CLASSIFYING COMPUTER GENERATED CHARTS

V. Shiv Naga Prasad, Behjat Siddiquie, Jennifer Golbeck and Larry S. Davis

Computer Vision Lab,
Dept. of Computer Science,
Univ. of Maryland - College Park,
College Park, MD 20742, U.S.A.
{shivnaga, behjat, golbeck, lsd}@cs.umd.edu

ABSTRACT

We present an approach for classifying images of charts based on the shape and spatial relationships of their primitives. Five categories are considered: bar-charts, curve-plots, pie-charts, scatter-plots and surface-plots. We introduce two novel features to represent the structural information based on (a) region segmentation and (b) curve saliency. The local shape is characterized using the Histograms of Oriented Gradients (HOG) and the Scale Invariant Feature Transform (SIFT) descriptors. Each image is represented by sets of feature vectors of each modality. The similarity between two images is measured by the overlap in the distribution of the features - measured using the Pyramid Match algorithm. A test image is classified based on its similarity with training images from the categories. The approach is tested with a database of images collected from the Internet.

1. INTRODUCTION

An average business computer user generates tens of charts and plots each week. This is in addition to the millions of potentially useful images of data-plots available on the Internet. We present an approach for classifying computer generated charts into different categories e.g. bar-charts, curve-plots, etc. This can be employed to automatically determine the type of chart depicted in a given image. Such a classification would be useful for further analysis of the data presented in the chart, semantic description of the chart, organizing a large collection of images of charts, etc.

Five types of charts are considered: bar-charts, curve-plots, pie-charts, scatter-plots and surface-plots. The objective is to model the structures of these categories so that a chart image can be classified into one of them. We assume that the chart has been isolated from the rest of the document using document-layout analysis, e.g., [5, 16]. The classification is challenging due to:

- Variability in the data being depicted. Consider the case of pie-charts - changes in the number of entities represented in the pie-chart and their relative quantities leads to variations in the structure of the pie-chart.
- Stylistic variation in terms of the color palette, shading, geometry, etc., makes structural analysis of the images difficult. E.g., pie-charts can be drawn in 3D with perspective distortion, by "exploding" the segments, with images overlayed within the segments.
- Presence of ancillary structure such as text, legends, axes, grids, etc.

In spite of this variability, each category has a distinctive primitive which is used to depict information. E.g., for bar charts a rectangle, for curve plots a salient curve. The classification is performed by characterizing the primitives and their spatial relationships. Shape and perceptual grouping features are used to compute statistics of the primitives present in each category of images. The classification is performed by measuring the similarity between the primitives present in a given test image and the learnt statistics. We introduce two novel features for characterizing the structural information based on: (a) region segmentation and (b) edge continuity. In addition, local shape is characterized using two state-of-the-art gradient descriptors - Histogram of Oriented Gradients (HOG) [6] and Scale Invariant Feature Transform (SIFT) [13]. The similarity between two images is quantified by the overlap in the distributions of these features, measured using the Pyramid Match algorithm [8]. An image is classified based on its similarity with training images of each category. The approach is tested with a database of images of charts collected from the Internet. Figures 7 and 8 show a subset of the database grouped according to the classification results.

1.1. Overview of the Approach and Related Work

Each image is processed in three stages:

1. *Preprocessing:* An open-source Optical Character Recognition (OCR) utility [1] is used to detect text in the image. A Gaussian-derivative kernel is used to compute the image gradients.

2. *Feature extraction:* The primitives present in the image are characterized using shape and perceptual grouping features.
3. *Distribution of the features:* A multi-resolution pyramidal histogram is computed for the obtained set of feature vectors.

During the training stage, the Pyramid Match algorithm [8] is used to compute the similarity between each pair of training images and a classifier is trained to classify based on these similarity scores. A test image is matched with each of the training images to obtain a vector of similarity scores, which is used for classifying it into one of the chart categories.

