Detecting File Fragmentation Point Using Sequential Hypothesis Testing

By

Anandabrata Pal, Husrev Sencar, Nasir Memon

Presented At

The Digital Forensic Research Conference

DFRWS 2008 USA Baltimore, MD (Aug 11th - 13th)

File Carving Using Sequential Hypothesis Testing

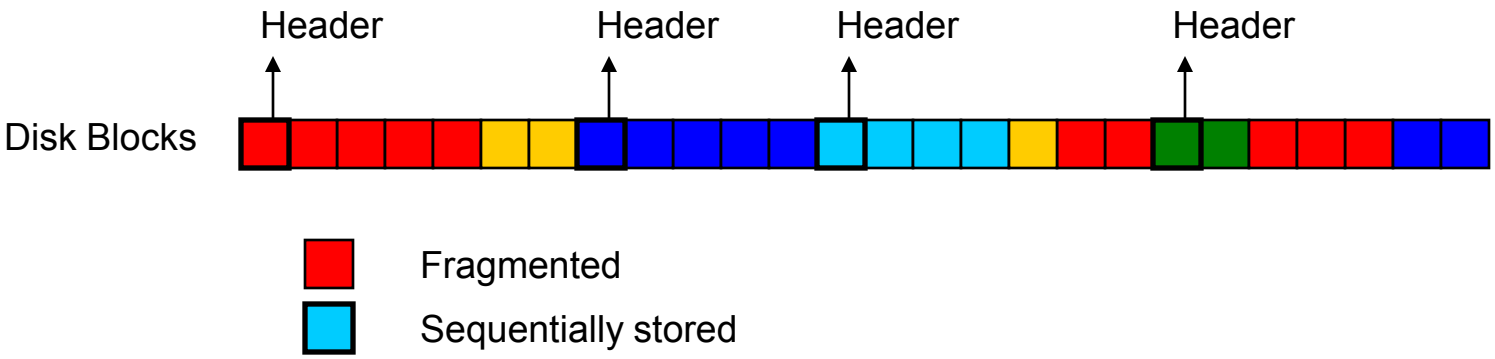
Anandabrata (Pasha) Pal, Taha Sencar and Nasir Memon

SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

Introduction

- File Carving: recovery without file system meta-data.
  - Recovery based on file structure/content
  - Great for sequentially stored data
  - Problems with fragmented data



Contents

- Data Recovery
- File Systems
- Fragmentation Problem
- File Carvers
- Sequential Hypothesis Testing
- Experiments and Results

Data Recovery

- Data Recovery attempts to recover data from a digital device.
- Used in:
  - Disaster Recovery
  - Digital Forensics
  - E-Discovery

File Systems

- Files are typically stored in file systems (FS)
  - Windows
    - FAT 12/16/32
    - NTFS
  - Mac
    - HFS
    - HFS+
- FS typically store data in clusters or blocks.

File Allocation in FAT Example

File System Structures

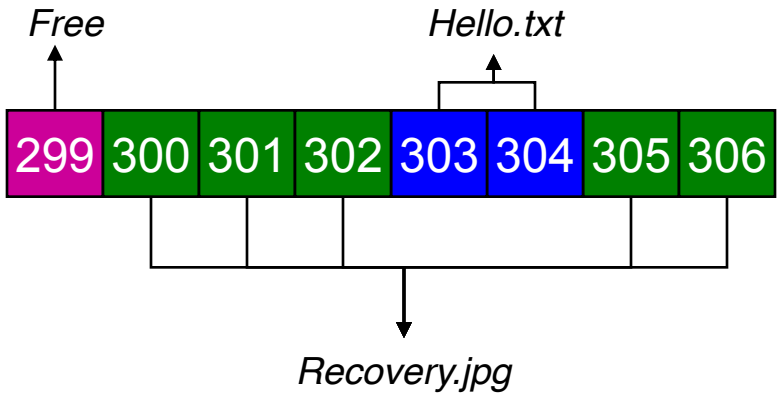
Root Table Entry

File name	*Starting Block*
Recovery.jpg	300
Hello.txt	303

FAT

Block Index	*Next Block*
299	Free
300	301
301	302
302	305
303	304
304	EOF
305	306
306	EOF

Disk Data Block Area



File Deletion in FAT Example

File System Structures

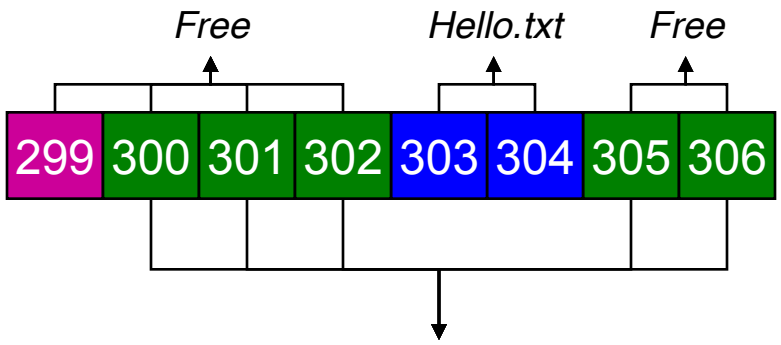
Root Table Entry

File name	*Starting Block*
?ecovery.jpg	300
Hello.txt	303

FAT

Block Index	*Next Block*
299	Free
300	Free
301	Free
302	Free
303	304
304	EOF
305	Free
306	Free

Disk Data Block Area



Traditional Recovery

- Use File System Information to recover “deleted” files
  - Most File Systems do not actually delete the files.
    - Thank god, this keeps us in business and research!
  - Simply removes link to file in the file table.
    - Sets blocks to “free/available”
  - Recovery occurs by going to starting cluster of deleted file, checking that subsequent clusters are unallocated and then recovering.

File System Problems

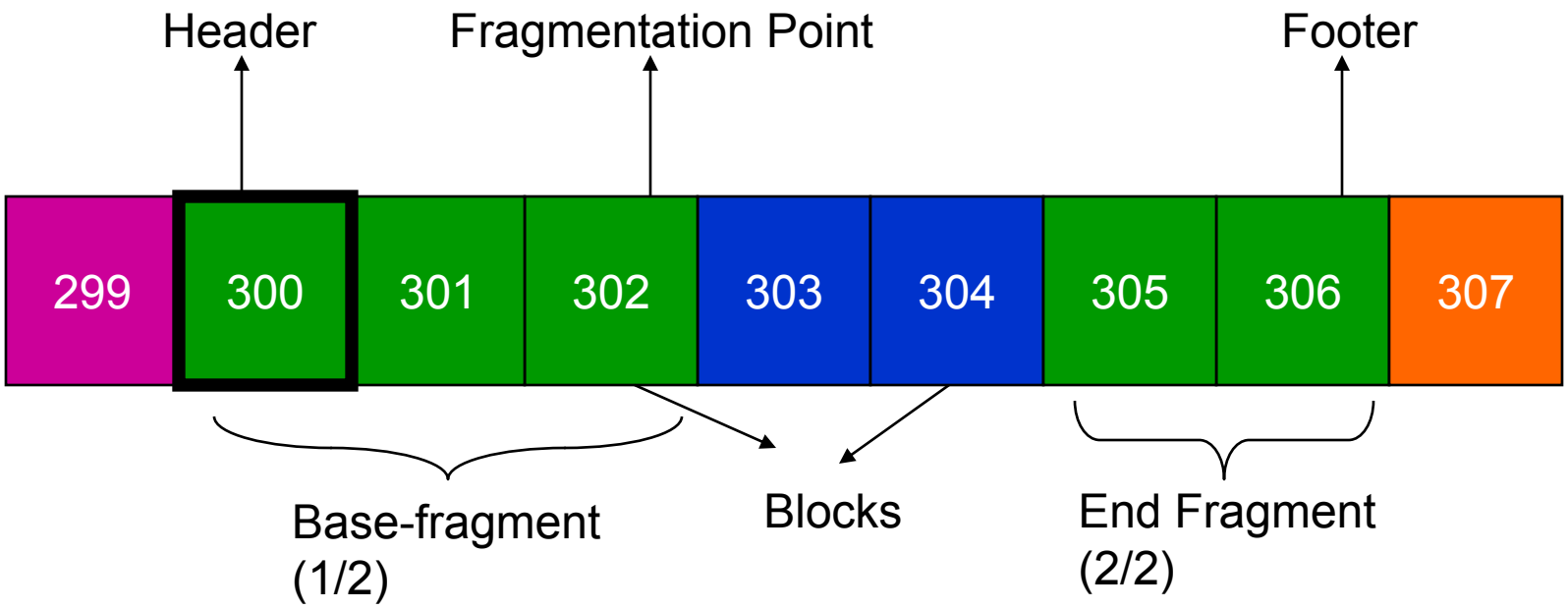
- Reasons why a FS should not be trusted
  - Corruption
  - Missing File Information
  - Intentionally Modified
  - Unknown
- So how do we recover data without using FS information?
  - File Carving!

File Carving (cont'd)

- Traditional File Carving:
  - Recover From Known Header to Footer:
    - Jpeg: "FFD8" header and "FFD9" footer
  - Recover From Known Header based on Size:
    - 24-bit BMP: "BM" header followed by size
- Carving occurs sequentially from the header
- Most file carvers can not handle fragmented files

External Fragmentation

- Fragmentation occurs when a file is not stored in sequential blocks.



How Important Is Fragmentation?

