Image Descriptors based on Curvature Histograms

Philipp Fischer[*] and Thomas Brox

Department of Computer Science
University of Freiburg
{fischer,brox}@cs.uni-freiburg.de

Abstract. Descriptors based on orientation histograms are widely used in computer vision. The spatial pooling involved in these representations provides important invariance properties, yet it is also responsible for the loss of important details. In this paper, we suggest a way to preserve the details described by the local curvature. We propose a descriptor that comprises the direction and magnitude of curvature and naturally expands classical orientation histograms like SIFT and HOG. We demonstrate the general benefit of the expansion exemplarily for image classification, object detection, and descriptor matching.

1 Introduction

Orientation histograms, such as SIFT [14] or HOG [4], are omnipresent in computer vision. They are the dominant descriptors in all recognition tasks and play a major role in structure from motion and image retrieval. This success is because the descriptor is invariant to local deformations on the one hand, and it

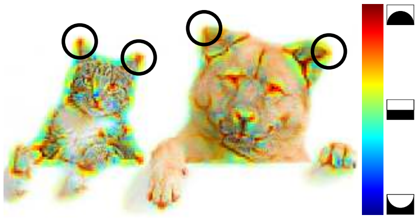

Fig. 1. Curvature is a valuable feature. The image is overlaid with its scalar curvature q, where the opacity is given by the gradient magnitude. The ear tips have high curvature. The average q values are: 62 for the cat's ears, 43 for the dog's ears.

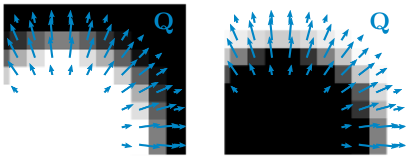

Fig. 2. The two images are considered equivalent when using the proposed curvature descriptors. The vector curvature Q is invariant to the sign of the gradient and always points from the interior to the exterior of the circle. The image shows the product of Q with the local gradient magnitude.

[*] Supported by a scholarship of the Deutsche Telekom Stiftung

is still descriptive due to the spatial grid of multiple histograms (cells) on the other hand. It allows recognition approaches to deal with a large set of natural variations without the need to model them explicitly.

However, the invariance to local deformations also frustrates discrimination of structures. HOG and SIFT locally simplify the image to straight lines. Therefore, the pointed tip of a cat's ear leads to almost the same representation as the roundish ear of a dog (see Fig. 1).

In this paper, we extend the idea of orientation histograms to curvature, i.e., the image is no longer locally simplified to straight lines but curved lines. Consequently, curvature histograms can distinguish local shape at a more detailed level. In contrast to previous work [16], we compute curvature with a per-pixel filter instead of operating on parametric curve segments. Hence, the approach is generally applicable and the descriptor can be computed basically as fast as a SIFT or HOG descriptor.

Moreover, we include the sign of the curvature (convex vs. concave). It is worth noting that the sign of the curvature is different from the sign of the gradient, which often is of little use in classification tasks. In contrast, the sign of the curvature allows the descriptor to get an idea where the interior or exterior of a (convex) object is, independent of the gradient direction; see Fig 2.

The richer set of features leads to significant improvements in descriptor performance. We demonstrate this with image classification experiments on Caltech 101, detection experiments on Pascal VOC 2007, and descriptor matching experiments on the dataset of Mikolajczyk et al. [15].

2 Related Work

The most closely related work is by Monroy et al. [16], where curvature histograms are used for object detection. In [6] Eigenstetter et al. use these features in combination with self-similarity. Our approach is different in the way how curvature is defined and computed. In [6, 16], the *chord-to-point distance accumulation* from [10] is used to measure curvature; see Fig. 3. This requires boundary segments, which Monroy et al. obtain performing a boundary probability computation pB [1]. Due to sparsity of such edge maps, some features in the image are lost for the descriptor. In contrast, we compute curvature densely based on the image gradient, which is significantly faster and preserves all relevant features. While Monroy et al. showed good results of their approach on images with strong line segments, our approach is more generally applicable, as we demonstrate in Section 4.

