Edge Image Description Using Angular Radial Partitioning

A.Chalechale, A. Mertins and G. Naghdy

IEE Proc.-Vis. Image Signal Processing,
2004

Slides by David Anthony Torres
Computer Science and Engineering — University of California at San Diego

Angular Radial Partitioning

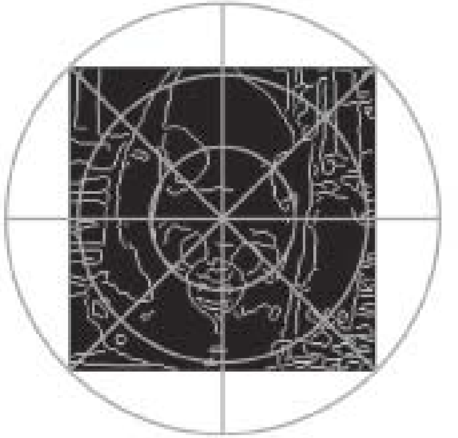
- Partition edge map into angular and radial bin.
- Partition into $M \times N$ bins
  - M radial parts
  - N angular parts
- Index a bin by (k,i)

$\rho = kR/M, \quad k = 1..M$

$\theta = 2\pi i/N, \quad i = 1..N$

Image: $I(\rho,\theta)$

Angular Radial Partitioning

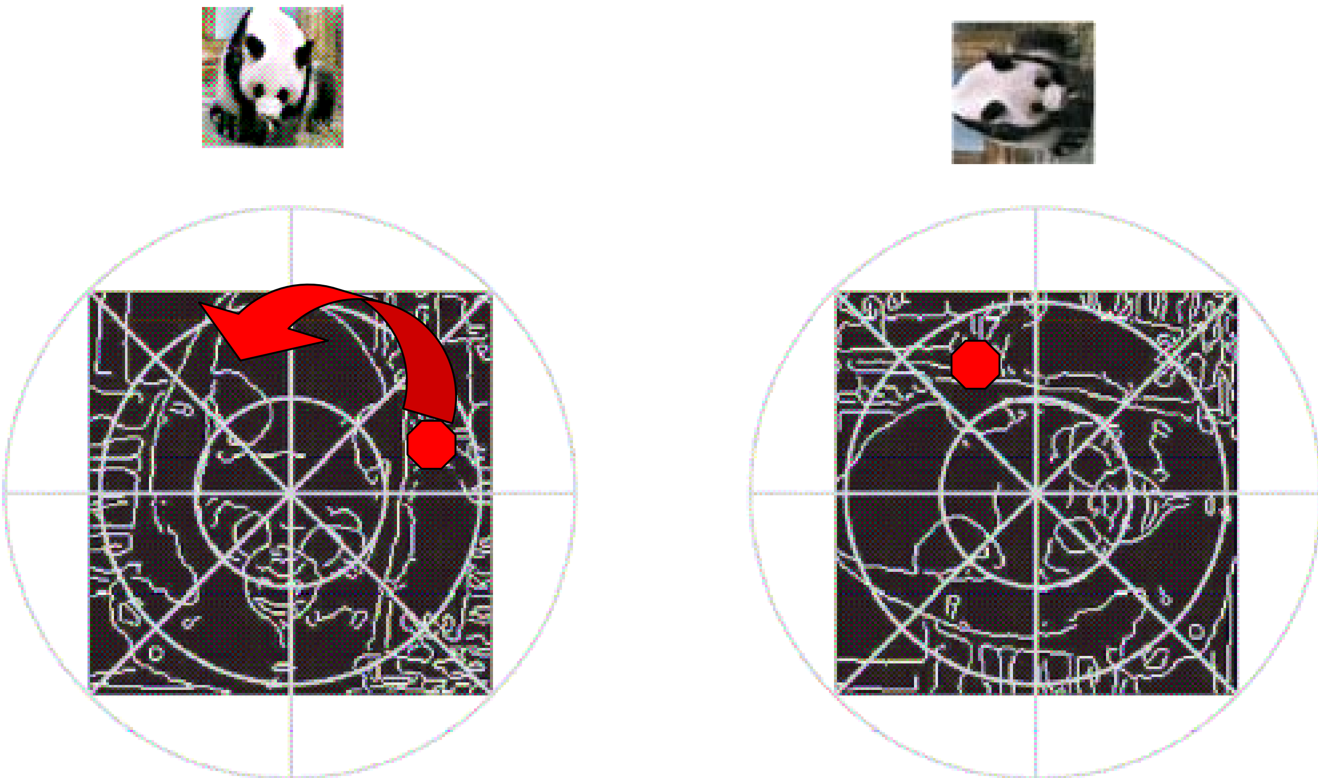


- Sector feature *f(k,i)* = # edge pixels in (k,i)

