.

Idelix PDT lens



Ashmore's GI 05 video.

Eye Tracking Research & Applications

- ETRA brings together top researchers from around the world, now in its 5th year.

ETRA proceedings covers.

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

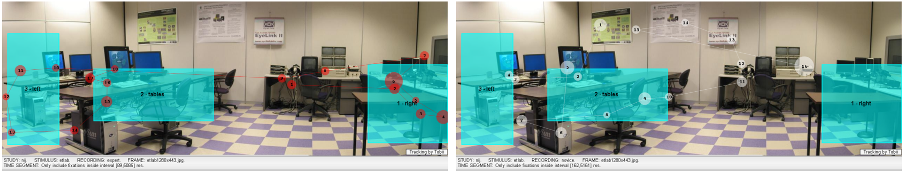
Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Mainly interactive in nature

- Scanpath comparison, average scanpath
- Accessibility
- Videoconferencing (with avatars)
- Gaze guidance for for education
- Peripheral color degradation
- Limbus detection for wearable eye tracker
- Kalman filtering for eye movement prediction

Motivation



Expert (left) and novice (right) scanpaths, captured at Clemson’s Eye Tracking Laboratory.

- Consider the scanpaths shown above.
- A single, quantifiable comparison metric is desirable.
- Solution: implementation of *scanpath comparison* algorithm based on string editing (Privitera & Stark, 2000).

Approach

- String editing was used by Privitera & Stark to compare human fixations with those predicted by automatic means (e.g., Itti’s saliency model (Itti et al., 1998)).

$s_1 = abcfeffgdc$ $s_2 = afbffdcdf$	`start`	`cost 0`
$s_1 = abcfeffgdc$ $s_2 = afeffdcdf$	`after substitution of first` b `by` e	`cost 1`
$s_1 = abcfeffgdc$ $s_2 = abcfeffdcdf$	`after insertion of` bc `after first` a	`cost 2`
$s_1 = abcfeffgdc$ $s_2 = abcfeffdc$	`after deletion of last` df	`cost 2`
$s_1 = abcfeffgdc$ $s_2 = abcfeffgdc$	`after insertion of` g	`cost 1`

Example of string editing (Privitera & Stark, 2000).

String editing framework

- Comparison relies on cost of three character operations: *deletion*, *insertion*, and *substitution*.
- Character transformation costs are based on Levenshtein distance, as illustrated below for strings $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$.

	a	f	b	f	f	d	c	d	f
a	0	1	2	3	4	5	6	7	8
b	1	1	1	2	3	4	5	6	7
c	2	2	2	2	3	4	4	5	6
f	3	2	3	2	2	3	4	5	5
e	4	3	3	3	3	3	4	5	6
f	5	4	4	3	3	4	4	5	5
f	6	5	5	4	3	4	5	5	5
g	7	6	6	5	4	4	5	6	6
d	8	7	7	6	5	4	5	5	6
c	9	8	8	7	6	5	4	5	6

Example of Levenshtein distance calculation.

String editing framework

- Comparison relies on cost of three character operations: *deletion*, *insertion*, and *substitution*.
- Character transformation costs are based on Levenshtein distance, as illustrated below for strings $s_1 = abcfeffgdc$ and $s_2 = afbffdcdf$.

	a	f	b	f	f	d	c	d	f
a	0	1	2	3	4	5	6	7	8
b	1	1	1	2	3	4	5	6	7
c	2	2	2	2	3	4	4	5	6
f	3	2	3	2	2	3	4	5	5
e	4	3	3	3	3	3	4	5	6
f	5	4	4	3	3	4	4	5	5
f	6	5	5	4	3	4	5	5	5
g	7	6	6	5	4	4	5	6	6
d	8	7	7	6	5	4	5	5	6
c	9	8	8	7	6	5	4	5	6

Example of Levenshtein distance calculation.

String editing implementation

- Implementation relies on constructing an array:

$$A[i][j] = min(A[i-1][j]+1, A[i][j-1]+1, A[i-1][j-1]+c(i,j))$$

where

$$c(i,j) = \begin{cases} 0, & s_1[i-1] = s_2[j-1] \\ 1, & \text{otherwise} \end{cases}$$

- Similarity coefficients are stored in a table, the Y-matrix, with one row and column for each image viewed by a subject (for all subjects).

String editing implementation

- Implementation relies on constructing an array:

  $$A[i][j] = min(A[i-1][j]+1, A[i][j-1]+1, A[i-1][j-1]+c(i,j))$$

  where

  $$c(i,j) = \begin{cases} 0, & s_1[i-1] = s_2[j-1] \\ 1, & \text{otherwise} \end{cases}$$

- Similarity coefficients are stored in a table, the Y-matrix, with one row and column for each image viewed by a subject (for all subjects).

Y-matrix structure

S_p	Subj. 1		Subj. 2	
	Pict1	Pict2	Pict 1	Pict 2
S1P1	**R**	**I**	**L**	**G**
S1P2		**R**	**G**	**L**
S2P1			**R**	**I**
S2P2				**R**

S_s	Subj. 1		Subj. 2	
	Pict1	Pict2	Pict 1	Pict 2
S1P1	**R**	**I**	**L**	**G**
S1P2		**R**	**G**	**L**
S2P1			**R**	**I**
S2P2				**R**

Structure of Y-matrices (Privitera & Stark, 2000).

- Since complete Y-matrices are too large for display, *parsing diagrams* are used to condense the data.

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

Same Subj.	Diff. Subj.		Same Subj.	Diff. Subj.
Repetitive	**L**ocal	← Same Image →	**R**epetitive	**L**ocal
Idiosyncratic	**G**lobal	← Diff. Image →	**I**diosyncratic	**G**lobal
S_p	*Ra*ndom		S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

Same Subj.	Diff. Subj.		Same Subj.	Diff. Subj.
Repetitive	**L**ocal	← Same Image →	**R**epetitive	**L**ocal
Idiosyncratic	**G**lobal	← Diff. Image →	**I**diosyncratic	**G**lobal
S_p	*Ra*ndom		S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

Same Subj.	Diff. Subj.		Same Subj.	Diff. Subj.
Repetitive	**L**ocal	← Same Image →	**R**epetitive	**L**ocal
Idiosyncratic	**G**lobal	← Diff. Image →	**I**diosyncratic	**G**lobal
S_p	*Ra*ndom		S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

Same Subj.	Diff. Subj.		Same Subj.	Diff. Subj.
Repetitive	**L**ocal	← Same Image →	**R**epetitive	**L**ocal
Idiosyncratic	**G**lobal	← Diff. Image →	**I**diosyncratic	**G**lobal
S_p	*Ra*ndom		S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

Same Subj.	Diff. Subj.		Same Subj.	Diff. Subj.
Repetitive	**L**ocal	← Same Image →	**R**epetitive	**L**ocal
Idiosyncratic	**G**lobal	← Diff. Image →	**I**diosyncratic	**G**lobal
S_p	*Ra*ndom		S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Parsing diagrams

- Parsing diagrams contain averages of similarity coefficients collected from Y-matrices and consist of four main entries:
  1. **R**epetitive: same viewer looking at same scene at different times;
  2. **L**ocal: different viewers looking at the same scene;
  3. **I**diosyncratic: same viewer looking at different scenes; and
  4. **G**lobal: different viewers looking at different scenes.
- A *Ra*ndom entry is included for significance testing.