There are many works on curvature computation in general. In the continuous setting, curvature is well defined [17], yet on discrete images there are multiple ways to estimate it. Most works on discrete curvature estimation operate on binary images or discrete contours [3, 11, 13]. In contrast, the work in [5] also applies to color images. It assumes a height field surface given by the image intensity, and calculates the principal curvatures on this surface. However, this is fundamentally different from the intuitive understanding of curvature in an

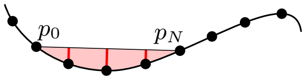

Fig. 3. Curvature computation by chord-to-point distance accumulation. The chord-to-point distances are shown as red lines. These are accumulated to obtain the local curvature at $p_{N/2}$ in [6, 16]

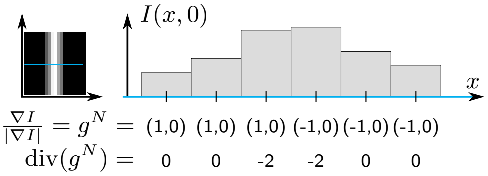
$I(x,0)$
x
$\frac{\nabla I}{|\nabla I|} = g^N =$ (1,0) (1,0) (1,0) (-1,0) (-1,0) (-1,0)
$\text{div}(g^N) =$ 0 0 -2 -2 0 0

Fig. 4. Profile of a vertical line in a discrete image. The sign of the normalized gradient changes at the center of the line. This leads to a non-zero curvature on straight lines when curvature is calculated via the divergence of the normalized gradient (in this example central differences were used). With the proposed descriptor the straight line does not expose curvature.

image, which is the curvature of isolines. Another classical application of curvature is the use of maxima of the norm of the Hessian to detect interest points. However, in this paper we compute the signed curvature densely on natural color images and curvature is represented by a curvature histogram to increase the expressiveness as a local image descriptor.

In a wider sense, also Carreira et al. [2] is related to our work. They suggested second order pooling of features in the context of semantic segmentation. This work shares the idea that important second order features need to be computed *before* pooling, as they get lost otherwise. While they propose the use of the feature vector covariance, we propose the use of second order derivatives. The two concepts are complementary.

3 Curvature

In general, the curvature of some curve is understood as how much it deviates from a straight line. Quantitatively, for one point on the curve it is defined as $\kappa = 1/r$, where r is the radius of a circle fitting the curve locally; hence straight lines have zero curvature.

As there are different ways to describe a curve, there are also different ways to define curvature, all leading to the same concept as stated above. We focus on the definition from [17] and assume that the curve X is parameterized by the arc-length t, such that equally spaced t map to equally spaced $X(t)$. Let $T(t)$ be the tangent and $N(t)$ the normal at $X(t)$. As in [17], we define curvature as the change of the normal along the tangent:

$$\kappa \cdot T = -\frac{dN}{dt} \qquad (1)$$

For an intuition, imagine how the normal rotates when you walk along a curve. The quicker the change, the higher the curvature. The sign of κ depends on the direction of the normal rotation.

3.1 Dense Vector Curvature on Images

When computing curvature in an image, we are limited by its resolution, and hence the underlying curve can never be reproduced accurately. Curvature computed on discrete images can only be an approximation of κ. In the discrete case, the image gradient must be approximated by considering a neighborhood with some spatial extent, for instance, by using finite differences. To distinguish a slightly curved line from a straight line, the observed area must be large enough. Hence, the maximum and minimum curvature that can be detected, are bounded by pixel size and neighborhood size.

Another important issue is illustrated in Fig. 4. While the divergence of the normalized gradient is a standard method to determine the curvature of the isolines, in the discrete case it can yield undesired curvature responses along straight lines.

