Background Information

• **What do we mean by presentation?**
  – “Look and feel” of the website in a browser

• **What is a presentation failure?**
  – Web page rendering ≠ expected appearance
Examples of Presentation Failures

Cross-browser Issues (XBIs)

Internationalization Issues

Mockup Driven Development

Regression Debugging
Background Information

• What do we mean by presentation?
  – “Look and feel” of the website in a browser

• What is a presentation failure?
  – Web page rendering ≠ expected appearance

• Importance of presentation
  – It takes users only 50 ms to form opinion about your website [Google research 2012]
  – Aesthetics impact users’ evaluation [Tractinsky et. al. 2006]
  – Impacts trustworthiness and usability [Lindgaard et. al. 2011]
Motivation

• **Manual detection is difficult**
  – Complex interaction between HTML, CSS, and Javascript
  – Hundreds of HTML elements + CSS properties
  – Labor intensive and error-prone

• **My research** – Automate debugging of presentation failures
Two Key Insights

1. Detect presentation failures

Expected appearance → Visual comparison → Presentation failures

Use computer vision techniques
Two Key Insights

2. Localize to faulty HTML elements

Test web page

Layout tree

Faulty HTML elements

Use rendering maps
Our Approach

**Goal:** Automatically detect and localize presentation failures in web pages

- P1. Detection
- P2. Localization

Test web page

Expected appearance

Pixel-HTML mapping

Report

Logo
P1. Detection

- Find visual differences (presentation failures)
- Compare oracle image (expected appearance) and test page screenshot
- **Simple approach:** strict pixel-to-pixel equivalence comparison
  - Drawbacks
    - Spurious differences due to difference in platform
    - Small differences may be “OK”
Perceptual Image Differencing (PID)

- Uses models of the human visual system
  - Spatial sensitivity
  - Luminance sensitivity
  - Color sensitivity

**Shows only human perceptible differences**

- \( \Delta \) : Threshold value for perceptible difference
- \( F \) : Field of view of the observer
- \( L \) : Brightness of the display
- \( C \) : Sensitivity to colors
P1. Detection – Example

Test web page for unexpected changes (e.g. GUM & Danch)

Oracle Visual comparison using PID

Apply clustering (DBSCAN)

Filter differences belonging to dynamic areas

A

B

C

Login
Username:
Password:
Sign in

Cellphone advertisement

Username:
Password:
Sign in

Advertisement box

News box

This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news! This is indeed a big news!
Special Regions Handling

• Special Regions = Content is not known (e.g. dynamic portions)
• Running PID directly can result in false positives
• Need to be handled “specially”
• Two types
  – Exclusion regions (e.g. advertisement box)
  – Dynamic text regions (e.g. news box)
P2. Localization

• Identify the faulty HTML element
P2. Localization

• Identify the faulty HTML element

Use rendering maps to find faulty HTML elements corresponding to visual differences

• Use R-tree to map pixel visual differences to HTML elements

• “R”ectangle-tree: height-balanced tree, popular to store multidimensional data
P2. Localization - Example
P2. Localization - Example
P2. Localization - Example

Map pixel visual differences to HTML elements

Result Set:
1. /html/body/…/tr[2]/td[1]/table[1]/td[1]
2. /html/body/…/tr[2]/td[1]
3. /html/body/…/tr[2]
4. /html/body/…/tr[2]/td[1]/table[1]/tr[1]
5. /html/body/…/tr[2]/td[1]/table[1]

Rank HTML elements in order of likelihood of being faulty based on the heuristics of HTML element relationships
Detection and Localization Empirical Evaluation

- **RQ1**: Accuracy of detecting and localizing?
- **RQ2**: Quality of the localization results?
- **RQ3**: Time required to detect and localize?
Experimental Protocol

• Approach implemented in “WebSee”
• Five real-world subject web applications
• For each subject application
  – Download page and take screenshot, use as the oracle
  – Seed presentation failure to create a variant
  – Run WebSee on oracle and variant
Subject Applications

<table>
<thead>
<tr>
<th>Subject Application</th>
<th>Size (Total HTML Elements)</th>
<th>Generated # test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gmail</td>
<td>72</td>
<td>52</td>
</tr>
<tr>
<td>USC CS Research</td>
<td>322</td>
<td>59</td>
</tr>
<tr>
<td>Craigslist</td>
<td>1,100</td>
<td>53</td>
</tr>
<tr>
<td>Virgin America</td>
<td>998</td>
<td>39</td>
</tr>
<tr>
<td>Java Tutorial</td>
<td>159</td>
<td>50</td>
</tr>
</tbody>
</table>
Empirical Evaluation

Detection Accuracy = 100%

Localization accuracy

<table>
<thead>
<tr>
<th>Gmail</th>
<th>92%</th>
</tr>
</thead>
<tbody>
<tr>
<td>USC</td>
<td>93%</td>
</tr>
<tr>
<td>USC CS</td>
<td></td>
</tr>
<tr>
<td>Craigslist</td>
<td>90%</td>
</tr>
<tr>
<td>Java Tutorial</td>
<td>94%</td>
</tr>
<tr>
<td>Virgin America</td>
<td>97%</td>
</tr>
</tbody>
</table>

Quality of localization

7 sec 3 min

Total time required

Rank = 4.8 (2%)

87 sec
Comparison with User Study

- Graduate-level students
- Manual detection and localization using Firebug

- Time
  - Students: 7 min
  - WebSee: 87 sec
Summary

• Techniques for automatically detecting and localizing presentation failures

• Use computer vision techniques for detection

• Use rendering maps for localization

• Empirical evaluation shows positive results
Thank you

Detection and Localization of Presentation (GUI) Failures in Web Applications

Sonal Mahajan and William G. J. Halfond

spmahaja@usc.edu
halfond@usc.edu