$$f(k,i) = \sum_{\rho=\frac{kR}{M}}^{\frac{(k+1)R}{M}} \sum_{\theta=\frac{i2\pi}{N}}^{\frac{(i+1)2\pi}{N}} I(\rho,\theta)$$

- A feature *f(k,i)* is shifted when we rotate by

$$\tau = 2\pi l / N \quad \text{for } l = 0, 1, 2, \ldots$$



Angular Radial Partitioning

- Denote image rotation by

$$I_\tau(\rho,\theta) = I(\rho,\theta\text{-}\tau)$$

- Feature rotates as well

$$f_\tau(k,i) = f(k,i\text{-}l)$$

Take 1-D Fourier Transform

- $$F(k,u) = \frac{1}{N}\sum_{i=0}^{N-1} f(k,i)e^{-j2\pi ui/N}$$

- $$F_\tau(k,u) = \frac{1}{N}\sum_{i=0}^{N-1} f_\tau(k,i)e^{-j2\pi ui/N}$$

Take 1-D Fourier Transform

$$F_\tau(k,u) = \frac{1}{N}\sum_{i=0}^{N-1} \underline{f_\tau(k,i)} e^{-j2\pi ui/N}$$

$$= \frac{1}{N}\sum_{i=0}^{N-1} f(k,i-l) e^{-j2\pi ui/N}$$

Take 1-D Fourier Transform

$$= \frac{1}{N} \sum_{i=0}^{N-1} f(k, i - l) e^{-j2\pi ui/N}$$

$$= \frac{1}{N} \sum_{i=-l}^{N-1-l} f(k, i) e^{-j2\pi u(i+l)/N}$$

Take 1-D Fourier Transform

$$= \frac{1}{N} \sum_{i=-l}^{N-1-l} f(k,i) e^{-j2\pi u(i+l)/N}$$

$$= e^{-j2\pi ul/N} F(k,u)$$

Rotational Invariance

- Transforms contain rotational invariance

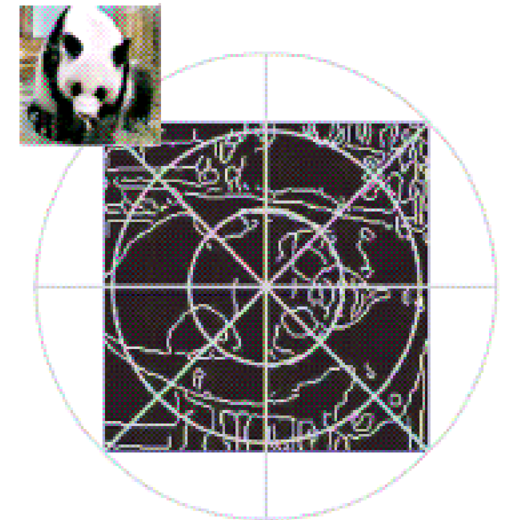
$$\left.F_\tau(k,u)\right| = e^{-j2\pi ul/N}F(k,u)$$

