April 2022

A METHOD TO ESTIMATE SKEW FOR IMAGES ACQUIRED VIA IMAGE SCANNERS

HP INC

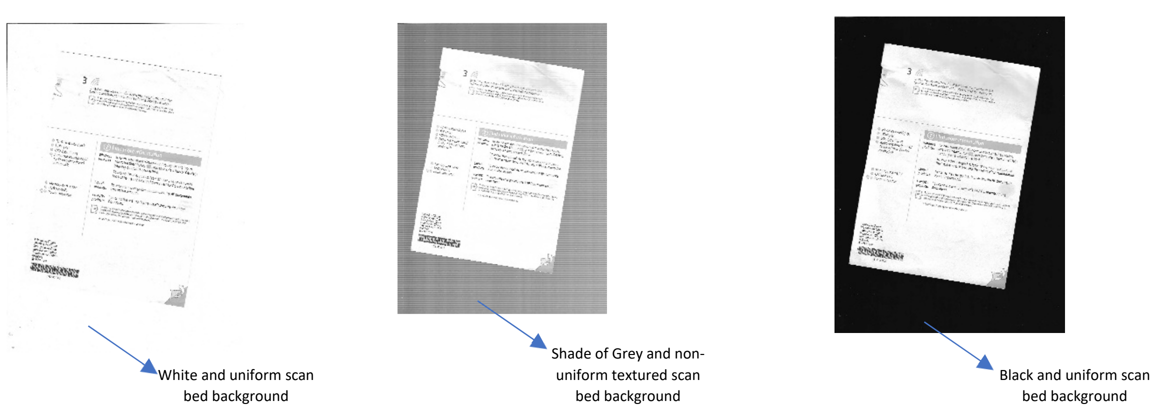
A method to estimate skew for images acquired via image scanners

Abstract

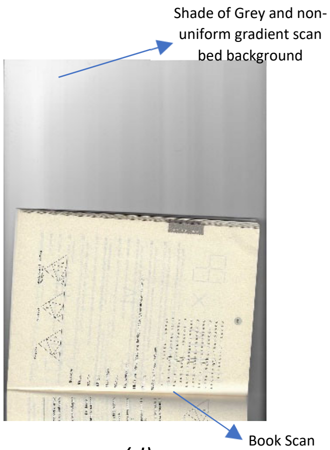
A document can be scanned via image scanners in two ways i.e., via flatbed or ADF. In flatbed, the user cannot place a document in perfect orientation, as a result, the image acquired is slightly skewed. Similarly, in the case of ADF, the skew in the image is introduced because of the mechanical nature of how it pulls the page from its input tray. Hence, there is a need for an algorithm to straighten the skew of the acquired image. This feature is very popular in scan solutions and almost always offered by default because of the frequent occurrence of the problem. We propose a new method of skew estimation and correction for images acquired via an image scanner. The method is not limited to just documents but works for all scan use cases i.e., documents, photos, book pages, cards, etc. It can work for use-cases with very little to no text if there is a representative pattern w.r.t orientation of content of an image. It is in-line in nature i.e., it only needs a band of a foreground image to estimate the skew of the image. The solution is scalable w.r.t resources of the user i.e. compute and memory as you can adjust its output precision of skew estimation based on resources at hand.

Literature Survey

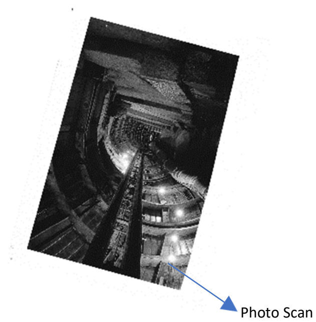
Image Skew estimation and correction is a well-studied problem. There are several ways to detect skew of the image, one of them is to rely on the 2D Fourier transform of the input image. For ex., algorithms cited in [1-4] rely on it to estimate the skew of the document. The algorithms cited in [1-2] rely predominantly upon the presence of a significant amount of text in scanned document images to estimate its skew, whereas the proposed solution can work for documents that mostly contain images or with images and only a few lines of text. The algorithms cited in [3-4] use Radon transform [6] to estimate the skew. The proposed solution differs from [4] in the following ways.



