Use of Generalized Hough Transform on interpretation of memory dumps

Paulo R. Nunes de Souza
Pavel Gladyshev
What is partial data?

• Incomplete data
• Partially corrupted data
• Data without required code/algorith
• Data of unknown source
Word puzzle

**SCIENCE OF SECRETS**

- ANTIBIOTIC
- BACTERIA
- BLOOD
- BRAINSTORM
- CHICKADEE
- COLLEGE
- COMMUNICATE
- DECODE
- E COLI
- ELECTRONIC
- EVIDENCE
- FEROCIOUS
- FLUORESCENT
- FORENSIC
- GELATINOUS
- GUNPOWDER
- HARD DRIVE
- HARVEST
- INTERNET
- INVESTIGATION
- JELLYFISH
- MERLIN
- PATTERNS
- PREDATOR

- SATELLITE
- SAXOPHONE
- SECRET AGENT
- SMART PHONE
- SOLDIERS
- SONOGRAM
- STROLL
- TERRORISM
- VIOLENT CRIME
- WIGGLEY

UCD Dublin
Requirements

• Tolerance to noisy data;
• Tolerance to partial data corruption;
• Flexibility to define the structure being searched.
• Suitable candidate: Hough Transform
Hough Transform
Encoding

User input (decimal number)
7562617 → Application A → 32-bit integer (hex.) → CPU Microcode

Little-endian encoding

Data in consecutive RAM cells: 79 65 73 00

User input (text)
y e s → Application B → character codes (hex.) → 79 65 73 00
Model

\[ T = \{ T_{bin}, T_{int8}, T_{ascii}, \ldots \} \]
Model

\[ P_{E_{T_b}}(o|i) = \sum_{\sigma \in T_{\alpha}} P_{E_{T_a}}(\sigma|i) \cdot P_{E_{T_{\beta}}} (\eta|\sigma) \cdots P_{E_{T_{\gamma}}} (o|\mu) \]

\[ P_{E_{T_b}}(o|i, t) = \begin{cases} P_{E_{T_t}}(o|i) & \text{if } i \in t \\ 0 & \text{if } i \notin t \end{cases} \]

\[ P_{SE_{T_b}}(w|k) = \prod_{j=0}^{q} \sum_{\forall \lambda \in k(j)} P_{E_{T_b}}(w(j)|\lambda, k(j)) \]

\[ P_{SE}(k|w) = \frac{P_{SE}(w|k) \cdot P_{SE}(k)}{P_{SE}(w)} \]
Data types

<table>
<thead>
<tr>
<th>Dalvik</th>
<th>Hough</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Char (Tc)</td>
</tr>
<tr>
<td>Char</td>
<td>Byte (Tb)</td>
</tr>
<tr>
<td>Byte</td>
<td>Short (Ts)</td>
</tr>
<tr>
<td>Short</td>
<td>Int (Ti)</td>
</tr>
<tr>
<td>Int</td>
<td>Float (Tf)</td>
</tr>
<tr>
<td>Float</td>
<td>Reference (Tr)</td>
</tr>
<tr>
<td>Class</td>
<td>Long (Tl)</td>
</tr>
<tr>
<td>Exception</td>
<td>Double (Td)</td>
</tr>
<tr>
<td>Object</td>
<td>Padding (Tp)</td>
</tr>
<tr>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>Double</td>
<td></td>
</tr>
</tbody>
</table>

\[
T = \{T\text{bin}, T\text{b}, T\text{c}, T\text{d}, T\text{f}, T\text{i}, T\text{l}, T\text{p}, T\text{r}, T\text{s}\}
\]
Probabilities $P_{se}(k|w,b)$
R-table

(a) First word ($a = 0$)

<table>
<thead>
<tr>
<th>Previous</th>
<th>Current</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_0$</td>
<td>$k_1$</td>
</tr>
<tr>
<td></td>
<td>$k_2$</td>
<td>$k_3$</td>
</tr>
<tr>
<td></td>
<td>$k_4$</td>
<td>$k_5$</td>
</tr>
<tr>
<td></td>
<td>$k_6$</td>
<td>$k_7$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Second word ($a = -4$)

<table>
<thead>
<tr>
<th>Previous</th>
<th>Current</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_0$</td>
<td>$k_1$</td>
<td>$k_2$</td>
</tr>
<tr>
<td>$k_3$</td>
<td>$k_4$</td>
<td>$k_5$</td>
</tr>
<tr>
<td>$k_6$</td>
<td>$k_7$</td>
<td>$k_8$</td>
</tr>
<tr>
<td>$k_9$</td>
<td>$k_a$</td>
<td>$k_b$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) Fourth and last word ($a = -12$)

<table>
<thead>
<tr>
<th>Previous</th>
<th>Current</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_8$</td>
<td>$k_9$</td>
<td>$k_a$</td>
</tr>
<tr>
<td>$k_b$</td>
<td>$k_c$</td>
<td>$k_d$</td>
</tr>
<tr>
<td>$k_e$</td>
<td>$k_f$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
### R-table

<table>
<thead>
<tr>
<th>ZZZZ FFFF IIII</th>
<th>0</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF IIII IIII</td>
<td>-4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IIII IIII CCPP</td>
<td>-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIII CCPP RRRR</td>
<td>-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCPP RRRR FFFF</td>
<td>-16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RRRR FFFF IIII</td>
<td>-20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIII IIII RRRR</td>
<td>-28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIII RRRR ZZZZ</td>
<td>-32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFFF IIII IIII CCPP RRRR FFFF IIII IIII RRRR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accumulation

If $C > C_t$, increment accumulation table at position $i+a$.

\[
C = P_{SE}(F|w(3),0) \cdot P_{SE}(F|w(4),1) \cdot P_{SE}(F|w(5),2) \cdot P_{SE}(F|w(6),3) \cdot P_{SE}(I|w(7),0) \cdot P_{SE}(I|w(8),1) \cdot P_{SE}(I|w(9),2) \cdot P_{SE}(I|w(a),3) \cdot P_{SE}(I|w(b),0) \cdot P_{SE}(I|w(c),1) \cdot P_{SE}(I|w(d),2) \cdot P_{SE}(I|w(e),3)
\]
Peak detection

• Local maxima above threshold $H$
• Positions of dump where structure was identified
Tests

- Randomly created 100 distinct dumps
- Each dump with 4 kb in size
- Randomly choose one of the types in T
- Then a random value of the chosen type was inserted in the dump
- Repeat this until a 4kb dump was created
- The last step was to insert instances of the structure of interest across the previously created dumps.
- The position and quantity of those structures were also randomly chosen
- Each dump and the positions of the structure of interest were saved to respective files
Tests

• For each of the 100 dumps, another 100 versions of them were created
• One file for each level of added noise. Starting at 0% for each 0.1% step until reaching 10%
• At the end, we have 101 versions of each of the 100 dumps
• Total of 10100 test dumps
Results
Conclusions

• Tolerance to noisy data
• Flexibility to identify structure of interest
• The structure of interest was correctly spotted in 99.8% of the tests with no noise
• The structure of interest was correctly spotted in 20% of the cases with 10% of noise.
• The downside is the high false positive rate.
• Applicable beyond memory realm
Follow ups

• Hough-Forensic DSI
• PNG filecarving
Thank you