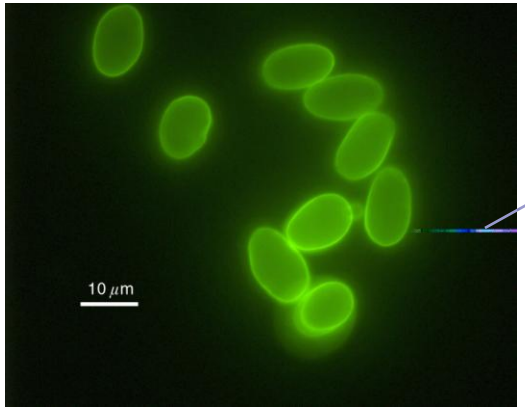
- From Simson Garfinkel's 350 disk corpus
  - Average fragmentation was 6%
  - Forensically important files had higher fragmentation rates:
    - Jpegs: 16%
    - PST: 58%
    - DOC: 17%
- Majority of fragments are bi-fragmented (two fragments) (40%)

Causes of Fragmentation

- Disk gets full after multiple additions and deletions
- Appending of data
- File System forces fragmentation
- Wear-Leveling in SSDs

Traditional File Carving on Fragmented Data

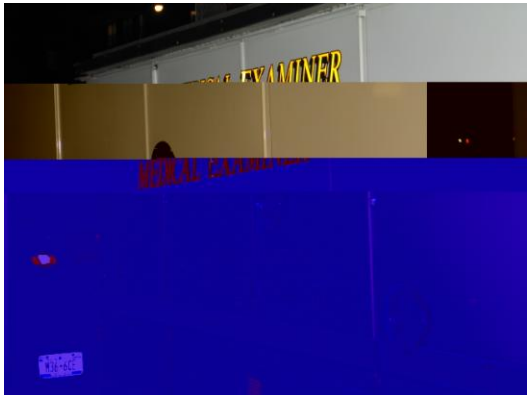
- Case 1: Data between two fragments



HTML data

- Case 2: Data incorrectly decoded and missing blocks

- Case 3: Data correctly decoded but wrong blocks



SmartCarving™

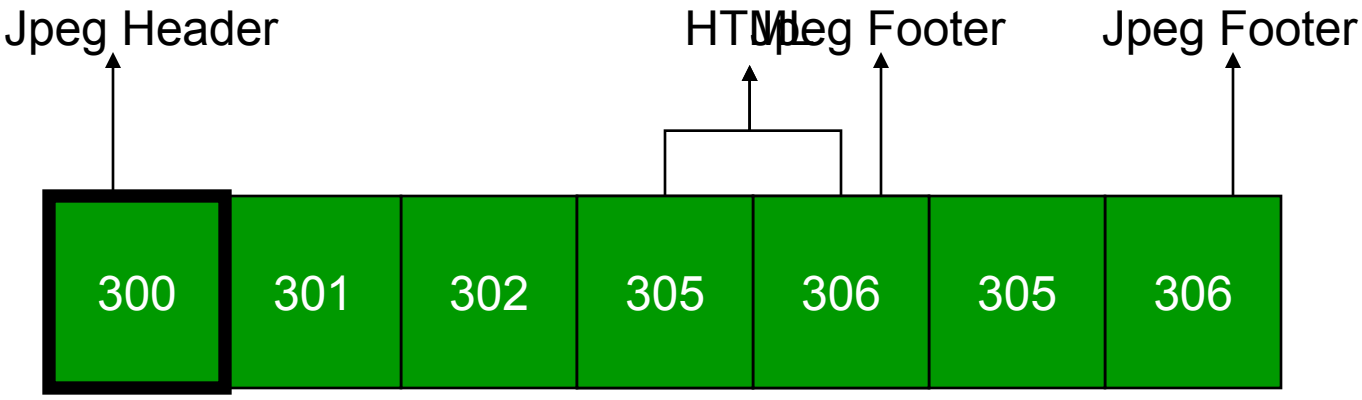
NYU·poly
POLYTECHNIC INSTITUTE OF NYU

Carving Techniques For Fragmentation

- Simple Carving: Removing full files or known data like text between fragments.
- Graph Theoretic Carving: Reformulating the fragmentation problem as a graph optimization problem.
- Bi-fragmented Gap Carving: Trying all possibilities between a header and footer

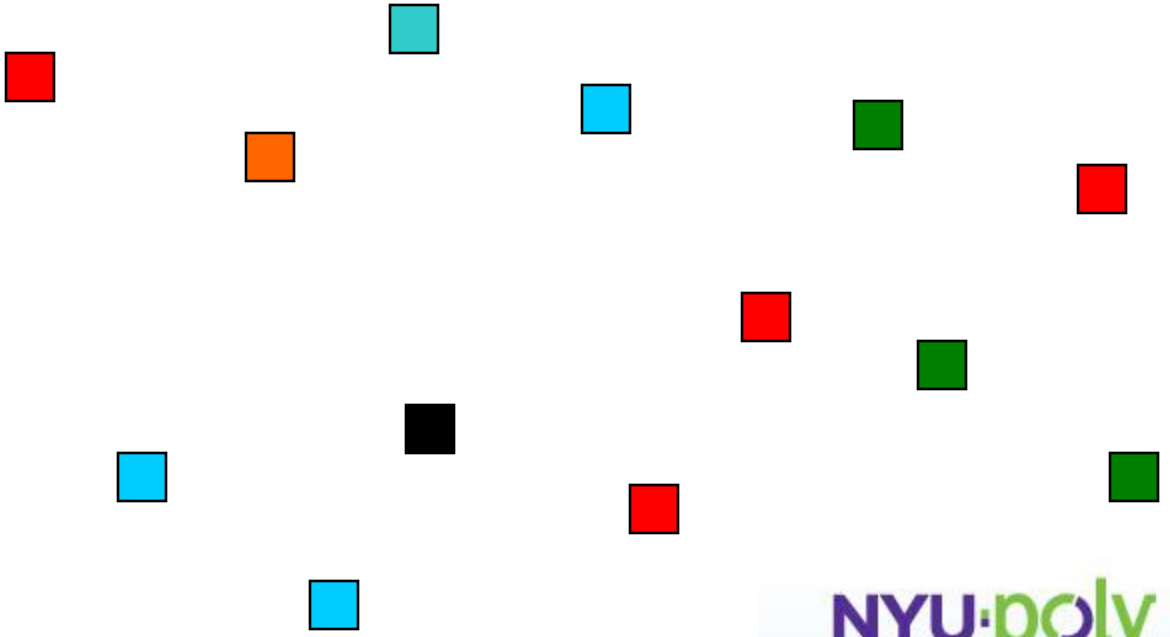
Simple Carving

- Identify and remove blocks that belong to another file and then carve.