The image primitives are characterized using:

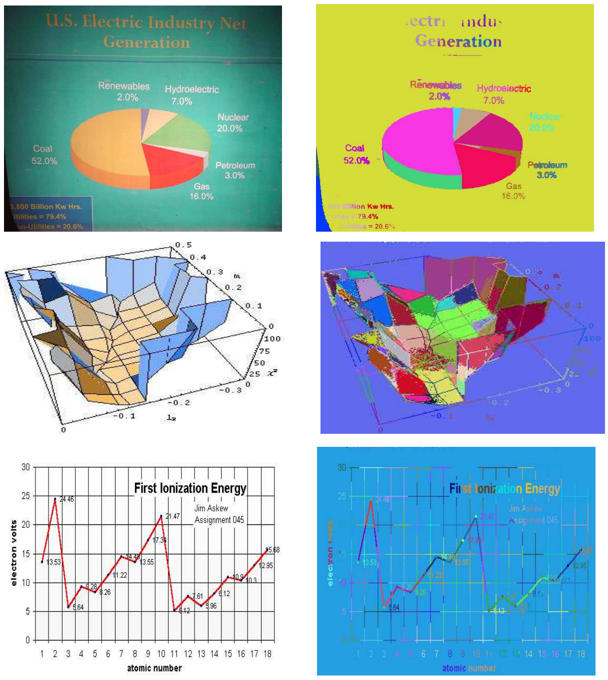
1. *Structure of salient regions present in the image.* A color clustering algorithm is used to obtain a segmentation of the image. Each of the obtained regions is characterized by the histograms of the boundary orientations and the edge distance-map within the segment (c.f. Section 2).
2. *Local shape of salient curves.* An approach proposed in [9] is used to enhance the gradients of salient curves present in the image. A localized version of the Radon transform is used to characterize the shape of the curves.
3. *SIFT*: SIFT and its variants have been shown to be effective for object detection and image retrieval applications. We use an implementation of SIFT proposed in [13].
4. *HOG descriptors*: HOGs consist of normalized histograms of image gradients w.r.t. various orientations, collected within rectangular regions in the image. They have been used as features for detecting pedestrians in [6], etc. The studies indicate that HOG features are robust to changes in illuminations and color-appearance.

2. REGION SEGMENTATION FEATURES

The region segmentation features are used to characterize image primitives that are salient regions, e.g., the pies in pie-charts. A large number of studies have been proposed for image segmentation based on color and texture, e.g., Normalized Cuts and its variants [18], Algebraic Multigrid [7], color clustering [10].

We use a split-and-merge color clustering algorithm for image segmentation. Each pixel is represented by a 5 dimensional vector consisting of its spatial coordinates and RGB values. Initially, all pixels in the image are put in a single cluster. In each iteration, clusters with variance greater than Δ_{split} are split into two clusters, and pairs of clusters whose union's variance is below Δ_{merge} are merged. This process is repeated until the cluster memberships reach an equilibrium, yielding a set of "super-pixels" for the image. Next, the intervening-contour feature proposed in [12] is used to merge adjacent clusters that are likely to belong to the same object in the image.

Fig. 1 shows examples of images and the corresponding segmentations. Due to the use of split-and-merge clustering, the number of clusters does not have to pre-specified. Regions with uniform color but having a "thin" structure, e.g. lines, are broken into multiple segments. The reason is that the intervening contour criterion assigns low likelihood for merging them.



(a) (b)

Fig. 1. Examples of region segmentation: (a) Image, (b) color map of the segmentation. (Best viewed in color.)

Two features are extracted from each region segment:

1. Histogram of the gradients' orientations (HOG) along the boundary of the segment. The orientation values are quantized into n_o uniform bins. The histograms are weighted by the gradient magnitudes.
2. Histogram of edge distance-map values within the segment. The Canny edge operator is used to compute an edge map of the image. The distance-map of the image quantifies the distance of each point on the image plane from the nearest edge. The presence of large distance-map values in the histogram indicates spatially extended segments with few or no "holes". In contrast, a majority of low distance-map values would indicate that the segment is thin, e.g. a curve or a scatter of small legends. The distance-map values are quantized into n_d bins of size Δ_d.

The HOG and the distance-map histograms are concatenated to form one feature vector for each region segment.

This gives a set, F_{seg}, of feature vectors for the image.

3. EDGE CONTINUITY FEATURES

Edge continuity is used to enhance the saliency of long continuous curves w.r.t. edges formed by scattered distributions and text. It has been studied in computer vision in the context of perceptual grouping and saliency. Shashua and Ullman constructed a network of edge elements for computing the saliency of edges based on curvatures [17]. Parent and Zucker used osculating circles to define edge continuity [14]. Guy and Medioni combined this with tensor voting to obtain saliency maps for images [9]. Williams et al. used a stochastic model to compute the probability that a contour connecting two edges on the image plane would pass through an intermediate point [20]. An edge affinity model was used for contour matching in [15]. We combine this with the tensor voting approach proposed in [9] for computing the saliency of the curves.