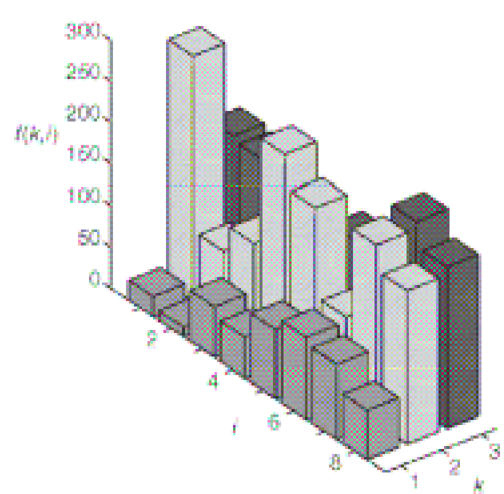
- Because their power-spectra are the same

$$\|F(k,u)\| = \|F_\tau(k,u)\|$$

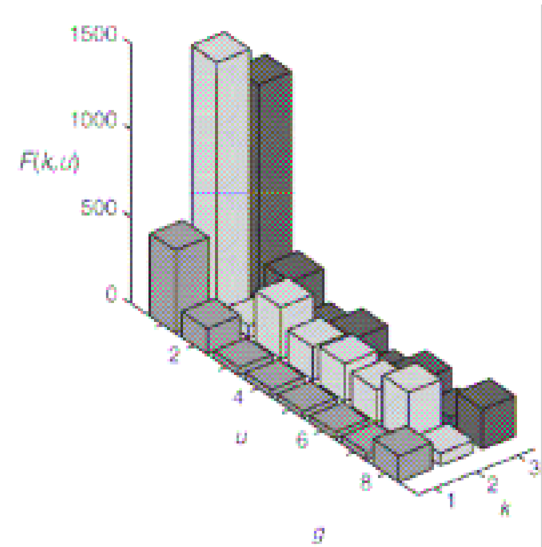
- Choose {‖F(k,u)‖} for k=0..M-1 and u=0..N-1 as image features.

	$f(k,i)$	$\|\|F(k,i)\|\|$

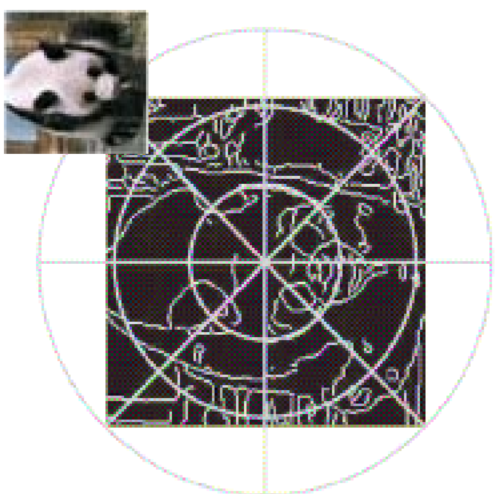


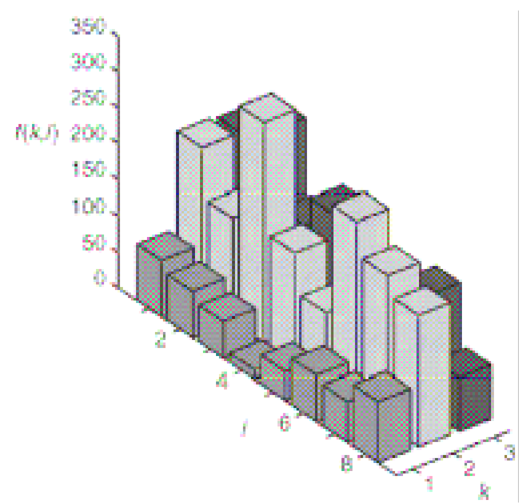


f(k,i)
0 50 100 150 200 250 300
2 4 6 8
i
1 2 3
k

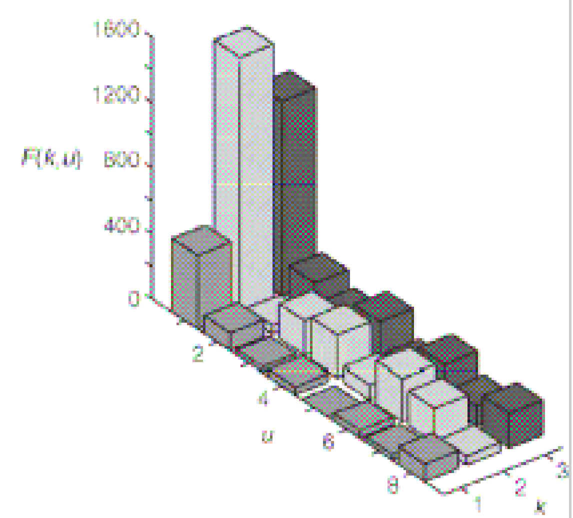


F(k,u)
0 500 1000 1500
2 4 6 8
u
1 2 3
k





f(k,i)
0 50 100 150 200 250 300 350
2 4 6 8
i
1 2 3
k



F(k,u)
0 400 800 1200 1600
2 4 6 8
u
1 2 3
k

Other approaches…

- **Moment Generating Functions**

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx\, dy$$

- ➢ Common image analysis technique
  - Treat the image as a distribution and generate moments.
  - Combine several different moments into a feature vector.

