

The International Arab Journal of Information
Technology, Vol. 7, No. 3, July 2010

Improved ccv method

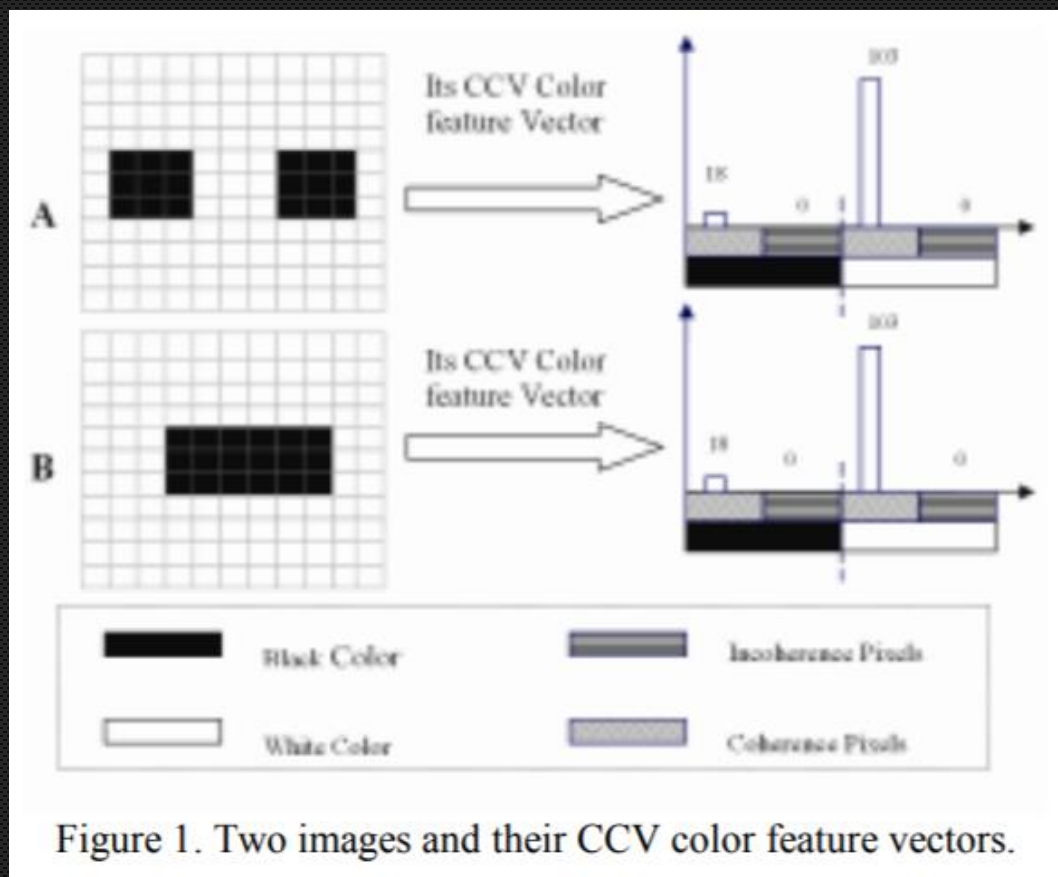
<Improving the Effectiveness of the
Color Coherence Vector>

이준희 정리

Alaa Al-Hamami¹ and Hisham Al-Rashdan² ¹Computer
Science Department, Amman Arab University for Graduate
Studies, Jordan ²Computer Science, Yarmouk University,
Jordan



필요성 1)



The two images A and B have the same CCV color feature vector as shown in Figure 1, because the amount of the coherence pixels in image A is the same as the amount of coherence pixels in image B. It doesn't tell us about the existence of dissimilarity between image A and image B in Figure 1.

