

# 블록 변환을 이용한 문서 영상의 기울어짐 교정

곽희규<sup>†</sup> · 김수형<sup>††</sup>

## 요 약

문서 영상의 기울기 교정은 각 화소 좌표들을 회전 변환함으로써 이루어진다. 본 연구에서는 인접한 검은 화소들의 사각형 영역인 블록 정보를 기반으로 임의의 각도  $\theta$  만큼 기울어진 문서를 빠르게 교정하는 방법을 제안한다. 블록 단위의 회전 변환을 이용하는 제안방법은 실수 연산이 현저히 줄어들기 때문에 화소 단위의 회전 변환을 이용하는 방법보다 기울기 교정에 따른 수행시간이 빨라진다. 제안 방법에서는 블록의 네 모서리 좌표만을 회전시킨 후, 블록의 경계선 내부에 위치하는 화소들을 찾아서 검은 화소로 결정한다. 블록 내부에 위치하는 화소들을 찾는 방법으로 두 가지를 제안하는데, 첫 번째는 임의의 행과 회전된 블록 경계선과의 교차점을 찾아서 두 교차점 사이의 검은 화소들을 찾는 방법이다. 두 번째는 경수 연산만을 사용하는 Bresenham의 선 그리기 알고리즘으로 네 경계선을 찾고 그 내부를 검은 화소로 채우는 방법이다.

제안 방법의 성능 평가를 위한 실험에서는 Pentium 200MHz PC에서 다양한 형태의 영문과 한글 문서 영상 2,016개를 사용하였다. 또한 화소 정보를 기반으로 하는 기존 방법과 제안 방법의 성능을 비교하여 제안 방법의 우수성을 입증하였다.

## Skew Correction for Document Images Using Block Transformation

Hee-Kue Kwag<sup>†</sup> · Soo-Hyung Kim<sup>††</sup>

## ABSTRACT

Skew correction for document images can be done using a rotational transformation of pixel coordinates. In this paper we propose a method which corrects the document skew, by an amount of  $\theta$  degrees, using block information, where the block is defined as a rectangular area containing adjacent black pixels. Processing speed of the proposed method is faster than that of the method using pixel transformation, since the number of floating-point operations can be reduced significantly. In the proposed method, we rotate only the four corner points of each block, and then identify the pixels inside the block. Two methods for inside pixel identification are proposed; the first method finds two points intersecting the boundary of the rotated block in each row, and determines the pixels between the two intersection points as the inside pixel. The second method finds boundary points based on Bresenham's line drawing algorithm, using fixed-point operation, and fills the region surrounded by these boundaries as black pixels.

We have measured the performance of the proposed method by experimenting it with 2,016 images of various English and Korean documents. We have also proven the superiority of our algorithm through performance comparison with respect to existing methods based on pixel transformation.

\* 본 연구는 한국과학재단 특정 기초 연구(98-0102-02-01-3)지원으로 수행되었음.

† 준 회원 : 전남대학교 대학원 전산학과

†† 정 회원 : 전남대학교 전산학과 교수 · 정보통신연구소  
논문접수 : 1999년 월 일, 심사완료 : 1999년 월 일

## 1. 서 론

문서 영상의 기울어짐 추정 및 교정은 문서 영상 처리와 이해의 성공 여부에 영향을 주기 때문에 중요한 문서 영상 처리 중의 하나이다. 문서 영상의 기울어짐은 디지털화 과정에서 흔히 발생하는 문제로, 스캐닝 과정에서 문서를 잘못 놓거나 자동 급지 장치(Automatic Document Feeder)의 다양한 속도 변이 때문에 보통 1%에서 3% 정도의 기울어짐이 발생하며 어떤 경우에는 10%에 이르는 경우도 있다. 영상의 기울어짐이 2~3%인 경우 문서 분석 및 문자 인식의 정확도가 떨어지며, 5% 이상이 되면 문서 영상 처리의 신뢰성이 급격히 떨어진다. 그러므로 문서 영역 분할 및 문자 부류 결정 이전에 기울어짐 정도를 추정하고 교정하는 과정이 필요하다. 본 논문에서는 임의의 각도  $\theta$  만큼 기울어진 문서를 효율적으로 교정(skew correction)하는 방법을 제안한다.

지난 수십 년간 문서 영상의 기울어짐 교정 문제를 해결하려는 많은 연구들이 이루어졌다. 이러한 연구들의 대부분은 영상의 화소(pixel) 정보를 기반[1-9]으로 하거나 연결요소(connected component)의 윤곽(contour) 정보를 기반[10]으로 한다. 화소를 기반으로 하는 방법들은 기울어진 각도에 대해 영상의 각 화소들을 회전시키는데, 회전하는 방향에 따라 직접 방법(direct method)과 간접 방법(indirect method)으로 나누어진다. 직접 방법은 기울어진 입력 영상의 검은 화소만을 회전시켜 출력 영상의 대응하는 화소를 얻는 방법으로 자리올림 문제(rounding problem)를 야기하는 단점을 가지고 있다. 이와는 반대로, 간접 방법은 출력 영상의 모든 화소의 값을 구하기 위해 입력 영상의 대응하는 화소를 탐색하는 방법으로 자리올림 문제는 발생하지 않지만 회전에 따른 시간 비용이 많이 소요되는 단점을 가지고 있다. 간접 방법에서 많은 시간 비용이 소요되는 단점을 개선하기 위해 회전 변환에 포함되는 실수 연산의 회수를 줄이고자 하는 시도[9]도 있었는데, 여전히 모든 화소를 고려한다는 것과 기울어진 각도가 큰 경우 추가적인 연산이 많아진다는 단점을 지니고 있다. 윤곽 정보를 기반으로 하는 방법은 연결요소들의 윤곽 정보만을 회전시키므로 영상의 기울어짐 교정을 빠르게 수행한다. 그러나 자리올림 문제에 의해 회전된 일련의 윤곽 정보를 인접한 화소 단위로 연결해야

하는 문제를 안고 있다. 기존 연구는 2장에 상세히 기술되어 있다.

