Recovery of Heavily Fragmented^ JPEG Files

SM Yiu
Department of Computer Science
The University of HK

Joint work with
Yanbin Tang, Junbin Fang, KP Chow, Jun Xu,
Bo Feng, Qiong Li, Qi Han

DFRWS 2016

^ Files with ≥ 3 fragments

Agenda

- ♬ Motivation
- ♬ Problem Descriptions
- ♬ Limitations of existing methods
- ♬ Our improved solution
- ♬ Experiments and conclusions

Motivation

More crime cases involve JPG files

e.g. A suspect forced a girl to take some dirty pictures

Later, he was caught, but before that, he deleted the evidence (jpg file) already!

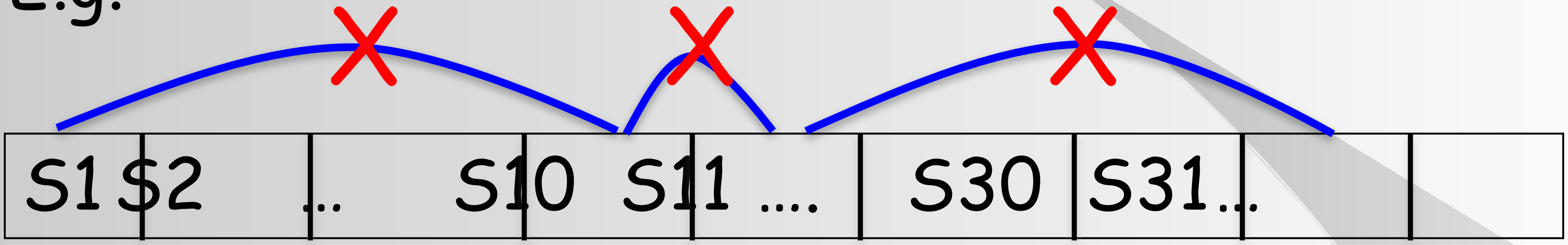
Q: How we can reconstruct the file from the deleted fragments from the storage media?

[Remark: of course, there are other applications]

Background

The same file may be put in the hard disk in parts (it happens for large files)

E.g.

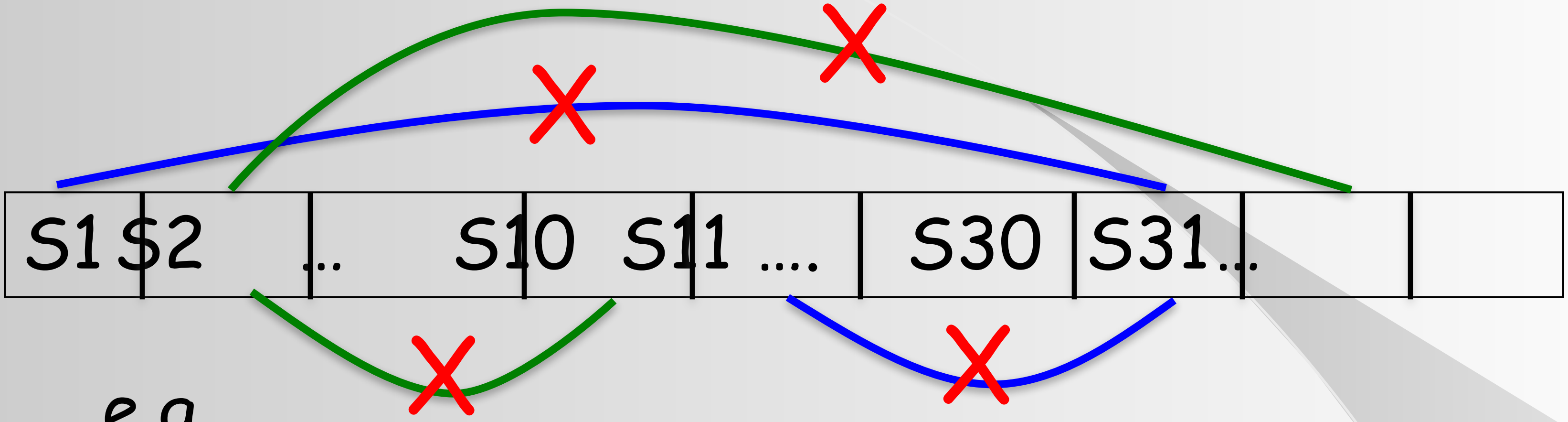


S1 S2 … S10 S11 …. S30 S31..

- Hard disk is divided into sectors (e.g. 1024 bit each)
- Where a file is stored is marked in a directory
  e.g. File 1: S1 -> S10 -> S11 -> S31

However, once a file is deleted, this chain information is deleted from the directory!!
=> Given one sector, not easy to tell what file it was belonging to.