[Solution_1] -> compare the number of coherence regions of the same color between the two images



필요성 2)

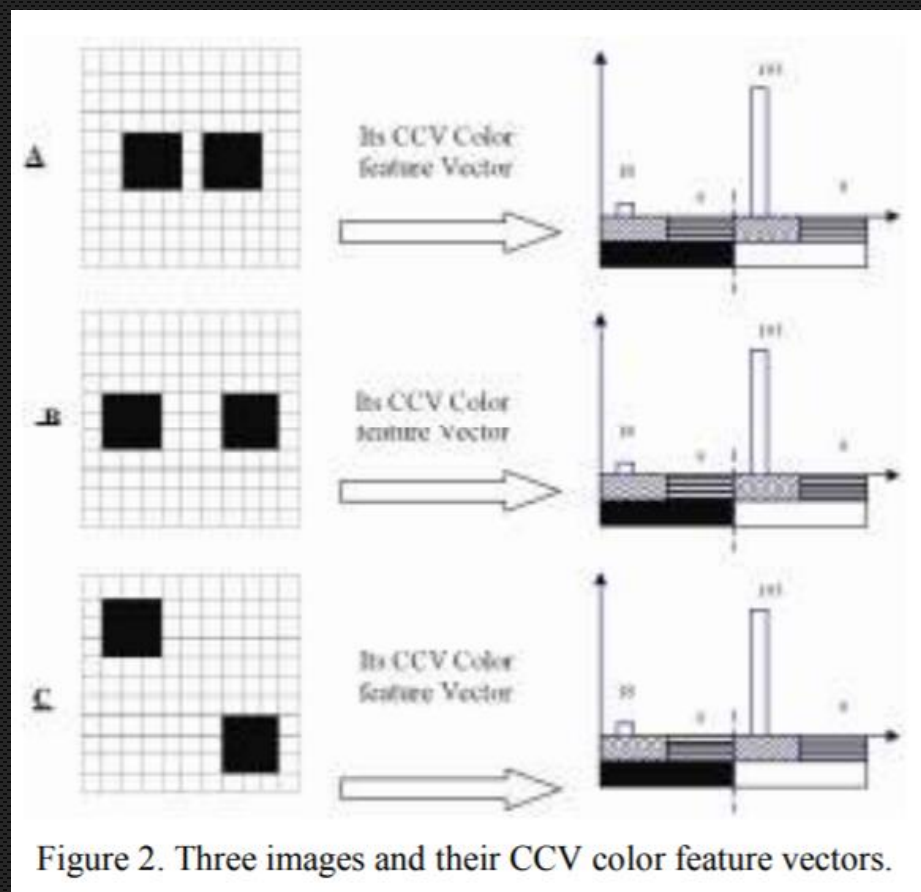


Figure 2. Three images and their CCV color feature vectors.

All of the three images in Figure 2 have the same CCV color feature vector and also the same number of coherence regions (each image has two black coherence regions and one white coherence region) although they differ in appearance.

[Solution_2] -> compare the location of coherence regions of the same color between the two images



접근방법 정리

1.〉 "Number" of Chunks

2.〉 "Location" of Chunks

2. (1)
distance

2. (2)
angle

2.}

"Location" of Chunks

2. (1)
distance

2. (2)
angle

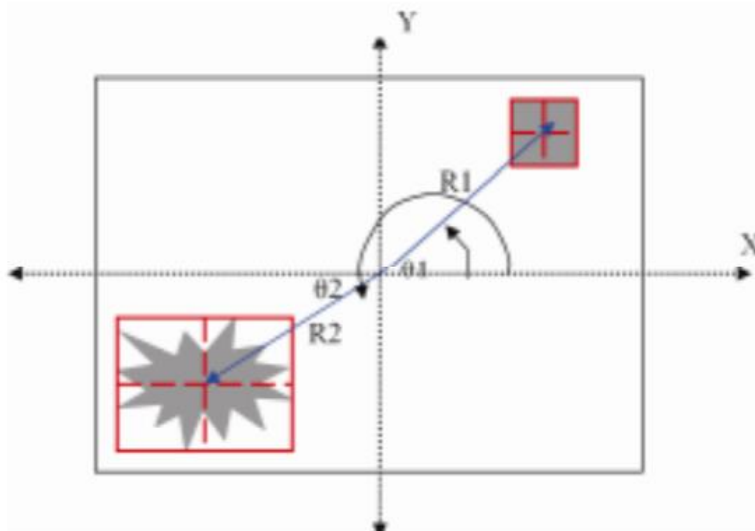


Figure 3. The process of determining the location of two gray coherence regions.

[1] Bounding box method + [2] Polar Coordinates

-> ① 해당 bounding box의 중앙점 좌표를 찾아 전체 이미지의 정 중앙 좌표와의 2.(1) distance를 구한다.

-> ② 각 bounding box등의 중앙점 좌표 기준으로 x축에서의 회전각을 구해 θ_1 와 $\theta_2 - \theta_1$ 로 각 chunk의 2.(2) angle 데이터를 구한다.

