Spatial-Temporal Feature Fusion for Human Fall Detection

Xin Ma(✉), Haibo Wang, Bingxia Xue, and Yibin Li

School of Control Science and Engineering, Shandong University, Jinan 250061, China
{maxin,liyb}@sdu.edu.cn, hbwang1427@gmail.com

Abstract. When suddenly falling to the ground, elderly people can get seriously injured. This paper presents a vision-based fall detection approach by using a low-cost depth camera. The approach is based on a novel combination of three feature types: curvature scale space (CSS), morphological, and temporal features. CSS and morphological features capture different properties of human silhouette during the falling procedure. All the two collected feature vectors are clustered to generate occurrence histogram as fall representations. Meanwhile, the trajectory of a skeleton point that depicts the temporal property of fall action is used as a complimentary representation. For each individual feature, ELM classifier is trained separately for fall prediction. Finally, their prediction scores are fused together to decide whether fall happens or not. For evaluating the approach, we built a depth dataset by capturing 6 daily actions (falling, bending, sitting, squatting, walking, and lying) from 20 subjects. Extensive experiments show that the proposed approach achieves an average 85.89% fall detection accuracy, which apparently outperforms using each feature type individually.

Keywords: Fall detection · Spatial-temporal feature · ELM

1 Introduction

When falling to the ground, elderly people need to be rescued as promptly as possible. Automatic fall detection becomes an emerging technique. Although wearable sensor has been used for fall detection [1], wearing a sensor will cause inconvenience to one's daily life. An unobtrusive technique is more favourable, but it needs to mount many ambient devices, such as vibration, sound sensor, infrared motion detector and pressure sensor, on room walls [2], which raises the cost of the solution, and may bring side effect to the human's health.

This work was supported in part by the National High Technology Research and Development Program of China under Grant No. 2015AA042307, Shandong Province Science and Technology Development Foundation under Grant No. 2014GGE27572, Shandong Province Independent Innovation and Achievement Transformation Special Fund under Grant No. 2014ZZCX04302, the Fundamental Research Funds of Shandong University under Grant No. 2015JC027, 2015JC051.

H. Zha et al. (Eds.): CCCV 2015, Part I, CCIS 546, pp. 438–447, 2015.
DOI: 10.1007/978-3-662-48558-3_44

Camera is the more convenient unobtrusive sensor for human fall detection [3]. Moreover, camera can not only capture human activities but also record contextual information, which may be significant for fall detection.

Shape analysis in 3D space is more robust to viewpoint and partial occlusion as compared to 2D shape features. With a reconstructed human volume, tracking the trajectory of the centroid and orientation of 3D human volume can detect falls [4]. Although fall detection becomes easier with 3D model, reconstructing the model is computationally demanding, and calibrating multiple cameras is still challenging [5]. Recently, Microsoft releases Kinect as a low-cost tool for 3D depth acquisition. Kinect is robust to the variation of visible lights, thus being able to work day and night. Moreover, the identity of the detected subject is well masked in the depth map of Kinect. Many depth-based applications have emerged, such as 3D skeleton analysis [6], 3D head detection [7] and 3D gait recognition [8].

In this paper, we present a new fall detection approach by using the depth map of Kinect. Unlike previous purely shape-based [9] or motion-based [10] approaches, the proposed approach bases off a combination of three spatial and temporal features: Curvature Scale Space (CSS) features [11], morphological and temporal features. Since the three types of features are different in terms of the number of features at each video frame, one Extreme Learning Machine (ELM) [12] classifier is separately trained for each feature. Only in the final decision, the prediction scores of the three classifiers are fused to predict whether fall happens or not.

The rest of the paper is organized as follows. Section 2 presents the proposed fall detection approach. Section 3 describes experimental results. Section 4 closes the paper with concluding remarks.

2 The Proposed Approach

Fig. 1 shows pipeline of the proposed fall detection approach. Three types of features (CSS, morphological and temporal) are extracted from each frame of the input depth videos. A bag-of-words model is then built for the CSS and morphological features, respectively. By mapping the collected feature vectors of a video clip to the words book, the histogram of occurrence counts is used to represent the video clip. Meanwhile, temporal features are directly vectorized as the third representation of a video clip (normalized to be 50-frames). An individual classifier is trained separately for each feature type, whose prediction score is combined to decide whether fall happens or not.