본 논문에서 제안하는 방법은 블록(block) 정보를 사용한다. 블록은 인접한 검은 화소들로 이루어진 사각형 영역으로, 제안 방법은 회전 변환을 블록 단위로 수행한다. 이것은 실수 연산이 포함되는 회전 변환이 블록의 네 모서리 좌표에만 적용되므로 빠르고 효율적이다. 특히, 제안 방법에서는 자리올림 문제가 발생하지 않으며 간접 방법에 의한 회전 영상과 거의 유사한 영상을 얻고 처리시간은 매우 빠르다. 다양한 형태의 영문 및 한글 영상에 적용한 실험을 통해 이를 입증하였다. 제안 방법의 자세한 내용은 3장에서, 실험 결과 및 분석 등은 4장에서 각각 기술한다.

## 2. 관련 연구

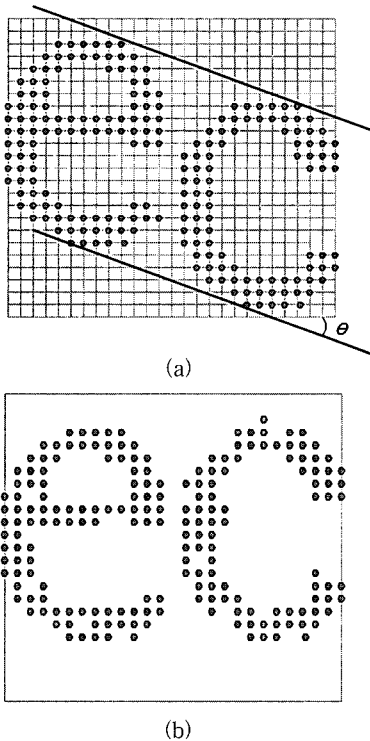
기울어진 문서를 교정하는 기존의 방법은 대부분 영상의 화소를 기반으로 하는 방법들이며, 연결요소의 윤곽을 기반으로 하는 방법도 있다. 화소를 기반으로 하는 방법은 직접 방법과 간접 방법으로 나누어진다.

### 2.1 직접 방법

직접 방법은 기울어진 문서 영상의 모든 검은 화소 좌표 값을 회전 행렬(rotational matrix)에 곱함으로써 출력 영상의 대응하는 화소의 좌표를 얻는다[6-8]. 기울어진 각도가  $\theta$ 일 때, 회전 변환은 다음과 같다.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

여기에서,  $(x, y)'$ 와  $(x', y)'$ 는 각각 기울어진 입력 영상과 출력 영상의 화소 좌표이고, 영상의 왼쪽 위를 회전의 중심인 원점으로 한다. 직접 방법의 주요한 단점은 이웃하는 연결요소들이 합쳐지거나 고립된 요소들이 분리되는 이산 평면(discrete plane)에서의 자리올림 문제(rounding problem)이다. 이로 인해 기울기 교정된 영상에서 객체(문자 등)들의 내부에 구멍(hole)들이 나타날 수 있다. (그림 1(a))는 기울어진 입력 영상을 보여 주고, (b)는 구멍들이 쉽게 발견되는 교정된 영상을 보여 준다. 이런 바람직하지 못한 변환은 문서의 영역 분할, 문자의 부류 결정과 같은 다음 단계의 문서 처리에 많은 문제를 야기한다.



(그림 1) 구멍이 발생하는 문제

2.2 간접 방법

간접 방법은 직접 방법의 반대 방향으로 수행된다. 출력 영상의 모든 화소  $(x', y')$ 에 대해 기울어진 입력 영상의 대응하는 화소  $(x, y)$ 를 찾아서 그 화소 값을  $(x', y')$ 의 화소 값으로 할당한다. 기울어진 각도가  $\theta$ 일 때, 회전 변환은 다음과 같다.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \end{pmatrix}$$

이 방법은 직접 방법의 단점인 자리올림 문제를 해결할 수 있다. 그러나 출력 영상의 모든 화소들에 회전 변환을 수행함으로써 실수 연산에 의한 시간 비용이 크다. 따라서 간접 방법은 직접 방법보다 좋은 영상을 생성하지만 보다 많은 처리 시간을 소비한다. 화소 정보를 기반으로 하는 대부분의 기울어짐 교정 방법들은 간접 방법을 채택하고 있다[1-5].

간접 방법의 회전 변환에서 각 화소에 대해 네 번의 곱셈과 두 번의 덧셈 실수 연산이 필요하다. 기본적으로

로 출력 영상의 화소 값은 왼쪽에서 오른쪽, 위에서 아래로 생성된다고 할 때, 출력 영상의 인접한 두 화소를 각각  $P_1(x_1, y_1)$ ,  $P_2(x_2, y_2)$ 라고 하면  $x_2 = x_1 + 1$ ,  $y_2 = y_1$ 과 같은 관계가 성립한다. 그리고 출력 영상에 대응하는 입력 영상의 인접 화소를 각각  $P_1(x_1, y_1)$ ,  $P_2(x_2, y_2)$ 라고 가정하면, 기울기 각도가  $\theta$ 인 회전 변환식에 의해 다음과 같은 관계를 유추해 낼 수 있다.