Note that the sectors belonging to the same file may not be consecutive in the hard disk



S1 S2 ... S10 S11 .... S30 S31...

e.g.
File 1: S1 -> S30 -> S11
File 2: S31 -> S2 -> S10

Think about it: if all these linkages are lost, can you still reconstruct the files?
I.e. Given S1, S2, S10, S11, S30, S31 and no other information, what we can do?

Another workflow for undeleted files/directory info still exists

Suspect's hard disk/storage media

Isolate sectors that belong to "deleted files"

There are existing tools (e.g. Oscar) doing this

Classify each sector into different file types

Word

Text

jpg

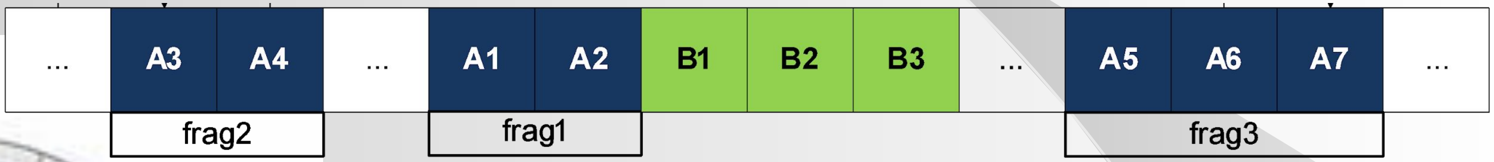
ppt

Note: sectors of same type may belong to multiple files

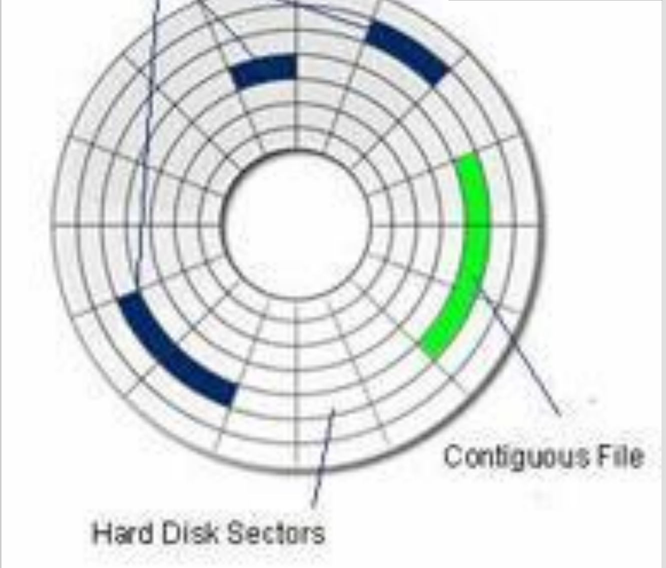
File carving / reconstruction

♫ Problem Description

Assuming that we identified some sectors from the harddisk that belong to jpg files (but the ordering is unknown),



…	A3	A4	…	A1	A2	B1	B2	B3	…	A5	A6	A7	…
	frag2			frag1						frag3			



Can we reconstruct (reorder the sectors) the jpg picture?

In this work, we assume: (1) all sectors still exist (not overwritten yet)*; (2) the directory info is gone.

* Header of a file can be found easily

♫ Limitations of existing methods

A naive solution: brute-force approach

- Checking all permutations of the sectors
- Look at each permutated file

N sectors (N can be thousands) => N! cases to consider: Too slow and not practical

One of the best existing solutions: Adroit Photo Forensics (APF) 2013

How it works?

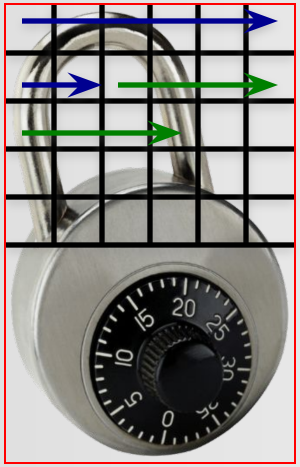
http://digital-assembly.com/products/adroit-photo-forensics/downloads/

Background

- jpg file is a compressed file
- The decompression starts from the 1st sector up to the end
- Each sector is decoding consecutive pixels

