Detecting Abnormal Gait

Christian Bauckhage   John K. Tsotsos   Frank E. Bunn

Centre for Vision Research

York University

Outline

Why Detecting Abnormal Gait?

- project in cooperation with industry (StressCam Ltd.)
- goal:
  - detecting *unusual, alarming* motion patterns
  - in real world indoor environments
- exemplary scenario:

Automatic Gait Analysis

- Observing human gait allows for:
  - person identification
  - activity recognition (walking, dancing, . . .)
- Usual approach to gait analysis:
  - silhouette based
    [Boyd(2004), Collins et al.(2002), Wang et al.(2002)]
- Main focus:
  - biometrics (identification at a distance)
- Our focus:
  - gait classification

How to Detect Abnormal Gait?

- Observations:
  - 2 class problem
  - shape dynamics are characteristic

  normal gait:

  abnormal gait:

- Approach:
  - shape based
  - using SVMs

Robust Shape Encoding (1)

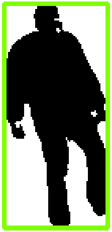
- Requirements for shape encoding:

  1. must capture characteristics of gait *across* individuals
  2. must be efficient
  3. must enable vector space embedding

Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:
  1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
  2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
  3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
  4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
  5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

  $\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$



Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:
  1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
  2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
  3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
  4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
  5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

  $\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$

Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:

1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

$\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$

Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:
  1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
  2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
  3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
  4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
  5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

  $\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$

Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:
  1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
  2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
  3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
  4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
  5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

  $\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$

Robust Shape Encoding (2)

- Assumption: shape $=$ set $\mathcal{S}$ of L pixels
- Encoding via bounding box splitting:
  1. compute bounding box $\mathcal{B}(\mathcal{S})$ of pixel set $\mathcal{S}$
  2. subdivide $\mathcal{B}(\mathcal{S})$ into n vertical slices
  3. for each resulting pixel set $\mathcal{S}_j$: compute box $\mathcal{B}(\mathcal{S}_j)$
  4. subdivide each $\mathcal{B}(\mathcal{S}_j)$ into m horizontal slices
  5. for each resulting pixel set $\mathcal{S}_{ij}$: compute box $\mathcal{B}(\mathcal{S}_{ij})$

  $\Rightarrow$ array **B** of boxes $\mathcal{B}(\mathcal{S}_{ij})$

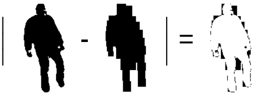
Robust Shape Encoding (3)

- Properties of the encoding scheme:

  1. complexity $\mathcal{O}(mn)$
  2. boxes $\equiv$ generalized pixels $\Rightarrow$ data compression
  3. homeomorphism between 2D lattice and shape

Robust Shape Encoding (4)

- Data compression:
  - compression rate: $1 - 4mn/2L$
  - normalized Hamming distance: $D = d_H(\mathcal{S}, \mathbf{B})/L$
  - Hamming distances for different $m \times n$:



(a) 8×4

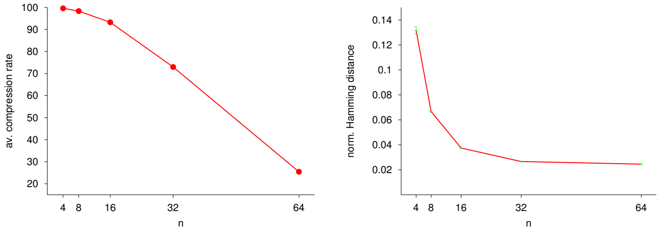
(b) 17×8

(c) 34×16

(d) 69×32

Robust Shape Encoding (5)

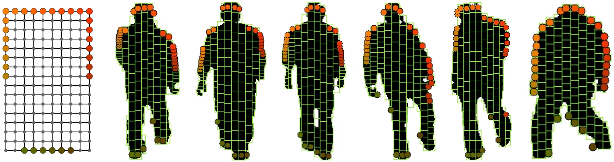
- Data compression:
  - experiments with 2170 shape images
  - average size $L \approx 20.000$ pixels
  - given n, set $m(n) = \left\lfloor \frac{h}{w} n \right\rfloor$
  - results:



$\Rightarrow$ encoding is accurate and storage & time efficient

Robust Shape Encoding (6)

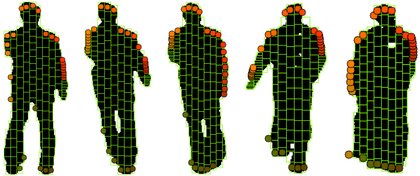
- Homeomorphism between 2D lattice and shape:
  - topology preserving mapping of shape onto lattice
  - example: 16 $\times$ 10 lattice



Vector Space Embedding (1)

- Boundary points of lattice capture shape dynamics:
  - normal gait

  - abnormal gait



Vector Space Embedding (2)

- Consider corresponding shape coordinates:
  - feature vector after normalization

$$\mathbf{r}_t = [\mu_x^{1,1}, \mu_y^{1,1}, \mu_x^{1,2}, \mu_y^{1,2} \ldots, \mu_x^{16,8}, \mu_y^{16,8}] \in \mathbb{R}^{60}$$