$$\begin{cases} x_2 = \cos \theta x_1 - \sin \theta y_1 \\ \leftarrow x_2 = x_1 + 1, y_2 = y_1 \\ = \cos \theta (x_1 + 1) - \sin \theta y_1 \\ = \cos \theta x_1 - \sin \theta y_1 + \cos \theta \\ \leftarrow x_1 = \cos \theta x_1 - \sin \theta y_1 \\ x_2 = x_1 + \cos \theta \end{cases}$$

$$x_2 = x_1 + \cos \theta, \quad y_2 = y_1 + \sin \theta$$

즉, 입력 영상에서  $P_2(x_2, y_2)$ 의 좌표는  $P_1(x_1, y_1)$ 와 두 오프셋(offset)  $(\cos \theta, \sin \theta)$ 에 의해 계산된다는 것이다. 이 변환식을 오프셋 가법(offset addition)이라고 하는데, 이러한 오프셋 가법을 이용할 경우 실수 연산 횟수가 간접 방법에 대해 반으로 줄어든다.

한편, 오프셋 가법 방법에서 회전 각도가 작은 경우 이산 평면의 성질 때문에 오프셋 값  $\sin \theta$ 가 거의 변하지 않는다. 따라서  $y$  좌표 값 계산에서 중복된 실수 연산이 존재하는데, Jiang 등[9]은 중복 연산을 제거함으로써 회전 연산 속도를 증가시켰다. 이것은 원점을 기준으로 하는 전형적인 회전 변환을 위치 변환(translation)과 국부적 회전(local rotation)변환으로 분해함으로써 가능하다. 국부적 회전 과정은 임의의 국부 점(local point)을 기준으로 연속된 화소들을 회전하는데, 회전에 따른 실수 연산을 정수 연산으로 대체하기 위해 오프셋 값들을 미리 계산하여 테이블에 저장한다. 국부 점에 대해 회전되는 화소를  $m$ 개라고 하면, 이 화소들의 좌표는 테이블에 저장된 오프셋을 국부 점에 더함으로써 얻어진다. 따라서 국부 점만이 회전 변환을 하게 되어 전체적으로 실수 연산의 횟수가  $1/m$ 로 줄어든다. 국부 점  $(x_1, y_1)$ 으로부터  $m$  화소만큼 떨어져 있는 국부 점  $(x_2, y_2)$ 의 계산식은 다음과 같다.

$$x_2 = x_1 + m \cdot \cos \theta, \quad y_2 = y_1 + m \cdot \sin \theta$$

이 방법에서는  $m$  값으로 16을 사용한다. 그러나 이 방법은 영상의 모든 화소에 대해 회전 변환을 고려한다는 점과 회전 각도가 큰 경우 오프셋 값  $\sin \theta$ 의 변화량이 커서 추가적인 연산이 필요하다는 단점을 가지고 있다. [9]에서 Jiang 등은 간접 방법의 성능을  $10^\circ$  이하에서는 16배,  $10^\circ$  이상에서는 12배정도 개선하였다.

### 2.3 윤곽 정보 지향적 방법

윤곽 정보를 기반으로 하는 방법에서는 회전 변환이 연결요소의 윤곽 정보에 적용된다. 모든 연결 요소는 색상, 시작 지점, 일련의 윤곽-열(contour-string)로 정의된다. 윤곽-열은 각각의 연결 요소들의 경계 정보를 기술한다. 실질적인 회전 과정은 다음과 같다.

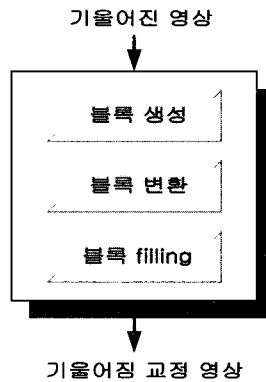
- 회전된 영상에서 시작 지점을 계산
- 윤곽-열에서 모든 윤곽 정보를 회전
- 최종적으로 새로운 시작 지점과 윤곽-열을 재구성

OCR 시스템에서는 비트맵 정보를 필요로 하기 때문에 연결 요소들의 윤곽 정보를 사용한다. 이 방법은 관련된 정보만을 회전시킨다는 점에서 효율적이다. 그러나 회전시 자리올림 문제에 의해 윤곽-열 정보가 화소 단위로 완전히 연결된 폐공간을 형성하지 못할 수 있기 때문에 항상 연결 여부를 검사해야 한다는 단점을 안고 있다.

### 3. 제안 방법

제안된 기울기 교정 방법은 화소 정보를 기반으로 하는 직접 및 간접 방법의 장점만을 취한 효율적인 방법으로, 인접한 검은 화소들의 사각형 영역인 블록 정보를 기반으로 한다. 회전 변환을 블록 단위로 수행하기 때문에 화소 단위로 수행하는 방법보다 실수 연산이 현저히 줄어들게 된다. 제안 방법에서는 블록의 네 모서리 좌표만을 회전시킨 후, 블록의 경계선에 둘러싸인 블록 내부에 위치하는 화소들을 검은 화소로 결정하면 된다. 블록 내부에 위치하는 화소들을 찾는 방법으로 두 가지를 제안한다. 첫 번째는 각각의 행에 대해 블록 경계선과의 두 교차점을 찾아서 두 점 사이에 위치하는 화소를 찾는 방법인데, 비교적 빠르고 교정 후의 영상이 간접 방법에 의한 것과 거의 유사하

다. 두 번째는 정수 연산만을 사용하는 Bresenham의 선 그리기 알고리즘[12]에 의해 회전된 블록의 네 경계선을 찾고, 경계선 내부를 검은 화소로 채우는 방법인데, 첫 번째 방법에 비해 영상의 질(quality)은 떨어지지만 수행속도가 더욱 빠르다. (그림 2)는 제안 알고리즘의 구성도이다.