1st sector

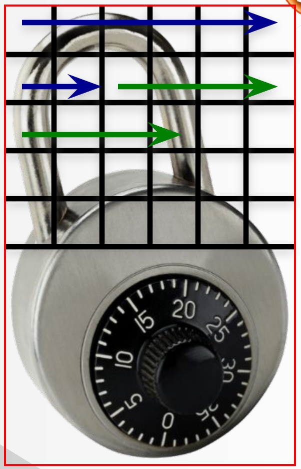
2nd sector



** Consecutive sectors should have pixels that are adjacent to each other!! **

1st sector

2nd sector



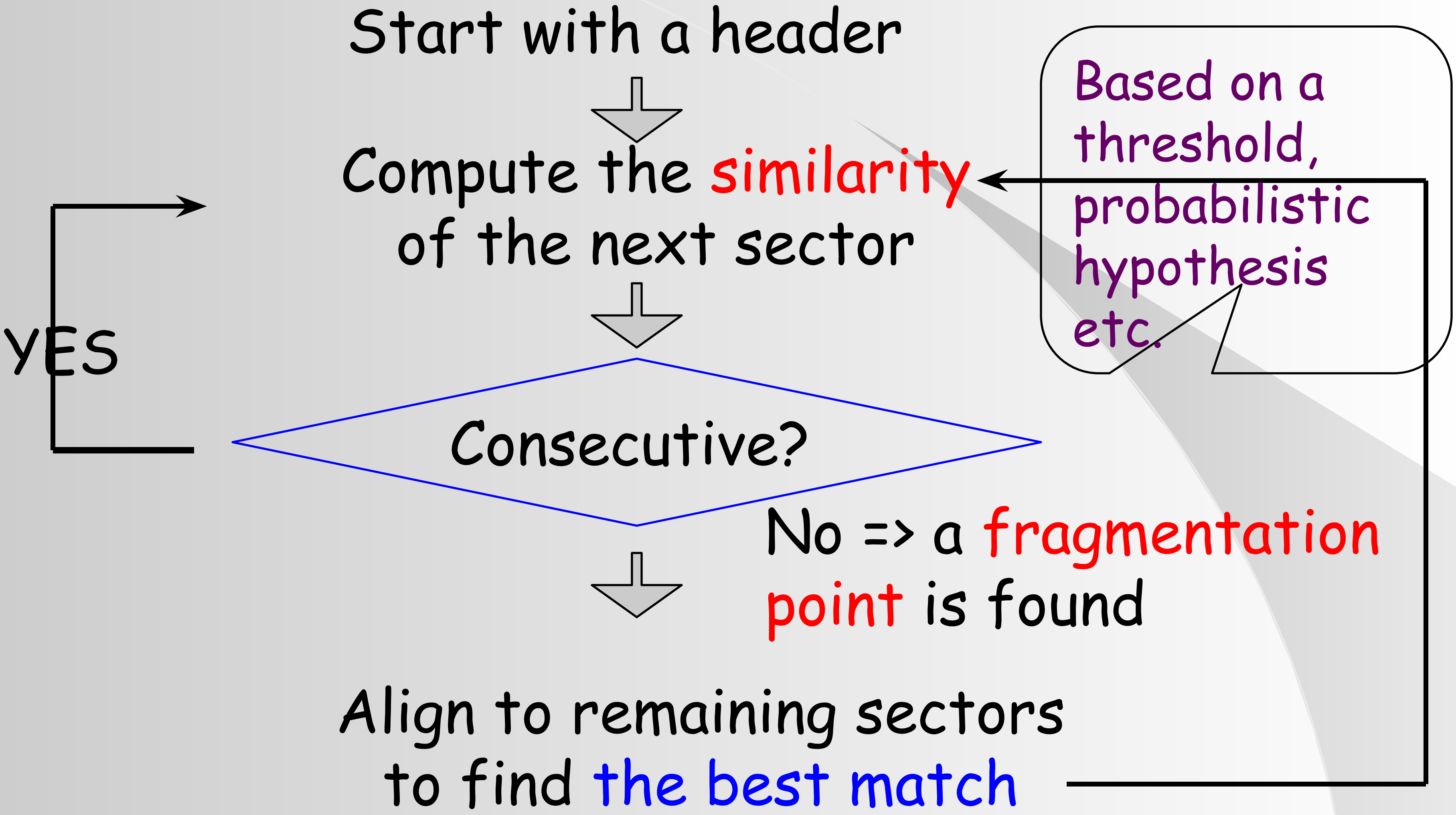
Key observation:
It two sectors are consecutive, the pixels along the connecting boundary are similar in values (color etc).

High level idea:

(1) Come up with a similarity measure to determine if two sectors are consecutive or not.

(2) Start from the header (easy to find by a signature string pattern), then use a heuristics to identify the next sector among the remaining sectors

Adroit Photo Forensic tool (APF)'s method



Start with a header

Compute the similarity of the next sector

Based on a threshold, probabilistic hypothesis etc.

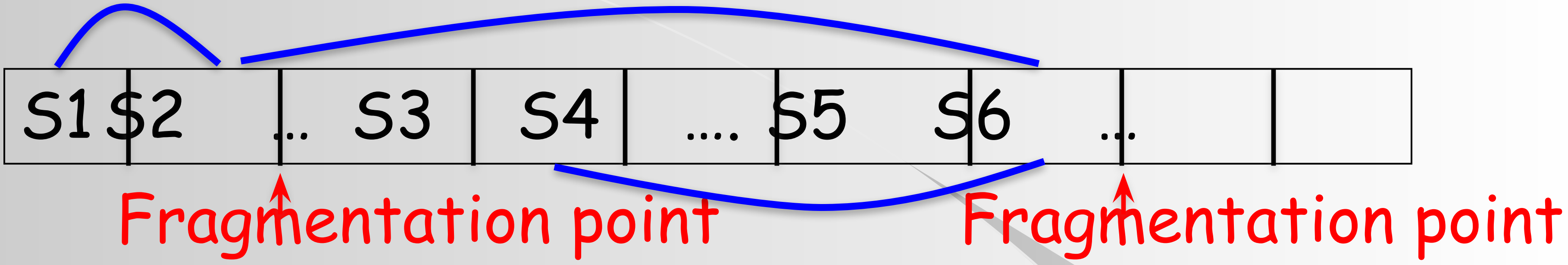
Consecutive?