2.}

"Location" of Chunks

2. (1)
distance

2. (2)
angle

원점과 chunk 중앙점 사이의 거리
(= absolute value)

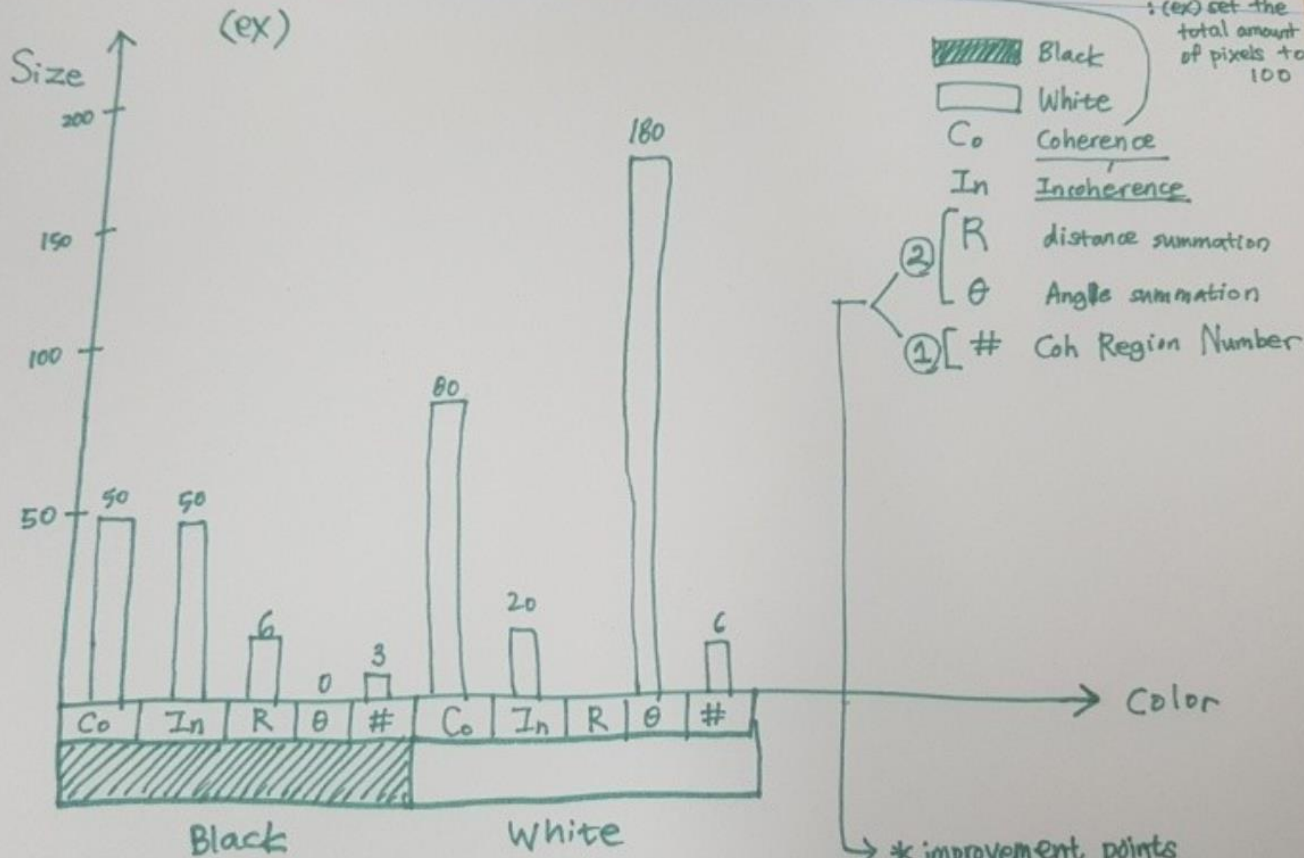
Table 1. The coherence region locations in Cartesian and polar coordinates.

Cartesian Coordinates	Equations 1 and 2	Polar Coordinates
(3, 0)	$R = \sqrt{x^2 + y^2}$	$3 < 0$
(-3, 0)		$3 < 180$
(0, 0)	$\theta = \tan^{-1} \left(\frac{y}{x} \right)$	$0 < 0$

x축에서의 회전각 (= argument)

(대략적인 예시 히스토그램)

SEOUL NATIONAL UNIVERSITY
1 GWANAK-RO, GWANAK-GU, SEOUL, 08826, KOREA



참고:

distance, angle 정보는
각 chunk 종류별로
summation
구해준 수치임

→ sum 말고 다른 값을
사용할 수도 있을 것 같
음. (하지만 굳이...?)