- Gait is temporal phenomenon:
  - stack feature vectors to model temporal context

$$\mathbf{s}_t = \mathbf{r}_t \oplus \mathbf{r}_{t-1} \oplus \ldots \oplus \mathbf{r}_{t-\Delta} \in \mathbb{R}^{60(\Delta+1)}$$

Experiments

- Data:
  - videos recorded in front of green screen
  - 7 subjects, each walking normal and abnormal
- Preprocessing:
  - background subtraction
  - 5×5 median
  - connected component analysis
- Training:
  - 7 videos of 4 subjects; 4 normal; 3 abnormal
  - SVM^{light} for training

Per Frame Results

- On training set:

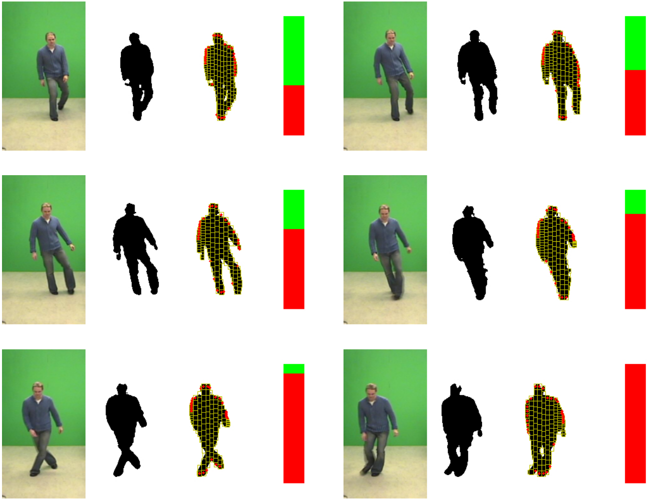
Δ	#frames normal	#frames abnormal	accuracy
1	1359	1128	73.5%
10	1323	1101	88.4%
20	1283	1071	93.9%

- On test set:

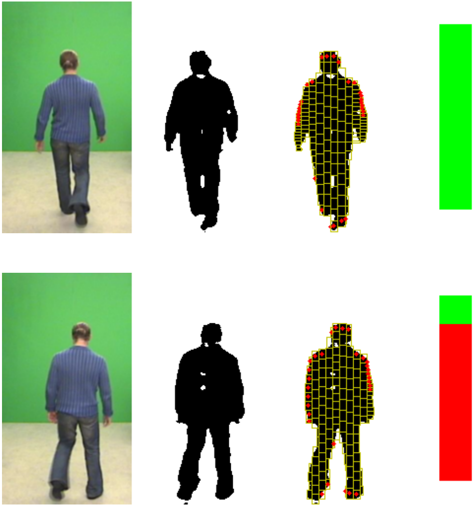
Δ	gait	#frames	accuracy
1	normal	1227	73%
	abnormal	1413	61%
10	normal	1195	77%
	abnormal	1382	70%
20	normal	1157	72%
	abnormal	1350	82%

Temporally Filtered Results

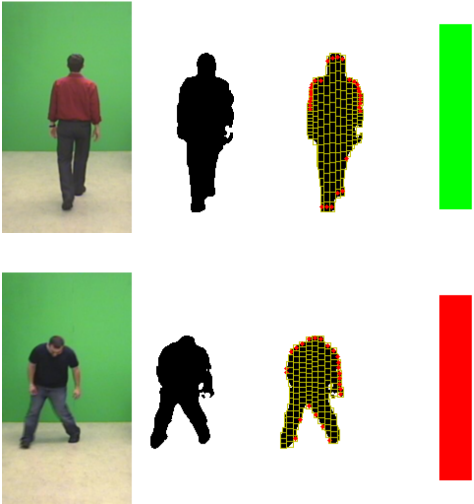
- Example: $\Delta = 20$



Temporally Filtered Results



Temporally Filtered Results



Summary

- Topic:
  - gait classification
- Approach:
  1. Homeomorphism between 2D lattices and shapes
     - robust, fast and storage efficient shape encoding
     - straightforward vector space embedding
  2. SVM classification
- Result:
  - satisfying per frame recognition rate
  - temporal smoothing of results improves applicability

Outlook

- Currently, focus on:
  - reliable segmentation of noisy signals
  - reliable tracking of several individuals
- Finally:
  - integration of segmentation, multiple people tracking and gait classification

References

- J.E. Boyd.
  Synchronization of oscillations for machine perception of gaits.
  Computer Vision and Image Understanding, 96(1): 35–59, 2004.
- R.T. Collins, R. Gross, and J. Shi.
  Silhouette-based human identification from body shape and gait.
  In *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pages 351–356, 2002.
- L. Wang, H. Ning, W. Hu, and T. Tan.
  Gait recognition based on procrustes shape analysis.
  In *Proc. ICIP*, volume III, pages 433–436, 2002.