YES

No => a fragmentation point is found

Align to remaining sectors to find the best match

An example: assuming S1 is the header:



S1 S2 .. S3 S4 .... S5 S6 ..

Fragmentation point

Fragmentation point

(1) Decompress S1, then S2, check the similarity between S1 and S2: Assume > threshold, then connect S1, S2.

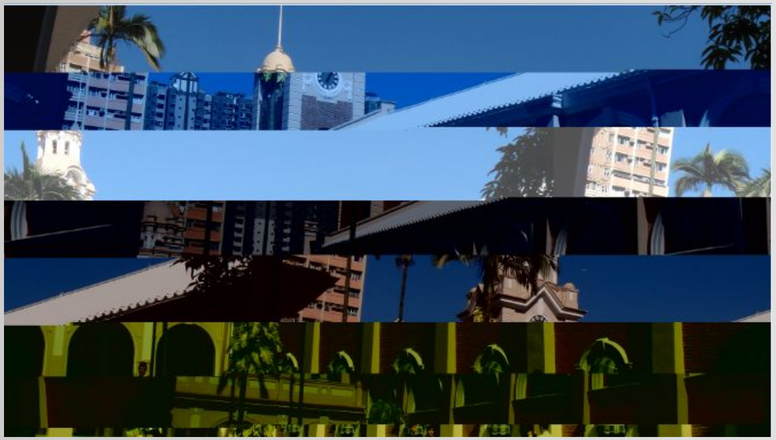
(2) Since the next one is not consecutive, we found a fragmentation point. Now, try to decompress S3, S4, S5, S6, compute similarity between (S2, S3) (S2, S4) (S2, S5) (S2, S6). Pick the highest score (say (S2, S5)), then connect them.

(3) Check similarity between S5 and S6, but assume it is smaller than the threshold => fragmentation point. Try to decompress S3, S4, compute similarity (S5,S3), (S5,S4), say (S5, S3) is higher, connect them etc…..

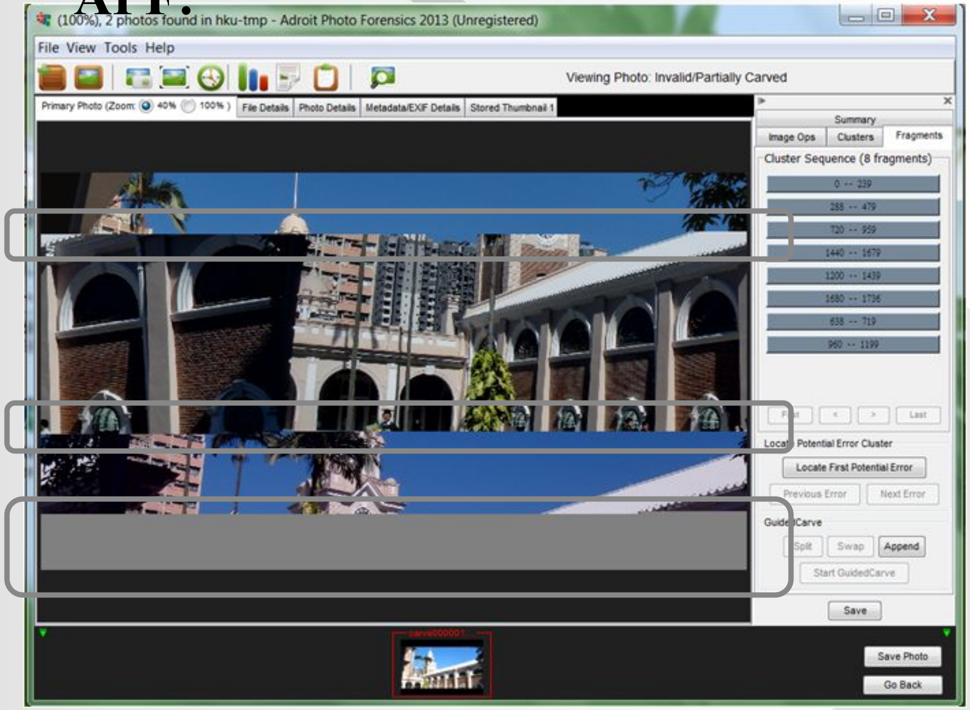
We found that their heuristics does not perform well if the no. of fragments is more (> 3)

Original JPEG file:

Input:
8 fragments with random order

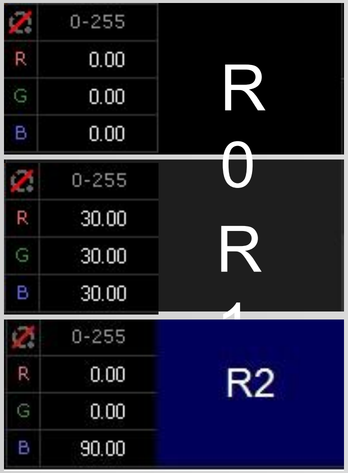


Recovered Result by APF:



Highlight of their Limitations

(1) Not so-good similarity measure



0-255	
R	0.00
G	0.00
B	0.00

0-255	
R	30.00
G	30.00
B	30.00

0-255	
R	0.00
G	0.00
B	90.00

SoD (Sum of Difference)

$$S = \sum_{i=1}^{n} |xi - yi|$$

$$\mathrm{SoD}_{R0R1} = |30| + |30| + |30| = 90$$

$$\mathrm{SoD}_{R0R2} = |0| + |0| + |90| = 90$$

RGB values:
R0 (0,0,0); R1 (30,30,30); R2(0,0,90)

Note: it is quite obvious that R0 is more similar to R1, then R2, but SoD cannot distinguish them!!

Another commonly used measure:

Euclidean Distance (ED)

$ED = 1/n \sqrt{}(x_i - y_i)^2$

We will also show that this measure is not always good.

(2) Fragmentation point detection problem

Using the best match candidate may not always give the correct answer.

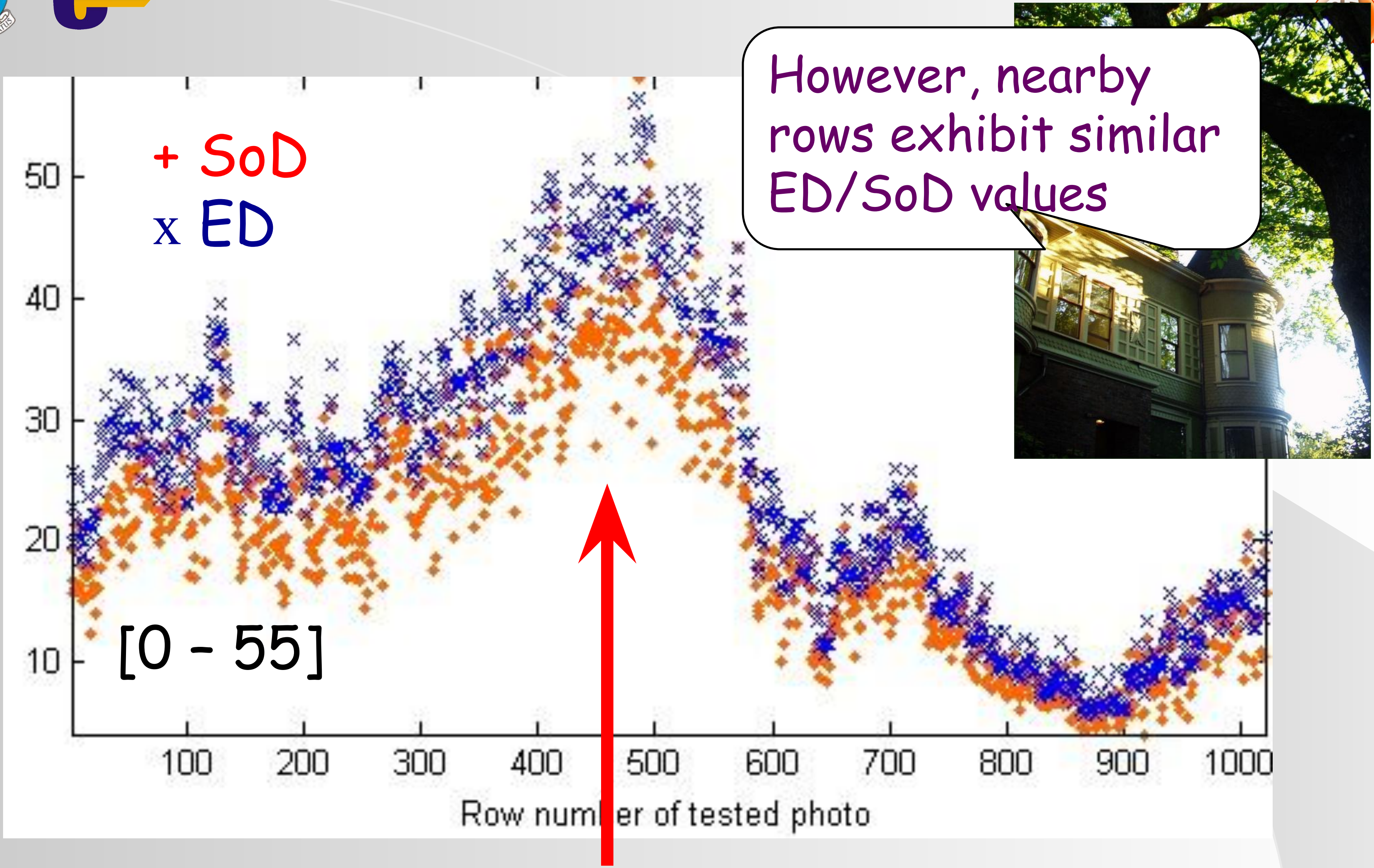
Note: we are not saying that both ED and SoD are always bad measures.

