IMAGE RECOVERY FROM BROKEN IMAGE STREAMS

Xiaoyu Zhao[a,*], *Chi Xu*[a], *Zheru Chi*[a], *Hong Yan*[b], *David Dagan Feng*[a], *and Gang Chen*[a]

[a]Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: enxyzhao@gmail.com
[b]Department of Computer Engineering and InformationTechnology
City University of Hong Kong, Kowloon, Hong Kong

ABSTRACT

This paper presents an approach for image recovery from broken image streams based on an image continuity model. The information to be recovered includes the header of an image (width, height, color space etc.) and the data stream which might have been partially damaged. We begin our discussion on common image formats and necessary information for the image recovery. We then define several measures on image contents to facilitate the image recovery. Using these measures, an effective recovery algorithm is proposed. Experimental results show that the algorithm can successfully recover images of different formats, even multi-frame images from broken image streams in most cases.

Index Terms — Image recovery, broken image streams, image formats, image continuity model

1. INTRODUCTION

An image file contains both a header which contains the information about the color space, image width, image height etc., and the image data (a binary stream). In image transmission, the image header might be damaged with the image data intact or partially damaged. In this situation, we simply cannot interpret the image file. Can we recover an image from a damaged image file? When the header of an image file is lost and the image data is partially damaged, can we recover the information of an image to interpret the image correctly? This paper try to answer these questions based on our research findings. Actually, we have tackled the problem in the following aspects:

(1). In the image format "Analyze", the image data is stored in a file with an extension of ".img" and the header information in another file with an extension ".hdr". When the header file is lost, we can recover the image by examining the image data file.

(2). In the DICOM standard [5], the binary file may contain several image frames, we can recover the images successfully without the header information.

(3). A raw image file, commonly a channel image data, usually stores an 8-bit grey image. In a raw image file, there is no header information (the raw data may be in a segment of main memory or in a file).

(4). In other commonly used image file formats, such as BMP, TIF, PPM, PGM, and PBM, uncompressed image data is stored. If the header is damaged, or the format is unknown, we can still recover the image.

2. GENERAL CONSIDERATION

At first, we need to formulate the problem to be tackled and do a necessary classification of damaged images.

Assume a broken binary stream file that contains a full or partial image data. Table 1 summarizes the features of a broken image stream.

Table 1: A broken image stream

Information	Description	Status
Stream	binary stream	known
Length	the length of broken stream file	known
Offset	the number of bytes from the beginning of the file to the beginning of the image data	unknown

We have to obtain sufficient header information from this stream in order to recover the image.

To recover an image's header, we have to determine the parameters summarized in Table 2.

Table 2: Header information

Parameter	Description
Pixel type	color space and data type
Alignment	Alignment by bytes: 1, 4 or 8
Width	image width
Height	image height

3. IMAGE MEASURES

We will first define several measures for recovering an image. Consider first a gray scale image I with a height of h and a width of w. The image is represented by

$$I = \begin{bmatrix} I_{11} & I_{12} & \ldots & I_{1w} \\ I_{21} & I_{22} & \ldots & I_{2w} \\ \ldots & \ldots & \ldots & \ldots \\ I_{h1} & I_{h2} & \ldots & I_{hw} \end{bmatrix}$$

Where I_{ij} $(i = 1,2,\ldots,h, j = 1,2,\ldots,w)$ is the grey scale of the pixel at location (i, j).

3.1. Crudeness

What is the main difference between an image data stream and a random data stream? Based on a continuity model, we believe that an image stream is smoother than a random stream and is continuous by nature. Therefore, we define a measure to compute the crudeness of a stream. Here crudeness is a function to estimate the vertical continuity of an image. We can use the crudeness function to estimate the number of bytes of a scan line.

3.1.1. Global Crudeness

A global crudeness measure f_{gc} is defined as

$$f_{gc} = \frac{\sum_{x=1}^{w} \sum_{y=1}^{h} \left| I_{x,y+1} - I_{x,y} \right|}{w*h}$$

where *w* and *h* are the width and height of the image, respectively.