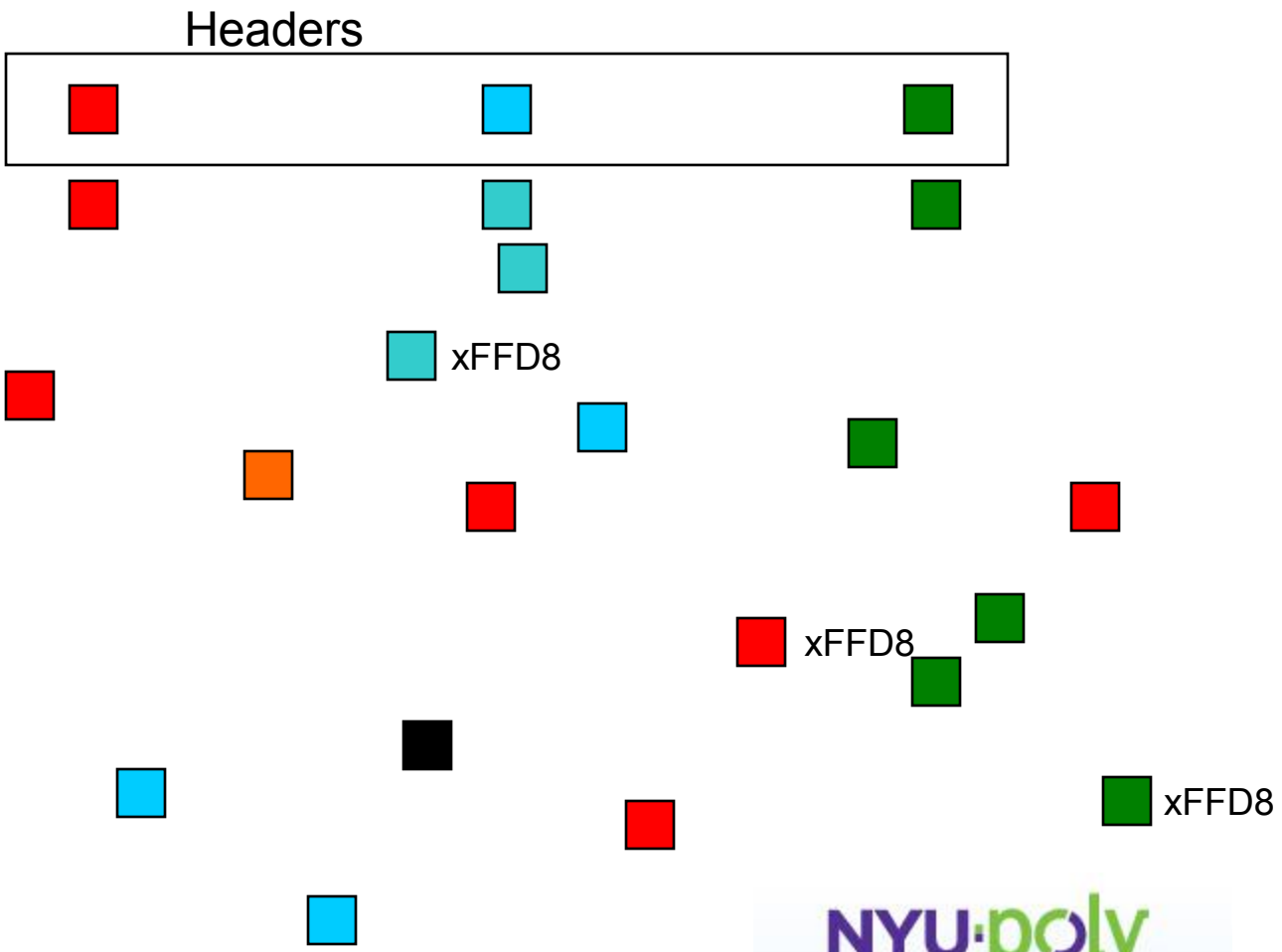
Graph Theoretic Carvers

- Assume blocks are completely randomized



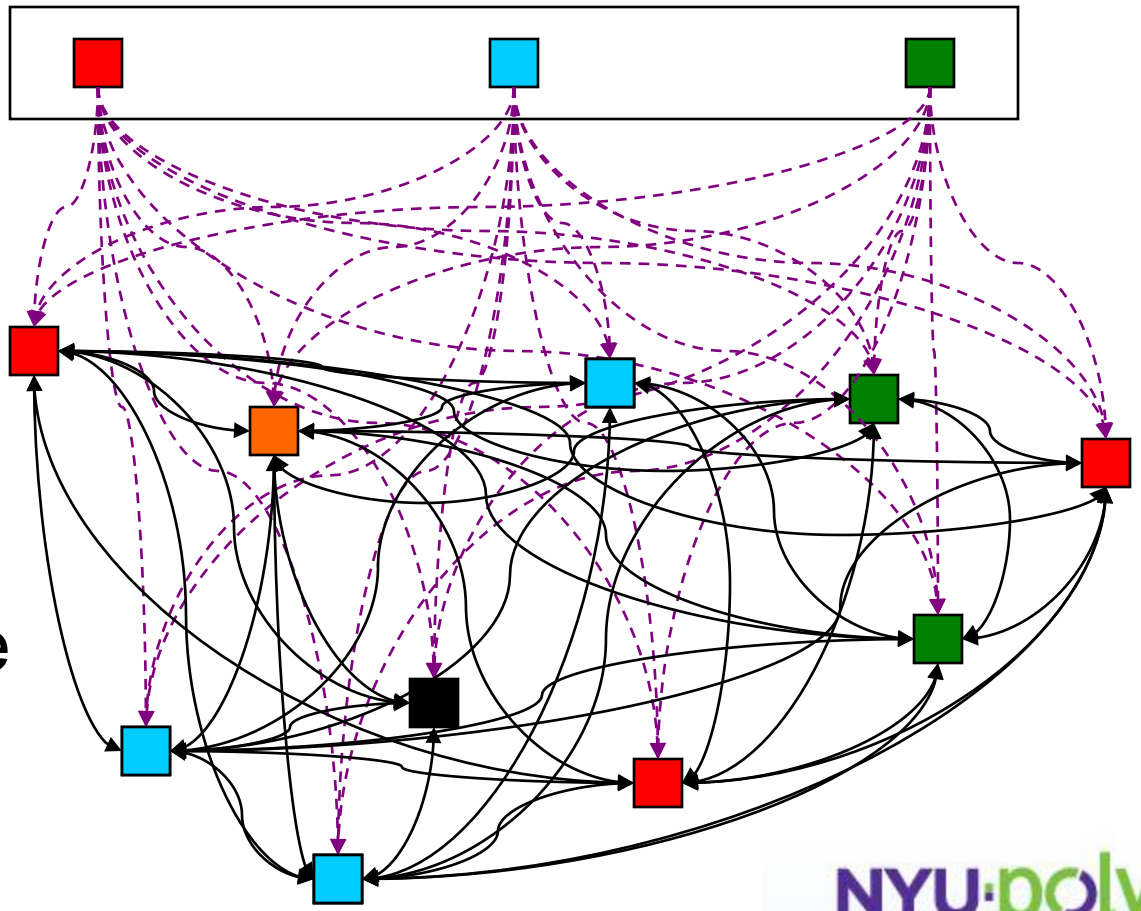
Graph Theoretic Carvers

- Identify headers using keywords/ signatures
- Jpeg: xFFD8



Graph Theoretic Carvers

- Create a fully connected weighted graph (other than headers)
- Each edge has a value defined as the matching metric.

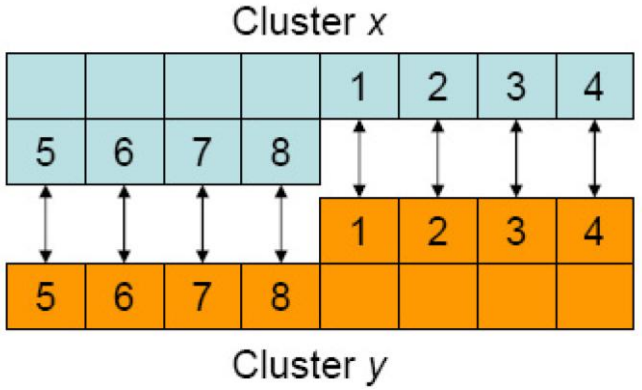


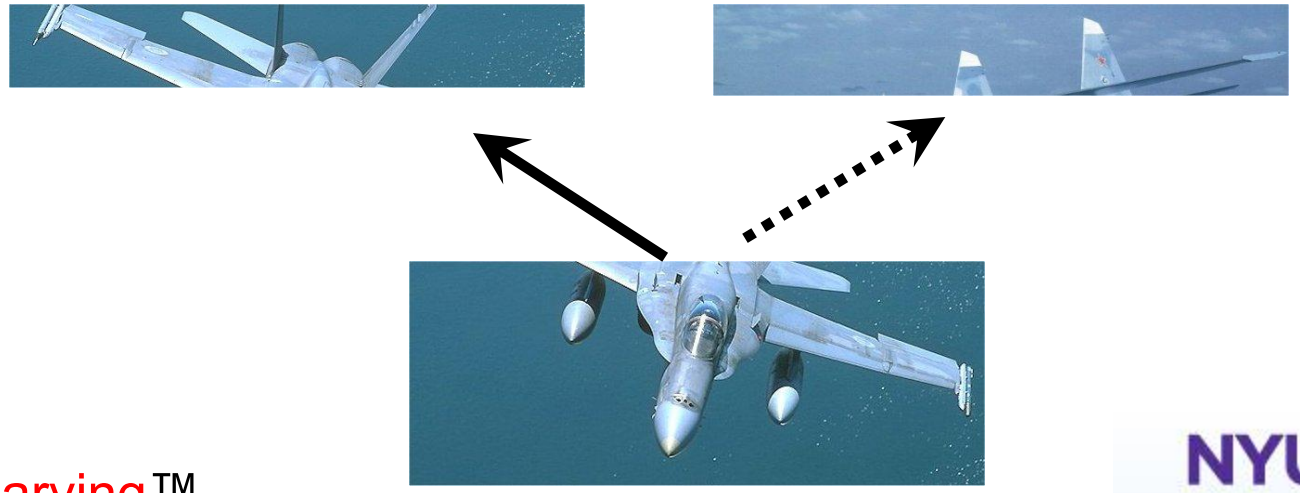
SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

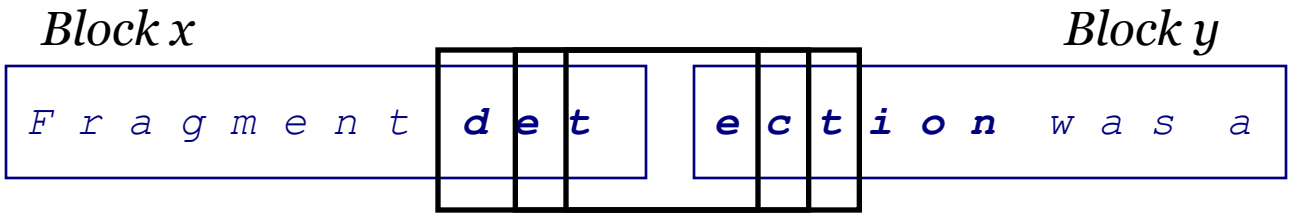
Matching Metric between blocks (Images)

- For images we look at boundary formed by addition of new block





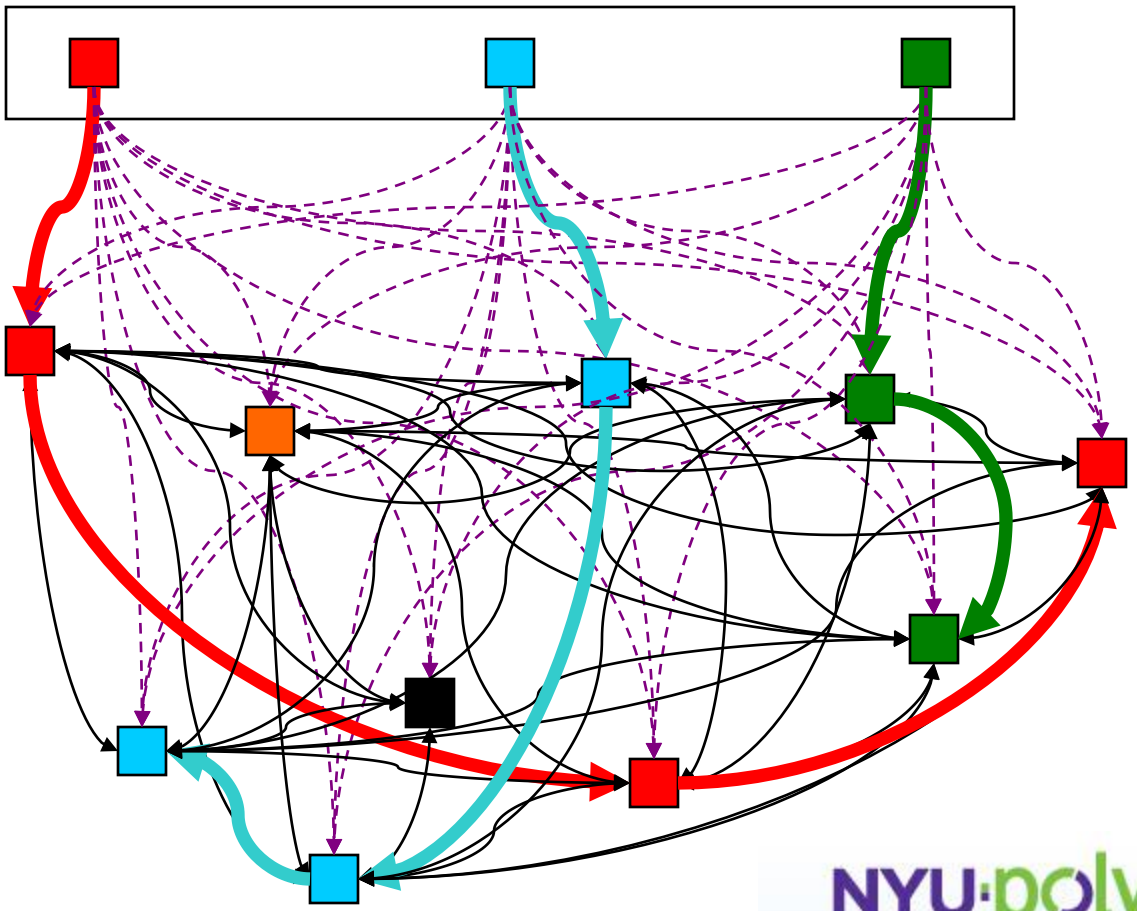
Matching metric between blocks (Text)



- Slide a window of size *d* from one fragment into the other
- At each position, use the window as context and calculate the score as the probability (p_i) of next symbol
- Matching Metric = ($p_0 * p_1 * \ldots p_d$)

Graph Theoretic Carvers

- Problem then becomes that of finding k-disjoint shortest paths based on the weight in a fully connected graph