♫ Our improved solution

Both SoD and ED focus on the absolute differences between the boundary pixels.

For regions with varying colors such as tree leaves, it may be falsely identified as fragmentation points

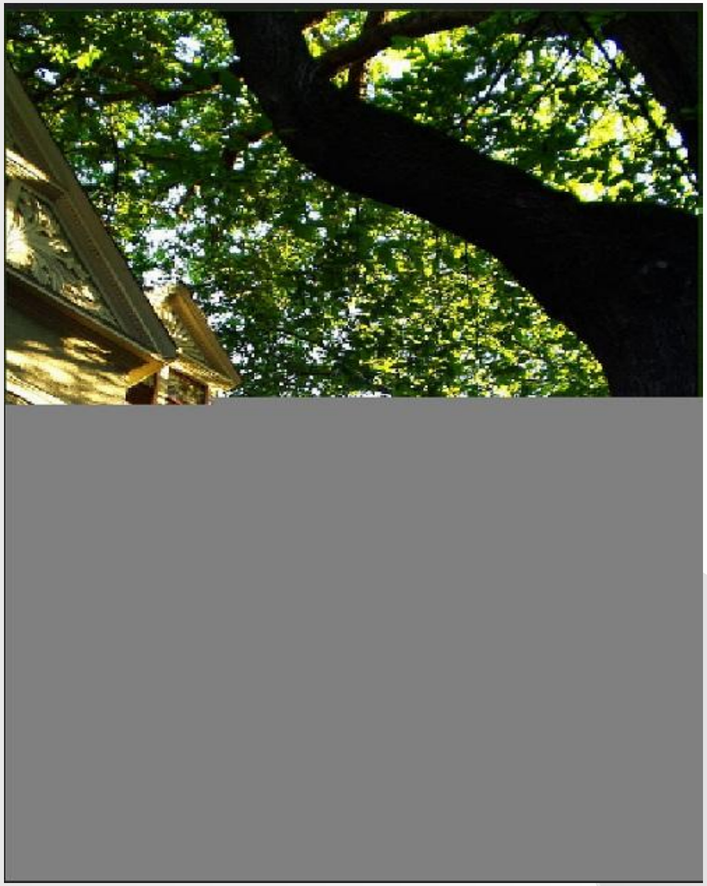
Contribution 1: We propose a variation of measure to tackle this issue.



However, nearby rows exhibit similar ED/SoD values

If fragmentation occurs here, likely not able to connect the sectors!