Other approaches…

- **Zernike Moments**
  - Less sensitive to noise.
  - More powerful in discriminating objects.
  - Used for shape descriptor in MPEG-7
  - Based on Zernike orthogonal polynomials

$$V_{nl}(x,y) = R_{nl}(r)e^{il\theta}$$

$$R_{nl}(r) = \sum_{s=0}^{(n-|l|)/2} (-1)^s \cdot \frac{(n-s)!}{s!((n+|l|)/2-s)!((n-|l|)2-s)!} r^{n-2s}$$

Other approaches…

- The Zernike Moment Invariant of an image *f*

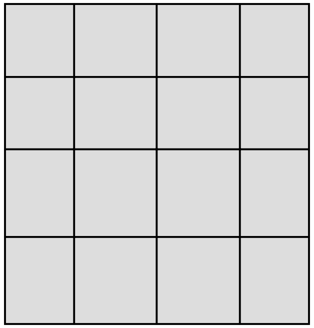
$$A_{nl} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^{\infty} [V_{nl}(r, \theta)]^* f(r \cos\theta, r \sin\theta) r \, dr \, d\theta$$

- Can approximate the image by

$$f(x, y) \approx \sum_{n=0}^{N} \sum_{\substack{l \\ n-|l| \text{ even}, |l| \le n}} A_{nl} V_{nl}(x, y)$$

- $\{A_{nl}\}$ are used for image matching.

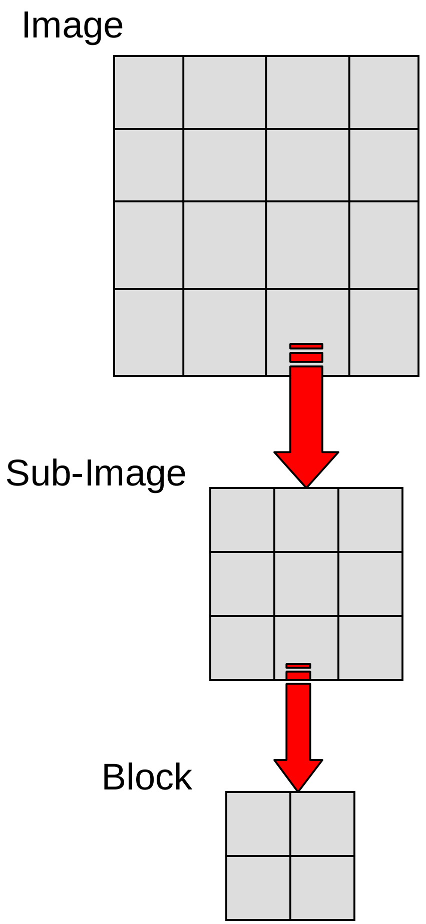
Edge Histogram Methods



- Partition image and build local edge histograms.
- **Histogram of Edge Directions**
  - Partition image into large local regions.
  - Partition each local region into small image patches.
  - Quantize each patch as horizontal, vertical, diagonal
    - Use filters for this.
  - Collect results into local region histograms.
  - Perform matching based on histograms.

Edge Histogram Methods

- **Application of Histogram Method: Edge Histogram Descriptor (EHD)**
  - Used in MPEG-7 to calculate frame similarity
  - Algorithm:
    - Divide the image into 16 sub-images
    - Divide each sub-image into blocks
    - Split block into 4 quadrants
    - Use 2x2 filter masks to bin each quadrant into vertical, horizontal, 45°diagonal, 135°diagonal, non-directional.
    - Collect a histogram of edge directions for each sub-image. (Gives you 16 x 5=80 bins)
    - Compare two histograms for similarity



Angular Radial Transformation

- Also used in MPEG-7 to retrieve/encode object information.
- Can describe complex objects (such as trademarks)
- ART is a transform defined on a unit disc.
- Consists of orthonormal sinusoidal basis functions .