2.1 Preprocessing

Given input videos the first step is to segment human body from background by using the adaptive Gaussian Mixture Model (GMM) [13]. Following it, silhouette is extracted by using a simple edge detector [14].

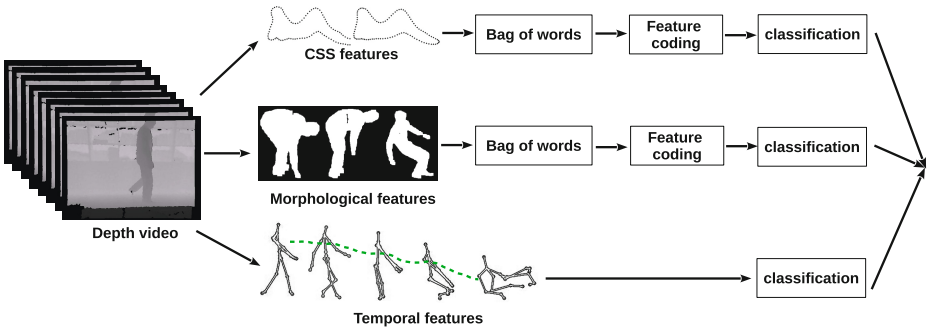


Fig. 1. Pipeline of the proposed fall detection approach. CSS, morphological and temporal features are extracted from input depth videos depicting various daily actions. A bag-of-words model is then built for the CSS and morphological features, respectively. The histogram of occurrence counts of the bagged words is used to encode each action. Meanwhile, temporal features are directly vectorized as the third action representation (normalized to be 50-frames). An individual classifier is trained separately for each feature type, whose prediction score is finally combined to decide whether fall happens or not.

The number of pixels on different silhouette is different, ranging from 10 to 50. To unify the pixel number for better CSS extraction, we uniformly sample 24 points on each extracted silhouette to form a compact silhouette. Then we process the compact silhouettes by normalizing their lengths to [0, 1] and smoothing them over time by averaging over the previous and the next four frames.

2.2 Spatial-Temporal Features

Curvature Scale Space (CSS). Curvature Scale Space (CSS) feature [11,15] is robust to translation, rotation, scaling and local deformation. Given a closed shape curve $\Gamma(x, y)$ with (x, y) at Cartesian coordinates, we re-parameterize $\Gamma(x, y)$ in terms of its arc length u: $\Gamma(u) = (x(u), y(u))$. The curvature κ of each Γ_σ is $\kappa(u, \sigma)$. The CSS image of Γ is defined at $\kappa(u, \sigma) = 0$, called the zero-crossing (ZP) point. There are two types of ZP: ZP_+ - the start point of a concavity arc where $\kappa(u, \sigma)$ changes from negative to positive, and ZP_- - the start point of a convexity arc where $\kappa(u, \sigma)$ changes from positive to negative. On a closed curve, ZP_+ and ZP_- always appear as a pair. The arc between a pair of ZP_+ and ZP_- is either concave (ZP_+, ZP_-) or convex (ZP_-, ZP_+). Since it is extracted from the curvatures at multiple scales, ZP is invariant to rotation, translation and uniform scaling. To make it further robust against local deformation, we resample the CSS features by curve interpolation. During the curve evolution, we keep increasing σ until Γ_σ shrinks to a circle-like shape, in which all ZPs disappear. On a CSS image, the (u, σ) coordinates of all ZPs form a set of continuous curves. The (u, σ) coordinates of the maxima point of each curve constitute our CSS feature vector.

Table 1. 15 morphological features used in the paper.