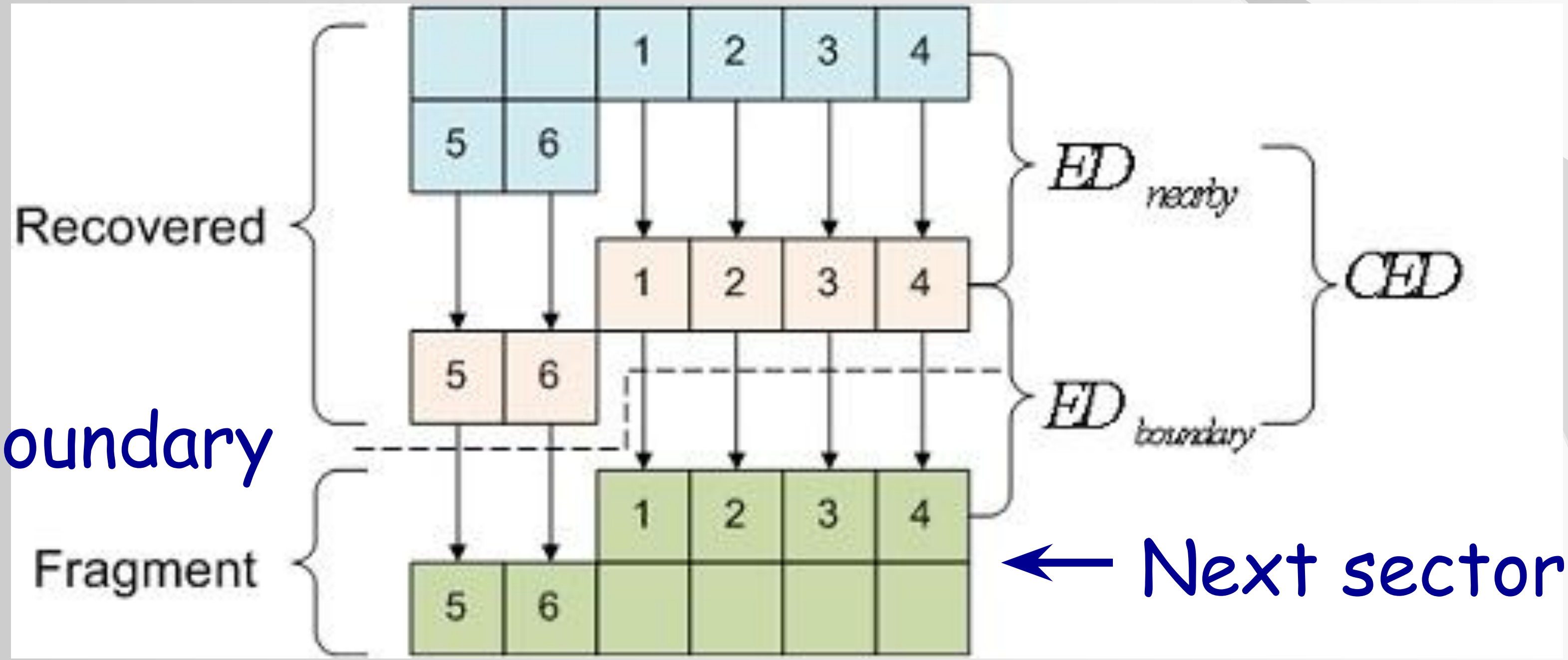
APF



Cut this into 4 fragments

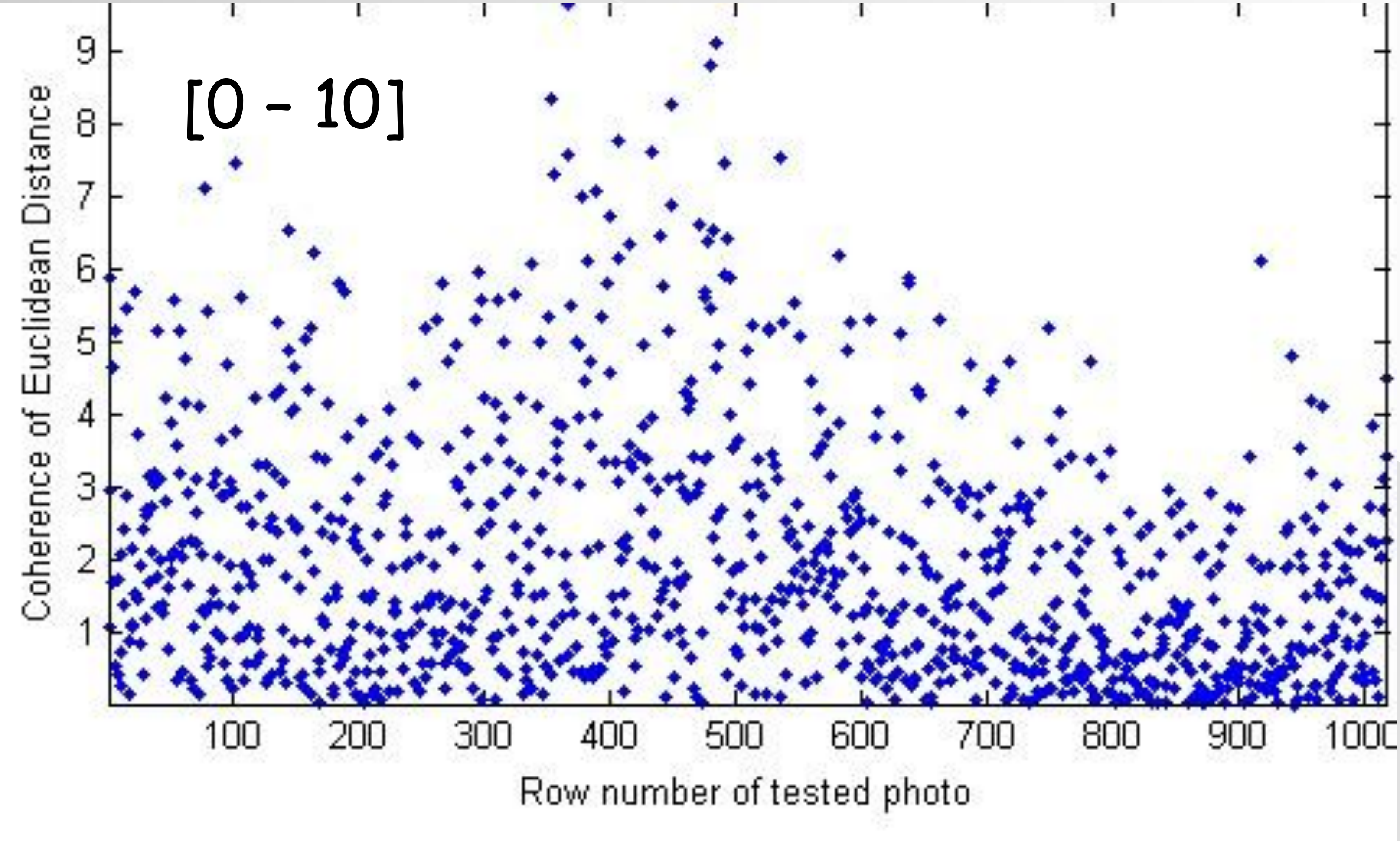
Coherence of Euclidean Distance (CED)