Each edge casts a vote at each of its neighboring edges for the possible orientations of the contours passing through them. The orientation and weights of the votes are computed using edge continuity constraints. Studies on edge grouping typically assume that curves with low curvature are more likely. In the model presented in [14], given two image points and the orientation of the edge at one of them, the most likely curve is assumed to be a circle passing through them. This is called the osculating circle. See Fig. 2.

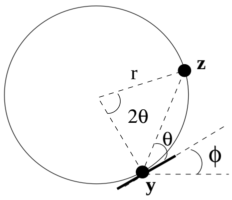


Fig. 2. Osculating circle given two points $\mathbf{y}$ and $\mathbf{z}$ lying on it and the tangent to the curve at $\mathbf{y}$.

The radius of the osculating circle for two points $\mathbf{y}$ and $\mathbf{z}$ is denoted by $r(\mathbf{y},\mathbf{z})$. It can be shown that

$$r(\mathbf{y},\mathbf{z}) = \frac{\|\mathbf{y}-\mathbf{z}\|}{2\sin\theta} \tag{1}$$

where θ is as shown in Fig. 2. The vote cast by $e_{\mathbf{y}}$ at $e_{\mathbf{z}}$ has orientation tangential to the osculating circle at $\mathbf{z}$, i.e. $\psi = 2\theta + \phi$. An affinity function for edge continuity was formulated through an analysis of contours in [15]. It is used to compute the weight for the vote cast by $e_{\mathbf{y}}$ at $e_{\mathbf{z}}$:

$$a(e_{\mathbf{y}}, e_{\mathbf{z}}) = \frac{1}{1+\exp(-\frac{r(\mathbf{y},\mathbf{z})-\lambda_1}{\lambda_2})}\|\Delta I(\mathbf{y})\|\,\|\Delta I(\mathbf{z})\| \tag{2}$$

$\Delta I(.)$ denotes the image gradient field. The constants λ_1 and λ_2 are assigned values 6 and 0.9, respectively [15].

The vote is defined to be the tensor $v(e_{\mathbf{y}}, e_{\mathbf{z}})$

$$v(e_{\mathbf{y}}, e_{\mathbf{z}}) = a(e_{\mathbf{y}}, e_{\mathbf{z}}) \begin{bmatrix} \cos^2\psi & \sin\psi\cos\psi \\ \sin\psi\cos\psi & \sin^2\psi \end{bmatrix} \tag{3}$$

The net vote cast at an edge is computed as

$$w(e_{\mathbf{z}}) = \sum_{e_{\mathbf{y}} \in N(e_{\mathbf{z}})} v(e_{\mathbf{y}}, e_{\mathbf{z}}) \tag{4}$$

Here $N(.)$ defines a $m \times m$ neighborhood around each edge.

When the orientations of the votes cast at an edge e are consistent, the rank of $w(e)$ is 1. Let $\lambda_1(e)$ and $\lambda_2(e)$ be the eigen values of $w(e)$ ($\lambda_1 \geq \lambda_2$). The gradient saliency map on the image edges is defined as

$$s(e) = \lambda_1(e) - \lambda_2(e) \tag{5}$$

For a long contiguous curve, the votes cast at its edges will be consistent, resulting in high saliency. In contrast, edges with randomly oriented neighbors get inconsistent votes, resulting in low saliency. Fig. 3 shows some images and the salient curves obtained.

Two set of features are extracted from the gradient saliency maps:

1. *HOG:* Histograms of the image gradients' orientations in the $m \times m$ neighborhood of each edge e with $s(e) \geq \Delta_{\text{sal}}$. The orientation values are quantized into n_o uniform bins. The histograms are weighted by the saliency values. The set of HOGs computed from the gradient saliency map is denoted by F_{HOGSal}.
2. *Local Radon transform* of the saliency map $s(.)$ at each edge e with $s(e) \geq \Delta_{\text{sal}}$. The Radon Transform has been studied in medical imaging for 3D reconstruction, and in the shape representation literature. It is computed by taking projections of a 2D figure on straight lines oriented at different angles w.r.t. the axes. See Fig. 4 for an illustration. The set of local Radon transforms computed from the gradient saliency map is denoted by F_{RadSal}.