(a) *(b)* *(c)*



(d)



(e)

Fig 1

- As shown in the above fig, there are a wide variety of scan bed backgrounds i.e., white (Fig.1(a)), shades of grey (Fig.1(b)), black (Fig.1(c)), etc. These backgrounds may not be uniform like in Fig 1(a)(c), they can be noisy, non-uniform, textured and gradient like in Fig 1(b)(d).
  The proposed solution performs 2D Fourier transform on a variance filtered image [6] which enables it to work with images with the above-mentioned wide variety of scan bed backgrounds.
- Cartesian to Polar coordinate transformation is used to estimate skew.
  - This enables us to support the wide variety of scan use cases like documents (Fig 1(a)(b)(c)), Photo (Fig 1(e)), and Book pages (Fig 1(d)).
  - This enables a user to adjust the precision of skew estimation w.r.t compute or memory resources at hand.
- The proposed solution doesn't need a full image to estimate skew, it can work in-line with a small chunk of the input image with foreground to estimate skew. Hence, it can be computationally efficient.

Problem Description

The problem statement is to reliably estimate the skew of the document being scanned from an image scanner. Some problems faced are described in the prior art section. The following is a summary of the problems faced.

1. Support for a wide variety of scan bed backgrounds with different monochromatic colors.
2. Support for a wide variety of scan bed backgrounds with or without noise. (As shown in fig 1).
3. Support for a wide variety of scan use cases i.e., documents, photos, book pages, Id cards, etc.
4. Support adjustable precision of skew estimation.
5. Support for inline skew estimation with a small chunk of the foreground image.