$$CED = |ED_{boundary} - ED_{nearby}|$$



Boundary

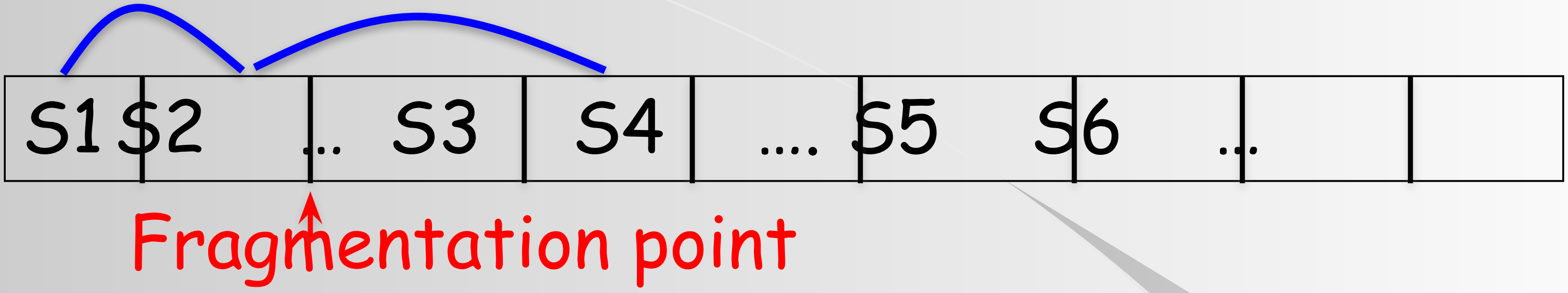
← Next sector


[0 - 10]
Coherence of Euclidean Distance
1
2
3
4
5
6
7
8
9
100
200
300
400
500
600
700
800
900
1000
Row number of tested photo

[Contribution 2: We also extend their candidate finding heuristics]

Instead of finding the "next" best candidate only, we keep m best next candidate, for each of these m candidate, we locate the next one, then use this look ahead step to re-confirm which of these m is the correct one.

A rough example: assuming S1 is the header:


S1 S2 … S3 S4 …. S5 S6 …

Fragmentation point

(1) Decompress S1, then S2, check the similarity (we use CED) between S1 and S2: Assume > threshold, connect S1, S2.

(2) Then, we found a fragmentation point. Now, try to decompress S3, S4, S5, S6, compute similarity between (S2, S3) (S2, S4) (S2, S5) (S2, S6). Pick the highest score (say (S2, S5), we do not connect them yet, but consider highest m scores. Say m = 2 (may be S5, S3)

(3) Consider the next sector of S5 and S3, say (S2, S5, S6) vs (S2, S3, S4): the overall score of (S2, S3, S4) is higher, then connect S2 with S3.

♫ Experiments

(1) CED vs SoD and ED

- Images are downloaded/taken from digital camera.
- CED, SoD, ED were used to connect adjacent rows.
- A false match (FM) = if two adjacent rows do not have the highest similarity measure.

100 (3648x2048) images

	CED	ED	SoD
# FM	1,829	115,142	128,805
FM rate	0.89%	56.22%	62.89%
# files w FM	59	100	100

(2) Carving performance

187 (87 sequential + 100 fragmented) pictures were randomly generated by digital camera

	# of files	Ours	APF
Sequential	87	87	87
2 fragments	79	78	65
3 fragments	18	17	11
4 fragments	2	1	2
6 fragments	1	1	0

For fragmented files, our method recover 97 files while APF can recover 78 files.

We analyzed three failure cases: 2 is due to a small fragment (not large enough for CED), 1 due to the dramatic change in color in picture.

♬ Conclusions

* We proposed a new jpg file carving algorithm

* The key ideas include a new similarity measure (CED) and top m best matches for fragmentation point matching.

* The performance of our method is shown to be better than APF in our experiments

* Unsolved problems

(a) 1000 low resolution (1024x768) images

	CED	ED	SoD
# FM	132,529	381,112	491,775
FM rate	17.26%	49.62%	64.03%
# files w FM	954	1000	1000

(b) Still cannot handle some “difficult” cases such as the ones with very varying colors/very similar colors

(c) How about some sectors overwritten….
[From the forensic point of view, we may need to get the whole picture, as long as the part of pictures can show evidence for the crime.]

(d) How about voice/videos (e.g. CCTV)…

‹ Thank you ›