Name	Interpretation
Area	the actual number of pixels in the foreground region
Perimeter	the distance between each adjoining pair of pixels around the border of the region
EquivDiameter	the diameter of a circle with the same area as the region
MajorAxisLength	the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region
MinorAxisLength	the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region
Eccentricity	the eccentricity of the ellipse that has the same second-moments as the region
Extent	the ratio of pixels in the region to pixels in the total bounding box
Solidity	the proportion of the pixels in the convex hull and also in the region
ConvexArea	the number of pixels in the convex hull of the region
smoothness	a measure of contour smoothness
compactness	a ratio of perimeter to the region area
Hausdorff Dimension	the number that represents the generalized fractal dimension of a 2D matrix
average radial ratio	the ratio of the radial distance of the contour over the radial distance of the minimally inscribing sphere
area overlap ratio	the area of the object over the area of the minimally inscribing circle
Stddis	the standard deviation of the distance of contour points normalized by the maximum distance

Morphological Features. By filling the extracted shape silhouette, we obtain a foreground human region whose properties might uniquely characterize the depicted action. Thus we measure the region properties by computing its various morphological values. To this end we first detect the bounding rectangle of the region, and then normalize the bounded rectangle to the same 60×80 size. On the normalized patch, 15 morphological values are measured as detailed in Table 1. The values capture various regional properties such as area, perimeter, eccentricity, extent, smoothness, compactness and etc.

Temporal Features. Along with the depth map, the Kinect SDK can provide the 3D coordinates of 20 skeleton points. For each action the trajectories of the skeleton points can be very different. Thus, it is necessary to apply the trajectory of these skeleton points for fall detection. Unfortunately, when a person falls to the ground, the Kinect SDK fails to detect most of the skeleton points except the shoulder center. Therefore, we only consider the trajectory information of the shoulder center (shown in Fig. 2). Let V denote the 3D coordinates of the shoulder center at time t

$$V = (x_t, y_t, z_t). \tag{1}$$

To reduce the influence of coordinate center, we calculate the relative coordinates as our temporal feature

$$F_t = \{x_t - x_{t-1}, y_t - y_{t-1}, z_t - z_{t-1} | t = 2, 3, ..., T\} \tag{2}$$

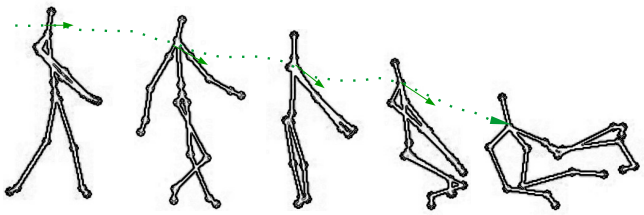


Fig. 2. Illustrating the temporal features. Kinect SDK could track the 3-D trajectory of upto 20 skeleton points. Among the points, the shoulder center is the only one that can be correctly tracked when a person falls to the ground. Thus, only the relative coordinates of the shoulder center is considered for fall detection. In particular, 50 frames are sampled from each sequence for the feature extraction, forming a vector of $3 \times 50 = 150$ dimensions.

where T is the sequence length. Throughout the paper, T is fixed at 50, indicating that we sample 50 frames from each sequence. Therefore, the temporal feature is a vector of $3 \times 50 = 150$ dimensions.

2.3 Feature Encoding

Both the CSS and the morphological features do not show obvious temporal consistency. Therefore, we neglect their temporal order, and use the Bag-of-Words (BoW) model [16] to generate distribution-based action representations. Since the numbers of CSS and morphological features can be different on each frame, an individual BoW model is separately built for each feature type.

In the first stage of BoW modeling, K-means clustering is applied over all feature vectors to generate a codebook. Each cluster center is a codeword, as a representative of similar feature vectors. Then by mapping the collected vectors of a video clip to the codebook, we have a histogram of occurrence counts of the words, which is the BoW representation of video action. Since both the CSS and the morphological features are in low-dimensional space (summarized in Table 2), building the BoW models is relatively fast.

The value of K is critical in the K-means clustering. We experimented with several values, and empirically found that fixing $K = 100$ is good enough for both the CSS and morphological features.

Table 2. Summary of the three feature types.