(그림 2) 제안 알고리즘의 구성도

#### 3.1 블록 생성

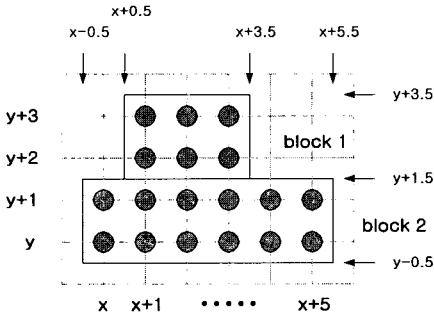
인접한 검은 화소들의 사각형 영역으로 정의되는 블록은 one-pass 알고리즘을 이용하여 문서 영상의 스캔 과정과 동시에 생성 가능하다[11]. 네 모서리 좌표에서 왼쪽-하부(lower-left) 모서리 좌표  $(x_{min}, y_{min})$ 와 오른쪽-상부(right-upper) 모서리 좌표  $(x_{max}, y_{max})$ 는 다음과 같다.

$$\begin{aligned} (x_{min}, y_{min}) &= (L - 0.5, B - 0.5), \\ (x_{max}, y_{max}) &= (R + 0.5, T + 0.5), \end{aligned}$$

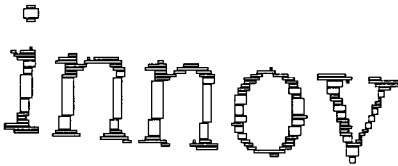
$$\text{where, } \begin{cases} L : \text{leftmost pixel} \\ B : \text{bottommost pixel} \\ R : \text{rightmost pixel} \\ T : \text{topmost pixel} \end{cases}$$

(그림 3)에서는 인접한 두 블록과 모서리 좌표들을 보여주고, (그림 4)에서는 블록들로 표현된 텍스트를 보여준다.

(그림 3)에서 block1의  $y_{min}$  와 block2의  $y_{max}$  는 동일한 좌표 값  $y + 1.5$ 를 갖는다. 따라서  $y_{min}$  과  $y_{max}$  는 회전 변환 후에도 같은 좌표 값을 갖기 때문에 자리올림에 의한 두 블록의 분리 문제는 발생하지 않는다.



(그림 3) 인접한 두 블록과 모서리 좌표들



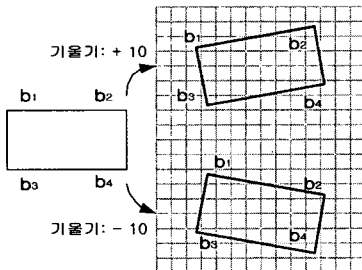
(그림 4) 블록으로 표현된 텍스트

### 3.2 블록 변환

블록 변환 과정에서는 블록의 네 모서리 좌표를 회전 변환하여 출력 영상의 대응하는 좌표를 얻는다. 이것은 기울어진 각도가  $\theta$  일 때, 다음과 같이 회전 변환 행렬을 블록의 각 모서리 좌표에 곱함으로써 이루어진다.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

여기에서  $(x, y)^t$  와  $(x', y')^t$  는 각각 기울어진 입력 영상과 출력 영상의 좌표이다. (그림 5)는 블록을 각각  $+10^\circ$ ,  $-10^\circ$  회전시켰을 때의 형태이며,  $b_1, b_2, b_3, b_4$ 는 블록의 네 모서리 좌표들이다.



(그림 5) 블록의 회전( $+10^\circ$ ,  $-10^\circ$ )

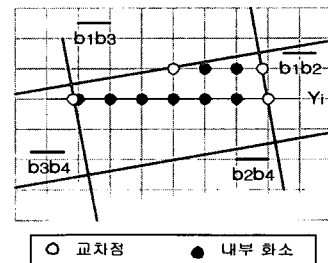
### 3.3 블록 filling

블록은 연속된 검은 화소들의 집합이기 때문에, 블록 filling 과정에서는 회전된 블록의 내부에 해당하는 화소들을 검은 화소로 결정한다. 그러나 회전된 블록이 두 좌표축에 평행하지 않기 때문에 블록 내부에 존재하는 화소들을 찾는 것은 간단하지 않다(그림 5). 이것은 회전하기 전 블록의 내부 화소들을 각각 회전 변환하여 결정할 수도 있지만, 제안 알고리즘은 회전된 블록의 경계를 먼저 찾고, 각각의 행에 대해 대응하는 경계선 내부에 위치하는 화소들을 추가적인 회전 연산 없이 찾을 수 있다. 회전된 블록을 채우는 방법으로 다음의 두 가지를 제안한다. 첫 번째는 임의의 행에 대해 블록 경계선과의 두 교차점을 찾는 방법이고, 두 번째는 정수 연산만을 사용하는 Bresenham의 선 그리기 알고리즘[12]에 의해 경계선을 찾은 후 내부를 채우는 방법이다.

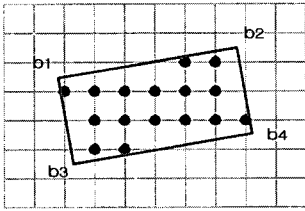
#### 3.3.1 블록 경계선과의 교차점 계산에 의한 방법

첫 번째 방법은 영상 내의 임의의 행과 회전된 블록의 경계선과의 교차점을 찾아 블록 내부에 위치하는 화소들을 찾는 방법인데, 단순하지만 임의의 행에 대해 네 직선과의 교차점을 구하는 연산이 계산상의 부담으로 존재한다. 그러나 두 교차점 사이에 존재하는 화소를 찾는다는 다른 연산이 필요치 않기 때문에 화소를 기반으로 하는 방법보다 빠르다. 특히, 블록이 크고 회전 각도가 작은 경우 연산 횟수가 줄기 때문에 더욱 효과적이다. (그림 6(a))는 회전된 블록의 네 경계선과 각 행의 교차점에 대해 블록이 채워지는 과정을 보여주고, (b)는 채워진 블록을 보여준다.

