Improving User Perceived Page Load Time using Gaze

Conor Kelton†, Jihoon Ryoo†, Aruna Balasubramanian, Samir R. Das
†Students with equal contribution
Motivation

• Websites exploding in number! (Over 1.1 B today)
• Performance of these sites is important:
  – Google Uses Page Speed as major ranking factor
  – Amazon Reports $1.6 B in profit per 1 second decrease in site load time
Hypothesis: Traditional Metrics for Page Load Time Do *Not* relate to the user experience

- If true, then the effect of optimizations on user **Quality of Experience (QoE)** is **uncertain**
Does `Window.OnLoad()` capture the user’s experience?

Amazon.com: 7.9 s (OnLoad)
Amazon.com: 1.5 s (ATF Loaded)
Gmail.com: 0.9 s (OnLoad)
Gmail.com: 5.1 s (ATF Loaded)
Does Window.onLoad() capture the user’s experience?

Similar Mismatches of user QoE to other PLT metrics such as **Speed Index**, and **DOMContentLoaded**.
The \textit{uPLT}: \textit{user-perceived Page Load Time}

- How to determine if users are actually experiencing this disconnect?

\textit{Real User Studies!}

\textit{When is the Page Loaded?} \rightarrow \textit{uPLT}
The uPLT User Study Logistics

- **Consistency:**
  - Website loads shown as videos to the user

- **Quality:**
  - Measure user’s reaction times
  - Filter out erroneous responses

Related Work
[CoNext ‘16]
User Study Results: uPLT Spread

- Narrow spread in 25th - 75th percentiles shows consensus among users
User Study Results: OnLoad vs uPLT

- OnLoad indeed over-to-under estimating user experience
uPLT Results in the Wild

- Overall Observation:
  \[ Corr(uPLT, OnLoad) = .46 \]
  \[ Corr(uPLT, Speed Index) = .44 \]

- Additional analyses across site categories/network conditions in paper
Our Goal: Optimize Web loads for uPLT

- Intuition: Loading objects important to users first should improve the user experience

- How to find objects important to the user?
Leveraging Gaze Tracking

Software Aided Commodity Webcam Tracking

- User Eye Gaze has been used to track user attention
- Low cost, personalized, gaze tracking becoming feasible
Like uPLT, *Gaze* also captured during real user studies!

- Webcam based tracker
- 50+ Lab participants, same 45 Web sites as uPLT study
- Goal: To find attention on Web objects from user Gaze tracks
Going from Gaze to Object Importances

- Human Gaze consists of rapid *saccades* interspersed with stable *fixations* which mark points of user attention
Going from Gaze to Object Importances

- Human Gaze consists of rapid *saccades* interspersed with stable *fixations* which mark points of user attention.

- Plotting fixations over the page captures a user’s attention.
Going from Gaze to Object Importances

- Human Gaze consists of rapid *saccades* interspersed with stable *fixations* which mark points of user attention

- Plotting fixations over the page captures a user’s attention
Going from Gaze to Object Importances

- Human Gaze consists of rapid *saccades* interspersed with stable *fixations* which mark points of user attention.

- Plotting fixations over the page captures a user’s attention.
Going from Gaze to Object Importances

- Human Gaze consists of rapid *saccades* interspersed with stable *fixations* which mark points of user attention.

- Plotting fixations over the page captures a user’s attention.

- Fixations overlap across users.
Gaze: Collective Fixation

- First Divide Web page into its **Visual Regions**
- Map the fixations of all users onto the visual regions
- **Collective Fixation** is the fraction of users who fixate on a region
Combining Collective Fixation Results

25% of Regions have at most .3 Collective Fixation on average

There are objects with low user attention!

25% of Regions have at least .9 Collective Fixation on average

A subset of objects have high user attention!
Webgaze
A Web Prioritization System for uPLT

**Offline Component**

- **WebGaze Servers**
  - Process gaze for collective fixation
  - Enlists users to collect gaze

- **Gaze Providers**
  - Supplies gaze data to

**Online Component**

- **Web Servers**
  - Provides site info to
  - Sends set of priority Web objects to

- **Web Users**
  - Deliver Web site with objects prioritized via HTTP/2 Server Push
Prioritization Details: Webpage Dependencies

- Web page objects exhibit object *dependencies* on one another
- WebGaze finds and prioritizes these dependencies
Prioritization Details: Server Pushes

- WebGaze pushes objects of high *Collective Fixation* and their *dependencies* with HTML
- HTTP/2 is Multiplexed: Resources will contest for bandwidth
- WebGaze Pushes only objects above a *Collective Fixation Threshold*
WebGaze User Study Implementation

- Download **same 45 pages from uPLT study** locally
- Serve from **HTTP/2 Push enabled Web server**
- Take **videos** of Website loads
- Host videos on **Microworkers to obtain uPLT** from real users
# WebGaze Evaluation Comparisons

<table>
<thead>
<tr>
<th>Default</th>
<th>Push All</th>
<th>Klotski [NSDI ‘15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Prioritization</td>
<td>Pushes all resources identified in the page load</td>
<td>Pushes all objects that can be loaded in a static user tolerance limit (5 seconds)</td>
</tr>
<tr>
<td>Default under HTTP/2</td>
<td></td>
<td>State of the art prioritization</td>
</tr>
</tbody>
</table>
WebGaze: Demonstration

Default

Push-All

Klotski

WebGaze
WebGaze: Demo uPLT Results

- Default: 12 seconds
- Push-All: 10 seconds
- Klotski: 9 seconds
- WebGaze: 7 seconds

Freeze frame of load process at 6 seconds
WebGaze: Performance Results
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!

CDF

Percentage decrease in median uPLT

12%
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!
- *Case studies and comparisons to PLT metrics in the paper*
WebGaze: Why We Do Better

- uPLT Improvements over Default come from **general prioritization**
- uPLT Improvements over Push-all come from **ATF prioritization**
- uPLT Improvements over Klotski come from **prioritizing the right set of ATF objects**
WebGaze: Why We Do Worse

- **Comparing to Push-All:** Pushing everything sometimes works!

- **Comparing to Klotski:** Klotski thresholds objects, preventing worst case push performances
WebGaze: Where to?

- Formally optimize the trade off between collective fixation and object size at the Webgaze Servers
- Using saliency to predict gaze, i.e. automatic gaze feedback
- WebGaze for Mobile
Conclusion

Webgaze

- www.gaze.cs.stonybrook.edu
- uPLT Results - Low Correlation with Traditional PLT Metrics
- Gaze Data - Subset of Web Objects Viewed Significantly!
- Side By Side Loads of Optimized Sites - uPLT Improvements up to 64%
- More Work to Come!
A Visually Oriented Metric: The Speed Index

Visual Completeness (VC) = ___

Time Interval (TI) = 0.1 s

Speed Index = \sum TI \left( 1 - \frac{VC}{100} \right)
Does Speed Index do a Better Job?

Marketwatch.com: 14.5s (Speed Index)

Marketwatch.com: 7.5s (Most ATF Rendered)

Energystar.gov: 3.7s (Speed Index)

Energystar.gov: 7.8s (ATF Rendered)
Speed Index vs. uPLT in the Wild

- Speed Index also not trending well with user experience
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!
WebGaze: Performance Results

- Delivering objects identified by gaze early does help!
- *Case studies and comparisons to PLT metrics in the paper*