Features	CSS	Morphological	Temporal
Original Dimension	2	15	3
# of Clustering Centers	100	100	N/A
# of Action representation	100	100	$3 \times 50 = 150$

2.4 Classification and Fusion

Extreme learning machine (ELM) [12] is a single-hidden-layer feed-forward neural network. Given samples $\{\mathbf{x}_j\}$ and their labels $\{\mathbf{t}_j\}$, ELM is modeled by

$$\sum_{i=1}^{L} \beta_i \cdot g(\omega_i \cdot \mathbf{x}_j + \mathbf{b}_i) = \mathbf{y}_j, j = 1, \cdots N, \tag{3}$$

where $g(x)$ is an activation function, L indicates the number of hidden neurons, and ω_i, $\mathbf{b}_i$ and β_i are input weights, biases and output weights of the ith hidden neuron, respectively.

Rewriting Eq. 3 in matrix form leads to

$$\mathbf{H}\beta = \mathbf{Y}, \tag{4}$$

where $\mathbf{Y} = [\mathbf{y}_1^T, \cdots, \mathbf{y}_N^T]^T$, and $\mathbf{H}$ is the hidden layer output matrix,

$$\mathbf{H} = \begin{bmatrix} g(\omega_1 \mathbf{x}_1 + \mathbf{b}_1) & \cdots & g(\omega_L \mathbf{x}_1 + \mathbf{b}_L) \\ \vdots & \cdots & \vdots \\ g(\omega_1 \mathbf{x}_N + \mathbf{b}_1) & \cdots & g(\omega_L \mathbf{x}_N + \mathbf{b}_L) \end{bmatrix}_{N \times L}. \tag{5}$$

The ith column of $\mathbf{H}$ is the output of the ith hidden neuron with respect to the inputs $\mathbf{x}_1, \cdots, \mathbf{x}_N$. At training stage, by randomly initializing $\{\omega_i\}$ and $\{\mathbf{b}_i\}$, β can be efficiently optimized via least squares.

An individual ELM classifier is separately trained for each feature representation. Let $\mathbf{Y}_c$, $\mathbf{Y}_m$ and $\mathbf{Y}_t$ denote the labels predicted with the CSS, morphological and temporal features, respectively. The final predicted labels $\mathbf{Y}$ are designed as the weighted combination of $\mathbf{Y}_c$, $\mathbf{Y}_m$ and $\mathbf{Y}_t$

$$\mathbf{Y} = w_c \mathbf{Y}_c + w_m \mathbf{Y}_m + w_t \mathbf{Y}_t, \tag{6}$$

where w_c, w_m and w_t stand for feature significance. Through watching experimental results, we find that morphological features yield better fall detection accuracy than the CSS and temporal features. Therefore, we empirically set $w_c = 0.2$, $w_m = 0.5$ and $w_t = 0.3$.

3 Experimental Results

3.1 Dataset

SDUFall dataset [1] consists of 6 daily actions captured from 20 subjects: falling, bending, squatting, sitting, lying, and walking. Each subject repeats the same action 10 times, with each time one or more of the following conditions changed: carrying or not carrying something, light turning on or off, random walking-in direction and random viewpoint to the Kinect camera. The camera was installed $1.5m$ high for the action capturing. A total of $6 \times 20 \times 10 = 1200$ video clips are collected. Video frame is at size of 320×240, saved at $30fps$ in the AVI format. The baseline sequence length is about 8 seconds.