	Same Subj.	Diff. Subj.
← Same Image →	**R**epetitive	**L**ocal
← Diff. Image →	**I**diosyncratic	**G**lobal
	S_p	*Ra*ndom

Same Subj.	Diff. Subj.
Repetitive	**L**ocal
Idiosyncratic	**G**lobal
S_s	*Ra*ndom

Structure of parsing diagrams (Privitera & Stark, 2000).

Related Work

- String editing has been employed in several studies where scanpath comparison was required, e.g.,
  - evaluation of web page design and television news layout (Josephson and Holmes, 2002; 2006),
  - amalgamation of numerous scanpaths into a single, representative (e.g., “expert”) scanpath (Hembrooke et al., 2006),
  - augmentation of string editing with Needleman-Wunsch weighting (West et al., 2006) (often used in bioinformatics); resultant *eyePatterns* is (not yet) freely available at: <http://www.juliamae.com/eyepatterns/>
- Common limitation of previous approaches is reliance on *a priori* registration of Regions Of Interest, ROIs, or fixation clusters.

Related Work

- String editing has been employed in several studies where scanpath comparison was required, e.g.,
  - evaluation of web page design and television news layout (Josephson and Holmes, 2002; 2006),
  - amalgamation of numerous scanpaths into a single, representative (e.g., “expert”) scanpath (Hembrooke et al., 2006),
  - augmentation of string editing with Needleman-Wunsch weighting (West et al., 2006) (often used in bioinformatics); resultant *eyePatterns* is (not yet) freely available at: `<http://www.juliamae.com/eyepatterns/>`
- Common limitation of previous approaches is reliance on *a priori* registration of Regions Of Interest, ROIs, or fixation clusters.

Related Work

- String editing has been employed in several studies where scanpath comparison was required, e.g.,
  - evaluation of web page design and television news layout (Josephson and Holmes, 2002; 2006),
  - amalgamation of numerous scanpaths into a single, representative (e.g., “expert”) scanpath (Hembrooke et al., 2006),
  - augmentation of string editing with Needleman-Wunsch weighting (West et al., 2006) (often used in bioinformatics); resultant *eyePatterns* is (not yet) freely available at: `<http://www.juliamae.com/eyepatterns/>`
- Common limitation of previous approaches is reliance on *a priori* registration of Regions Of Interest, ROIs, or fixation clusters.

Related Work

- String editing has been employed in several studies where scanpath comparison was required, e.g.,
  - evaluation of web page design and television news layout (Josephson and Holmes, 2002; 2006),
  - amalgamation of numerous scanpaths into a single, representative (e.g., “expert”) scanpath (Hembrooke et al., 2006),
  - augmentation of string editing with Needleman-Wunsch weighting (West et al., 2006) (often used in bioinformatics); resultant *eyePatterns* is (not yet) freely available at: `<http://www.juliamae.com/eyepatterns/>`
- Common limitation of previous approaches is reliance on *a priori* registration of Regions Of Interest, ROIs, or fixation clusters.

Related Work

- String editing has been employed in several studies where scanpath comparison was required, e.g.,
  - evaluation of web page design and television news layout (Josephson and Holmes, 2002; 2006),
  - amalgamation of numerous scanpaths into a single, representative (e.g., “expert”) scanpath (Hembrooke et al., 2006),
  - augmentation of string editing with Needleman-Wunsch weighting (West et al., 2006) (often used in bioinformatics); resultant *eyePatterns* is (not yet) freely available at: `<http://www.juliamae.com/eyepatterns/>`
- Common limitation of previous approaches is reliance on *a priori* registration of Regions Of Interest, ROIs, or fixation clusters.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_S, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_S, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_S, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on *k*-means clustering, which requires *a priori* estimation of *k* (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_s, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_s, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_s, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Implementation

- Automatic fixation clustering is required.
- Privitera and Stark (2000) relied on k-means clustering, which requires *a priori* estimation of k (number of clusters).
- Santella and DeCarlo (2004) presented an alternative with *mean shift* fixation clustering.
- Mean shift's only existing user-adjustable parameter is σ_s, which can epistemically be set to match the extent of the retinal dimension of the human fovea (about 5° visual angle).
- The mean shift process is composed of two stages:
  1. Shift fixations into denser configurations until they can be easily separated,
  2. Administer a distance threshold clustering algorithm.

Iteration

- The process continues by repeatedly moving a point $\mathbf{x}_i$ to a new location, $\mathbf{s}(\mathbf{x}_i)$, the weighted mean of nearby points, based on the kernel function K:

$$\mathbf{s}(\mathbf{x}_i) = \frac{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)\mathbf{x}_j}{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)}$$

  where K is typically a multivariate zero-mean Gaussian and covariance $\sigma^2\mathbf{I}$.

- Applying the above to fixations, the following zero-mean spatiotemporal Gaussian kernel can be used:

$$K([\mathbf{x}_i, t_i]) = \exp\left(\frac{x_i^2 + y_i^2}{\sigma_s^2} + \frac{t_i^2}{\sigma_t^2}\right)$$

  where σ_s and σ_t determine local support of the Gaussian kernel in both spatial (dispersion) and temporal extent.

Iteration

- The process continues by repeatedly moving a point $\mathbf{x}_i$ to a new location, $\mathbf{s}(\mathbf{x}_i)$, the weighted mean of nearby points, based on the kernel function K:

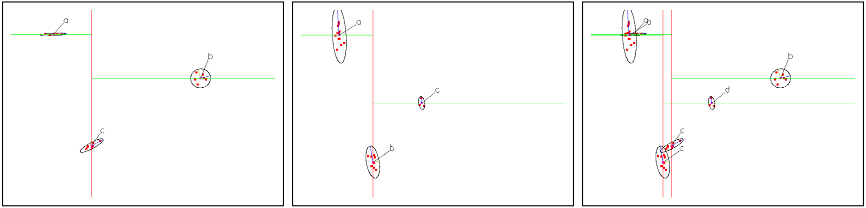
$$\mathbf{s}(\mathbf{x}_i) = \frac{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)\mathbf{x}_j}{\sum_j K(\mathbf{x}_i - \mathbf{x}_j)}$$

  where K is typically a multivariate zero-mean Gaussian and covariance $\sigma^2\mathbf{I}$.
- Applying the above to fixations, the following zero-mean spatiotemporal Gaussian kernel can be used:

$$K([\mathbf{x}_i, t_i]) = \exp\left(\frac{x_i^2 + y_i^2}{\sigma_s^2} + \frac{t_i^2}{\sigma_t^2}\right)$$

  where σ_s and σ_t determine local support of the Gaussian kernel in both spatial (dispersion) and temporal extent.

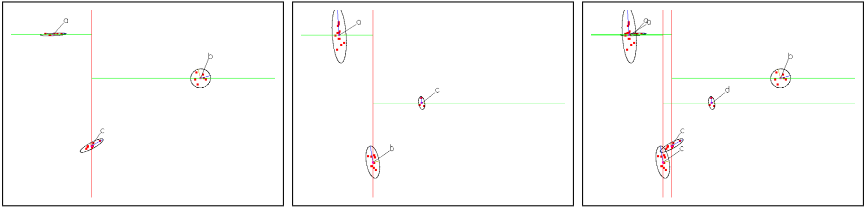
Preliminary Results



Distinct scanpaths with overlapping clusters.