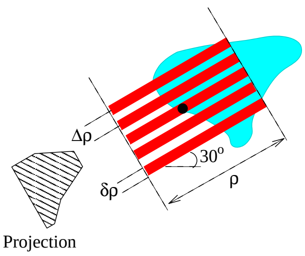


Fig. 4. Illustration of Radon transform: local projections are computed along various orientations - here shown at 30°. Each projection is performed within $\rho \times \delta\rho$ bands placed at $\Delta\rho$ intervals.

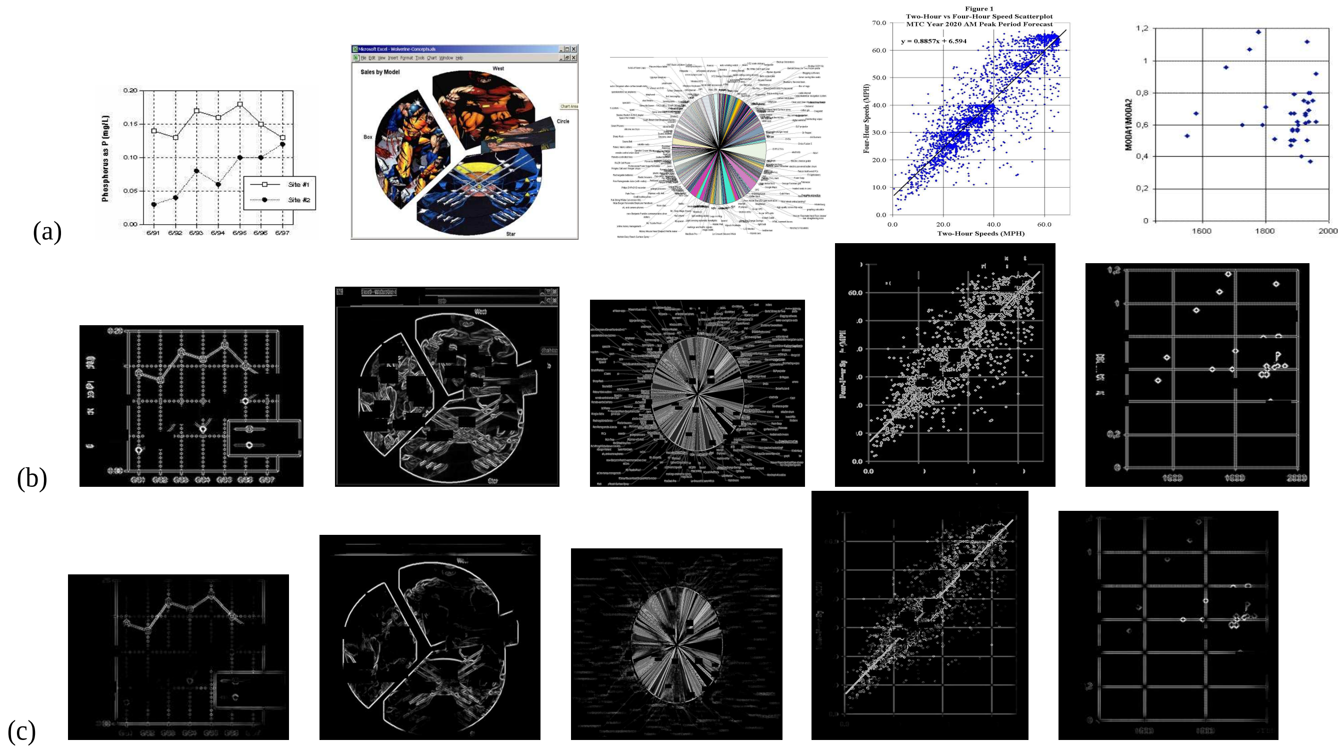


Fig. 3. Examples of salient edge: (a) Image, (b) Gradient magnitude map with edges of detected text suppressed, (c) Edge saliency map. (Best viewed in color.)