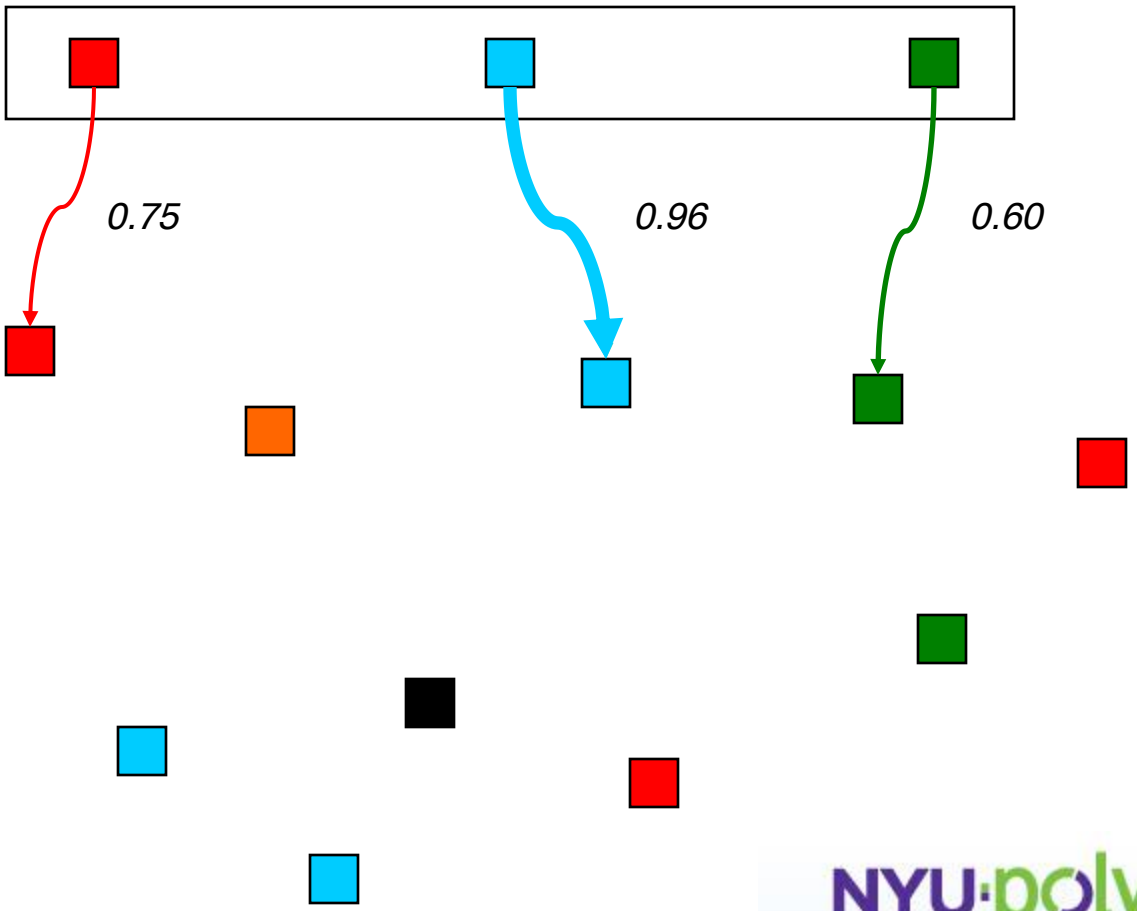


SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

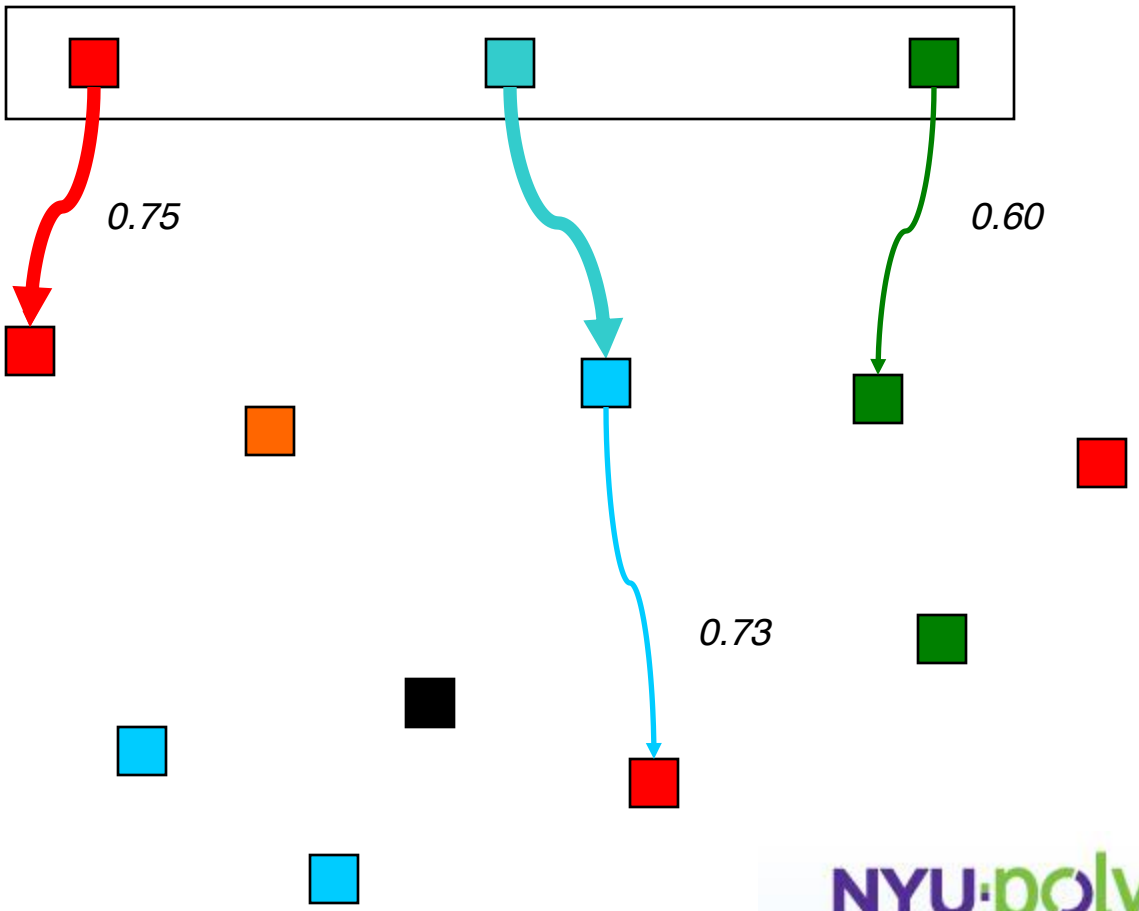
Parallel Unique Path

- For each header find best match (using matching metric)
- Then choose the best of the best matches.



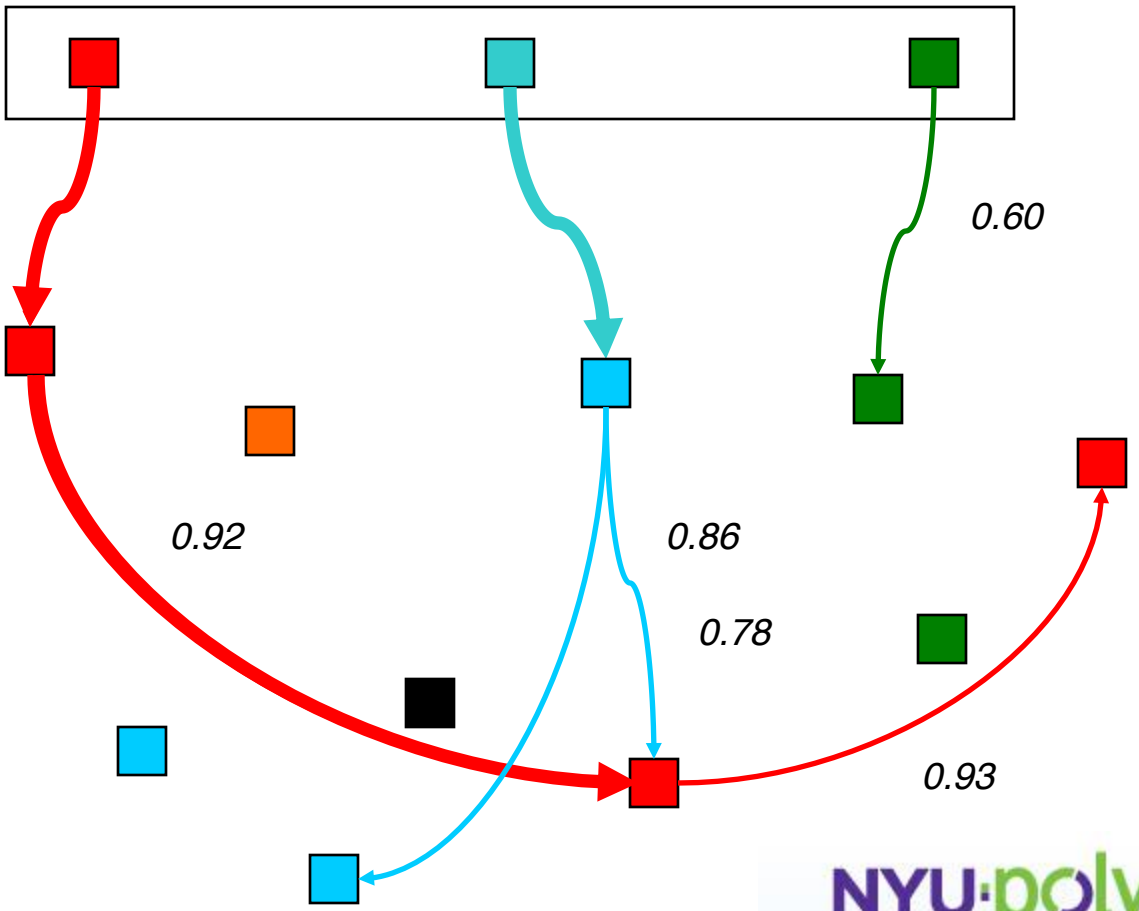
Parallel Unique Path

- Find best match for recently added node
- Then choose the best of the best matches again.