[1] http://www.sucro.org/homepage/wanghaibo/SDUFall.html

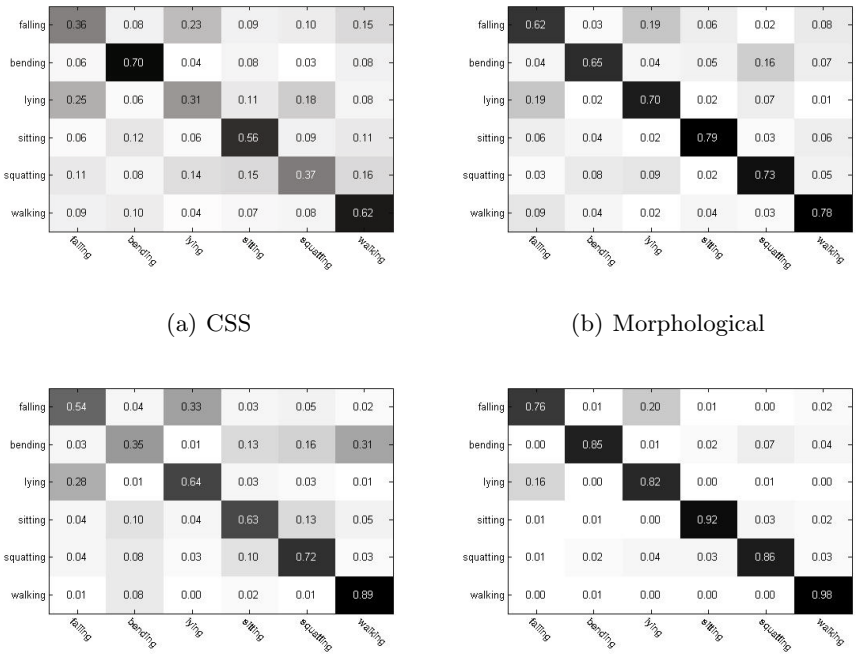
3.2 Settings

The CSS feature extraction was implemented in C++ while all the other modules were implemented in MATLAB. All the experiments were conducted with MATLAB 7.10 (R2010a) on a PC with Intel (R) Core (TM) i3-2120 CPU and 2.00 GB RAM. 5-folder cross validation on a per subject basis was repeated many times until every 5 subjects have been used as the test set.

The proposed approach is compared with using CSS [9], morphological and temporal feature individually. 100 cluster centers are applied for both the CSS and morphological clustering. The number of neurons in ELM is fixed at 80.

3.3 Results

Fig. 3 shows the confusion matrices of using (a) CSS features, (b) morphological features, (c) temporal features, and (d) the proposed approach that fuses the



(a) CSS (b) Morphological

(c) Temporal (d) Proposed

Fig. 3. Classification confusion matrices of using (a) CSS features, (b) morphological features, (c) temporal features, and (d) the proposed approach that fuses the three features. Among the three features, morphological feature is most discriminative. Fusing the three features greatly reduces the misclassification rates as compared to using each feature individually. It is also shown that falling and lying are likely to be mutually misclassified, indicating that their dissimilarities are only subtle.

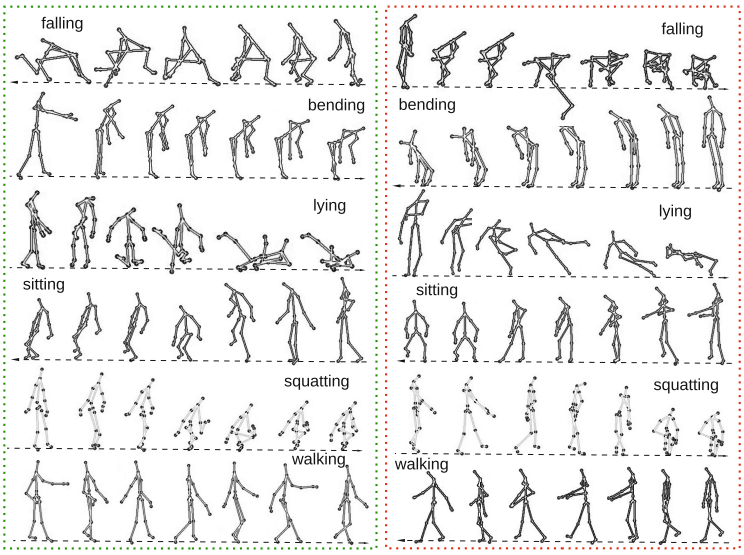


Fig. 4. Action samples that are most likely to be correctly classified (left, green-rectangle enclosed) and mis-classified (right, red-rectangle enclosed). The arrow under each sequence indicates the sequence order. Note that in order to increase the variation of our dataset, each subject repeats the same action by walking in from different directions.

Table 3. Classification accuracy (+ standard deviation) in distinguishing fall from non-fall actions.