Solution Description

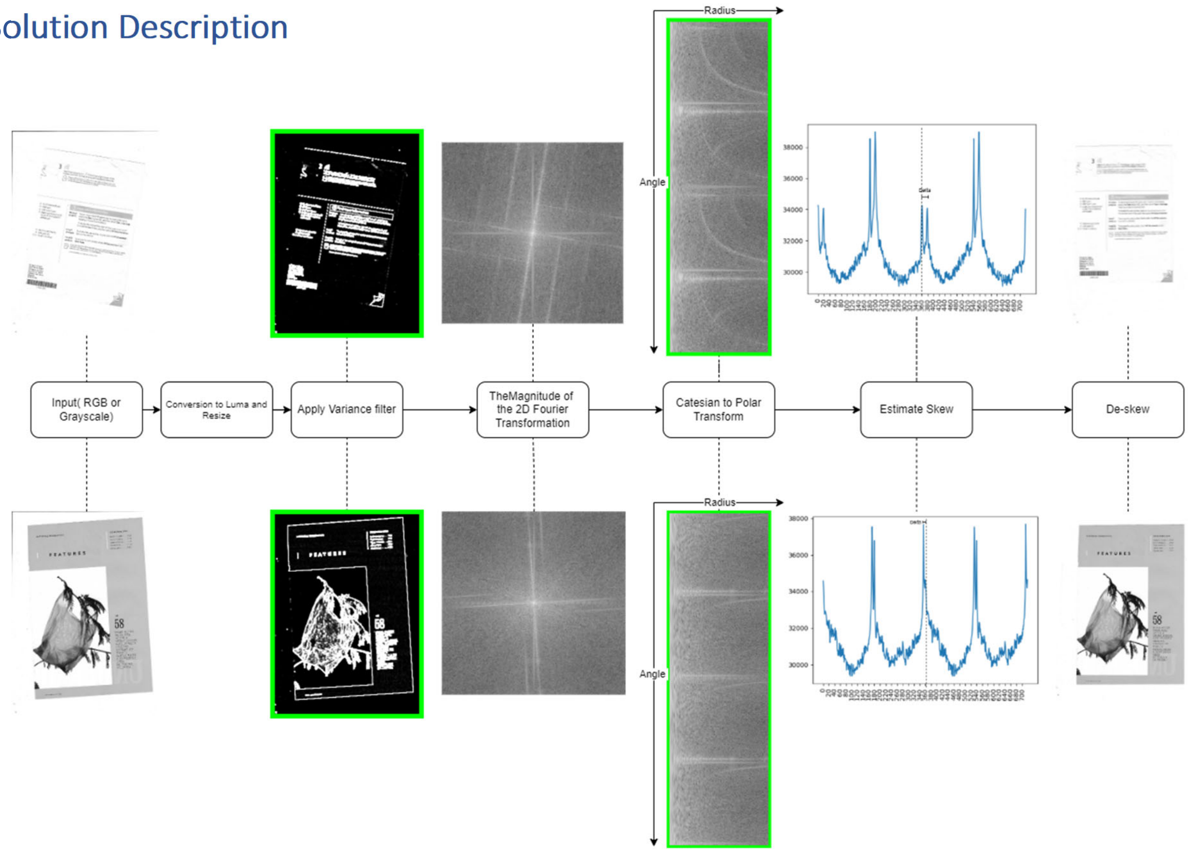


Fig 2

This pipeline supports an RGB or greyscale input image. **The components highlighted in green show unique aspects of the proposed solution pipeline.** The following are stages in the pipeline.

1. **Conversion to luma and resize:** if the input image is an RGB image then its luma or pixel intensity component is extracted. If the input image is a greyscale image, then no transformation is done. The luma or greyscale image thus obtained is downscaled by a factor of 0.25 along both its width and height. This step allows us to be efficient w.r.t computation and storage.
2. **Apply variance filter:** A variance filter (procedure described in [6]) is applied to the resized luma image to obtain an edge image shown in fig 2. Variance filter emphasizes the edges in the image at the same time being resilient to texture or gradient noise induced by scan bed background. This step takes care of problems 1 and 2 as described in the problem description.
3. **The magnitude of the 2D Fourier transforms:** Variance filtered image undergoes 2D Fourier transform [7] which transforms it from a spatial to a frequency domain, the magnitude of each component is computed and for sake of visualization, it is shifted such that DC-component is at the center. Fig2 shows two examples of the result of this

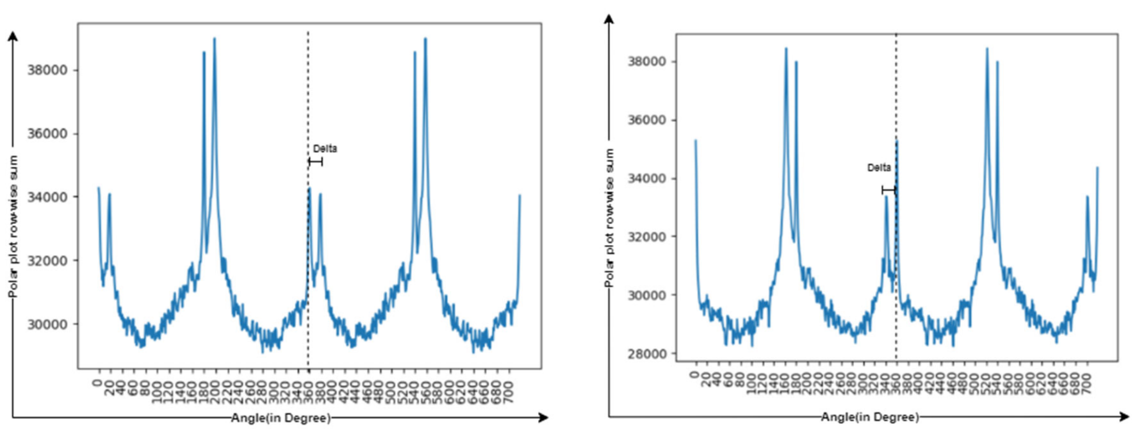
transformation. The intensity of each pixel in the transformed image is indicative of the magnitude of a respective frequency component.