임의의 행은 블록의 네 경계선과 교차하는데, (그림 6(a))에서는 행  $Y_1$ 에 대해  $\overline{b_1b_2}$ ,  $\overline{b_1b_3}$ ,  $\overline{b_2b_4}$ ,  $\overline{b_3b_4}$ 의 순서로 교차한다.  $Y_1$  행과 교차하는 네 경계선 중



(a)



(b)

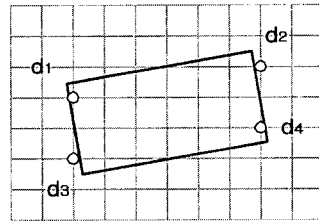
(그림 6) 회전된 블록 채우기(제안 방법 1)

가운데 두 개 실질적으로 블록의 경계와 만나므로, 두 경계선  $\overline{b_1b_3}$ 와  $\overline{b_2b_4}$ 에서 교차점이 나온다. 따라서 두 교차점 사이에 위치하는 화소들이 회전된 블록 내부의 검은 화소로 결정된다.

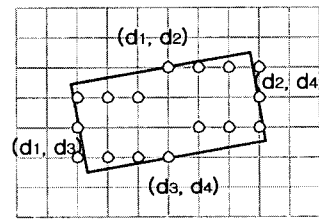
### 3.3.2 Bresenham 알고리즘을 이용한 방법

이 방법은 네 경계선 상에 위치하는 화소들을 먼저 찾고, 행별로 경계선 내부에 존재하는 화소들을 채우는 방법이다. 이 방법은 블록 경계선 상에 위치하는 화소들을 정수 연산만을 사용하는 Bresenham의 선 그리기는 알고리즘을 사용하여 찾기 때문에 전체적인 수행시간이 더욱 빨라진다. 회전된 블록의 네 모서리 좌표를 연결하는 경계선을 각각  $\overline{b_1b_2}$ ,  $\overline{b_1b_3}$ ,  $\overline{b_2b_4}$ ,  $\overline{b_3b_4}$  이라고 하자(그림 6). 먼저, 회전 변환이 적용된 네 모서리 좌표에 가장 가까운 화소를 찾는다. (그림 7(a))에서 보듯이 이것들은 각각  $d_1$ ,  $d_2$ ,  $d_3$ ,  $d_4$  이고, 이 화소들을 쌍으로 Bresenham 알고리즘에 적용하여 네 경계선에 해당하는 화소들을 찾는다. (그림 7(b))의 원(circle)들이 찾아진 경계선 상의 화소들이다.

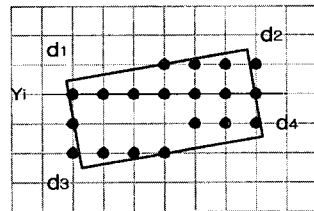
블록의 경계선 상에 위치한 화소들을 찾은 다음, 행별로 이동하면서 경계 화소들로 둘러싸인 블록 내부에 위치하는 화소들을 결정한다. 임의의 행에 대해 항상 블록의 두 경계선이 교차하므로 블록의 경계 채우기 과정에서 y축 좌표 각각에 대한 경계선 상의 x축 좌표를 이용한다. 즉, 임의의 행과 교차하는 두 경계선 상의 x축 좌표가 시작점과 끝점이 된다. (그림 7(c))는 임의의 행  $Y_1$ 에 대해 내부 화소들이 결정되는 과정을 보여주고, (그림 7(d))는 채워진 블록을 보여준다. 첫 번째 방법에 의한 결과(그림 6(b))와 두 번째 방법에 의한 결과(그림 7(d))를 비교해 보면, 후자가 더 많은 화소들을 블록 내부의 화소로 결정함을 알 수 있다.



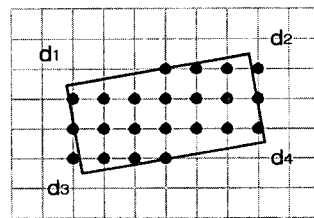
(a)



(b)



(c)



(d)

(그림 7) 회전된 블록 채우기(제안 방법 2)

제안된 두 방법의 차이는 경계선 상에 위치하는 화소를 찾는데 실수 연산을 사용하느냐, 정수 연산만을 사용하느냐 하는 것이다. 실수 연산을 사용하면 시간 비용은 증가하지만 정확한 영상을 얻을 수 있고, 정수 연산을 사용하면 시간 비용은 훨씬 감소하는 대신 자리올림 문제 등에 의한 오차를 유발할 수 있다.