Parallel Unique Path

- Repeat process
- Now a block is the best match for two files
- Choose the better of the two and continue

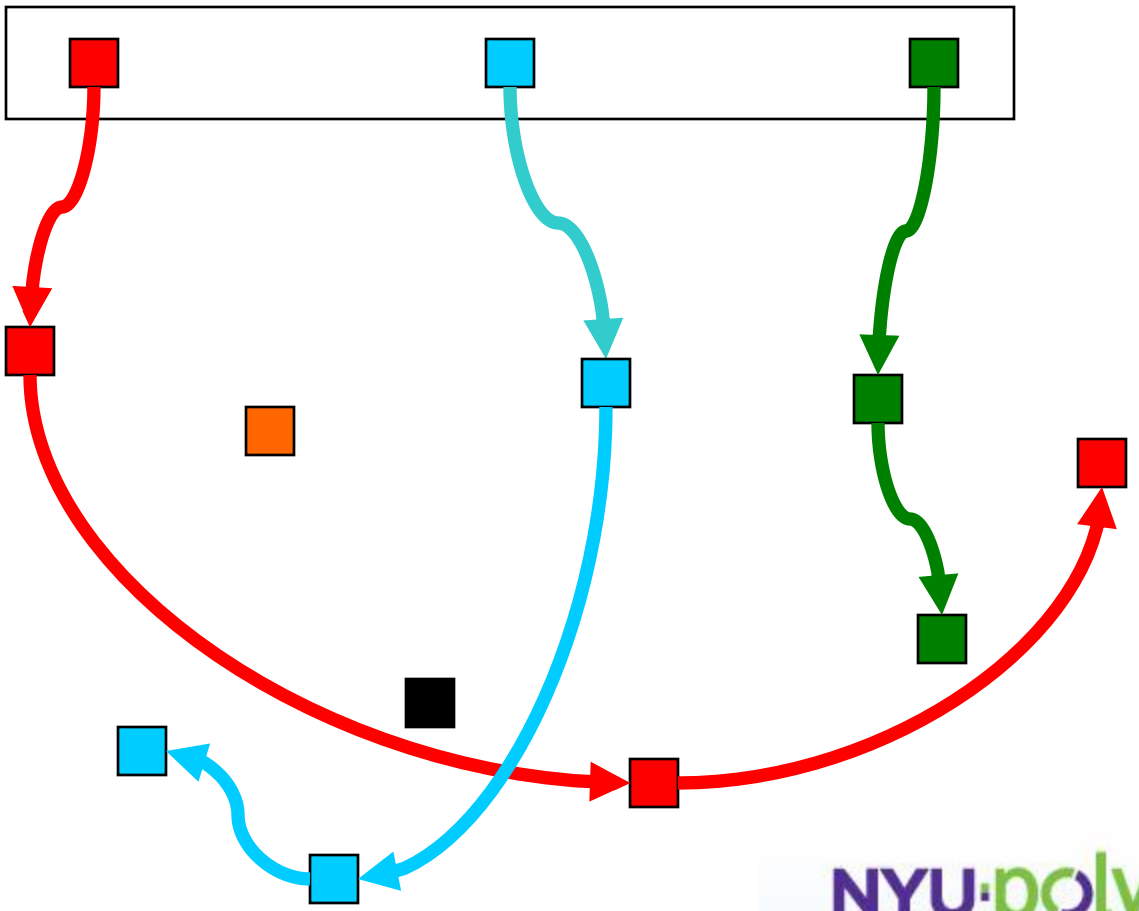


SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

Parallel Unique Path

- Repeat until all files are built or no more nodes can be chosen.

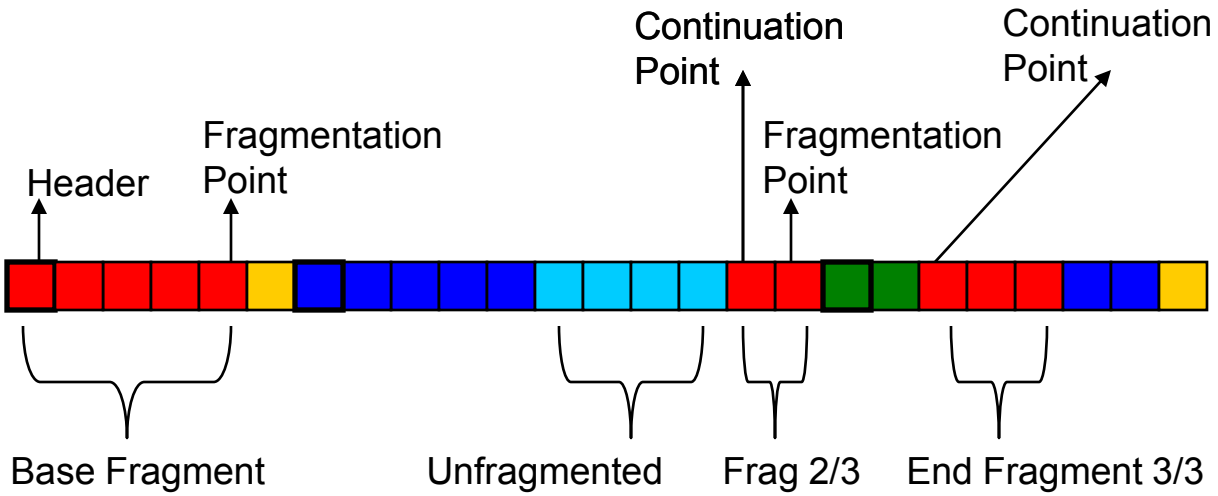


Problems with PUP

- Does not scale
  - n^2 weights are pre-generated
- Disk Geometry is not taken into account
  - Files normally do not have most of their blocks scattered randomly on the disk

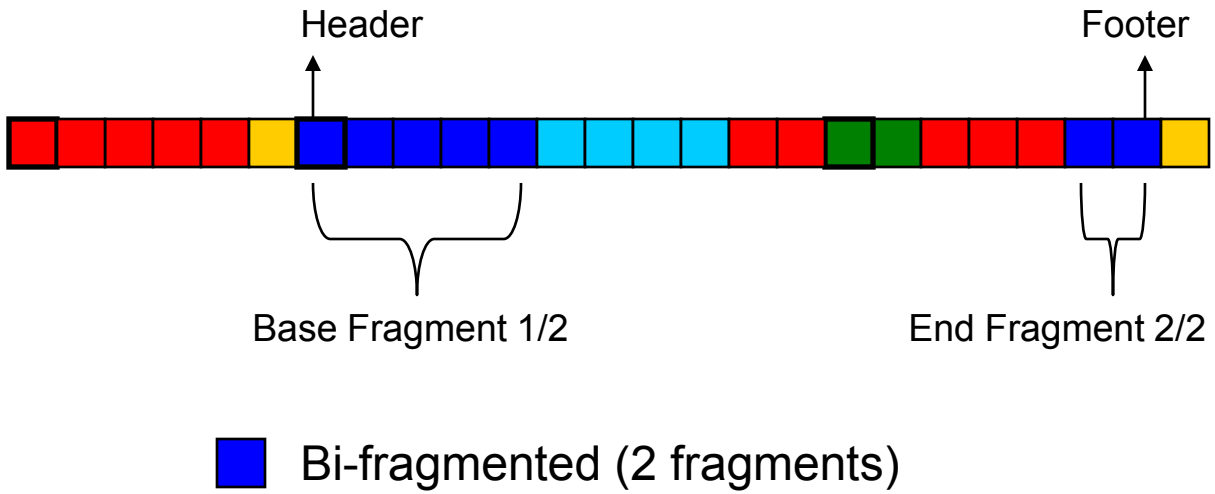
A More Realistic Visualization of Fragmentation

- Garfinkel showed fragmentation normally does not result in a huge number of fragments.
- This means that a large number of sequential blocks belong together even in a fragmented file



Bi-fragment Gap Carving (BGC)

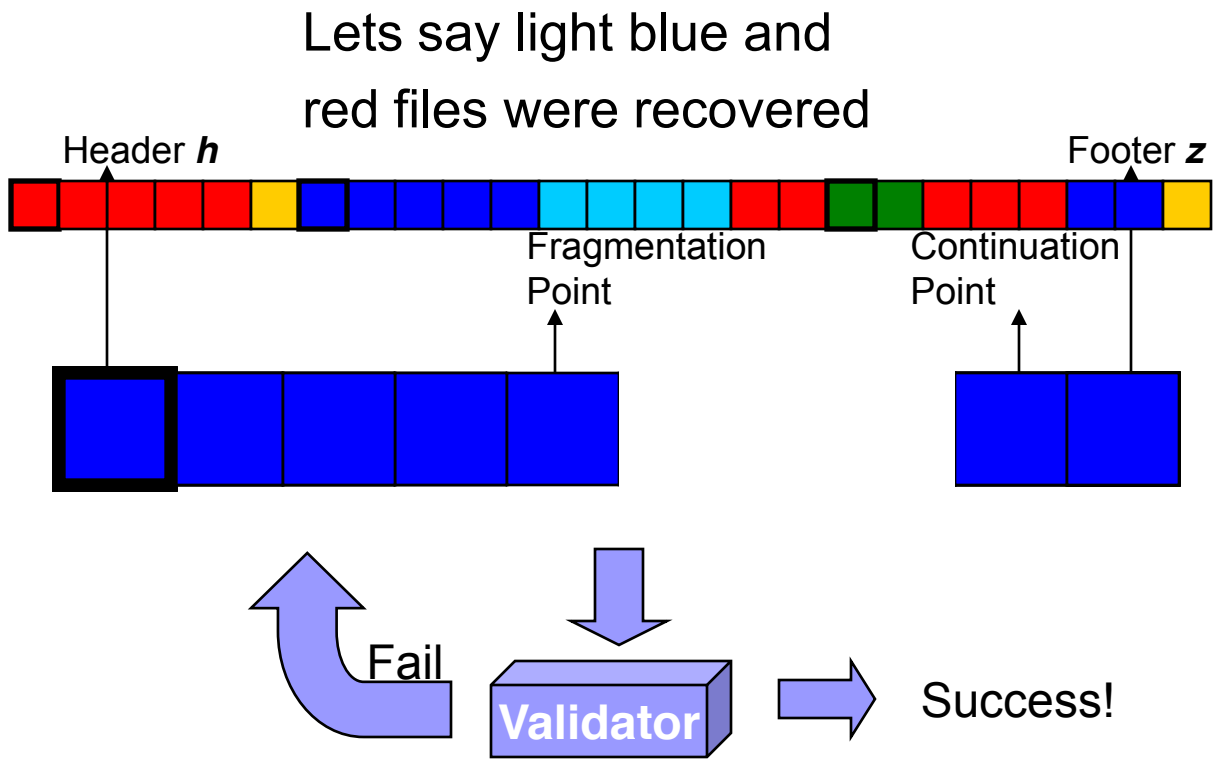
- BGC can only be used for files that are fragmented into two fragments.
- Requires a validator/decoder to determine success.