3.1.2. Crudeness on Selected Columns

To speed up the processing, we can evaluate the crudeness on selected columns as given by

$$f_{sc} = \frac{\sum_{x \in S} \sum_{y=1}^{h} \left| I_{x,y+1} - I_{x,y} \right|}{w*l}$$

$$S = \{s_1, s_2, ..., s_l,\} \subset \{1,2,..., w\}$$

For example, S can be a set of all odd columns $\{1,5,9,...\}$.

3.2. Separability

Separability is a measure defined to estimate the horizontal continuity of an image. The measure can be used to determine the offset of the stream.

3.2.1. Basic Separability (f_{bs})

We calculate the basic separability f_{bs} between column x and column $x' = (x-1)$ column. If x = 1, then let $x' = w$.

$$f_{bs}(x) = \frac{\sum_{y=1}^{h} \left| I_{x',y} - I_{x,y} \right|}{h}, \quad x' = \begin{cases} x-1, \text{if } (x > 1) \\ w, \text{if } (x = 1) \end{cases}$$

3.2.2. Delta Separability (*ds*)

In many images, pixel intensity changes within a small range even if these pixels are in a homogeneous region. In this case, we define a delta separability (f_{ds}) as

$$f_{ds}(x) = \frac{\sum_{y=1}^{h} dta(\left| I_{x',y} - I_{x,y} \right|)}{h}, \quad dta(t) = \begin{cases} t, \text{if } (t > d) \\ 0, \text{if } (t \le d) \end{cases}$$

where x' is defined as in Section 3.2.1 and d is a preset threshold.

3.2.3. Sign Separation (*ss*)

The sign separability (f_{ss}) is defined as

$$f_{ss}(x) = \frac{\sum_{y=1}^{h} sig(\left| I_{x',y} - I_{x,y} \right| - d)}{h}, \quad sig(t) = \begin{cases} 1, \text{if } (t > 0) \\ 0, \text{if } (t \le 0) \end{cases}$$

3.3. Invariability

Invariability measure is used to determine whether it is a smooth area where no significant change of pixel intensities is observed. The following definitions are based on columns. Similarly, we can define the measures based on rows. If a column or row is of a constant intensity C, obviously this line is an invariable line. To handle the pixels in an area are of intensities within a small range, we define the invariability as

$$f_i(y) = \frac{\sum_{x=1}^{w} sig(\left| \left| I_{x,y} - m \right| - d \right|)}{h}$$

where *m* is the mean value of pixel intensities of row *y* and d is a preset threshold (e.g., 5). The sig function is the same as in Section 3.2.3.

3.4. Boundary Function

3.4.1. Frame Boundary (fb)

In order to break the frames in a multi-frame data streams, we define a boundary function (f_{fb}) as

$$f_{fb}(y) = \frac{\sum_{x=1}^{w} slb(x, y)}{h}, \quad slb(x, y) = \begin{cases} 1, \text{if } (x, y) \in S_b \\ 0, \text{else} \end{cases}$$

where S_b is the set of pixels on the boundary lines generated by the JSEG [2] or *ncut* [4] segmentation algorithms.

3.4.2. Region Boundary (rb)

k image lines near the image margins are used to determine whether there is a clear boundary within these lines. A clear boundary means there are noisy lines in the margins. We define a region boundary f_{rb} as

$$f_{rb}(y) = \min(y \in \{1,2,...,k\} \ \& \ list(y) = 1)$$

$$list(y) = \begin{cases} 0, & if\,(sv(y) = sv(1)) \\ 1, & if\,(sv(y) != sv(1)) \end{cases}$$

$$sv(y) = \frac{\sum_{x=1}^{w} I_{x,y}}{w} - m, \qquad m = \frac{\sum_{x=1}^{w}\sum_{y=1}^{k} I_{x,y}}{w * k}$$

4. RECOVERY OF IMAGES BASED ON EVALUATION FUNCTIONS

Our proposed image recovery approach consists of several steps illustrated in Fig. 2.

