

# Inversion detection in text document images

Hamid Pilevar<sup>1</sup> A G Ramakrishnan<sup>2</sup>

Medical Intelligence and Language Engineering Lab

Department of Electrical Engineering

Indian Institute of Science, Bangalore - 560 012

<sup>1</sup> Assistant Professor <sup>2</sup> Associate Professor

<sup>1</sup> [pilevar@ee.iisc.ernet.in](mailto:pilevar@ee.iisc.ernet.in), <sup>2</sup> [ramkiag@ee.iisc.ernet.in](mailto:ramkiag@ee.iisc.ernet.in)

## Abstract

OCR makes it possible for the user to edit or search the document's contents. In this paper we describe a special water fill technique for detecting the upside down text document. Each character has a upside and downside filling capacities. A character may have two sides or one side filling capacity or zero filling capacity. The total upside and downside capacities for the scanned page calculated and the page with bigger downside capacity decided to be upright. The merit of the algorithm is that it requires only simple arithmetic operations per image pixel. Our experimental results, based on detecting inversion for 100 documents demonstrate a high detection performance of more than 98%, indicating the validity of the proposed methods.

**Keywords:** OCR; document processing; Inversion detection

## 1. Introduction

Optical character recognition is an important component of any document image analysis and recognition system [1]. Feature extraction involves identification of the bounding box of the character or symbol and locating its position with reference to the zone boundaries, the detection of zone boundaries is explained in [3]. An optical character recognition (OCR) algorithm must be robust in performance across large variations in style, font, shape, size, slant, background, quantization-level and general image quality. In real-time system applications [2], an additional constraint is ease of computation and fast recognition. Most of the OCR systems assume that the document is placed in upright position. However, there may be occasions where the document may not be placed on the scanner in an upright position. We are working on an automated book reader for the visually challenged, and the above is a very likely scenario in this application.

In this paper, we address this issue by proposing a robust method for detecting the upside down text document images.

## 2. Inversion Detection in Text Document Images

A special water fill technique is used to detect text documents that have been scanned upside down (inverted). This exploratory step comes after the character boxes are distinguished and right before character recognition process. Some of the characters have the potential of being filled from bottom or top or both sides. Also, some of them have some filling capacity, while the others don't. For discovering such characters we look at each character box and mark the right and left side edges, then look for concave or convex shape of the characters. If there is any concave part, then, it will be the character's jar or filling place. It is clear that, if the concave part is on the upper side then it is known as upper side jar and if it is on the bottom side then it will be bottom or down side jar. As shown in figure1, the capacity jars may be located in both sides of characters for example character "H", or one side of character, such as "U". If the capacity jar is smaller than  $\epsilon$ , then it will be neglected. For example, the capacity of upper jar of letter "h" is smaller than  $\epsilon$ , where  $\epsilon$  is selected to be 15 pixels. As we can see in figure1, for letters such as "O", there isn't any jar capacity, while some of the characters have only one side filling capacity, for example letter "U" has only down side jar.

The fill capacity of upside and then downside jars have been calculated as follows:

First, the fill capacities of the letters in each line and ultimately in the scanned page have been calculated. The upside fill capacities of the letters and number is defined set U, which has been called Upside Fill Jar Set or simply set U. We have called the set of down side fill capacities of the letters and number Downside Fill Jar Set or simply set D.

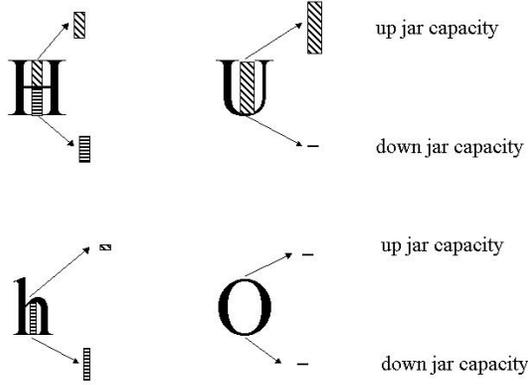


Fig 1: Upside and downside filling jars

Let  $U_1$  be the up fill capacity of letter "A",  $U_2$  be the up fill capacity of letter "B",  $U_3$  be the up fill capacity of letter "C" and so on for other capital letters up to  $U_{26}$  for capital letter "Z". The fill capacity of lower case letter "a" is known by  $U_{27}$ , "b" by  $U_{28}$  and at so on till letter "z" is known by  $U_{52}$ . Numbers are started from  $U_{53}$  for digit "0" and ends by  $U_{62}$  for digit "9". Therefore the set of up side fill capacities is defined as follows:

$$U = \{U(C_i)\} , C_i \in S$$

where:  $S = \{A, B, \dots, Z, a, b, \dots, z, 0, 1, \dots, 9\}$

The same manner has been used for down side jar capacities. The set of down side fill capacities is defined as follows:

$$D = \{D(C_i)\} , C_i \in S$$

where:  $S = \{A, B, \dots, Z, a, b, \dots, z, 0, 1, \dots, 9\}$

It is clear that the value of the filling capacities for some of the members in sets U or D are zeros; for example, the upsides jar capacity of letter "A" is  $U_1$  which is zero, the up and down side capacities of letter "O" are  $U_{15}$ ,  $D_{15}$  which are both equal zeros. The letter "U" has the maximum jar capacity in set S (figure1), namely a value of 120 pixels.