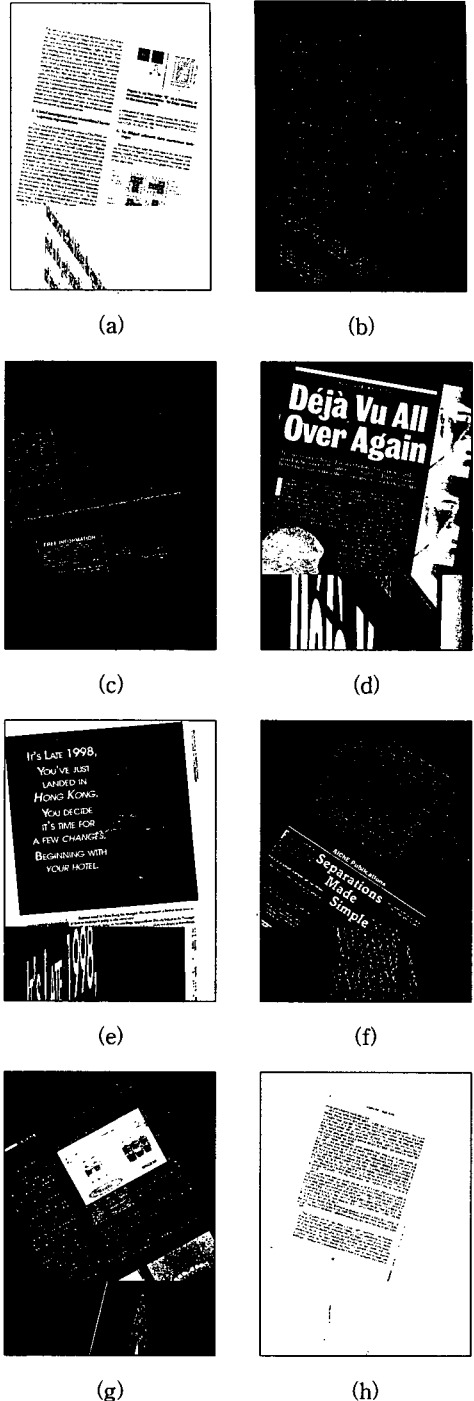
#### 4. 실험 결과 및 분석

실험에 사용된 문서 영상들은 A4 크기의 한글 및 영문으로 구성되는 저널(journal), 잡지, 논문 등을 300dpi 해상도로 스캐닝하였다. 이러한 문서 영상들은 단순 텍스트 문서, 다양한 폰트, 그림, 사진 등을 포함하는 문서, 다양한 구조(layout)를 가진 문서 영상, 전표(form) 영상 등으로 분류될 수 있다. 제안 방법의 실험을 위해 4가지 종류의 테스트 문서 집합을 사용하였는데, 기울어짐이 없는 영상을 8개 방향(-0.5°, -5°, -10°, -29°, 5°, 10°, 27°, 43°) 또는 5개 방향(-0.5°, -5°, -10°, 5°, 10°)으로 회전시켰다. 제안 방법의 실험에 사용되는 한글 및 영문 영상은 다음과 같이 총 2,016개의 영상으로 구성된다.

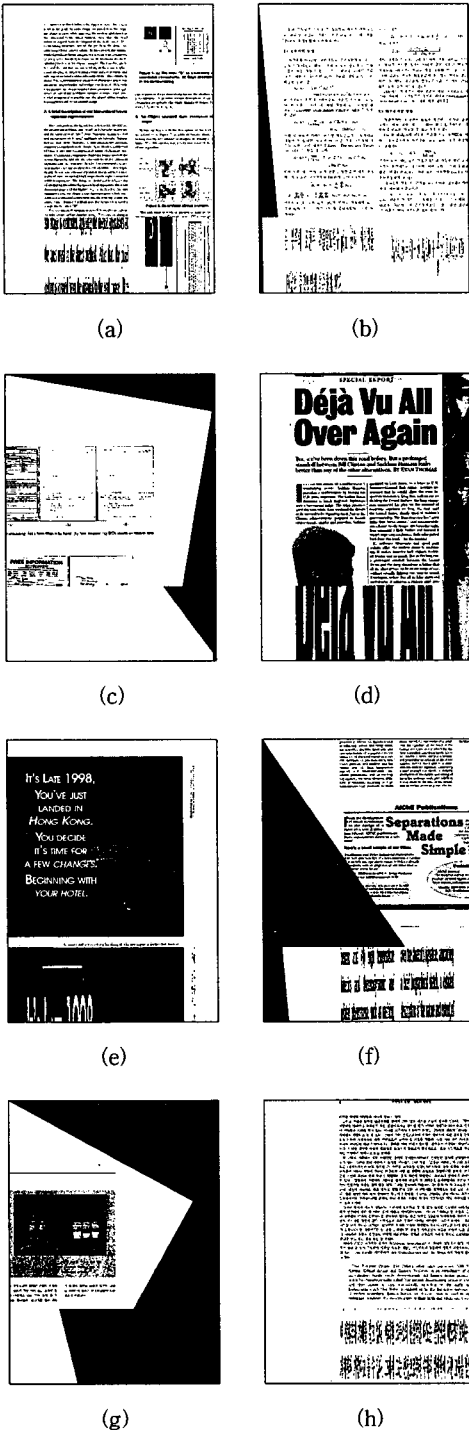
- 영문 잡지 및 저널 영상 : 100개의 원본+100×8 방향 회전 영상 = 900개
- 한글 잡지 영상 : 100개의 원본+100×8방향 회전 영상 = 900개
- 한글 논문 영상 : 18개의 원본+18×5방향 회전 영상 = 108개
- 영문 논문 영상 : 18개의 원본+18×5방향 회전 영상 = 108개

제안된 알고리즘의 우수성을 입증하기 위해 화소 정보를 기반으로 하는 직접 방법 및 간접 방법을 제안 방법과 동일한 환경 하에서 구현하여 정량적인 비교를 하였다. 제안 방법 및 기존 알고리즘들을 구현하는 환경은 Pentium 200MHz PC이며, 수행시간은 임의의 영상에 대해 5번 수행하여 가장 빈도수가 높은 수행 시간으로 결정하였다. 총 2,016개의 영상에 대해 기울기 교정 방법들을 적용한 결과, 화소 정보를 기반으로 하는 직접 및 간접 방법은 각각 4.1초, 31.9초를 기록하였고, 제안 방법 1과 제안 방법 2는 각각 1.8초, 0.7초를 기록하여 제안 방법이 매우 빠르게 수행됨을 알 수 있었다.