- The left scanpath proceeds in a roughly clockwise order, the middle counter-clockwise.
- Without taking overlap into account, i.e., labeling the scanpaths independently, scanpath labels would be $s_1 = s_2 = abc$.
- Taking overlap and sequential order into account, the resulting labels are generated as $s_1 = abc$ and $s_2 = acd$, as shown at right, indicating (spatial) cluster overlap at two locations.

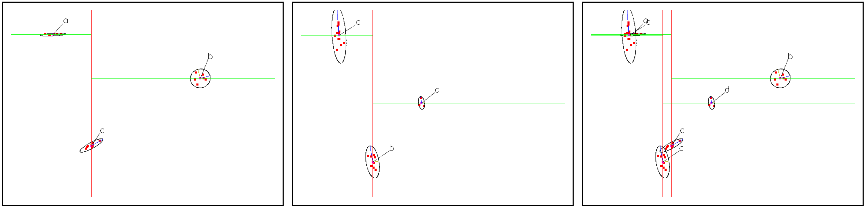
Preliminary Results



Distinct scanpaths with overlapping clusters.

- The left scanpath proceeds in a roughly clockwise order, the middle counter-clockwise.
- Without taking overlap into account, i.e., labeling the scanpaths independently, scanpath labels would be $s_1 = s_2 = abc$.
- Taking overlap and sequential order into account, the resulting labels are generated as $s_1 = abc$ and $s_2 = acd$, as shown at right, indicating (spatial) cluster overlap at two locations.

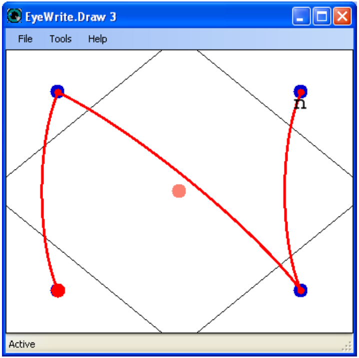
Preliminary Results



Distinct scanpaths with overlapping clusters.

- The left scanpath proceeds in a roughly clockwise order, the middle counter-clockwise.
- Without taking overlap into account, i.e., labeling the scanpaths independently, scanpath labels would be $s_1 = s_2 = abc$.
- Taking overlap and sequential order into account, the resulting labels are generated as $s_1 = abc$ and $s_2 = acd$, as shown at right, indicating (spatial) cluster overlap at two locations.

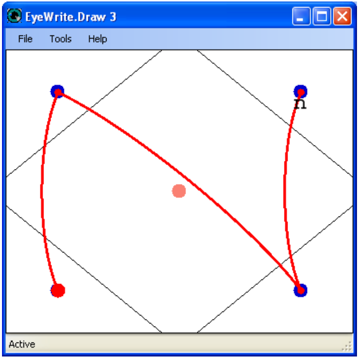
Motivation



Screenshot of EyeWrite: gestural interface (w/Jake Wobbrock @ UW).

- Gestures with gaze—like Palm Pilot's graffiti.
- Initial study involves eye typing, but other gestures for interface control are also possible.

Motivation



Screenshot of EyeWrite: gestural interface (w/Jake Wobbrock @ UW).

- Gestures with gaze—like Palm Pilot's graffiti.
- Initial study involves eye typing, but other gestures for interface control are also possible.

Hypothesis

1. EyeWrite should be faster/more accurate than eye typing with a keyboard of about the same size
2. EyeWrite's corner areas are larger targets than small keyboard keys.
3. Although each character stroke is longer than single keyboard key (4 per letter, on average), eye tracker inaccuracy and gaze jitter work in EyeWrite's favor in this instance.
4. EyeWrite should also take up less screen real-estate.

Hypothesis

1. EyeWrite should be faster/more accurate than eye typing with a keyboard of about the same size
2. EyeWrite's corner areas are larger targets than small keyboard keys.
3. Although each character stroke is longer than single keyboard key (4 per letter, on average), eye tracker inaccuracy and gaze jitter work in EyeWrite's favor in this instance.
4. EyeWrite should also take up less screen real-estate.

Hypothesis

1. EyeWrite should be faster/more accurate than eye typing with a keyboard of about the same size
2. EyeWrite's corner areas are larger targets than small keyboard keys.
3. Although each character stroke is longer than single keyboard key (4 per letter, on average), eye tracker inaccuracy and gaze jitter work in EyeWrite's favor in this instance.
4. EyeWrite should also take up less screen real-estate.

Hypothesis

1. EyeWrite should be faster/more accurate than eye typing with a keyboard of about the same size
2. EyeWrite's corner areas are larger targets than small keyboard keys.
3. Although each character stroke is longer than single keyboard key (4 per letter, on average), eye tracker inaccuracy and gaze jitter work in EyeWrite's favor in this instance.
4. EyeWrite should also take up less screen real-estate.

Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
   - he achieved this by performing several calibration sequences until he was happy with accuracy
   - he also very consciously maintained his head steady (acted like an "experienced" eye tracking user, which is somewhat unusual).

Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
   - he achieved this by performing several calibration sequences until he was happy with accuracy
   - he also very consciously maintained his head steady (acted like an "experienced" eye tracking user, which is somewhat unusual).

Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
    - he achieved this by performing several calibration sequences until he was happy with accuracy
    - he also very consciously maintained his head steady (acted like an "experienced" eye tracking user, which is somewhat unusual).

Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
   - he achieved this by performing several calibration sequences until he was happy with accuracy
   - he also very consciously maintained his head steady (acted like an "experienced" eye tracking user, which is somewhat unusual).

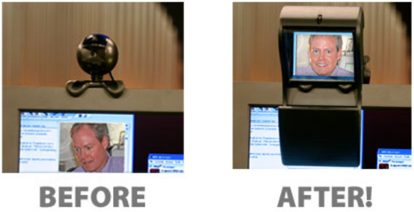
Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
   - he achieved this by performing several calibration sequences until he was happy with accuracy
   - he also very consciously maintained his head steady (acted like an "experienced" eye tracking user, which is somewhat unusual).

Pilot study

1. Just completing longitudinal study now.
2. Our expectations were that time was required to learn the alphabet.
3. Comments from one participant anecdotally suggest that keyboard is faster *initially*.
4. It should be noted that this particular user was intent on getting good accuracy
   - he achieved this by performing several calibration sequences until he was happy with accuracy
   - he also very consciously maintained his head steady (acted like an “experienced” eye tracking user, which is somewhat unusual).

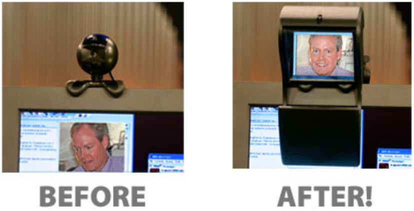
Motivation



Bodelin's "se2e" device.

- I saw an ad for this "Eye 2 Eye" webcam "teleprompter".
- The ad said this device lets you see the person you're talking to and that it's worth the $100.
- But it just looks like a simple periscope (and a rather bulky one).
- Eye contact in video conferencing is of course an old problem.
- There must be a better way...

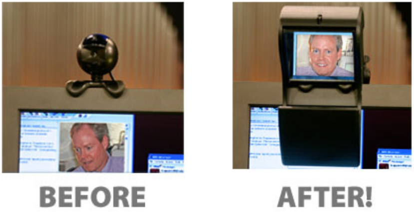
Motivation



Bodelin's "se2e" device.