**Definition 1:** Fill capacity of up side jars  
The fill capacity of up side jars of a page of document can be calculated as follows:

$$U_c = \sum_{i=1}^n u(C_i) , C_i \in S$$

where: n is the total number of characters in the page document

**Definition 2:** Fill capacity of down side jars  
The fill capacity of down side jars of a page can be calculated as follows:

$$D_c = \sum_{i=1}^n d(C_i) , C_i \in S$$

where: n is the total number of characters in the page document

However, our experiments shows that in most of the tested documents  $D_c > 2 U_c$ . For clarity, we define an inverted document as follows:

**Definition 3:** Upside down text document images  
If the fill capacity of downside  $D_c$  is bigger than fill capacity of upside  $U_c$  ( $D_c > U_c$ ), then we can say that the page has been scanned upright; otherwise, we say the text document page has been scanned upside down and it must be rotated by 180 degrees.

## 2.1. Experimental Results 1

In scanned document figure2, the threshold  $\epsilon$  is selected to be 15 pixels. In Table1, the number of upside and downside jars for each line and for the page has been shown. As we can see, the number of downside jars in all of the six lines are more than the upside jars. The number of upside jars in the page is 26 and number of downside jars are 57, which is more than twice. It is good to mention that in average, the capacity of downside jars are more than upside jars, therefore the difference between downside and upside jars in this example is much more than two times.

Jars	upside	downside
Line 1	2	8
Line 2	6	7
Line 3	5	12
Line 4	4	13
Line 5	5	11
Line 6	4	6
Page	26	57

Table1: The number of upside and downside jars for scanned document figure 2

This paper concerns the extraction of rotation invariant texture features and the use of such features in script identification from document images. Rotation invariant texture features are computed based on an extension of the popular multi-channel Gabor filtering technique, and their effectiveness is tested with 300 randomly rotated samples of 15 Brodatz textures. These features are then used

Fig 2: The marked water fill jar letters. Number of upside jars are 26 and downside jars are 57.  $\downarrow$  indicates Up side Jar letters and  $\uparrow$  indicates Down side Jar letters. Threshold  $\in = 1$

Document Recognition or Optical Character Recognition software (OCR software), can be used to automatically capture and store your documents digitally. The scanning software handles all types of documents such as forms, e-mails, invoices, paper faxes etc. Using character recognition the data capture software automatically interprets, verifies and transfers the information, from any source to any system. Since almost no invoice or form look alike, the document recognition software has to be self-learning,

Fig 3: The marked water fill jar letters. Number of upside jars are 37 and downside jars are 75.  $\downarrow$  indicates Up side Jar letters and  $\uparrow$  indicates Down side Jar letters. Threshold  $\in = 15$ .

## 2.2. Experimental Results 2

In the document shown in figure 3, the threshold  $\in$  is again selected to be 15 pixels. In Table 2, number of upside and downside jars for each line and for the page has been shown. The number of downside jars in all of the six lines are more than that of the upside jars, while the number of upside jars in the entire page is 37 and number of downside jars are 75. Once again the downside jars are twice in number.

Jars	upside	downside
Line 1	5	14
Line 2	8	10
Line 3	7	13
Line 4	6	8
Line 5	5	16
Line 6	6	14
Page	37	75

Table 2: The number of upside and downside jars for scanned document figure 3

Our results, based on analyzing around 100 document images containing about 3000 lines and 150,000 characters, demonstrate a high performance of more than 98%, indicating the validity of the proposed method.

## 3. Discussion

The effectiveness and performance of our algorithm have been tested on samples collected from various images of legal documents belonging to one city. Around 100 document pages are scanned at 300 dpi and binarized using the two-stage method described in [4]. Determining the valley points in the horizontal and vertical projection profiles segments the textual lines. It is assumed that all the characters of a text line are of the same font size. Our experiments show that 98% of inversion detections are correct therefore, the proposed algorithm performs satisfactorily.

## 4. References

- [1] G. Nagy, Optical character recognition--theory and practice, in Handbook of Statistics-Classification, Pattern Recognition and Reduction of Dimensionality, Krishnaiah, P. R. and Kanal L. N. eds, pp. 621-649 (1982).
- [2] S.N. Srihari, High-performance reading machines, Proc. IEEE 80, 1120-1132 (1992).
- [3] Dhanya, D., Ramakrishnan, A.G.: Script identification in printed bilingual documents. Published in the same (2002).
- [4] Dhanya, D.: Bilingual OCR for Tamil and Roman scripts. Master's thesis, Department of Electrical Engineering, Indian Institute of Science (2001).