4. SIFT

The Scale Invariant Feature Transform (SIFT) is a local appearance based feature employed for image registration and object detection [13]. A large number of key-points are detected based on extrema in the image's scale-space. Each key-point is described using statistics of the image gradients in its neighborhood. The sizes of the neighborhoods are determined from local scale-space characteristics. This technique was extended by describing the key-points with the Principle Component Analysis (PCA) of the local image gradients, called PCA-SIFT [11]. A number of object recognition and image retrieval approaches have been proposed using SIFT and PCA-SIFT, e.g. [19, 8], etc. The descriptors have been shown to be robust to affine, scale and illumination variation. We use the SIFT descriptor implemented in [13]. Fig. 5 shows examples of images and the detected SIFT key-points. The set of SIFT feature vectors obtained for an image are denoted by F_{SIFT}.

5. GRADIENT DESCRIPTORS

Two descriptors are employed:

1. *Histograms of Oriented Gradients (HOG):* A number of variations of HOG have been used for key-point description e.g. [4, 11, 13], pedestrian detection e.g. [6]. We employ the descriptors used in [6]. The HOGs are computed in $m \times m$ neighborhoods around each edge; the orientations are quantized into n_o uniform bins. The set of HOG vectors computed for an image are denoted by F_{HOG}.
2. *Distance Map Histograms:* The distance map values in the neighborhood of an edge indicate the presence/absence of neighboring edges. The distance map histograms are computed in $m \times m$ neighborhoods around each image edge. The distances are quantized into n_d bins with size Δ_d. This gives a set, F_{DT} of feature vectors.

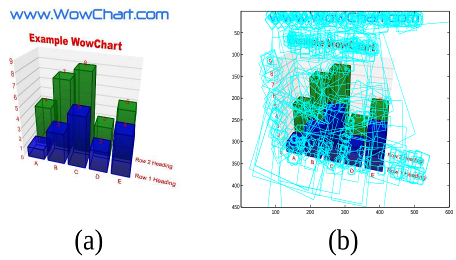


Fig. 5. Examples of SIFT key-point detection: (a) Image, (b) key-point locations, orientation and scales indicated with squares. (Best viewed in color.)

6. CLASSIFICATION

The classification is performed in the following stages:

1. The feature sets (F's) are computed for each image in the database.
2. A multi-resolution histogram is computed for each feature modality for each image.
3. For each feature modality, the Pyramid Match algorithm is used to compute similarities between all possible pairs of images. For a database of M images, this results in a $M \times M$ similarity matrix for each feature modality. The similarity values quantify the images' relative positions in an embedded space. Let l be the number of feature modalities. Each image is represented by an lM dimensional vector, $\mathbf{f}_{\mathrm{sim}}$, obtained by concatenating the similarity values from the feature modalities.
4. For each pair of chart categories, a boosting technique [2] is used to determine the most discriminative components from the lM dimensions. The discriminative components for all pairs of categories are combined to obtain a low dimensional representation of the image similarities, denoted by $\mathbf{f}_{\mathrm{redSim}}$. The components of this vector represent the training images that are the most discriminative members of each category.
5. A multi-class Support Vector Machine (SVM) [3] is trained to classify the $\mathbf{f}_{\mathrm{redSim}}$'s into the chart-categories.

6.1. Pyramid Match

The Pyramid Match algorithm was proposed for computing the similarity between two set of features vectors [8]. The sets can have different cardinalities. The approach has been shown to be efficient and effective for image retrieval applications. Given a set of feature vectors, a pyramid (multi-resolution) histogram is computed. The scale of quantization bins at each level of the histogram is half that of the previous level's bins. The match between two pyramid histograms is measured by the overlap in the bins, with higher weight given to overlap at finer scales. See [8] for details. Fig. 6 shows the similarity matrix obtained for HOG features of the curve saliency maps.

7. EXPERIMENTAL RESULTS AND SUMMARY

The approach was tested with a database of chart images collected from the Internet. The database consisted of 124 bar-charts, 117 curve-plots, 130 pie-charts, 158 scatter-plots and 124 surface-plots - 653 images in total. The experiments were conducted in 5 trials, in each trial $\frac{1}{5}$ of the data was used for testing and the rest was used for training[1]. Tab. 1 shows the classification results. Figures 7 and 8 show a subset of the charts database grouped according to the computed labels.