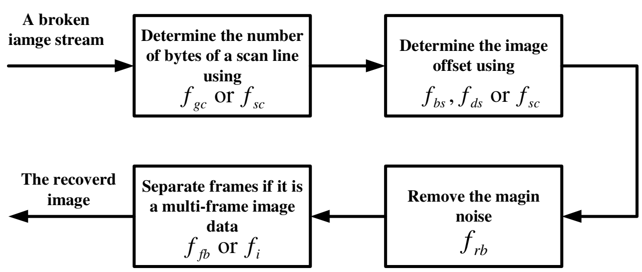


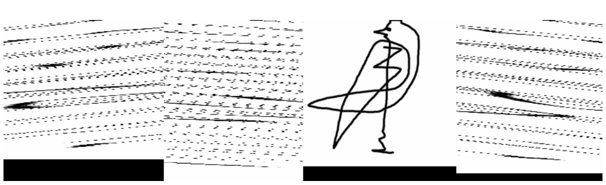
Figure 1: The proposed approach for image recovery.

4.1 Determination of the number of bytes on a scan line

The first and the most important step is to determine the number of bytes on a scan line (*nbl*). When the pixel type is Gray(UINT8) and alignment is 1, *nbl* is equal to the width of the image. Consider a gray scale image first and assume the possible image frames to be $\{I^1, I^2, I^3, ..., I^k\}$.

Assume a binary image stream. Suppose it is an 8-bit gray image with a stream length of 46200. By the factor decomposition, we have $46200 = 2 \times 2 \times 2 \times 3 \times 5 \times 7 \times 11$

There are several possible ways for setting the image width and height, e.g., four settings illustrated in Fig. 2.



(a) (b) (c) (d)

Figure 2: Four images formed from four different setting of image width and length. (a) width=231 and height = 200; (b) width=200 and height=231; (c) width=220 and height=210; (d) width=210 and height=220.

From the image formed with different settings as shown in Fig. 1, we can see that the third setting is most likely the correct one. Now the question is how the computer knows which is the proper setting.

The crudeness measure (f_{gc} or f_{sc}) defined in Section 3.1 can be used to determine a possible setting of the image width and height. For each possible setting, the crudeness function is computed. The one with the minimum crudeness measure is the optimal solution. f_{sc} is more efficient than f_{gc}, while f_{gc} is likely to be more accurate than f_{sc}.

4.2 Determination of the image offset

If the beginning of an image data is not known, then we have to determine the offset of the image stream.

(a) (b)

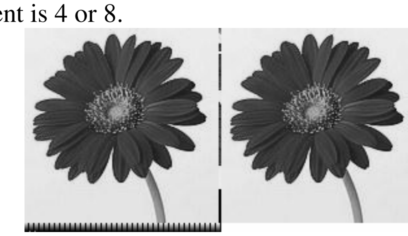
Figure 3: An image with a known starting point and an image with the starting point unknown.

In this case, an image cannot be successfully recovered even we know the width of the image. In Figure 3(a), the zebra is separated into two parts mistakenly.

The separability measure (f_{bs}, f_{ds} or f_{ss}) defined in Section 3.2 can be used to determine the offset of an image data. Assume that each position ($x = 1,2,...,w$) is a possible separation. The position with the maximum separation measure is the optimal solution.

4.3 Removing margin noise

Margin noise may be introduced in two ways: (1) header data might be interpreted as the image data and appears as noise in the top or bottom regions as shown in Fig. 4; (2) the right or left margin of an image is filled with noise when the alignment is 4 or 8.



(a) (b)

Figure 4: An image with margin noise (a) and that with the margin noise removed (b).