- I saw an ad for this "Eye 2 Eye" webcam "teleprompter".
- The ad said this device lets you see the person you're talking to and that it's worth the $100.
- But it just looks like a simple periscope (and a rather bulky one).
- Eye contact in video conferencing is of course an old problem.
- There must be a better way...

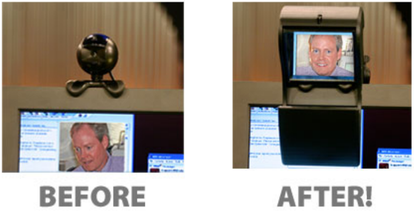
Motivation



Bodelin's "se2e" device.

- I saw an ad for this "Eye 2 Eye" webcam "teleprompter".
- The ad said this device lets you see the person you're talking to and that it's worth the $100.
- But it just looks like a simple periscope (and a rather bulky one).
- Eye contact in video conferencing is of course an old problem.
- There must be a better way...

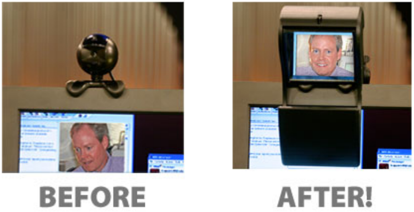
Motivation



Bodelin's "se2e" device.

- I saw an ad for this "Eye 2 Eye" webcam "teleprompter".
- The ad said this device lets you see the person you're talking to and that it's worth the $100.
- But it just looks like a simple periscope (and a rather bulky one).
- Eye contact in video conferencing is of course an old problem.
- There must be a better way...

Motivation



Bodelin's "se2e" device.

- I saw an ad for this "Eye 2 Eye" webcam "teleprompter".
- The ad said this device lets you see the person you're talking to and that it's worth the $100.
- But it just looks like a simple periscope (and a rather bulky one).
- Eye contact in video conferencing is of course an old problem.
- There must be a better way...

Hypothesis

1. Gaze-driven avatars should “solve” eye contact problem.
2. Just need good avatars.
3. Ideally, each person should have a locally-stored texture of their face that can be mapped onto an MPEG-4 “talking head” 3D facial skeleton object.
4. Note that MPEG-4’s head object already has sub-specifications for eyeball objects—they just need to rotate.

Hypothesis

1. Gaze-driven avatars should “solve” eye contact problem.
2. Just need good avatars.
3. Ideally, each person should have a locally-stored texture of their face that can be mapped onto an MPEG-4 “talking head” 3D facial skeleton object.
4. Note that MPEG-4’s head object already has sub-specifications for eyeball objects—they just need to rotate.

Hypothesis

1. Gaze-driven avatars should "solve" eye contact problem.
2. Just need good avatars.
3. Ideally, each person should have a locally-stored texture of their face that can be mapped onto an MPEG-4 "talking head" 3D facial skeleton object.
4. Note that MPEG-4's head object already has sub-specifications for eyeball objects—they just need to rotate.

Hypothesis

1. Gaze-driven avatars should “solve” eye contact problem.
2. Just need good avatars.
3. Ideally, each person should have a locally-stored texture of their face that can be mapped onto an MPEG-4 “talking head” 3D facial skeleton object.
4. Note that MPEG-4’s head object already has sub-specifications for eyeball objects—they just need to rotate.

Pilot study

Avatar conferencing and hotspots.

1. Results inconclusive.
2. No apparent difference between fixed gaze, networked gaze, and random gaze.
3. Study needs to be re-run...

Pilot study

Avatar conferencing and hotspots.

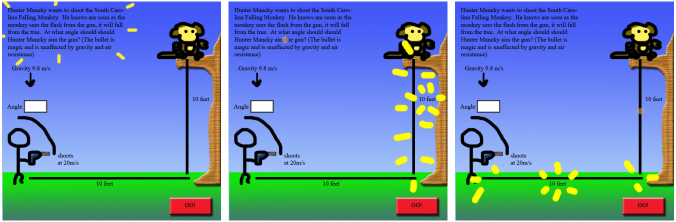
1. Results inconclusive.
2. No apparent difference between fixed gaze, networked gaze, and random gaze.
3. Study needs to be re-run...

Pilot study

Avatar conferencing and hotspots.

1. Results inconclusive.
2. No apparent difference between fixed gaze, networked gaze, and random gaze.
3. Study needs to be re-run...

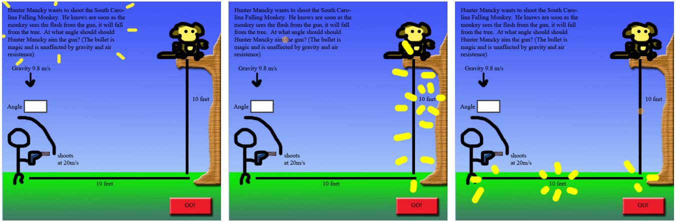
Motivation



In this storyboard for a gaze-guiding adventure physics lesson, the hunter wants to shoot the monkey. As soon as the monkey sees the flash from the gun, it will fall from the tree. At what angle should the hunter aim the gun?

- Objective is to guide attention to the essential components of the problem.
- Initial angle is obtained via $\tan^{-1}(\text{height}/\text{distance})$.

Motivation



In this storyboard for a gaze-guiding adventure physics lesson, the hunter wants to shoot the monkey. As soon as the monkey sees the flash from the gun, it will fall from the tree. At what angle should the hunter aim the gun?

- Objective is to guide attention to the essential components of the problem.
- Initial angle is obtained via $\tan^{-1}(\text{height}/\text{distance})$.

Hypothesis

The proposed research examines the following questions:

1. Does gaze-guidance help manage attentional distribution and improve learning?
2. Does gaze-guidance influence affective responses of students?
3. How do individual differences in cognitive ability, prior knowledge, and spatial visualization ability interact with the effects of cueing?

Hypothesis

The proposed research examines the following questions:

1. Does gaze-guidance help manage attentional distribution and improve learning?
2. Does gaze-guidance influence affective responses of students?
3. How do individual differences in cognitive ability, prior knowledge, and spatial visualization ability interact with the effects of cueing?

Hypothesis

The proposed research examines the following questions:

1. Does gaze-guidance help manage attentional distribution and improve learning?
2. Does gaze-guidance influence affective responses of students?
3. How do individual differences in cognitive ability, prior knowledge, and spatial visualization ability interact with the effects of cueing?

Pilot Study

1. No significant performance results yet...
2. Low power experiment ($n = 10$, between-subjects).
3. Some trends evident for the harder physics problems.
4. Four of five participants indicated they found gaze-guidance helpful for finding or clarifying things in the picture.

Pilot Study

1. No significant performance results yet...
2. Low power experiment ($n = 10$, between-subjects).
3. Some trends evident for the harder physics problems.
4. Four of five participants indicated they found gaze-guidance helpful for finding or clarifying things in the picture.

Pilot Study

1. No significant performance results yet...
2. Low power experiment ($n = 10$, between-subjects).
3. Some trends evident for the harder physics problems.
4. Four of five participants indicated they found gaze-guidance helpful for finding or clarifying things in the picture.

Pilot Study

1. No significant performance results yet...
2. Low power experiment ($n = 10$, between-subjects).
3. Some trends evident for the harder physics problems.
4. Four of five participants indicated they found gaze-guidance helpful for finding or clarifying things in the picture.