Angular Radial Transformation

- From each image we extract ART coefficients: ψ_{mn}

$$\psi_{mn} = \int_0^{2\pi} \int_0^1 V_{mn}^*(\rho, \theta) f(\rho, \theta) \rho \, d\rho \, d\theta$$

- $f(\rho,\theta)$ is the image.
- $V_{nm}(\rho,\theta)$ is the basis function

Angular Radial Transformation

- The basis function: $V_{mn}(\rho, \theta) = A_n(\theta) R_m(\rho)$

- Consists of an angular component: $$A_n(\theta) = \frac{1}{2\pi} e^{jn\theta}$$

- And a radial component: $$R_m(\rho) = \begin{cases} 1 & m = 0 \\ 2\cos(\pi m \rho) & m \neq 0 \end{cases}$$

Angular Radial Transformation

- ART Algorithm
  - Normalize image to a set dimension
  - Perform edge detection
  - Calculate ψ_{mn} for m=0..M, n=0..N according to

$$\psi_{mn} = \int_0^{2\pi} \int_0^1 V_{mn}^*(\rho, \theta) f(\rho, \theta) \rho \, d\rho \, d\theta$$

  - Scale coefficients by $|\psi_{00}|$ to normalize.
  - Perform matching on the features ψ_{mn}.

Results

Performance Measure: ANMRR

- Use Average Normalized Modified Retrieval Rank (ANMRR)
- Incorporates recall, precision and rank information.
- Defines as:

$$ANMRR = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q)$$

  - Average NMRR score for all queries 1,2,…,Q

Performance Measure: ANMRR

- *NMRR* is the normalized *MRR* score

$$NMRR(q) = \frac{MRR(q)}{K + 0.5 - 0.5 * NG(q)}$$

- *NG(q)* is the number of ground truth images for query q.
- *K=min(4NG(q), 2GTM)* where *GTM* is *max{NG(q)}*

Performance Measure: ANMRR

- *MRR* is an adjusted average rank measure

$$MRR(q) = AVR(q) - 0.5 - \frac{\mathrm{NG}(q)}{2}$$

- And *AVR* is the average rank of images in a query

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank(k)}{NG(q)}$$

  - *NG(q)* is the number of ground truth images for a query *q*.

Performance Measure: ANMRR

- A numerical example:
  Suppose a query *q* has 10 similar images in the database (*NG=10*).
  If we find 6 of the top 20 retrievals (K=20) in the ranks
  1,5,8,13,14,18 then:

  AVR=14.3
  MRR=8.8
  NMRR=0.5677

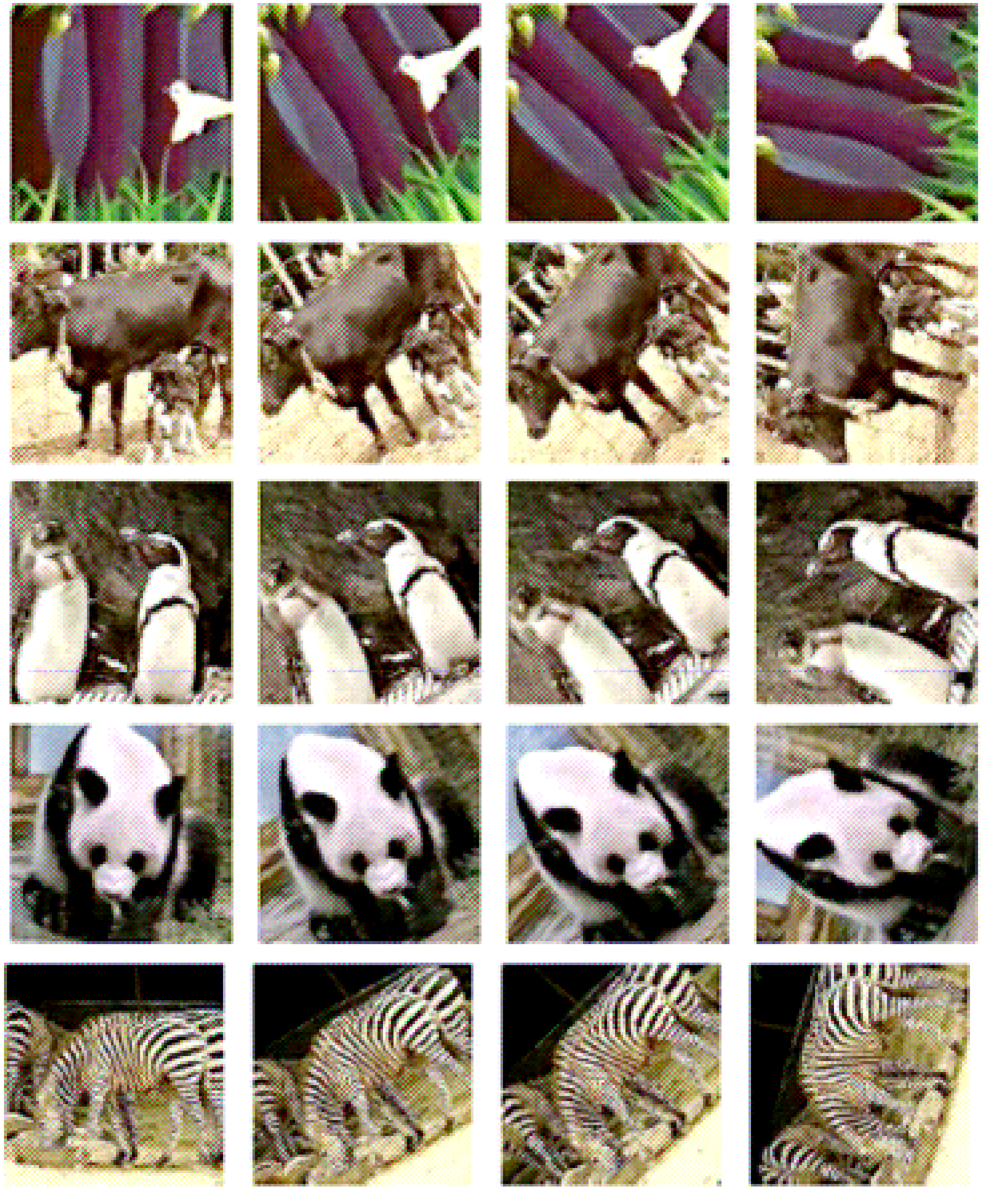
$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank(k)}{NG(q)}$$

$$MRR(q) = AVR(q) - 0.5 - \frac{NG(q)}{2}$$

$$NMRR(q) = \frac{MRR(q)}{K + 0.5 - 0.5 * NG(q)}$$

- NMRR and ANMRR are always in [0,1]
- The smaller the ANMRR the better

Test Inputs:



Varying Angular and Radial Partitions

Table 1: ANMRR of ARP method with three radial and varying angular partitions using main database (4320 images)

Normalised size	3 × 4 partitions	3 × 6 partitions	3 × 8 partitions	3 × 9 partitions	3 × 12 partitions	3 × 24 partitions	3 × 36 partitions	3 × 72 partitions
101 × 101	0.2921	0.2269	0.1538	0.1853	0.1922	0.1660	0.1661	0.1679
129 × 129	0.2625	0.1847	0.1090	0.1402	0.1441	0.0934	0.1010	0.0865
201 × 201	0.2072	0.1250	0.0552	0.0806	0.0832	0.0464	0.0451	0.0257
257 × 257	0.2216	0.1334	0.0694	0.0903	0.0931	0.0538	0.0518	0.0293

Table 2: ANMRR of ARP method with 12 angular and varying radial partitions using main database (4320 images)

Normalised size	5 × 12 partitions	7 × 12 partitions	10 × 12 partitions	15 × 12 partitions	18 × 12 partitions
101 × 101	0.1263	0.1107	0.1002	0.0952	0.0910
129 × 129	0.0974	0.0786	0.0716	0.0646	0.0566
201 × 201	0.0443	0.0317	0.0244	0.0217	0.0200
257 × 257	0.0512	0.0361	0.0269	0.0253	0.0223