4.3.1. Top or Bottom Noise Lines

The header information occupies different number of bytes for different formats adopted. We should set the upper limit of bytes the header information will occupy in order to make our method work. In most cases, image data has a header of shorter than 1024 bytes. So we may set the upper limit to 1024 bytes. Using the boundary measure (f_{rb}) defined in Section 3.4.2, we can determine the number of lines for the margin noise and hence determine height of the image.

4.3.2. Left or Right Noise Lines

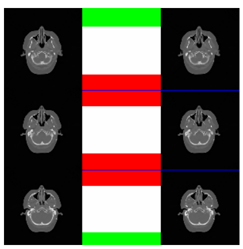
Some image formats, such as BMP, require a scan line with bytes of multiples of 4. If the width equal to 29, there are three bytes (32-29=3) useless in each line. We can remove the right or left margin by using (f_{rb}).

4.4 Separation of a multi-frame image data

Some image formats, such as DICOM, contain multi-frame image data, where individual frames need to be separated from the image data.

4.4.1. Buffer cache

In the first case, a buffer zone exists between two consecutive frames as shown in Figure 5.



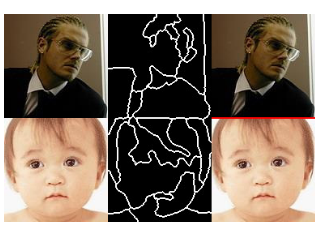
(a) (b) (c)

Figure 5: Multi-frame image data with a buffer between two consecutive image frames. (a) is the whole multi-frame image, and (b) is invariable area between the frames. (c) is the separated result.

We can determine the buffer region using the invariability measure (f_i) defined in Section 3.3. Except for the first (top) and the last (bottom) regions, the boundary is set in the middle of each buffer region.

4.4.1. Clear Boundary between two image frames

In the second case, there is a clear boundary between two image frames as shown in Fig. 6.



(a) (b) (c)

Figure 6: A clear boundary between two images.(a) is an image consisted of two frame with clear boundary and (b) is the boundary which is generated by JSEG segmentation algorithm. (c) is the separated result.

The frame boundary measure (f_{fb}) defined in Section 3.4.1 can be used to determine the separation line of two consecutive image frames.

5. EXPERIMENTAL RESULTS

We conducted simulation on 2691 images among which 2050 images have clear objects and 641 images are texture-based. Table 3 shows the result on the estimation of the number of bytes of a scan line on 2691 images.

Table 3: Scan line estimation with crudeness measures

	f_{gc}	f_{sc}
Accuracy	97.03%	98.66%

We computed the offset of the image stream for all 2691 images. These images all have an offset of 8 bytes.

Table 4: Experimental result on the estimation of offset

	f_{ss} $(d=5)$	f_{ss} $(d=1)$
accuracy	87.89%	64.36%

We tested a DICOM image which has 34 frames (part of the image is shown in Fig. 5), the frames can be separated correctly by using the f_i measure.

6. CONCLUSION AND FUTURE WORK

This paper firstly discusses the problems in image recovery with damaged or/and missing image data in transmission. Several measures have been defined to facilitate the image recovery by using our proposed image recovery approach. Testing on 2691 real images showed very promising results.

7. ACKNOWLEDGEMENTS

The work described in this paper was supported by the research grant from The Hong Kong Polytechnic University (project code: G-U285).

8. REFERENCES

[1] X. Zhao, G. Chen, and J. Li, *Image Processing using UIMP*, Zhejiang University Publisher, Hangzhou, China, January 2007.

[2] Y. Deng, B.S. Manjunath and H. Shin "Color image segmentation", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '99, Fort Collins, CO, vol.2, pp.446-51, June 1999.

[3] H. Fu, Z. Chi, and D. Feng, "Attention-Driven Image Interpretation with Application to Image Retrieval," Pattern Recognition, Vol. 39, Issue. 9, pp. 1604-1621, 2006.

[4] Shi J,Malik J.Normalized cuts and image segmentation.IEEE Trans Pattern Analysis and Machine Intelligence, Augest 2000

[5] http://medical.nema.org/