Motivation

Real-time spatiochromatic peripheral degradation.

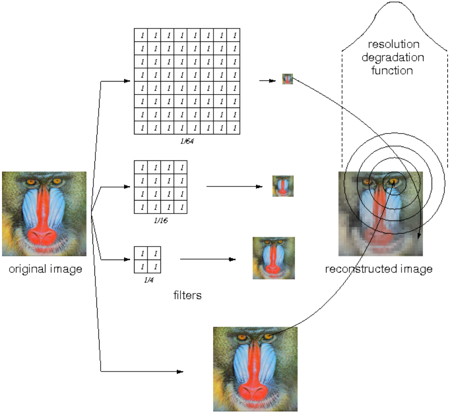
- Peripheral color degradation has not been widely studied.
- Are there implications for computer graphics or compression?

Motivation

Real-time spatiochromatic peripheral degradation.

- Peripheral color degradation has not been widely studied.
- Are there implications for computer graphics or compression?

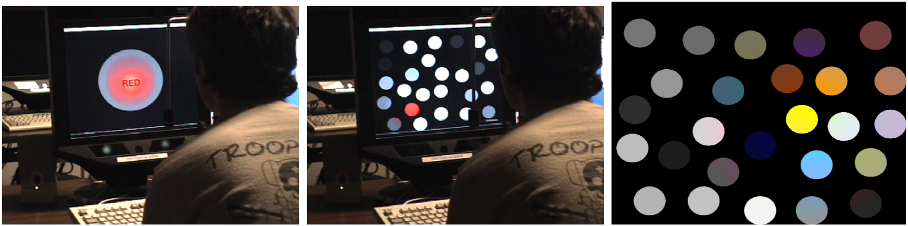
Approach



Per-fragment mipmap LOD bias selection illustration.

- GPU implementation is now trivial.

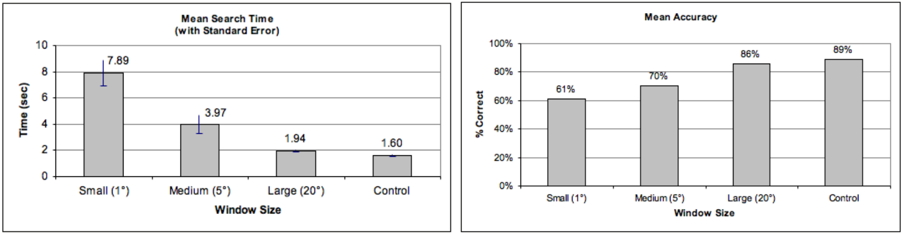
Experiment



Physical setup. Participant viewing red target (left) with a 5° gaze-contingent color mask and locating the target in color-degraded field (middle), GCD (right).

- The independent variable was visual window size, visual angle 1°, 5°, 20°, and control.

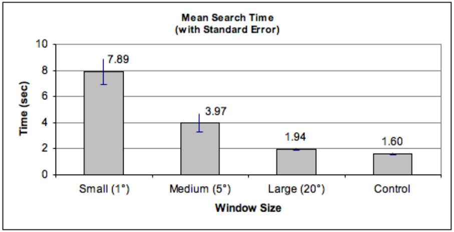
Results

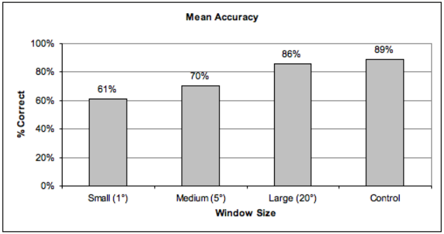


Mean search time (in sec; left) and accuracy (right) vs. window size.

- Reduction of peripheral chrominance significantly degrades search performance when window is $\leq 5^\circ$.
- For visual search, vision is more sensitive to peripheral color than spatial detail.
- To achieve performance comparable to that of a uniform color display, a window of 20° is required.

Results

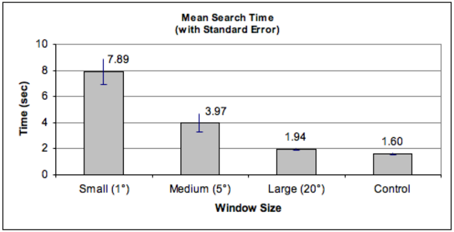


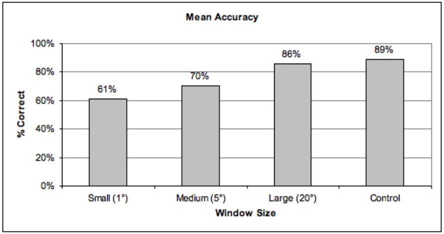


Mean search time (in sec; left) and accuracy (right) vs. window size.

- Reduction of peripheral chrominance significantly degrades search performance when window is $\leq 5^\circ$.
- For visual search, vision is more sensitive to peripheral color than spatial detail.
- To achieve performance comparable to that of a uniform color display, a window of 20° is required.

Results

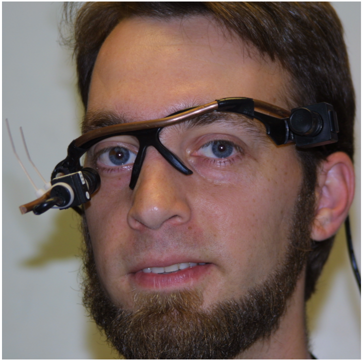




Mean search time (in sec; left) and accuracy (right) vs. window size.

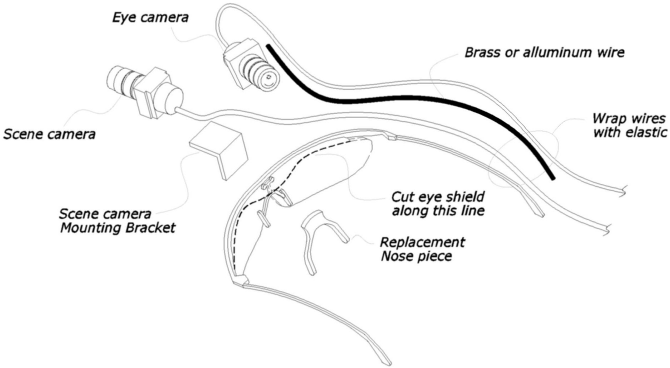
- Reduction of peripheral chrominance significantly degrades search performance when window is $\leq 5^\circ$.
- For visual search, vision is more sensitive to peripheral color than spatial detail.
- To achieve performance comparable to that of a uniform color display, a window of 20° is required.

Motivation



Our “DIY” wearable eye tracker from relatively inexpensive (∼$700) off-the-shelf components.

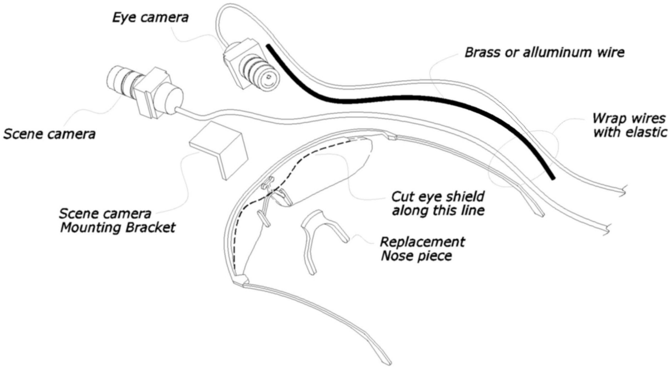
Approach



Eye tracker assembly.