Bi-fragment Gap Carving Steps

Step 1: Identify Header (***h***) and footer (***z***)

Step 2: Brute Force all possible combinations by excluding blocks and validating (in essence trying all possible fragmentation and continuation points).



SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

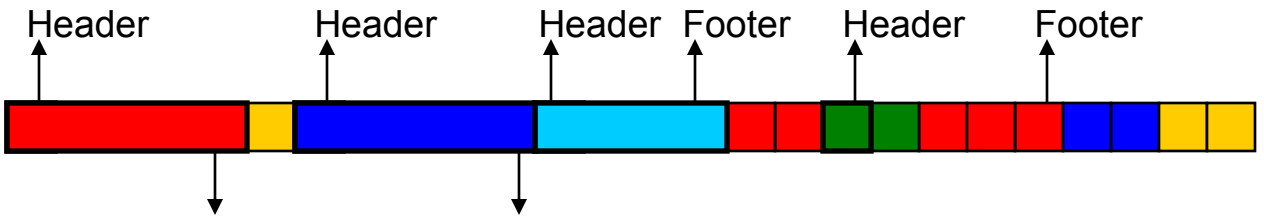
Bi-fragment Gap Carving Problems

- Won't work with > 2 fragments
- Worst case n^2
  - n^2 whenever missing or corrupt blocks
- Validation does not always mean correct reassembly.

Importance of Disk Geometry

- BGC shows disk geometry is important to take into consideration.
  - Blocks are not scattered all over the disk typically and a large number of consecutive blocks will *probabalistically* belong together.
- So how do we use this information?
  - We propose a new technique called SmartCarving.

SmartCarving - Collation



Step 1: **Collation**

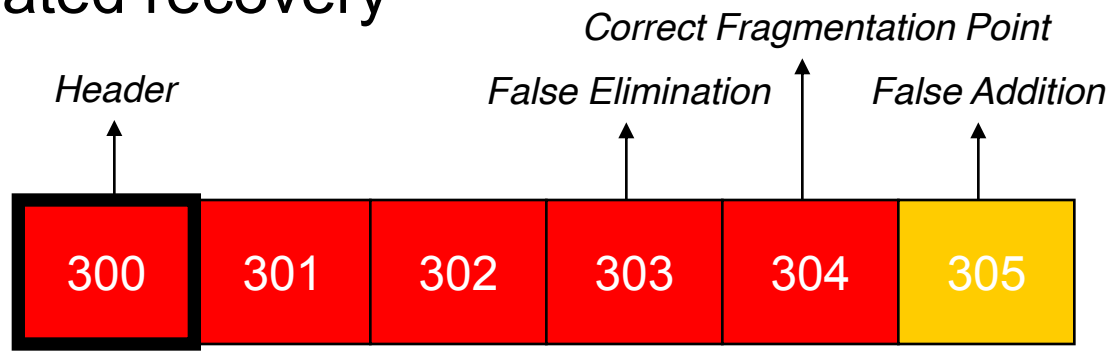
a) Identify Headers, Footers

b) Merge Sequential Blocks and find Fragmentation Point

c) Identify Blocks including loose or unattached blocks

Reliable Fragmentation Point Detection

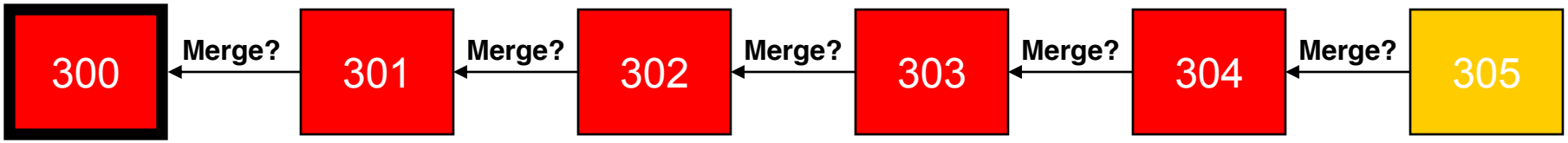
- Accurate fragmentation point detection is crucial for automated recovery



- Blocks have to be analyzed individually before being merged into a fragment.
- So how does one do this?
  - Sequential Hypothesis Testing!

Fragmentation Point Detection using Sequential Hypothesis Testing

- Core Problem



- Generic Case
  - A fragment consisting of blocks b_1, b_2 … b_{n-1}
  - Does the next block b_n belong to the same fragment?
- Formally we have two hypotheses
  - H_0 : b_n belongs with blocks b_1, b_2… b_{n-1} in sequence
  - H_1 : b_n does not belong with blocks b_1, b_2… b_{n-1} in sequence

Fragmentation Point Detection using SHT (cont'd)

- We define a decision statistic λ
  - We accept one of the hypotheses if λ is decisive.
  - Otherwise we don't make a decision and consider the next sequential block(s).
- SHT will adaptively decide the number of blocks before a decision can be made.
  - **Y** = $Y_1, Y_2, \ldots Y_n$ where Y_i = Matching Metric($b_1 \ldots b_{i-1}, b_i$)

Probability of blocks not belonging to fragment

$$\Lambda(\mathbf{Y}) = \frac{Pr(\mathbf{Y}|H_1)}{Pr(\mathbf{Y}|H_0}.$$

Probability of blocks belonging to fragment

SHT (cont'd)

- 3 possible outcomes

$$outcome\ of\ test = \begin{cases} H_1, & \Lambda(\mathbf{Y}) > \tau^+ \\ H_0, & \Lambda(\mathbf{Y}) < \tau_- \\ inconclusive, & \tau^- < \Lambda(Y) < \tau^+ \end{cases}$$

- When the matching metrics Y_i are independent

$$\Lambda(Y) \equiv \frac{Pr(\mathbf{Y}|H_1)}{Pr(\mathbf{Y}|H_0)} = \prod_{i=1}^{n} \frac{Pr(Y_i|H_1)}{Pr(Y_i|H_0)}.$$

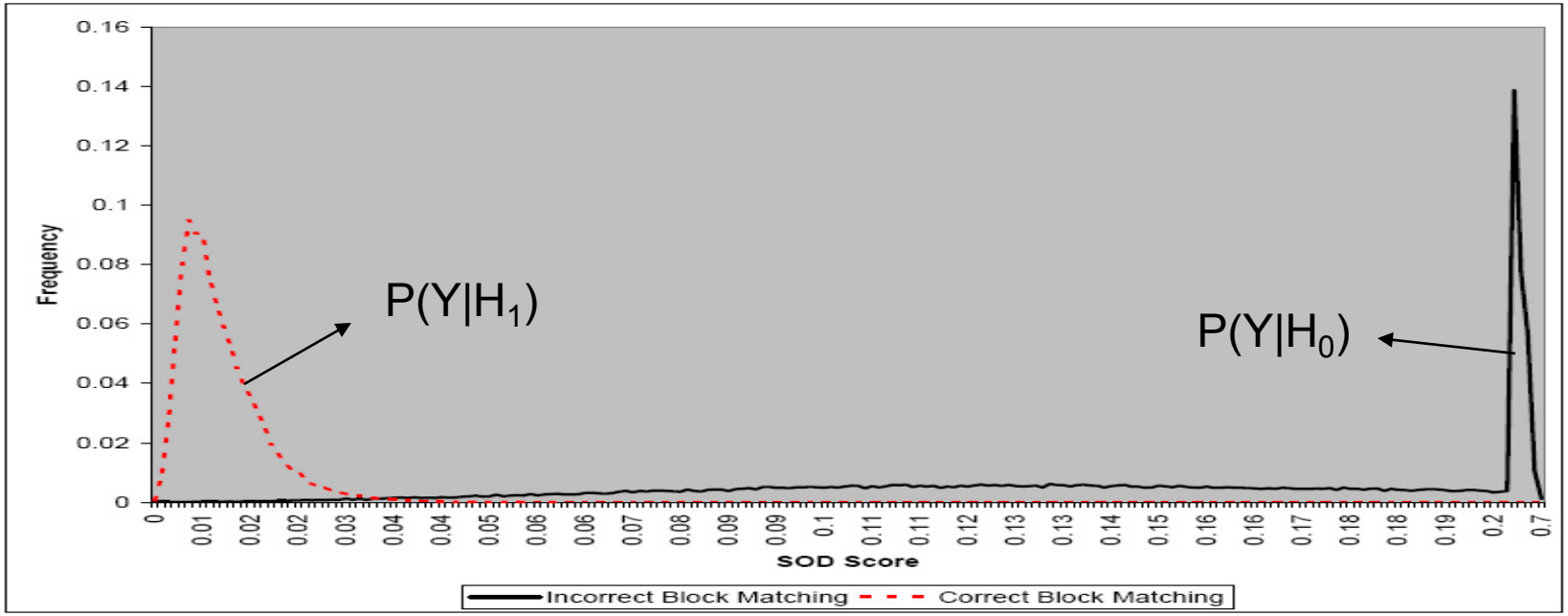
$$\tau^+ = \frac{1 - P_{fe}}{P_{fa}}; \text{and}; \tau^- = \frac{P_{fe}}{1 - P_{fa}}.$$

P_{fe} = Probability of false elimination

P_{fa} = Probability of false addition.