(그림 8)은 실험 데이터의 예를 보여주고 (그림 9)는 기울기 교정된 영상을 보여준다. 각각의 영상은 (a) 영문 논문, (b) 한글 논문, (c) 표, (d) 영문 잡지, (e) 광고지, (f) 저널, (g) 한글 잡지, (h) 한·영 혼용 문서로 분류된다.

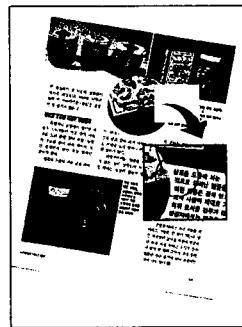


(그림 8) 입력 문서 영상의 예

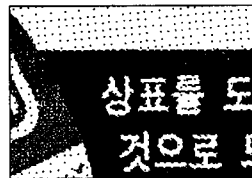


(그림 9) 기울기 교정된 문서 영상의 예

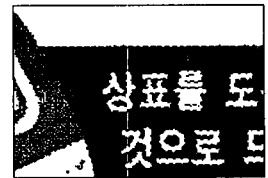
(그림 10)은  $+10^\circ$  기울기를 가지는 한글 텍스트에, 화소를 기반으로 하는 방법 및 제안 방법을 적용한 결과를 보여준다. (그림 10(a))는 기울어진 입력 영상이고 (그림 10(b))는 직접 방법을 사용하여 교정한 영상의 일부분을 보여주는데, 단점인 자리올림 문제에 의해 부적절한 구멍이 발생함을 쉽게 발견할 수 있다. (그림 10(c))는 간접 방법을 사용하여 얻은 영상이고 (그림 10(d))는 제안 방법 1에 의해 얻어진 영상이다. (그림 10)의 (c)와 (d)는 교정된 영상이 거의 유사하며 자리올림 문제에 의해 구멍이 발생하는 일은 없다. (그림 10)(e)는 제안 방법 2를 사용하여 얻어진 영상인데, 전체적으로 검은 화소의 밀도가 높아지고 분리된 연결요소가 합쳐지는 결과를 낳았다. 이것은 Bresenham의 선 그리는 알고리즘이 정수 연산만을 사용하여 수행시간은 빠르지만 자리올림 문제가 발생함을 보여준다.



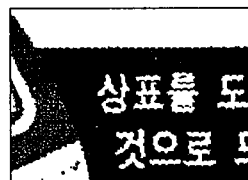
(a)



(b)



(c)



(d)



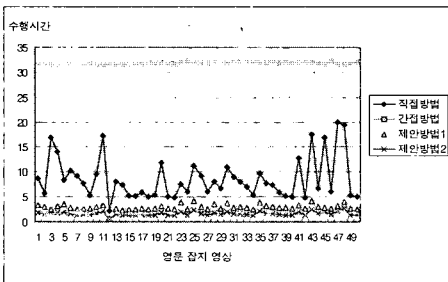
(e)

(그림 10) 기울기 교정 방법들의 결과

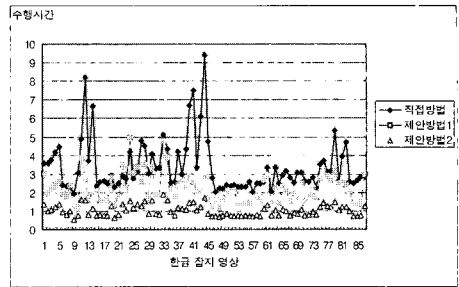


(그림 11)은 기울기 교정 방법들의 수행시간에 대한 분석 결과를 보여 준다. (그림 11(a))는 +10° 기울어진 영문 잡지 영상 50개에 적용한 결과이다. 제안 방법 2가 가장 빠르고, 제안 방법 1, 직접 방법, 간접 방법의 순서로 나열되어, 제안된 방법들이 화소 정보를 사용하는 직접 및 간접 방법보다 수행시간이 빠름을 알 수 있다. 직접 방법은 입력 영상의 검은 화소에 대해 회전 변환을 수행하기 때문에 화소의 밀도에 민감하게 반응하여 밀도가 높은 경우 교정 시간이 많이 소요되었다. 간접 방법의 경우에는 출력 영상의 모든 화소를 고려하기 때문에 많은 시간 비용이 요구되었다. 평균 수행시간은 제안 방법 2와 제안 방법 1은 각각 1.6초, 2.9초이고, 직접 방법이 8.5초, 간접 방법이 31.8초이다. (그림 11(b))는 한글 잡지 영상 87개에 적용한 결과로, 제안 방법 2는 1.0초, 제안 방법 1은 2.2초, 직접 방법이 3.5초 걸렸다. 분석 결과에는 나타나지 않았지만 간접 방법은 31.8초를 기록했다. 제안 방법 1의 경우에는 크기가 작은 블록이 많이 생기면 교차점 계산의 부담이 크기 때문에 수행시간이 직접 방법에 비해 많이 걸릴 때도 있다. (그림 11(c))는 +10°, -10°, +27°, -29°의 기울기를 가진 영문 및 한글 논문 영상에 적용한 결과인데, 기울기별 18개의 영상을 사용하였다. 평균 수행시간은 제안 방법 2와 제안 방법 1이 각각 0.3초, 0.5초이고, 직접 방법은 1.8초, 간접 방법은 31.8초를 기록하여 기울어짐 정도에 관계없이 제안 방법의 성능이 빠름을 알 수 있다.