- Basically same hardware design described by Li (2006).
- Inexpensive commercial off-the-shelf (COTS) components.
- Digital video minicams from DejaView (Reich et al., 2004).

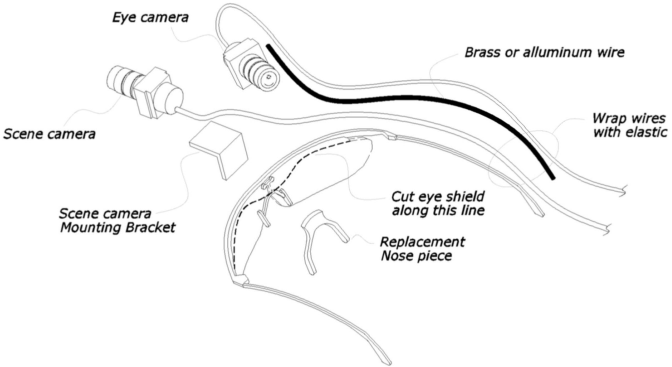
Approach



Eye tracker assembly.

- Basically same hardware design described by Li (2006).
- Inexpensive commercial off-the-shelf (COTS) components.
- Digital video minicams from DejaView (Reich et al., 2004).

Approach

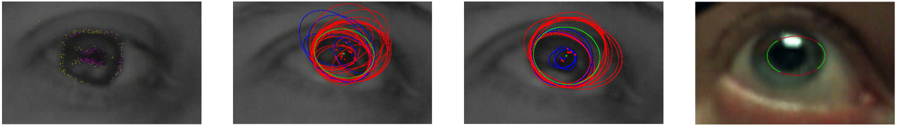


Eye tracker assembly.

- Basically same hardware design described by Li (2006).
- Inexpensive commercial off-the-shelf (COTS) components.
- Digital video minicams from DejaView (Reich et al., 2004).

Approach

- Erroneous ellipses spannnig both pupil and limbus are generated by fitting to all feature points.
- Distinguishing between points on the pupil and limbus reduces number of erroneous ellipses.



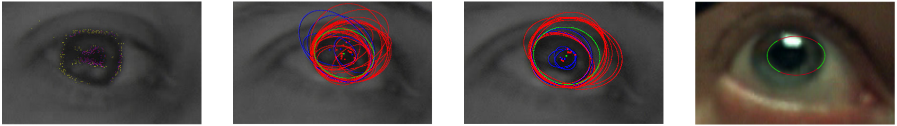
(a) (b) (c) (d)

Ellipse fitting.

(a) *Detection of feature points on both pupil and limbus.*
(b) *Ellipses fit to sets of 5 point selecte at random.*
(c) *Luminance-based delineation of feature points.*
(d) *Proper detection of erroneous ellipses.*

Approach

- Erroneous ellipses spannnig both pupil and limbus are generated by fitting to all feature points.
- Distinguishing between points on the pupil and limbus reduces number of erroneous ellipses.



(a) (b) (c) (d)

Ellipse fitting.

(a) *Detection of feature points on both pupil and limbus.*
(b) *Ellipses fit to sets of 5 point selecte at random.*
(c) *Luminance-based delineation of feature points.*
(d) *Proper detection of erroneous ellipses.*

Signal modeling

- For networked transmission of eye movement coordinates, first-order velocity-based prediction is required.
- To do so, use the discrete Kalman filter for human interactive motion tracking and adapt a similar model for eye movements.
- The discrete Kalman filter is defined by the state-space model of eye movement motion,

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k,$$

and the discrete-time process observation (measurement),

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{e}_k,$$

where (5 definitions follow):

Signal modeling

- For networked transmission of eye movement coordinates, first-order velocity-based prediction is required.
- To do so, use the discrete Kalman filter for human interactive motion tracking and adapt a similar model for eye movements.
- The discrete Kalman filter is defined by the state-space model of eye movement motion,

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k,$$

and the discrete-time process observation (measurement),

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{e}_k,$$

where (5 definitions follow):

Signal modeling

- For networked transmission of eye movement coordinates, first-order velocity-based prediction is required.
- To do so, use the discrete Kalman filter for human interactive motion tracking and adapt a similar model for eye movements.
- The discrete Kalman filter is defined by the state-space model of eye movement motion,

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k,$$

and the discrete-time process observation (measurement),

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{e}_k,$$

where (5 definitions follow):

Signal modeling

1. $\mathbf{x}_k$ is the $(n \times 1)$ vector state of the eye movement discrete-time process at time t_k modeling the current gaze position and velocity,

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix}$$

2. $\mathbf{\Phi}_k$ describes the $(n \times n)$ state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
3. $\mathbf{w}_k$ is the $(n \times 1)$ timewise uncorrelated zero-mean white noise sequence with known covariance structure,
4. $\mathbf{e}_k$ represents the $(m \times 1)$ measurement error (assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence), and
5. $\mathbf{H}_k$ describes the $(m \times n)$ ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time t_k,

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Signal modeling

1. $\mathbf{x}_k$ is the $(n \times 1)$ vector state of the eye movement discrete-time process at time t_k modeling the current gaze position and velocity,

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix}$$

2. $\mathbf{\Phi}_k$ describes the $(n \times n)$ state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
3. $\mathbf{w}_k$ is the $(n \times 1)$ timewise uncorrelated zero-mean white noise sequence with known covariance structure,
4. $\mathbf{e}_k$ represents the $(m \times 1)$ measurement error (assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence), and
5. $\mathbf{H}_k$ describes the $(m \times n)$ ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time t_k,

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Signal modeling

1. $\mathbf{x}_k$ is the ($n\times 1$) vector state of the eye movement discrete-time process at time t_k modeling the current gaze position and velocity,

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix}$$

2. $\mathbf{\Phi}_k$ describes the ($n\times n$) state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
3. $\mathbf{w}_k$ is the ($n\times 1$) timewise uncorrelated zero-mean white noise sequence with known covariance structure,
4. $\mathbf{e}_k$ represents the ($m\times 1$) measurement error (assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence), and
5. $\mathbf{H}_k$ describes the ($m\times n$) ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time t_k,

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Signal modeling

1. $\mathbf{x}_k$ is the $(n \times 1)$ vector state of the eye movement discrete-time process at time t_k modeling the current gaze position and velocity,

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix}$$

2. $\mathbf{\Phi}_k$ describes the $(n \times n)$ state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
3. $\mathbf{w}_k$ is the $(n \times 1)$ timewise uncorrelated zero-mean white noise sequence with known covariance structure,
4. $\mathbf{e}_k$ represents the $(m \times 1)$ measurement error (assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence), and
5. $\mathbf{H}_k$ describes the $(m \times n)$ ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time t_k,

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Signal modeling

1. $\mathbf{x}_k$ is the $(n \times 1)$ vector state of the eye movement discrete-time process at time t_k modeling the current gaze position and velocity,

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{pmatrix}$$

2. $\mathbf{\Phi}_k$ describes the $(n \times n)$ state transition matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function,
3. $\mathbf{w}_k$ is the $(n \times 1)$ timewise uncorrelated zero-mean white noise sequence with known covariance structure,
4. $\mathbf{e}_k$ represents the $(m \times 1)$ measurement error (assumed to be a white sequence with known covariance structure and uncorrelated with the $\mathbf{w}_k$ sequence), and
5. $\mathbf{H}_k$ describes the $(m \times n)$ ideal (noiseless) connection between the state vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at time t_k,

$$\mathbf{H}_k = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Signal filtering

The Kalman recursion is composed of 5 computation steps:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

   where:
   (a) $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and
   (b) $\mathbf{R}_k$ is the measurement error covarianace matrix,
2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

   where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ is the difference of the real measurement and the measurable components of the process state vector.
3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-.$$

Signal filtering

The Kalman recursion is composed of 5 computation steps:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

   where:
   (a) $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and
   (b) $\mathbf{R}_k$ is the measurement error covarianace matrix,
2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

   where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ is the difference of the real measurement and the measurable components of the process state vector.
3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-.$$

Signal filtering

The Kalman recursion is composed of 5 computation steps:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

   where:
   (a) $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and
   (b) $\mathbf{R}_k$ is the measurement error covarianace matrix,

2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

   where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ is the difference of the real measurement and the measurable components of the process state vector.