Sensitivity to Rotation

Table 3: ANMRR of the ARP method using selected data sets (2160, 1440, 720 and 480 images)

Partitions	10° rotated	15° rotated	30° rotated	45° rotated
3 × 12	0.0764	0.0789	0.0617	0.0972
3 × 8	0.0508	0.0557	0.0579	0.0472

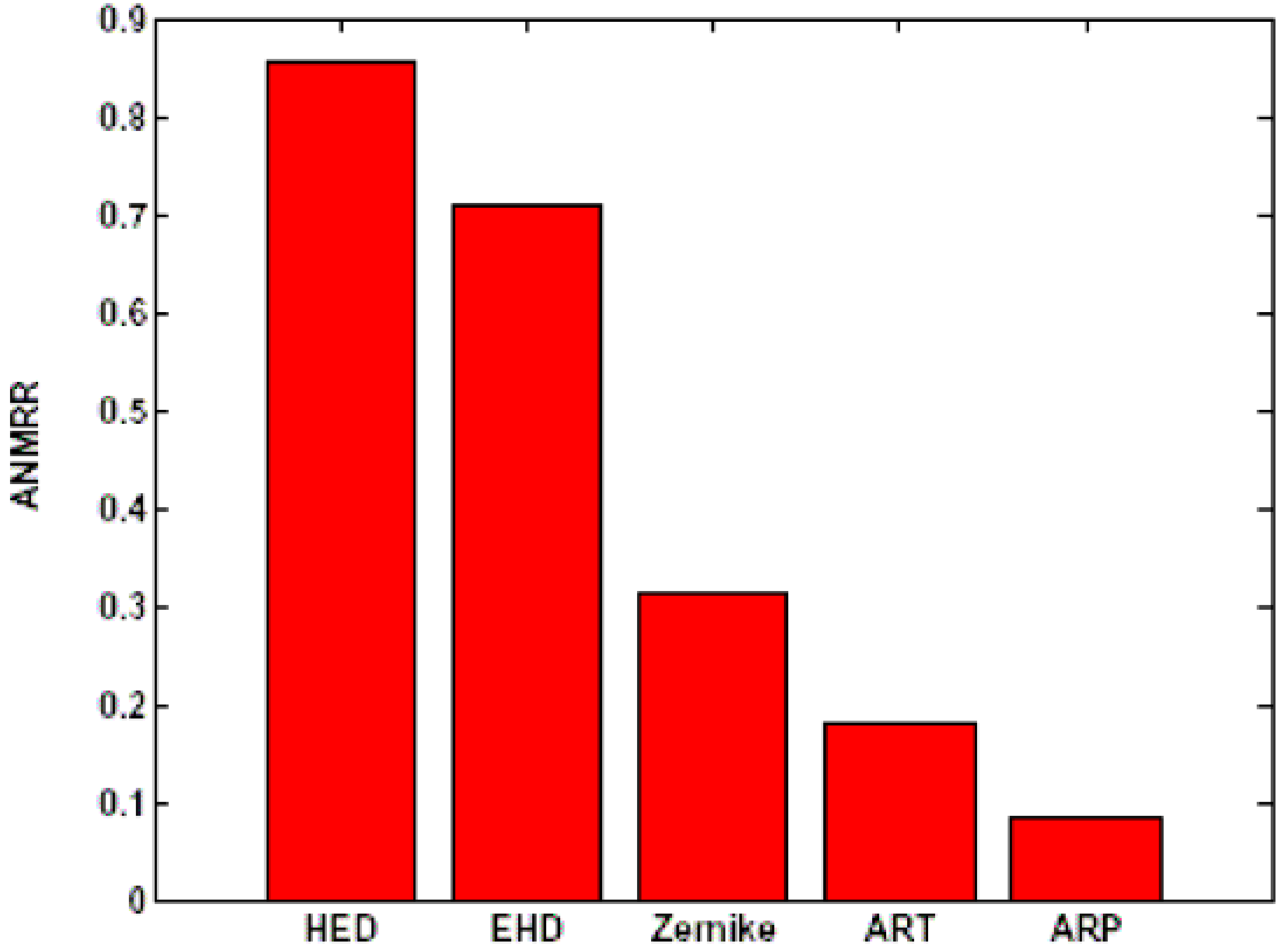


Figure 3: Retrieval results of different methods with ANMRR.