Following the idea of Eq. 1, the aim is to determine the change of the gradient (or normal) along the tangent. Locally, the tangent coincides with the curve itself. At each point (x, y) we extract the gradient $g = (g_x, g_y)^T = \nabla I$ using standard linear differencing filters. This yields also the tangent $\phi = (-g_y, g_x)^T$ orthogonal to the gradient. Further, let $g^N = g/|g|$ and $\phi^N = \phi/|\phi|$ denote their normalized forms.

The Jacobian matrix of the normalized gradient (i.e. its differential) describes the change of gradient direction:

$$J(g^N) = \begin{pmatrix} \left(g_x^N\right)_x & \left(g_x^N\right)_y \\ \left(g_y^N\right)_x & \left(g_y^N\right)_y \end{pmatrix}$$

Thus, $J(g^N) \cdot \phi^N \in \mathbb{R}^2$ gives the gradient change in tangent direction. According to (1), the resulting vector is expected to be $-\kappa \cdot T$ and hence parallel to the tangent. On a discrete image, however, small deviations occur. Thus we project this vector to the tangent to obtain the scalar q approximating κ:

$$q = -\left(\left(\phi^N\right)^\top \cdot J(g^N) \cdot \phi^N\right) \in \mathbb{R} \qquad (2)$$

Note that this scalar curvature can be positive or negative. The sign depends on both the convexity and the sign of the gradient.

For deriving a descriptor, we suggest combining the curvature with the orientation of the gradient. In this scope, we are interested in decoupling the sign of the curvature from the sign of the gradient to distinguish convex and concave curves (see Fig. 2). To this end, we multiply q with the gradient vector g^N

$$Q = q \cdot g^N \in \mathbb{R}^2 \qquad (3)$$

and obtain a vector Q, which we call *vector curvature*.

Intuitively, driving along the curved isoline with a car, the vector Q points in the direction of the centrifugal force that acts on the passenger. Its magnitude is proportional to this force, assuming a constant velocity.

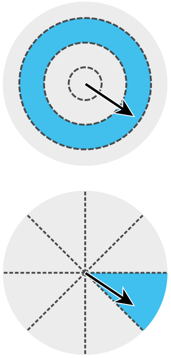

Fig. 5. The curvature vector Q can be binned to marginal histograms for magnitude and direction. This results in 12 bins.

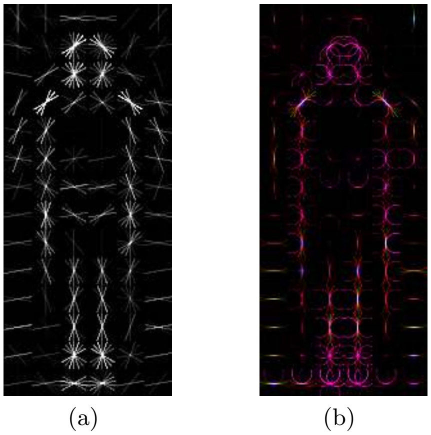

Fig. 6. Positive weights of feature descriptors trained for pedestrian detection using a linear SVM. In (a) HOG features were used, (b) shows a curvature descriptor. High convexity is learned pointing upwards in the head region and downwards in the feet region.

3.2 Feature Binning

With the above procedure, we compute the vector curvature Q densely for every pixel in an image and bin it into sparse per-pixel histograms as illustrated in Fig. 5. We quantize the orientations of Q into 8 bins. This histogram is partially redundant to the orientation histogram, but the sign depends on the direction of the curvature rather than the direction of the gradient. Our experiments show that the combination of orientation, sign and magnitude of curvature makes a strong feature.

Following the standard procedures to assemble SIFT and HOG descriptors [4, 8, 16], we pool these dense features into spatial histograms. The image is divided into a grid of square cells, and for each cell the sparse curvature histograms (one for each pixel) are aggregated by weighting them with their respective gradient magnitude. For detection, we follow the soft binning approach by linearly interpolating between cell centers just as in HOG. Figure 6 shows visualizations of trained HOG and curvature descriptors.