[1]Values of the parameters used in the experiments:

$\Delta_{\mathrm{split}} = 500$	$\Delta_{\mathrm{merge}} = 300$	$m = 41$	$n_o = 8$
$\rho = 41$	$\delta\rho = 5$	$\Delta\rho = 5$	♯bands= 5
$\Delta_d = 3$	$n_d = 6$	$\Delta_{sal} = 0.1$	

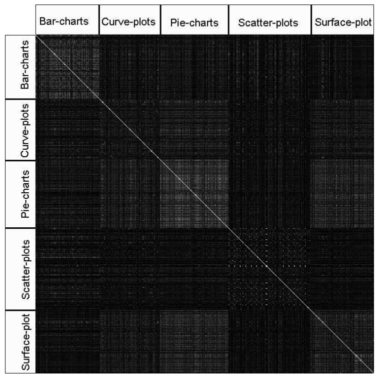


Fig. 6. Similarity matrix for the images computed for HOG of curve saliency maps (F_{HOGSal}).

	Bar-charts	Curve-plots	Pie-charts	Scatter-plots	Surface-plots
Bar-charts	**112 (90%)**	2	2	4	4
Curve-plots	7	**89 (76%)**	8	10	3
Pie-charts	2	6	**108 (83%)**	1	13
Scatter-plots	10	10	0	**136 (86%)**	2
Surface-plots	3	7	6	3	**105 (84%)**

Tab. 1. Confusion matrix for the classification

The following observations can be made about the results:

- One of the bar-charts that are misclassified as a curve-plot has a line plotted over the bars. 3 of the bar-charts misclassified as surface-plots are 3D bar-charts with large cuboids as primitives.
- 5 of the curve-plots misclassified as pie-charts have solid colors filled in between the curves. It is likely that the region segmentation features for these images resemble those for pie-charts resulting in the errors. This highlights the need for better shape features for the region segments.
- 5 of the pie-charts misclassified as surface-plots have dense imagery or text printed on the charts. The texture and lines present in these images are likely to be a cause for the errors.
- 6 of the scatter-plots misclassified as curve-plots have lines of best fit drawn on them.
- Of the surface-plots misclassified as scatter-plots, one image has a scattered distribution of globules plotted

in addition to a surface, and another case is a 4D plot depicted as a scattered distribution of legends in 3D whose color depends upon the 4^{th} dimension.

7.1. Summary

We presented an approach for classifying charts into various categories based on the shape and spatial relationships of their primitives. Sets of features based on region segmentation, curve saliency, HOG and SIFT are extracted from the image. Similarity between images is measured by the overlap in the distributions of these features - computed using the Pyramid Match algorithm. The categorization is accomplished by boosting the features and classifying using SVMs. The classification results indicate the utility of the approach. The features used for the classification are generic, admitting their applicability for other document classification tasks. We are currently exploring the features for semantic analysis of the charts.

8. REFERENCES

[1] GOCR, http://jocr.sourceforge.net/download.html.

[2] MSU Graphics and Media Lab, Computer Vision Group, http://graphics.cs.msu.ru.

[3] OSU Support Vector Machines (SVMs) Toolbox, http://www.csie.ntu.edu.tw/c̃jlin/libsvm.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(4):509–522, 2002.

[5] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical Report ITC-irst-TR9703-09, ITC-IRST, Povo, Trento, Italy, 1998.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR-2005*, 2005.

[7] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV-2003*, pages 716–723, 2003.

[8] C. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV-2003*, 2003.

[9] G. Guy and G. Medioni. Inferring global perceptual contour from local features. *Int'l J. Computer Vision*, 20(1-2):113–133, 1996.

[10] B. Heisele, U. Kressel, and W. Ritter. Tracking non-rigid, moving objects based on color cluster flow. In *Proc. 1997 IEEE Conf. Computer Vision and Pattern Recognition (CVPR '97)*, pages 257–261, 1997.

[11] Y. Ke and R. Sukthankar. Pca-sift: A more discriminative representation for local image descriptors. In *CVPR-2004*, 2004.

[12] T. K. Leung and J. Malik. Contour continuity in region based image segmentation. In *Proc. European Conf. Computer Vision (ECCV'98)*, volume 1, pages 544–559, 1998.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[14] P. Parent and S.W.Zucker. Trace inference, curvature consistency and curve detection. *IEEE Trans. Pattern Anal. and Machine Intell.*, 11(8):823–839, Aug. 1989.