Method	Fall vs Non-Fall Accuracy
CSS	$62.07 \pm 4.50\%$
Morphological	$76.96 \pm 6.11\%$
Temporal	$72.76 \pm 4.89\%$
Proposed	$85.89 \pm 5.02\%$

three features. Among the three features, morphological feature has the lowest misclassification rates since it calculates different statistics of the shape. Fusing the three features further reduces the misclassification rates ad the merits of the three feature types are combined. It is thus proven that the three features are mutually beneficial. It is also shown in the figure that falling and lying are likely to be mutually misclassified, indicating that their dissimilarities are subtle. Fig. 4 shows action samples that are likely to be correctly classified and misclassified. The misclassified falling action is largely attributed to the inaccurate track of the skeleton trajectory.

By treating the other five activities (sitting, walking, squatting, lying, and bending) as a single nonfall class, we are able to calculate the fall-versus-nonfall classification accuracy based off the results of Fig. 3. Table 3 shows the calculated

accuracy. Fusing the three features in the proposed approach significantly outperforms using each feature individually.

4 Conclusions

In this paper, we presented a new vision-based fall detection approach that uses only a low-cost Kinect camera. The approach is based off the fusion of three independent features. The CSS and morphological features capture different properties of human silhouette. But since the two features have no explicit temporal consistency, we cluster all collected feature vectors to generate occurrence histogram as the representation of an action. Meanwhile, we integrate the trajectory of a skeleton point that captures the temporal property of an action as a complimentary feature.

Extensive evaluation shows that the proposed approach achieves an average 85.89% accuracy in distinguishing fall from five other daily activities (walking, lying, sitting, squatting, and bending). However, it should be pointed that the proposed approach increases the computation complexity compared to the methods with only one kind of feature. In the future, we will optimize the weights for fusing the three prediction results which is set empirically. Moreover, we will capture more subtle daily activities such as eating, calling, laughing and carrying objects. Meanwhile, we will also integrate our dataset to other publicly available RGBD dataset for more general and precise fall detection.

References

1. Shany, T., Redmond, S., Narayanan, M., Lovell, N.: Sensors-based wearable systems for monitoring of human movement and falls. IEEE Sensors Journal **12**(3), 658–670 (2012)
2. Doukas, C., Maglogiannis, I.: Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components. IEEE Transactions on Information Technology in Biomedicine **15**(2), 277–289 (2011)
3. Popoola, O., Wang, K.: Video-based abnormal human behavior recognition-a review. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **42**(6), 865–878 (2012)
4. Yu, M., Naqvi, S., Rhuma, A., Chambers, J.: One class boundary method classifiers for application in a video-based fall detection system. IET Computer Vision **6**(2), 90–100 (2012)
5. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. IEEE Transactions on Information Technology in Biomedicine **15**(2), 290–300 (2011)
6. Planinc, R., Kampel, M.: Introducing the use of depth data for fall detection. Personal and Ubiquitous Computing **17**(6), 1063–1072 (2013)
7. Nghiem, A.T., Auvinet, E., Meunier, J.: Head detection using kinect camera and its application to fall detection. In: Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications, pp. 164–169 (2012)

8. Parra-Dominguez, G., Taati, B., Mihailidis, A.: 3D human motion analysis to detect abnormal events on stairs. In: Proceedings of the Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 97–103 (2012)
9. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE Journal of Biomedical and Health Informatics **18**(6), 1915–1922 (2014)
10. Mirmahboub, B., Samavi, S., Karimi, N., Shirani, S.: Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Transactions on Biomedical Engineering **60**(2), 427–436 (2013)
11. Mokhtarian, F.: Silhouette-based isolated object recognition through curvature scale space. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(5), 539–544 (1995)
12. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of IEEE International Joint Conference on Neural Networks, vol. 2, pp. 985–990 (2004)
13. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (1999)
14. Ding, L., Goshtasby, A.: On the canny edge detector. Pattern Recognition **34**(3), 721–725 (2001)
15. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(8), 789–805 (1992)
16. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 524–531 (2005)