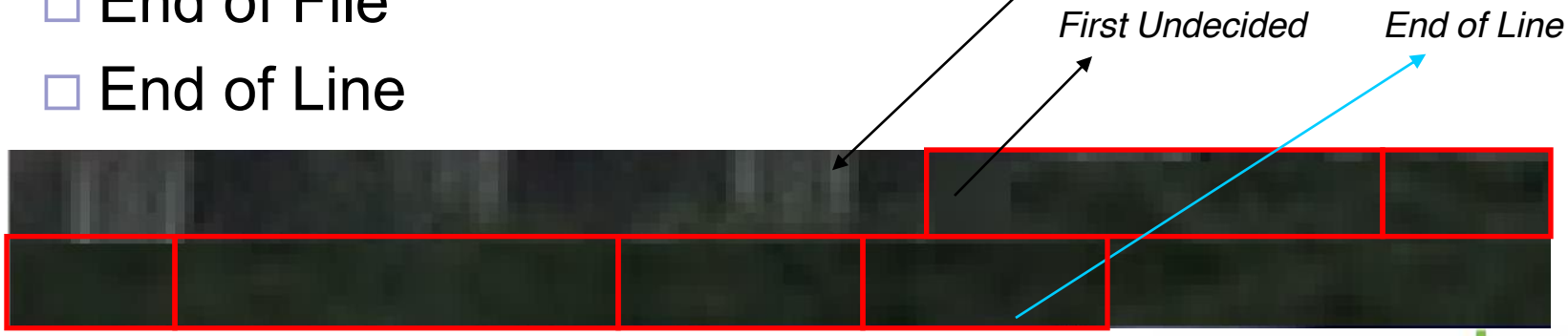
Building Models for use with SHT

- Models built using a training set.
- Jpeg: clear distinction between the scores that result in “correct mergings” and those that are “incorrect”



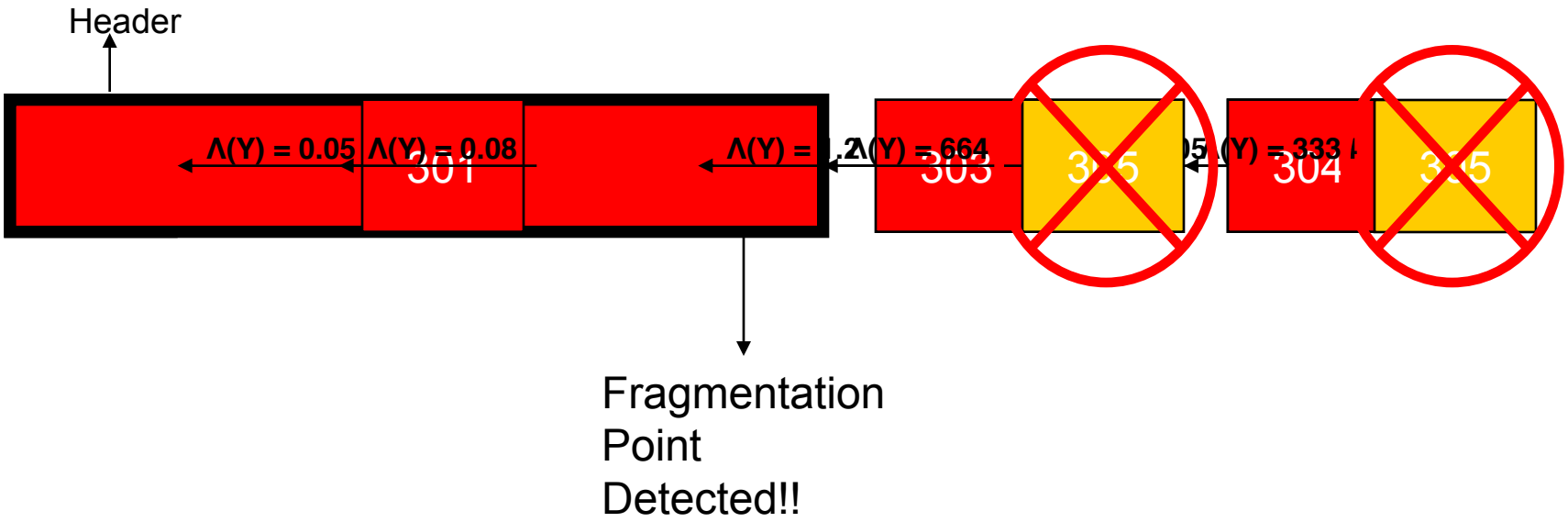
SHT Stopping Conditions

- Fragmentation Point Detection for a fragment stops when
  - H_0 is true (block decidedly does not belong to fragment)
  - End of File
  - End of Line



Fragmentation Point Detection Example

$T^+ = 1000$
$T^- = 0.06$



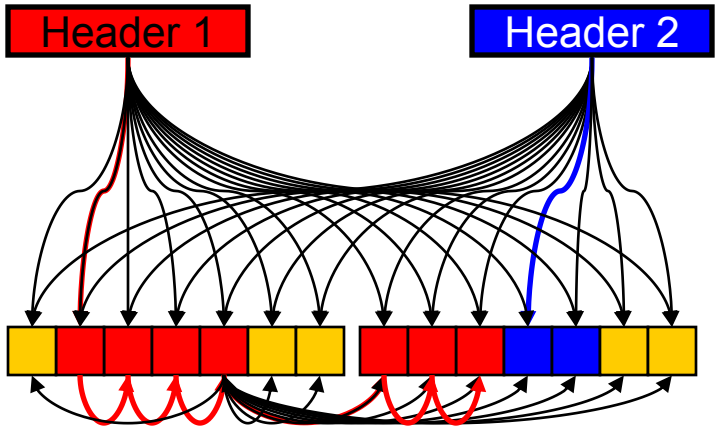
SmartCarving - Reassembly

Step 2: **Reassembly**

a) Find Continuation Point from those in Bin
   - i) Check Within **x** number of blocks
   - ii) Prioritize based on geometry

b) Find next fragmentation point or end of file
   - i) Using Sequential Hypothesis Testing

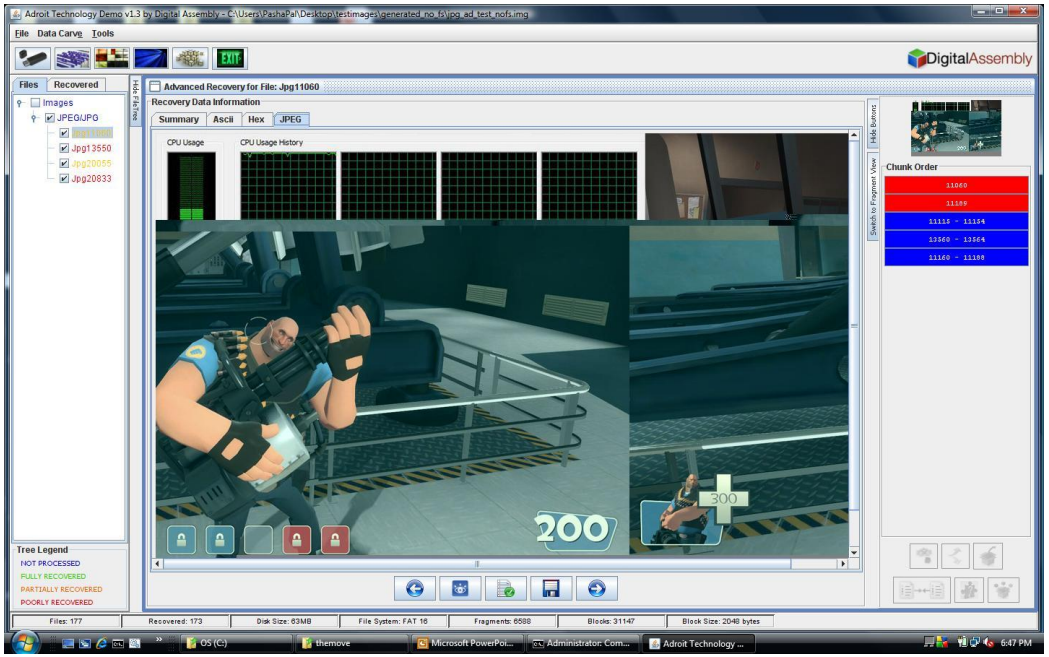
c) If not end of file repeat (a) – (b)



d) We can do this in Parallel using PUP!!!
   - i) Find the best of the best continuation points and then do SHT on that
   - ii) Repeat process exactly as in PUP

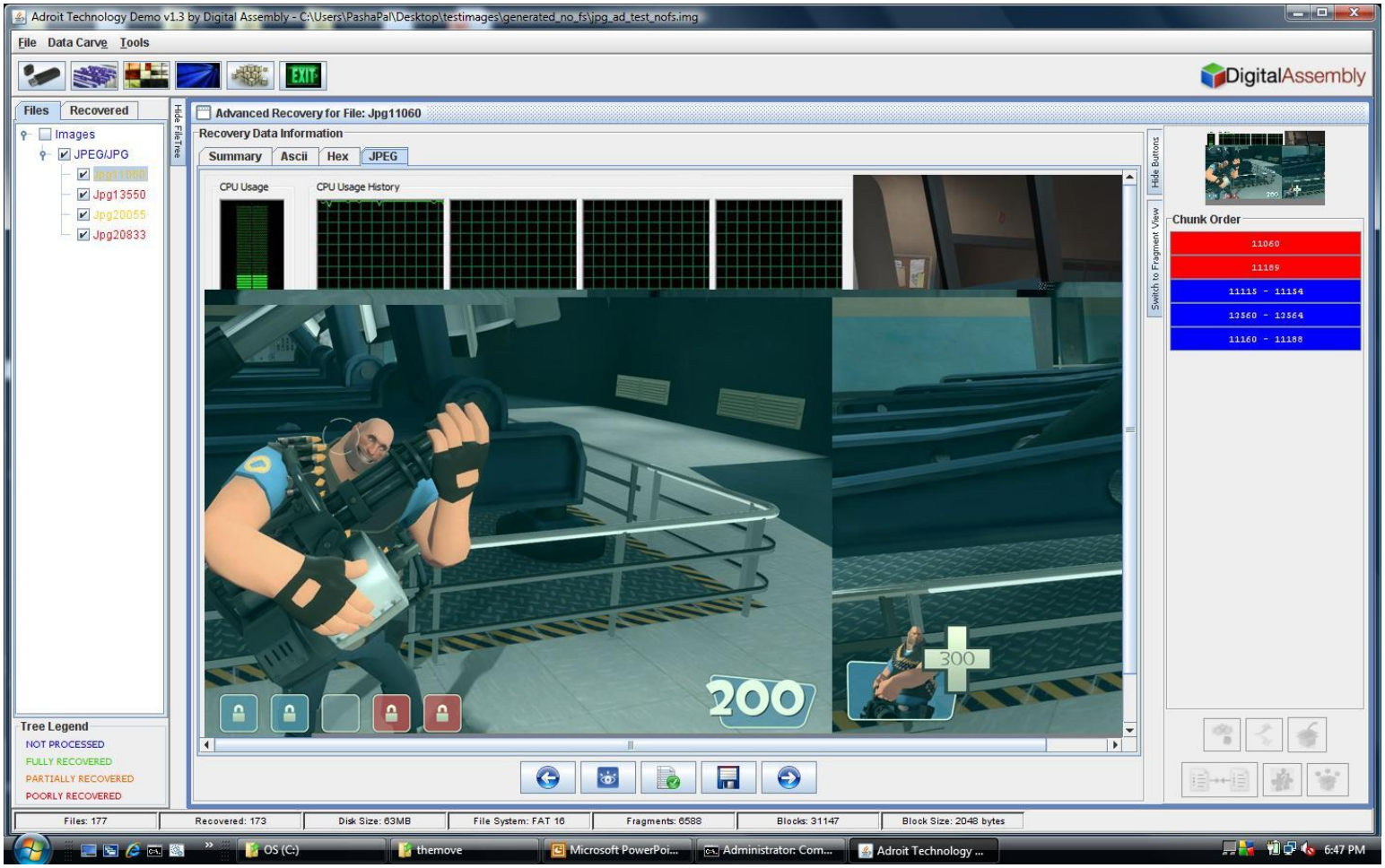
SmartCarving – User Interaction

- Step 3 - Allow user to specify changes
  - Indicate blocks don't belong together
  - Manually merge blocks
  - Move blocks between bins
- Reiterate based on user input.