The feature computation on one image of size 300×250px takes around 0.3 seconds with HOG. Adding our curvature, increases the computation time to 0.7 seconds. The approach from [16] takes about 35 seconds due to the costly extraction of boundary segments.

4 Experiments

To demonstrate the benefits of curvature histograms in a wide range of tasks and images, we tested it in three major areas: image classification, object detection, and descriptor matching.

Curvature is a second order feature and hence a natural expansion to first order features like HOG and SIFT. In isolation, second order features are not meaningful enough to exceed the performance of a good first order feature. Thus, we compared gradient-only features to the version that includes gradient and curvature histograms. A comparison to the boundary curvature from [16] was done in the same way.

4.1 Image Classification

For the experiments on image classification we used the Caltech 101 dataset and the VLfeat implementation of spatial pyramid matching [12, 18]. The baseline method is based on dense SIFT descriptors. It trains a k-means based bag-of-words classifier with an SVM. By collecting SIFT descriptors from random positions and scales, it builds a dictionary using k-means clustering. Using this dictionary, for every training image a spatial histogram is built with one bin per *word* (i.e. one k-means cluster). Spatial binning is achieved by dividing the image into 2 × 2 and 4 × 4 cells and generating histograms for each cell of each division. A χ^2 kernel map is applied before training the SVM on the histograms. For further details, we refer to the public source code.

We compared this baseline to our curvature-augmented approach which was added in the form of a second spatial pyramid (i.e. using a second dictionary). For training the SVM and during testing, both histograms were concatenated.

Results. For every category, we randomly chose 15 training images and 15 test images. We repeated the whole training and testing process for 10 random sets to report statistics on the accuracy in Fig. 8(a). The *accuracy* refers to the share of test images that were classified correctly, i.e., the sum of the confusion matrix

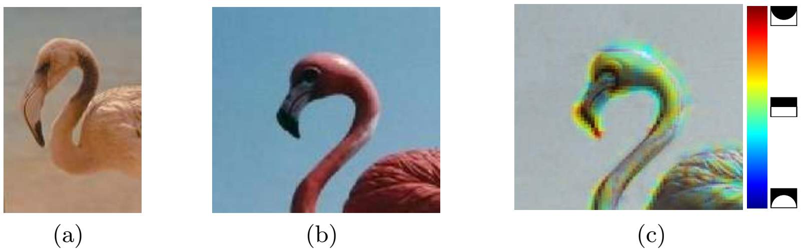
(a) (b) (c)

Fig. 7. Two flamingo images from Caltech are shown in (a) and (b). The typical shape of the flamingos' beaks and necks enables reliable recognition when using curvature. The scalar curvature of (b) is shown in (c). Note the high curvature at the beak's tip.