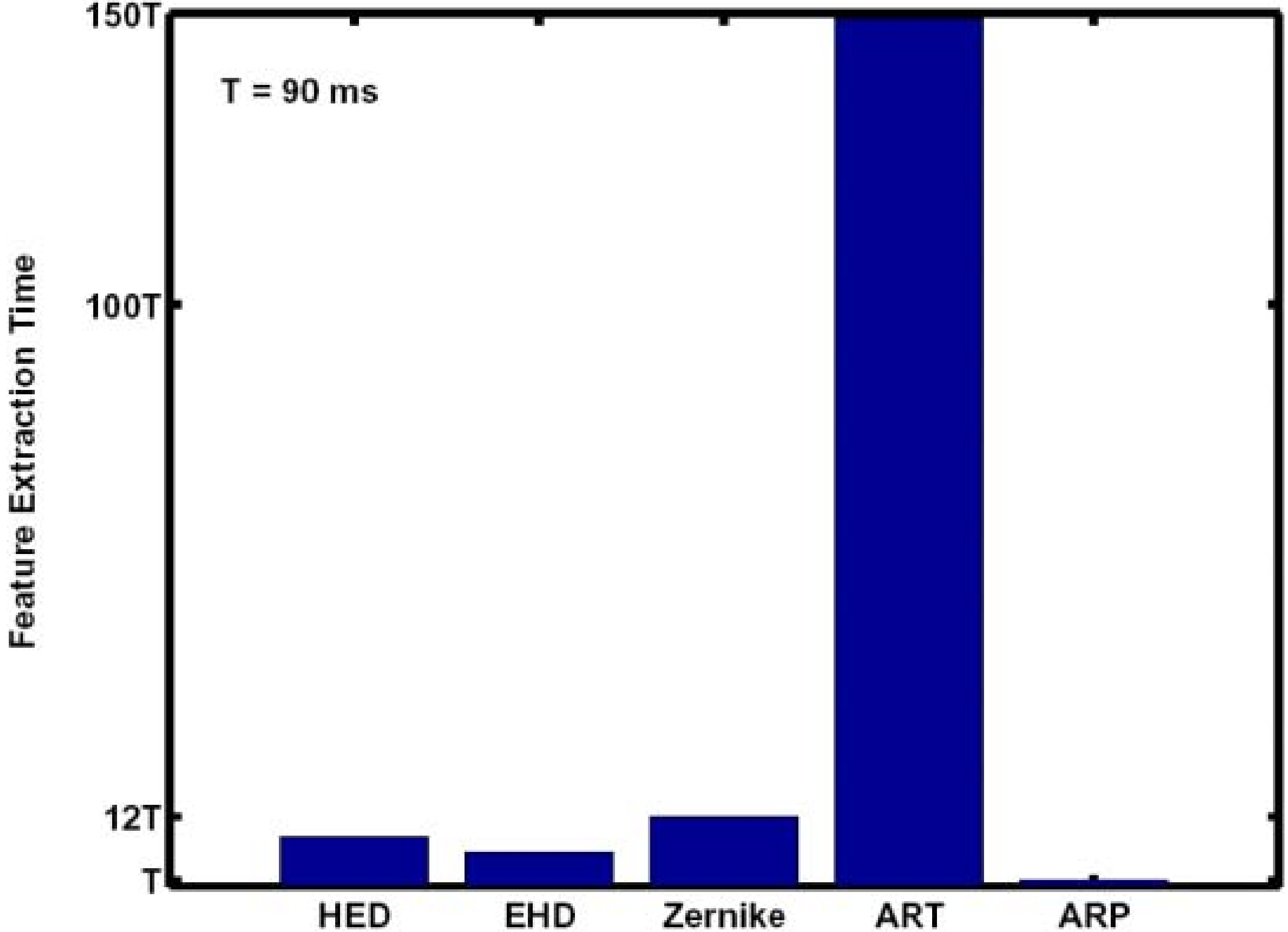


Figure 4: Comparison of feature extraction time.