3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-.$$

Signal filtering

The Kalman recursion is composed of 5 computation steps:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

   where:

   (a) $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and

   (b) $\mathbf{R}_k$ is the measurement error covarianace matrix,

2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

   where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ is the difference of the real measurement and the measurable components of the process state vector.

3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-.$$

Signal filtering

The Kalman recursion is composed of 5 computation steps:

1. Compute gain (blending factor):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

   where:
   (a) $\mathbf{P}_k$ is the covariance matrix of the estimation error (the negative superscript denotes projection from the previous step; see below), and
   (b) $\mathbf{R}_k$ is the measurement error covarianace matrix,

2. Update estimate with measurement $\mathbf{z}_k$ (correction step):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-)$$

   where $\mathbf{z}_k - \mathbf{H}_k \cdot \hat{\mathbf{x}}_k^-$ is the difference of the real measurement and the measurable components of the process state vector.

3. Update the error covariance matrix:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-.$$

Signal filtering

4. Project ahead (prediction step):

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{\Phi}_k \hat{\mathbf{x}}_k.$$

5. Obtain best estimate of the error covariance matrix:

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k,$$

where $\mathbf{Q}_k$ is the covariance matrix of the white acceleration,

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T].$$

For translational motion of constant velocity and random acceleration, derive:

$$\mathbf{Q}_k = \frac{a^2 \Delta t}{6} \begin{pmatrix} 2\mathbf{I}(\Delta t)^2 & 3\mathbf{I}\Delta t \\ 3\mathbf{I}\Delta t & 6\mathbf{I} \end{pmatrix} s,$$

where $\mathbf{I}$ is the 2×2 identity matrix, a is the spectral amplitude of the white noise, and s denotes unit seconds.

Signal filtering

4. Project ahead (prediction step):

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{\Phi}_k \hat{\mathbf{x}}_k.$$

5. Obtain best estimate of the error covariance matrix:

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k,$$

where $\mathbf{Q}_k$ is the covariance matrix of the white acceleration,

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T].$$

For translational motion of constant velocity and random acceleration, derive:

$$\mathbf{Q}_k = \frac{a^2 \Delta t}{6} \begin{pmatrix} 2\mathbf{I}(\Delta t)^2 & 3\mathbf{I}\Delta t \\ 3\mathbf{I}\Delta t & 6\mathbf{I} \end{pmatrix} s,$$

where $\mathbf{I}$ is the 2×2 identity matrix, a is the spectral amplitude of the white noise, and s denotes unit seconds.

Parameter estimation

- Eye movement analysis (characterization into fixations, saccades, and smooth pursuits), as well as prediction, by the Kalman filter depends on the appropriate specification of the stochastic process, described by the matrices $\mathbf{P}_k$, $\mathbf{Q}_k$, and $\mathbf{R}_k$.
- The recursive update of the matrix $\mathbf{P}_k$ at time t_k relies on the *a priori* estimates $\mathbf{P}_k^-$ prior to assimilation of the measurement at time t_k and the initial determination of $\mathbf{P}_0^-$.
- The estimation error is defined as $\mathbf{x}_k - \hat{\mathbf{x}}_k^-$ where vector $\mathbf{x}_k$ is the (unknown) ideal process state vector. The associated error covariance matrix is then

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T].$$

Parameter estimation

- Eye movement analysis (characterization into fixations, saccades, and smooth pursuits), as well as prediction, by the Kalman filter depends on the appropriate specification of the stochastic process, described by the matrices $\mathbf{P}_k$, $\mathbf{Q}_k$, and $\mathbf{R}_k$.
- The recursive update of the matrix $\mathbf{P}_k$ at time t_k relies on the *a priori* estimates $\mathbf{P}_k^-$ prior to assimilation of the measurement at time t_k and the initial determination of $\mathbf{P}_0^-$.
- The estimation error is defined as $\mathbf{x}_k - \hat{\mathbf{x}}_k^-$ where vector $\mathbf{x}_k$ is the (unknown) ideal process state vector. The associated error covariance matrix is then

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T].$$

Parameter estimation

- Eye movement analysis (characterization into fixations, saccades, and smooth pursuits), as well as prediction, by the Kalman filter depends on the appropriate specification of the stochastic process, described by the matrices $\mathbf{P}_k$, $\mathbf{Q}_k$, and $\mathbf{R}_k$.
- The recursive update of the matrix $\mathbf{P}_k$ at time t_k relies on the *a priori* estimates $\mathbf{P}_k^-$ prior to assimilation of the measurement at time t_k and the initial determination of $\mathbf{P}_0^-$.
- The estimation error is defined as $\mathbf{x}_k - \hat{\mathbf{x}}_k^-$ where vector $\mathbf{x}_k$ is the (unknown) ideal process state vector. The associated error covariance matrix is then

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T].$$

Parameter estimation

- Since $\mathbf{P}_k^-$ is updated during the Kalman recursion, it is sufficient to determine $\mathbf{P}_0^-$, although generally this is a difficult task. For the discrete-time Kalman model of human motion, $\mathbf{P}_0^-$ takes the form

$$\mathbf{P}_0^- = \begin{pmatrix} \sigma_{p,p}^2 & \sigma_{p,v}^2 \\ \sigma_{p,v}^2 & \sigma_{v,v}^2 \end{pmatrix}$$

  where $\sigma_{p,p}$ is the standard deviation of the position estimation error and $\sigma_{p,v}^2 = E[(\mathbf{p}_k - \hat{\mathbf{p}}_k^-)(\mathbf{v}_k - \hat{\mathbf{v}}_k^-)]$ since $\mathbf{x}_k = (\mathbf{p}_k, \mathbf{v}_k)^T$.
- Over time, $E[\mathbf{p}_k - \hat{\mathbf{p}}_k^-] = E[\mathbf{v}_k - \hat{\mathbf{v}}_k^-] = 0$ and position and velocity estimation errors can be assumed to be independent and uncorrelated, leading to $\sigma_{p,v} = 0$ (Kohler, 1997).