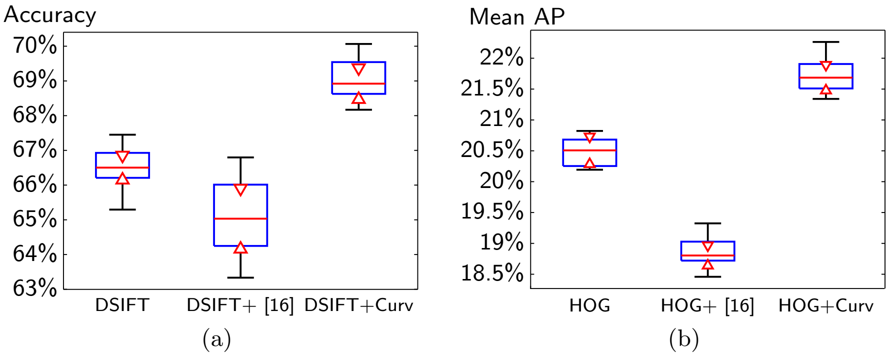

Fig. 8. Statistical results. The blue boxes show the inter-quartile range, the red line is the median. The total extent of the results is given by the black whiskers, while the red triangles enclose the range of significance[2].
(a) Results of Caltech101 evaluation. The average accuracies are 66.5% for SIFT only, 65.1% for SIFT with boundary curvature, and 69.1% for SIFT and vector curvature. This is an incrase of 2.6% over the baseline using our approach. There is some variance but the differences are clearly significant.
(b) Summarized results of Pascal VOC 2007 evaluation. When changing the random seeds, the mean AP over all classes varies by about 1%. Still, the boxplots show that the mean AP distributions do not overlap and that they differ significantly. Table 1 shows a per-class performance overview.

diagonal. The average increase in accuracy of our method over the baseline is 2.6%. While the curvature features from [16] work well on some images, they have problems on other images, where the extraction of good boundary segments is harder. On average, the accuracy is slightly lower than the baseline.

The proposed vector curvature improves performance on almost all classes. For some classes it is particularly useful: on the classes *pyramid*, *flamingo head* and *electric guitar*, the curvature extension outperforms baseline SIFT by a large margin. Objects belonging to these classes expose sharp corners, which help recognition. Figure 7 shows two sample images. As shown in Fig. 7(c), the flamingo exposes high curvature at the tip of its beak. Also the neck has a typical curvature. This information is largely ignored by orientation histograms.

4.2 Object Detection

For the evaluation in object detection, we trained filter masks for all 20 object classes used in the Pascal VOC 2007 challenge [7]. For every class, we clustered the training examples in 3 aspect ratios, such that in total 60 filter masks were trained. The training was iterated and used random negatives for the first round.

[2] Two distributions differ significantly (by 95%), if the triangle intervals (i.e. notches) do not overlap (cf. MATLAB 2013a boxplot function)

	aeropl	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog
HOG	18.1	46.8	1.8	6.5	14.7	38.2	42.3	4.6	11.9	17.3	8.3	2.5
HOG+ [16]	22.1	39.9	3.4	5.9	11.6	33.6	39.1	7.7	9.8	11.5	8.3	5.1
HOG+Curv	20.9	47.3	2.7	7.4	15.6	40.9	44.8	4.3	13.7	18.0	6.5	2.8

	horse	mbike	person	plant	sheep	sofa	train	tvmon	$\overline{\text{avg}}$	Δ
HOG	38.2	33.0	27.6	5.2	13.0	20.2	31.8	27.9	20.50	0
HOG+ [16]	36.7	26.6	26.8	2.6	8.4	18.4	29.1	30.3	18.85	-1.65%
HOG+Curv	41.8	35.5	29.7	6.0	15.0	19.9	33.0	28.9	21.74	+1.24%

Table 1. Pascal VOC 2007 Object Detection Challenge. Average results of 10 iterations of training and testing. While our approach improves the overall result, the approach from [16] on average decreases the performance over the baseline.

Retraining was based on the hard-negatives. Testing was done in a multi-scale sliding window fashion.

As in image classification, we compared gradient-only features (HOG in this case) to the curvature augmented versions, which add either our vector curvature or the boundary curvature from [16]. For Pascal we also ran the whole training and testing 10 times with different random seeds for sampling negative examples.

Results. While a combination of HOG and our vector curvature increases the mean AP by 1.24% over the baseline, the method from [16] combined with HOG decreases the mean AP by 1.65%. To show statistical significance, the results are summarized using boxplots in Fig. 8(b). A detailed per-class overview can be found in Table 1. The proposed curvature extension of HOG increases the AP for 18 of 20 classes.

As demonstrated in [16], boundary curvature performs well on datasets with clean boundaries, such as the ETHZ shape dataset [9]. The proposed vector curvature descriptor is more generally applicable as it does not fully depend on strong boundaries. Moreover, it includes important cues provided by the curvature direction.