[15] V. S. N. Prasad, L. S. Davis, S. D. Tran, and A. Elgammal. Edge affinity for pose-contour matching. *Computer Vision and Image Understanding*, 104(1):36–47, Oct. 2006.

[16] Dae-Seok Ryu, Seong-Whan Lee, and Sun-Mee Kang. Parameter-independent geometric document layout analysis. *ICPR-2000*, 04, 2000.

[17] A. Shashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *ICCV-1988*, pages 321–327, 1988.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 22(8):888–905, 2000.

[19] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV-2003*, 2003.

[20] L. R. Williams and K. K.Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *Int'l J. Computer Vision*, 34(2-3):81–96, 1999.

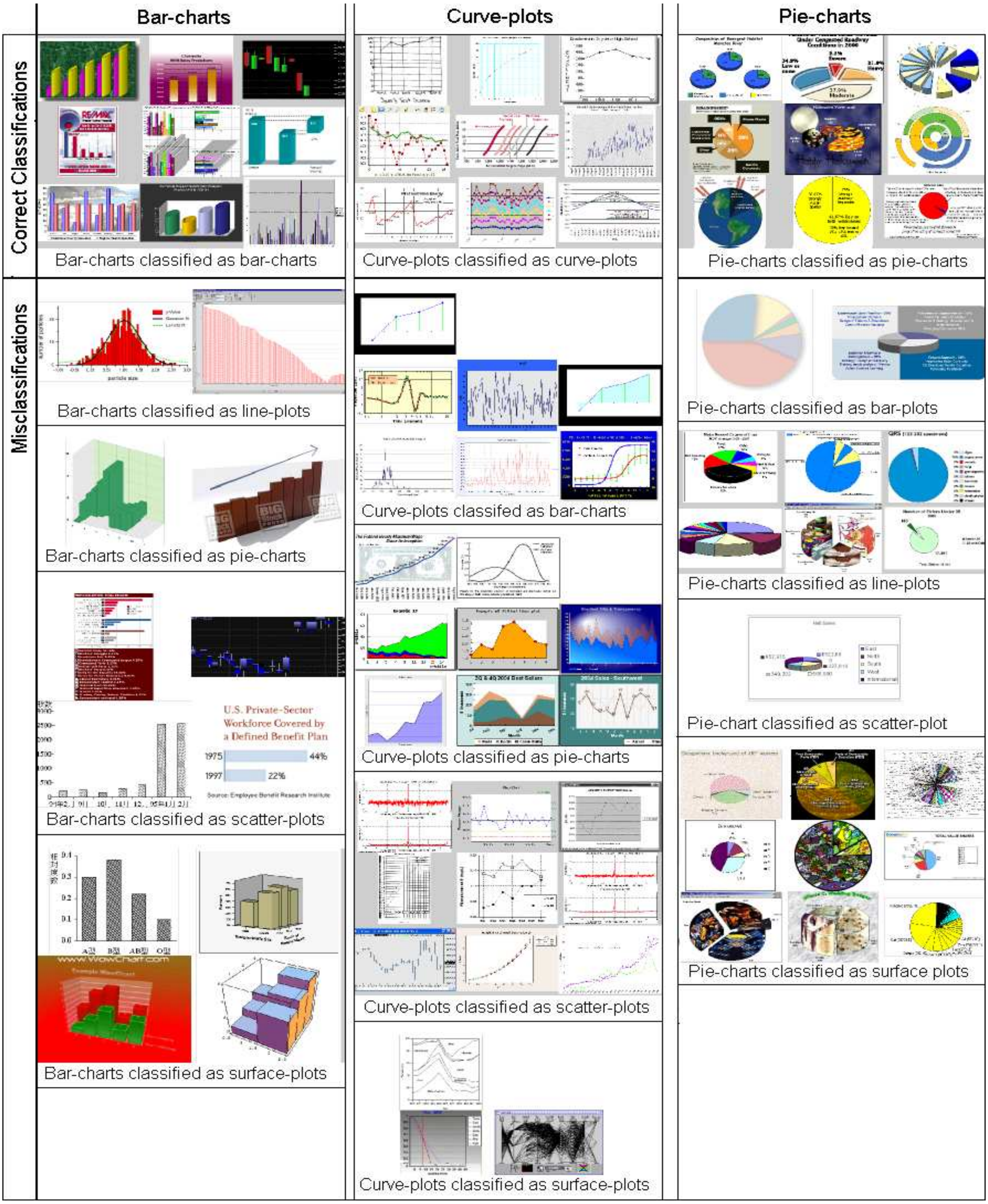


Fig. 7. Subsets of the classification results for bar-charts, curve-plots and pie-charts