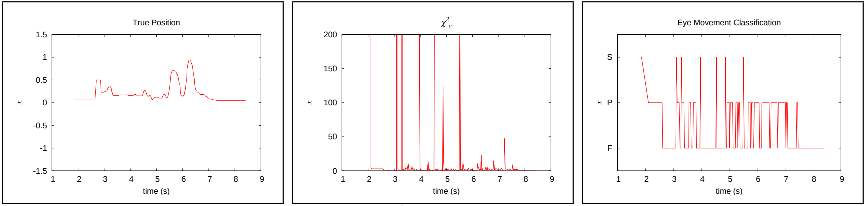
Parameter estimation

- Since $\mathbf{P}_k^-$ is updated during the Kalman recursion, it is sufficient to determine $\mathbf{P}_0^-$, although generally this is a difficult task. For the discrete-time Kalman model of human motion, $\mathbf{P}_0^-$ takes the form

$$\mathbf{P}_0^- = \begin{pmatrix} \sigma_{p,p}^2 & \sigma_{p,v}^2 \\ \sigma_{p,v}^2 & \sigma_{v,v}^2 \end{pmatrix}$$

  where $\sigma_{p,p}$ is the standard deviation of the position estimation error and $\sigma_{p,v}^2 = E[(\mathbf{p}_k - \hat{\mathbf{p}}_k^-)(\mathbf{v}_k - \hat{\mathbf{v}}_k^-)]$ since $\mathbf{x}_k = (\mathbf{p}_k, \mathbf{v}_k)^T$.
- Over time, $E[\mathbf{p}_k - \hat{\mathbf{p}}_k^-] = E[\mathbf{v}_k - \hat{\mathbf{v}}_k^-] = 0$ and position and velocity estimation errors can be assumed to be independent and uncorrelated, leading to $\sigma_{p,v} = 0$ (Kohler, 1997).

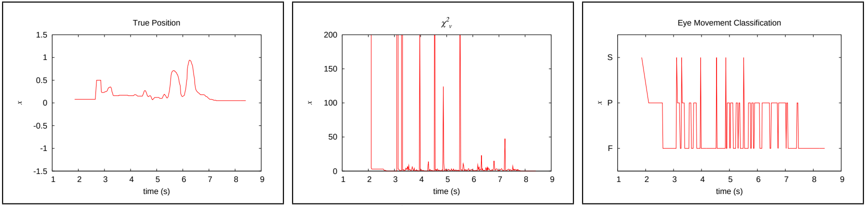
Preliminary results



True x-position, $\chi^2(dx/dt)$, classification.

- Filter appears to be working fairly well except during saccadic intervals—tends to overshoot prediction.
- Classification appears possible with position-variance used to detect fixations first, then check “velocity-variance” to disambiguate saccades from smooth pursuits.

Preliminary results



True x-position, $\chi^2(dx/dt)$, classification.

- Filter appears to be working fairly well except during saccadic intervals—tends to overshoot prediction.
- Classification appears possible with position-variance used to detect fixations first, then check "velocity-variance" to disambiguate saccades from smooth pursuits.

For Further Reading I

Ashmore, M., Duchowski, A. T., & Shoemaker, G. (2005). Efficient Eye Pointing with a Fisheye Lens. In *Proceedings of Graphics Interface.* Victoria, BC, Canada.

Danforth, R., Duchowski, A., Geist, R., & McAliley, E. (2000). A Platform for Gaze-Contingent Virtual Environments. In *Smart Graphics (Papers from the 2000 AAAI Spring Symposium, Technical Report SS-00-04)* (p. 66-70). Menlo Park, CA.

Duchowski, A., Medlin, E., Cournia, N., Gramopadhye, A., Melloy, B., & Nair, S. (2002). 3D Eye Movement Analysis for Visual Inspection Training. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 103-110). New Orleans, LA.

For Further Reading II

Duchowski, A., Medlin, E., Gramopadhye, A., Melloy, B., & Nair, S. (2001). Binocular Eye Tracking in VR for Visual Inspection Training. In *Virtual Reality Software & Technology (VRST).* Banff, AB, Canada.

Duchowski, A. T., & Çöltekin, A. (2007, November). Foveated Gaze-Contingent Displays for Peripheral LOD Management, 3D Visualization, and Stereo Imaging. *Transactions on Multimedia Computing, Communications and Applications*, *3*(4). (Accepted for Publication)

Duchowski, A. T., Cournia, N., Cumming, B., McCallum, D., Gramopadhye, A., Greenstein, J., et al. (2004, March 22-24). Visual Deictic Reference in a Collaborative Virtual Environment. In *Eye Tracking Research & Applications (ETRA).* San Antonio, TX.

For Further Reading III

Duchowski, A. T., Marmitt, G., Desai, R., Gramopadhye, A., & Greenstein, J. (2003). Algorithm for Comparison of 3D Scanpaths in Virtual Reality. In *Vision Sciences Society.* Sarasota, FL. ((Posters.))

Hembrooke, H., Feusner, M., & Gay, G. (2006). Averaging Scan Patterns and What They Can Tell Us. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 41). San Diego, CA.

Itti, L., Koch, C., & Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *20*(11), 1254-1259.

Josephson, S., & Holmes, M. E. (2002). Visual Attention to Repeated Internet Images: Testing the Scanpath Theory on the World Wide Web. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 43-49). New Orleans, LA.

For Further Reading IV

Josephson, S., & Holmes, M. E. (2006). Clutter or Content? How On-Screen Enhancements Affect How TV Viewers Scan and What They Learn. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 155-162). San Diego, CA.

Kohler, M. (1997). *Using the Kalman Filter to track Human Interactive Motion—Modelling and Initialization of the Kalman Filter for Translational Motion* (Tech. Rep. Nos. #629, Informatik VII). University of Dortmund.

Li, D. (2006). *Low-Cost Eye-Tracking for Human Computer Interaction*. Unpublished master's thesis, Iowa State University, Ames, IA. (Techreport TAMU-88-010)

Marmitt, G., & Duchowski, A. T. (2002, September 2-6). Modeling Visual Attention in VR: Measuring the Accuracy of Predicted Scanpaths. In *EuroGraphics (Short Presentations).* Saarbrücken, Germany.

For Further Reading V

Murphy, H., & Duchowski, A. T. (2001, September 4-7). Gaze-Contingent Level Of Detail. In *Eurographics.* Manchester, UK.

Privitera, C. M., & Stark, L. W. (2000). Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *22*(9), 970-982.

Reich, S., Goldberg, L., & Hudek, S. (2004, October 15). Deja View Camwear Model 100. In *CARPE'04: Proceedings of the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences* (p. 110-111). New York, NY: ACM Press.

For Further Reading VI

Sadasivan, S., Greenstein, J. S., Gramopadhye, A. K., & Duchowski, A. T. (2005). Use of Eye Movements as Feedforward Training for a Synthetic Aircraft Inspection Task. In *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems* (p. 141-149). Portland, OR.

Santella, A., & DeCarlo, D. (2004). Robust Clustering of Eye Movement Recordings for Quantification of Visual Interest. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 27-34). San Antonio, TX.

Vembar, D., Iyengar, N., Duchowski, A. T., Clark, K., Hewitt, J., & Pauls, K. (2004, June 8-9). Effect of Visual Cues on Human Performance in Navigating Through a Virtual Maze. In *EuroGraphics Symposium on Virtual Environments (EGVE).* Grenoble, France.

For Further Reading VII

West, J. M., Haake, A. R., Rozanski, E. P., & Karn, K. S. (2006). eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences. In *Eye Tracking Research & Applications (ETRA) Symposium* (p. 149-154). San Diego, CA.