The raw numbers already indicate the benefit of adding vector curvature to the overall descriptor. Some more distinct advantages can be seen when looking at actual samples. Figure 9 shows false-positives of the bus category. The top row lists the top ranked false-positives detected by HOG that did not occur when using curvature. The bottom row analogously shows the top ranked false-positives detected with curvature that did not occur with HOG. Note how the false-positives obtained by adding curvature all show buses and are only due to suboptimal bounding boxes or duplicate detections. This indicates that by adding curvature we obtain an additional performance gain that is not measured by the standard evaluation criterion.

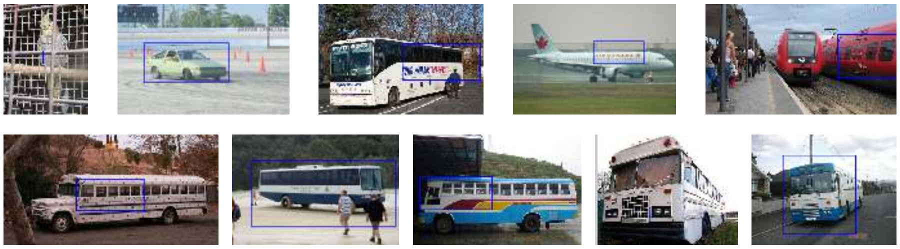

Fig. 9. By adding curvature we obtain an additional performance gain that is not measured by the standard evaluation criterion: The figure shows exclusive top false-positives buses for HOG without (top row) and with curvature (bottom row). Note how the false-positives obtained by adding curvature all show buses and are only due to suboptimal bounding boxes or duplicate detections.

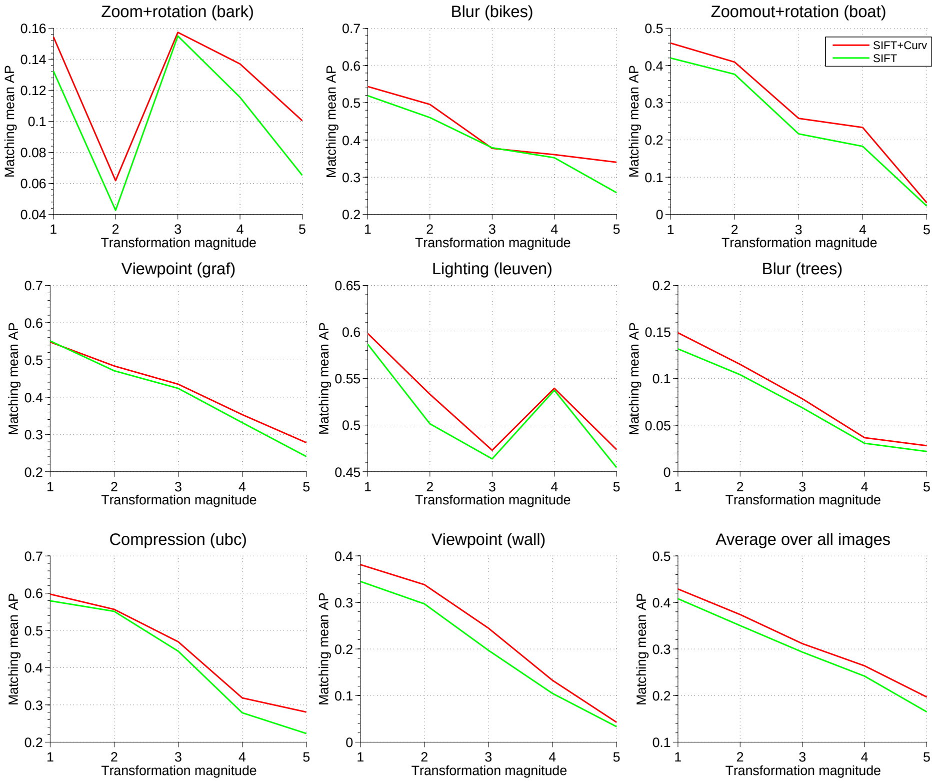

Fig. 10. Mean average precision on the Mikolajczyk dataset. The bottom right plot shows the average over the whole dataset. SIFT features (green) are consistently outperformed by the curvature expanded version (red).