* improvement points
for previous CCV methods

{ ① Comparing the "number" of
② Comparing the "location" }

of the coherence regions of the
same color between two images

Normalization



이렇게 정규화 시킨 데이터들을 가지고 두 이미지 사이의 유사도를 check! (by Euclidean Distance)

$$\delta = \frac{X - D_{\min}}{D_{\max} - D_{\min}}$$

$$\delta = \frac{X}{D_{\max}} \quad (X \geq 0)$$

: 이렇게 해서 0~1사이의 값으로 만들어준다

* Normalization

1. **size**
 $\rightarrow \frac{(\text{size of } Co)}{(\text{size of image})}, \frac{(\text{size of } In)}{(\text{size of image})}$
 ← 내가 전에 제안했던 방향이 맞!!
 : 이렇게하면 서로 이미지 크기가 달라도 비교가능~~!!
2. **"number" of chunks** (같은 색깔 내게서 chunk의 개수)
 $\rightarrow \frac{\text{색깔 A의 chunk의 개수}}{(\text{max \# of coh regions})}$
3. **"distance" of chunks**
 $\rightarrow \frac{\text{색깔 A의 chunk들의 distance 합}}{(\text{max distance val for all possible coherence regions})}$
 img size threshold val
 (max \# of coh regions)
 X (원점으로부터 max distance val)
4. **"angle" of chunks**
 $\rightarrow \frac{\text{색깔 A의 chunk들의 angle 합}}{(\text{max angle val for all possible coherence regions})}$
 (max \# of coh regions)
 X (max angle val)
 359°

대략적인 모델 수립 후 결정해야 할 사항

- (1) 1. The Coherence size threshold.
- (2) 2. The weight factors of the coherence-incoherence pixels size , coherence regions number, coherence regions distance and coherence regions angle.

- 그 후 각 이미지 feature간 거리는 Euclidean Distance로 구함



두 이미지가 동일한 인물을 담고 있다고 판단하기
위한 유사도 판단 threshold도 따로 설정해 주어야 함
(3)

그 외 현재 우리가 결정해야 할 것



1. Improved method 성능 실험 할 조합들

- 사용 가능한 선택지들 :

- [1] 각 coherence region들 center coordinate을 CCV에 추가하는 방법
(from 석주오빠가 찾은 논문)
- [2] 각 color chunk들의 location + number값을 추가하는 방법
(내가 이 PPT에 정리한 내용 / location과 number 둘 중 하나만 취할 수도 있음.
혹은 location에서도 angle과 distance 둘 중 하나만 사용할 수 있음. 조합은 만들기 나름일 듯 함.)
- [3] parallel 처리 방법 (속도를 높이기 위함)
http://www2.ic.uff.br/iwssip2010/Proceedings/nav/papers/paper_172.pdf
-> 속도가 크게 문제되지 않는다면 굳이 애까지 해야할까 싶긴 함...
// 위 [1], [2]는 정확도를 높이기 위한 방법들. [3]은 속도 향상을 위한 방법

이렇게 세 개로 실험 대상을 나누자는 게 아니라, 여기서 언급된 여러 방법들을 어떻게 섞어서 어떤 조합들로 나눌 것인지 결정!

2. 1에서 결정한 조합들의 성능 실험에 있어서 기준으로 삼을 지표들

- PPT 전반에 걸쳐 다른 논문에서는 정석적으로 Precision과 Recall을 사용해 성능을 측정함.
우리도 똑같이 할 것인가?? 혹은 소요시간 등을 평가항목에 추가? Or 또 다른 지표 활용??

- (참고) 논문에서 사용한 성능평가 지표

$$\text{Precision} = \frac{\text{the number of retrieved images that are relevant}}{\text{the number of retrieved images}}$$

$$\text{Recall} = \frac{\text{the number of retrieved images that are relevant}}{\text{the total number of retrieved images}}$$



다음으로 내가 할 일

논문에서 소개한 이론적인 방법들을 어떻게 파이썬 코드로 구현할 수 있을지 구글링해보면서 구체화시켜보기!!!