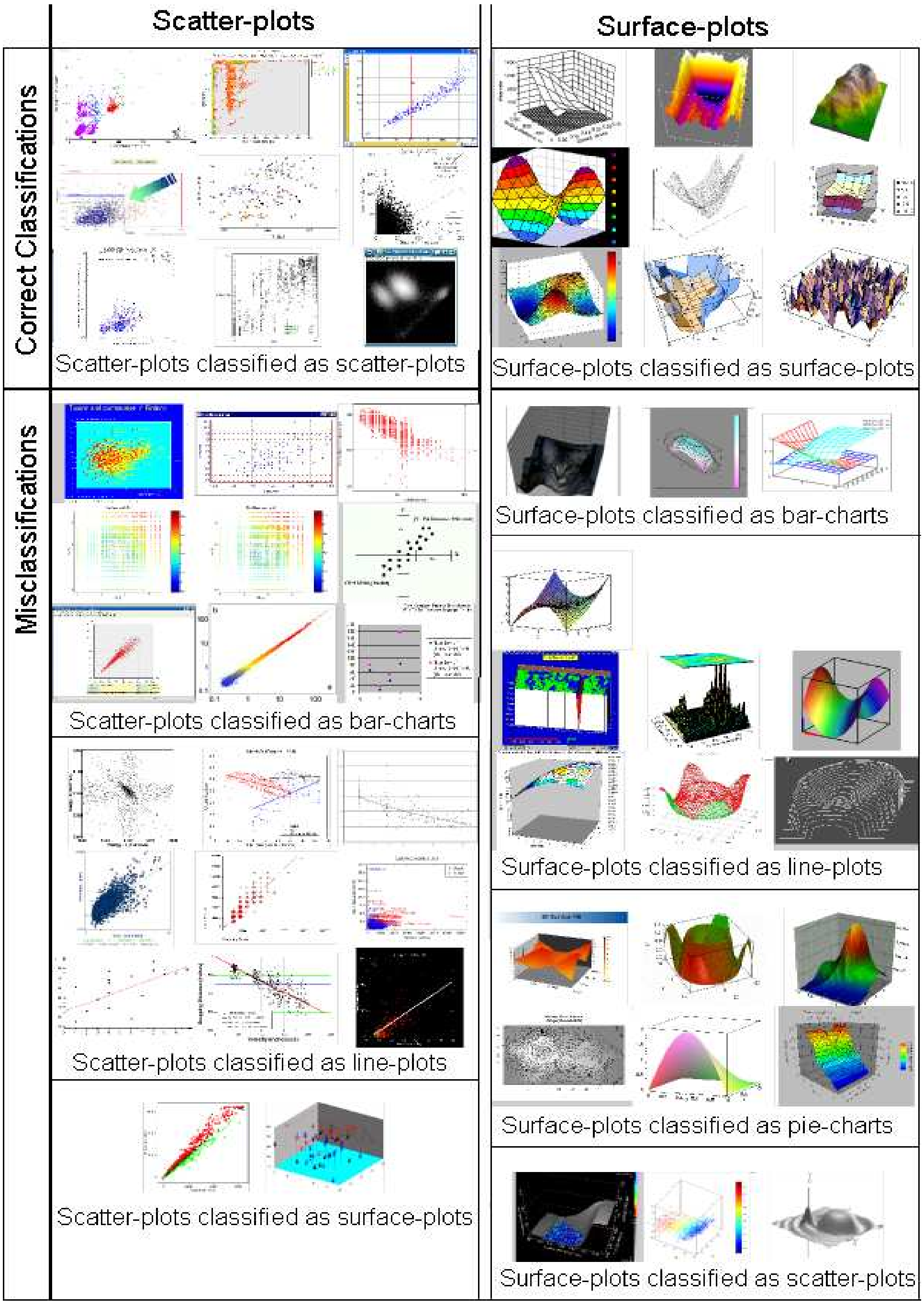


Fig. 8. Subsets of the classification results for scatter-plots and surface-plots