NYU·poly
POLYTECHNIC INSTITUTE OF NYU

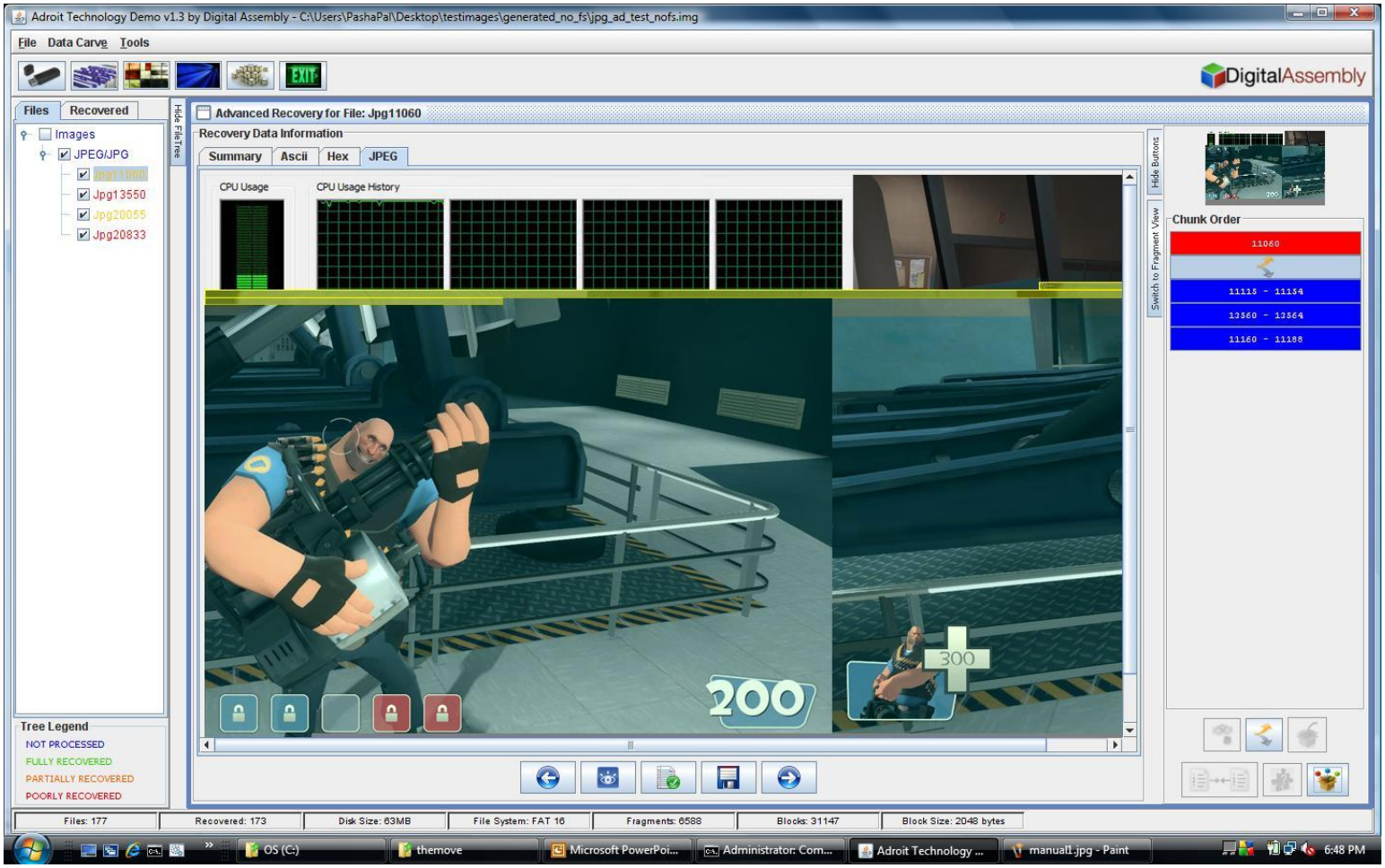
SmarCarving - User Interaction (II)



SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

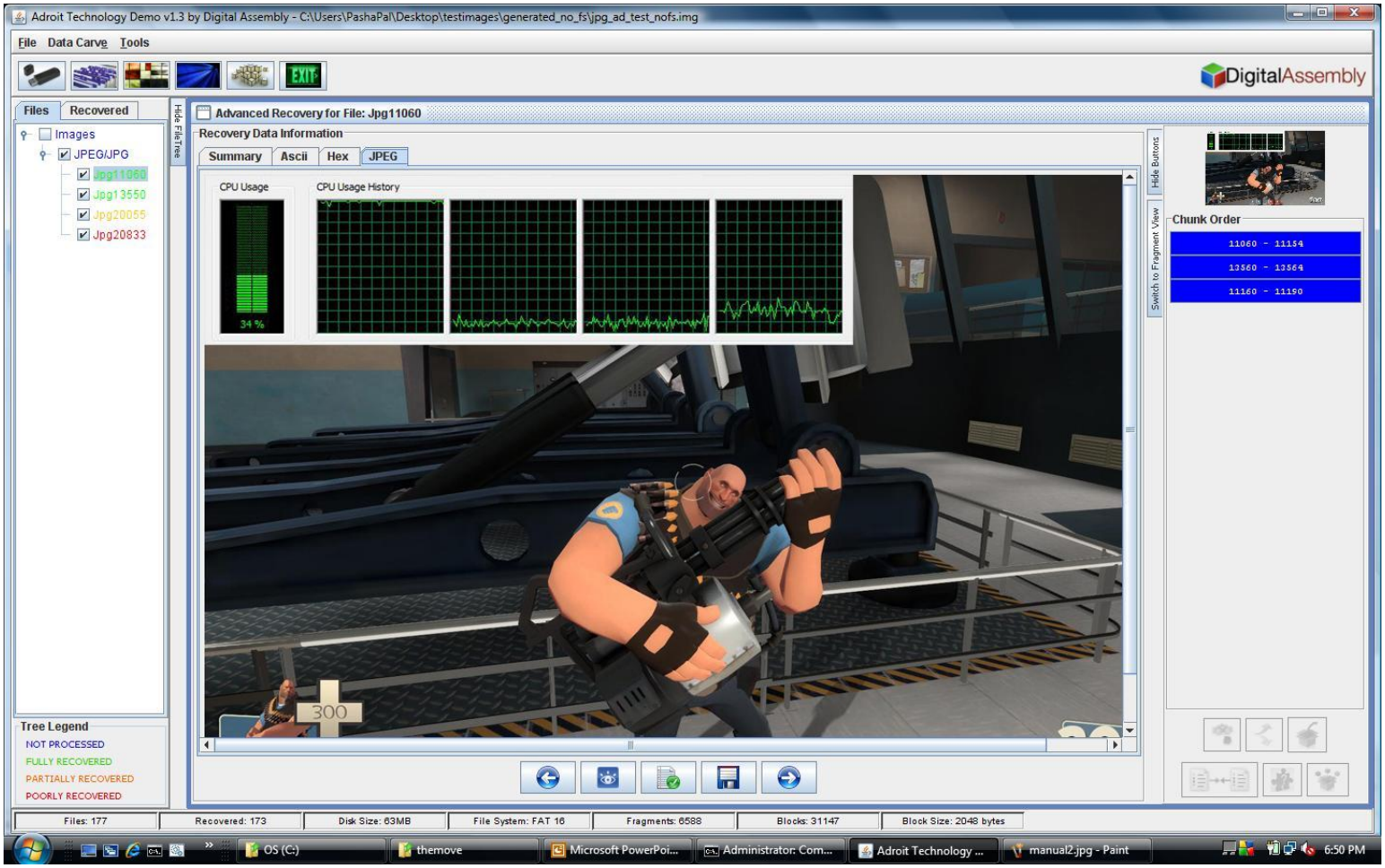
SmartCarving – User Interaction (III)



SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

SmartCarving – User Interaction (IV)



SmartCarving™

NYU·poly
POLYTECHNIC INSTITUTE OF NYU

SmartCarving - Advantages

- Don’t have to pre-compute all weights.
- Recovers multi-fragmented files
- Scales to millions of blocks
- Keys to success:
  - Finding continuation points
  - SHT to merge consecutive blocks and find fragmentation points

Experiments and Results

Disk Images	***DFRWS06***	***DFRWS07***	***Real-1***	***Real-2***	***Gen-1***
Fragmented Images	7	17	20	21	24
Bi-fragment Gap Carving					
Recovered Fragmented	7	1	-	-	-
Time Taken	9 sec	hours	-	-	-
SHT – PUP					
Recovered Fragmented	7	16*	18	16	20
Time Taken	13 sec	3.6 min	3.5 min	3.2 min	2.5

*Some images were not fully recovered due to missing or corrupt blocks.

Future Work

- Currently conducting SHT experiments with Office and HTML
- Work needs to be done to score video, audio and other formats.
- Work needs to be done to optimize continuation point algorithms for different file types.
- SSDs and file carving, how difficult will the problem be?