4. **Cartesian to Polar transformation:** The magnitude image from step 3 can be interpreted as a matrix of shape (h, w). Where $h\ =\ height\ of\ the\ image, w =$ $width\ of\ the\ image$ undergoes non-linear transformation(refer [8]) to get an image which could be interpreted as a matrix of shape $(MaxAngle, MaxRadius)$.

$$\text{where,}\ \ \text{MaxRadius} = \left(\frac{\sqrt{h^2 + w^2}}{2}\right)$$
$$\text{MaxAngle} = 360 * \text{k}, \text{k} \in \mathbb{N}$$

the parameter k controls the precision of skew estimation outcome, for ex., in fig. 2, $k = 2$, this gives the algorithm a margin of error of ± 0.5 degrees, to reduce this margin of error k value must be increased accordingly. This would increase the size of the polar image. Hence, has an impact on computational and storage aspects of the algorithm and makes it flexible to user needs. Solves problems 3 and 4 as described in the problem description.

5. **Estimate Skew:**



(a) Plot for image with + (clockwise) skew *(b) Plot for image with – (anti-clockwise) skew*

Fig 3

Polar transformed image from the previous stage can be interpreted as a matrix denoted as $\boldsymbol{P}$, undergoes a linear transformation as follows,

$$\boldsymbol{p} = \mathbf{P1},\ \text{where}\ 1 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$Shape\ of\ \mathbf{1}\ =\ (MaxRadius, 1)$$

$$\text{Shape of}\ \boldsymbol{p} = (MaxAngle, 1)$$

$\boldsymbol{p}$, is a vector and can be interpreted as a row-wise sum of the matrix, $\boldsymbol{P}$. The length of vector $\boldsymbol{p}$, is $MaxAngle$. The following equations show how to compute, Content Skew Angle, which ranges between $[-45,45]$

$$\boldsymbol{p}' = (p_i)_{i \in \{(MaxAngle/4)-j,\ (MaxAngle/4)+j\}, i \neq (MaxAngle/4)\}}, where\ j = 45 \times k$$

$$\text{Delta} = \left(\text{argmax}(\mathbf{p}') - \frac{MaxAngle}{4}\right)$$

$$\text{Content Skew Angle} = \frac{Delta}{k}$$

6. **De-skew:** Content Skew Angle from the previous step can be used to compute De − skew Angle,

$$\text{De} - \text{Skew Angle} = -\text{Content Skew Angle}$$

The above equation suggests that if the content is oriented in a clockwise orientation, then it must be rotated anti-clockwise and vice-versa to de-skew it.

The whole pipeline can function in-line with just a chunk of a foreground image to estimate its skew.

Advantages

1. Supports in-line processing use case i.e., can work with just chunk of foreground portion of the image to estimate skew.
2. Supports not just documents but also photos, books, cards, etc.
3. Supports scans with different scan bed background colors with or without noisy i.e., textured or gradient.
4. Supports precision control of skew estimation, in use-cases with very limited resources w.r.t compute and memory can work at lower precision and in case of vast resources, it can operate in high precision. Hence, the solution is scalable.

References

1. https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm
2. http://www.fmwconcepts.com/imagemagick/textdeskew/index.php
3. https://patents.google.com/patent/US9591176B2
4. https://www.cin.ufpe.br/~cabm/visao/artigos/A%20new%20algorithm%20for%20skew%20detection%20and%20correction.pdf
5. https://en.wikipedia.org/wiki/Radon_transform#:~:text=In%20mathematics%2C%20the%20Radon%20transform,the%20function%20over%20that%20line.
6. https://www.researchgate.net/publication/252020532_Variance_filter_for_edge_detection_and_edge-based_image_segmentation
7. https://en.wikipedia.org/wiki/Fourier_transform
8. https://mathinsight.org/polar_coordinates_mapping

Disclosed by Rithvik Kumar T and Sumit Nath, HP Inc.