4.3 Descriptor matching

In this experiment we evaluated the feature description performance in image matching tasks. We use the standard matching dataset by Mikolajczyk et al. [15], which contains different transformations such as zoom, rotation, blur and lighting changes. The dataset has previously been used for two different tasks: local interest point/region detection and local descriptor matching. We are interested in descriptor matching and intentionally disregard the detector performance by using the same region detector for both descriptors. We chose the MSER (maximally stable extremal regions) detector, because it is among the best detectors in [15]. The image patches of the detected elliptical regions were normalized to a uniform patch size and rotated to the dominant gradient orientation, which is the standard procedure. Given two images and their regions to be matched, we computed the descriptors of the normalized patches. All possible descriptor pairs were then ranked by their Euclidean distance. As in [15], a pair was considered to be a correct match if the overlap error of the region ellipses is less than 40% when transformed by the ground truth mapping. Subsequently, for each image pair, we computed a precision-recall graph and its average precision (AP).

Results. The dataset contains 8 image categories with 5 image pairs each. The matching results for all 40 image pairs are given in Figure 10, while the last graph shows the average performance over all categories. The curvature augmented version performs consistently better than SIFT alone.

5 Conclusions

We have presented an elegant expansion of the popular SIFT and HOG descriptors by curvature histograms. The new curvature descriptor can be computed as easily as SIFT and HOG. We have demonstrated the general applicability of the descriptor on very diverse tasks: image classification in a bag-of-features style, object detection with sliding windows, and descriptor matching. The expansion of SIFT or HOG descriptors by curvature consistently increases performance on all three tasks. This demonstrates that the benefits of curvature are not limited to certain datasets or object classes. The features are universal and can be used in various areas of visual recognition. Our source code will be made publicly available for research purposes.

Acknowledgments

The work was partially funded by the ERC Starting Grant VideoLearn.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI 33(5), 898–916 (May 2011)

2. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: ECCV. pp. 430–443. Springer-Verlag, Berlin, Heidelberg (2012)
3. Coeurjolly, D., Miguet, S., Tougne, L.: Discrete curvature based on osculating circle estimation. In: IWVF. pp. 303–312 (2001)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. pp. 886–893 (2005)
5. Deng, H., Zhang, W., Mortensen, E.N., Dietterich, T.G., Shapiro, L.G.: Principal curvature-based region detector for object recognition. In: CVPR (2007)
6. Eigenstetter, A., Ommer, B.: Visual recognition using embedded feature selection for curvature self-similarity. In: NIPS. pp. 386–394 (2012)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI 32(9), 1627–1645 (2010)
9. Ferrari, V., Tuytelaars, T., Gool, L.J.V.: Object detection by contour segment networks. In: ECCV. pp. 14–28 (2006)
10. Han, J.H., Poston, T.: Chord-to-point distance accumulation and planar curvature: A new approach to discrete curvature. Pattern Recognition Letters 22(10), 1133–1144 (2001)
11. Kerautret, B., Lachaud, J.O.: Robust estimation of curvature along digital contours with global optimization. In: DGCI. LNCS, vol. 4992, pp. 334–345. Springer (Apr 2008)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. pp. 2169–2178 (2006)
13. Liu, H., Latecki, L.J., Liu, W., Bai, X.: Visual curvature. In: CVPR (2007)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (Nov 2004)
15. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.J.V.: A comparison of affine region detectors. IJCV 65(1-2), 43–72 (2005)
16. Monroy, A., Eigenstetter, A., Ommer, B.: Beyond straight lines - object detection using curvature. In: ICIP. pp. 3561–3564 (2011)
17. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge University Press (2006), http://books.google.de/books?id=LFzGwdi0KtsC
18. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)