Scale Invariance without Scale Selection

Iasonas Kokkinos
Department of Statistics, UCLA
jkokkin@stat.ucla.edu

Alan Yuille
Department of Statistics, UCLA
yuille@stat.ucla.edu *

Abstract

In this work we construct scale invariant descriptors (SIDs) without requiring the estimation of image scale; we thereby avoid scale selection which is often unreliable.

Our starting point is a combination of Log-Polar sampling and spatially-varying smoothing that converts image scalings and rotations into translations. Scale invariance can then be guaranteed by estimating the Fourier Transform Modulus (FTM) of the formed signal as the FTM is translation invariant.

We build our descriptors using phase, orientation and amplitude features that compactly capture the local image structure. Our results show that the constructed SIDs outperform state-of-the-art descriptors on standard datasets.

A main advantage of SIDs is that they are applicable to a broader range of image structures, such as edges, for which scale selection is unreliable. We demonstrate this by combining SIDs with contour segments and show that the performance of a boundary-based model is systematically improved on an object detection task.

1. Introduction

Local image descriptors evaluated at interest points have been very successful for many visual tasks related to object detection [21]. An important issue is how to deal with changes in image scale. Typically this is done in a two-stage process which first extracts a local estimate of the scale and then computes the descriptor based on an appropriately sized image patch. This strategy is limited in two respects. First, for most places in the image it is hard to obtain reliable scale estimates, with the exception of symmetric structures, such as blobs or ridges. However, we would not like to limit ourselves to the few structures for which scale estimation is reliable, as other structures, e.g. edges can be useful for object detection. Second, even if scale estimation is reliable, it does not necessary indicate the scale where the most useful appearance information resides, as shown in Fig. 1. Context is most informative and can only be incorporated by considering multiple scales.

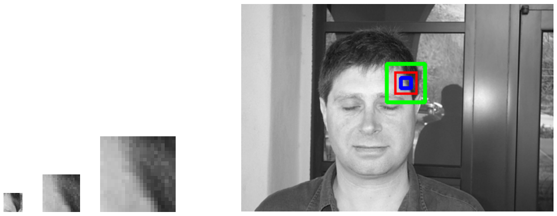


Figure 1. Our goal is to extract scale-invariant information around generic image structures where scale selection can be unreliable, e.g. edges. The local image structure that is used by most scale selection mechanisms is often not informative about the scale of the structure, which becomes apparent from the image context.

We propose a method to compute scale invariant descriptors (SIDs) that does not require scale selection. For this we use a combination of log-polar sampling with spatially varying filtering that converts image scalings and rotations into translations. Scale invariance is achieved by taking the Fourier Transform Modulus (FTM) of the transformed signals as the FTM is translation invariant. Our experiments show that SIDs outperform current descriptors when tested on standard datasets.

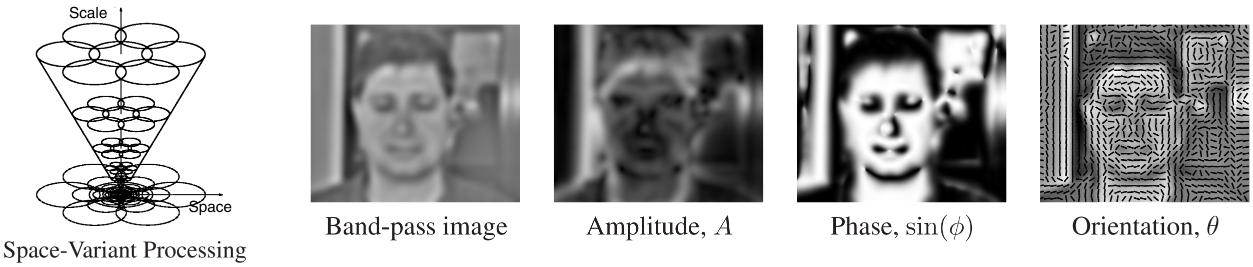
By freeing us from the need for scale selection, SIDs can be used in a broader setting, in conjunction with features such as edges and ridges [18]. Such features can be used to construct intuitive object representations as they are related to semantically meaningful structures, namely boundaries and symmetry axes. However, they have been limited since they do not come with scale estimates, so it has been hard to use them for scale-invariant detection. We address this by augmenting contour segments with SIDs, and use them in a flexible object model for scale-invariant object detection.