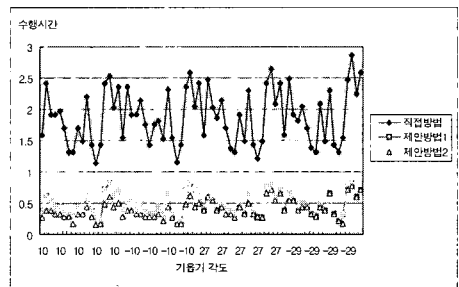
모든 실험 결과를 비교해 볼 때, 수행시간 면에서 제안된 방법들의 성능이 화소 정보를 기반으로 하는 방법들보다 효율적이다. 그러나 정확도 면에서 볼 때, 간접 방법과 제안 방법 1에 의한 영상이 자리올림 문



(a)



(b)



(c)

(그림 11) 기울기 교정 방법들의 성능 비교

제가 발생하지 않고 정확하다. 따라서 빠르고 정확한 제안 방법 1의 성능이 가장 우수함을 알 수 있다.

### 5. 결 론

본 논문에서는 블록 변환을 이용한 영문 및 한글 문서 영상들에 대한 빠른 기울어짐 교정 알고리즘을 제안하였다. 제안 방법은 화소 정보를 기반으로 하는 방법보다 빠르고, 특히 정확성이 높은 간접방법과 거의 유사한 양질의 영상을 생성한다. 제안 방법은 인접한 검은 화소들의 사각형 영역인 블록을 생성하고 블록의 네 모서리 좌표에만 회전 변환을 수행하며, 회전된 블록의 내부에 위치하는 화소를 찾아서 검은 화소로 채우는 과정으로 구성된다. 이처럼 제안 방법은 실수 연산이 포함되는 회전 변환을 블록의 모서리 좌표에만 적용하기 때문에 수행시간이 매우 빨라진다. 제안 방법에서 블록 내부의 화소를 찾는 방법으로 두 가지를 제안하였는데, 각 행마다 블록 경계선과의 교차점을 구하여 두 교차점 사이에 존재하는 화소를 찾는 첫 번째 방법과 Bresenham의 알고리즘을 사용하여 경계선

상의 화소를 찾고 내부를 채우는 두 번째 방법이다. 두 번째 방법은 정수 연산만을 사용하기 때문에 수행 시간은 매우 빠르지만 자리올림 문제가 발생함을 실험을 통해 알 수 있었다. 따라서 빠르고 정확한 영상을 생성하는 제안된 첫 번째 방법이 더 우수하다. 총 2,016개의 영상에 대해 기울기 교정 방법들을 적용한 결과, 제안한 첫 번째 및 두 번째 방법은 각각 1.8초, 0.7초를 기록하였고, 화소 정보를 기반으로 하는 직접 및 간접 방법은 각각 4.1초, 31.9초를 기록하였다.

**참 고 문 헌**

[1] A. Amin, S. Fischer, A.F. Parkinson and R. Shiu, "Comparative study of skew detection algorithms," *Journal of Electronic Imaging*, Vol.5, No.4, pp.443-451, 1996.

[2] Avanindra and S. Chaudhuri, "Robust detection of skew in document images," *IEEE Trans. Image Processing*, Vol.6, No.2, pp.344-349, 1997.

[3] H. S. Baird, "The skew angle of printed documents," *Proc. SPSE 40th Conf. and Symp. Hybrid Imaging Systems*, pp.21-24, May. 1987.

[4] S. C. Hinds, J. L. Fisher and D. P. D'Amato, "A document skew detection method using run-length encoding and the Hough transform," *Proc. 10th Int. Conf. on Pattern Recognition*, pp.464-468, 1990.

[5] D. X. Le, G. Thoma and H. Weschler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition*, Vol.27, No.10, pp.1325-1344, Oct. 1994.

[6] C. L. Yu, Y. Y. Tang and C. Y. Suen, "Document skew detection based on the fractal and least squares method," *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, pp.1149-1152, Aug. 1995.

[7] B. Gatos, N. Papamarkos and C. Chamzas, "Skew detection and text line position determination in digitized documents," *Pattern Recognition*, Vol.30, No.9, pp.1505-1519, 1997.

[8] C. Sun and D. Si, "Skew and slant correction for

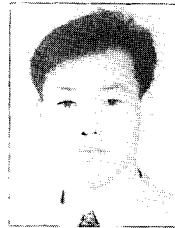
document images using gradient direction," *Proc. 4th Int. Conf. on Document Analysis and Recognition*, pp.142-146, 1997.

[9] H.F. Jiang, C.C. Han and K.C. Fan, "A fast approach to the detection and correction of skew documents," *Pattern Recognition Letters*, Vol.18, pp.675-686, 1997.

[10] M. Ali, "An object/segment oriented skew-correction technique for document images," *Proc. 4th Int. Conf. on Document Analysis and Recognition*, pp.671-674, 1997.

[11] B. Yu and A. K. Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern Recognition*, Vol.29, No.10, pp.1599-1629, 1996.

[12] D. Hearn and M.P. Baker, 'Computer graphics', 2nd Ed., Prentice Hall, 1994.



**곽 희 규**

e-mail : hkkwag@chonnam.chonnam.ac.kr  
 1996년 전남대학교 전산학과 졸업 (학사)  
 1998년 전남대학교 전산통계학과 대학원 졸업(이학석사)  
 1998년~현재 전남대학교 전산학과 박사과정

관심분야 : 패턴인식, 컴퓨터비전.



**김 수 형**

e-mail : shkim@chonnam.chonnam.ac.kr  
 1986년 서울대학교 컴퓨터공학과 졸업(학사)  
 1988년 한국과학기술원 전산학과 졸업(공학석사)  
 1993년 한국과학기술원 전산학과 졸업(공학박사)

1990년~1996년 삼성전자 멀티미디어연구소 선임연구원  
 1997년~현재 전남대학교 전산학과 조교수  
 관심분야 : 패턴인식, 컴퓨터비전, 신경망학습