2. Previous Work

Image descriptors summarize image information around points of interest using low-dimensional feature vectors that are designed to be both distinctive and repeatable. Two seminal contributions have been SIFT descriptors [21] and Shape Contexts [26]; these have been followed by several

[*] This work was supported by NSF Grant 0413214

978-1-4244-2243-2/08/$25.00 ©2008 IEEE



Space-Variant Processing

Band-pass image

Amplitude, A

Phase, $\sin(\phi)$

Orientation, θ

Figure 2. Front-End Analysis. Left: We combine log polar sampling with a spatially increasing filter scale to guarantee scale invariance (taken from [20]). Right: Features extracted from a band-pass filtered image using the Monogenic Signal.

extensions and refinements, such as Geometric Blur [3, 4], PCA-SIFT [12] and GLOH [23]. Please see [23] for an up-to-date review, comparisons and more extensive references; we will compare in detail our descriptor to related ones in the following sections.

Typically, a two stage approach is used to deal with changes in image scale. First, a front end system is used to estimate local image scale, e.g. by using a scale-adapted differential operator [19, 18]. Then, the estimated scale is used to adapt the descriptor. As argued in the introduction, this two-stage approach is often problematic and the only work known to us that addresses this is [6]. There the authors obtain stable measures of scale along edges by extracting descriptors at multiple scales and choosing the most stable one; however this requires an iterative, two-stage front-end procedure for each interest point.

3. Front-End Analysis

3.1. Image Sampling with the Log-Polar Transform

In order to design scale-invariant descriptors we exploit the fact that the log polar transform

$$\hat{I}(r, u) = I(x_0 - \sigma_0^r \cos(u), y_0 - \sigma_0^r \sin(u)) \tag{1}$$

converts rotations and scalings of the image I around (x_0, y_0) into translations of the transformed image $\hat{I}$; r and u are log-polar coordinates and σ_0 is a scaling constant. This transform is extensively used in image registration (see e.g. [31] and references therein), while in [27] it is argued that this logarithmic sampling of the image is similar to the sampling pattern of the human visual front-end. We apply this sampling strategy to our problem, by setting x_0, y_0 in (1) equal to the location of an interest-point, and consider the construction of a scale-invariant descriptor around it.

A practical concern is that directly sampling the image around each point is impractical, because we would need too many samples to avoid aliasing. Therefore we remove high frequency components by band-pass filtering the image before extracting features from it.

Further, as shown in Fig. 2, we use spatially varying filtering and sample the image (or its features) at a scale that is proportional to the distance from the center of the log-polar sampling grid. As we show in App. A, this guarantees that scaling the image only scales the features and does not distort them in any other way. This allows us to then use a sparse log-polar sampling of the image features and convert scalings/rotations into translations.

Comparing to other image descriptors that use a log-polar sampling strategy, in the GLOH descriptor of [23], the histogram of the image gradient at several orientations is computed by averaging the gradient within each compartment of a log polar grid. Such descriptors can be redundant, since typically a single orientation is locally dominant. Further, a front-end detector is used to determine the descriptor's scale, which as mentioned can be problematic. The work of the authors in [3] is also closely related, as they increase the smoothing with the distance from the center. However their approach leads to distortions due to scale changes, while in our work we guarantee that apart from being translated, the signal does not get distorted.

3.2. Feature Extraction

Having described our sampling strategy, we now describe the features that are being sampled. Specifically, we compute the image orientation, phase and amplitude, as shown in Fig. 2, which largely capture local image structure. The phase of an image provides symmetry-related information, indicating whether the image looks locally like an edge ($\phi = 0$) or a peak/valley ($\phi = \pm\pi/2$). The amplitude A is a measure of feature strength (contrast), and its orientation θ indicates the dominant direction of image variation. To compute these we use the Monogenic signal of [7], as described below.

3.2.1 The Monogenic Signal

In order to estimate the amplitude and phase of an 1-D signal a well established method is based on the Analytic Signal, obtained via the Hilbert transform [11]. However the extension of the Analytic Signal to 2D had only been partial, until the introduction of the Monogenic Signal in [7].

Following [7], we obtain the local amplitude a, orientation θ and phase ϕ measurements of a 2D signal h by:

$$A = \sqrt{h^2 + h_x^2 + h_y^2},\ \theta = \tan^{-1}\frac{h_y}{h_x},\ \phi = \tan^{-1}\frac{h}{\sqrt{h_y^2 + h_x^2}}$$

$$h_{\{x,y\}} = \mathcal{F}^{-1}(\sqrt{-1}\frac{\omega_{\{x,y\}}}{\sqrt{\omega_x^2 + \omega_y^2}}H),$$

where $H = \mathcal{F}(h)$ is the 2D Fourier transform of h and ω_x, ω_y are horizontal/vertical frequencies. A simple implementation can be found at [16].

Apart from being theoretically sound, the Monogenic Signal is also efficient; prior to the generalization of the Hilbert transform to 2D, earlier approaches would first preprocess the image with a set of orientation-selective filters [25, 10, 22, 24, 13] and then essentially treat the output of each filter as an 1-D signal. Instead, the Monogenic Signal only requires filtering with a single band-pass filter with no orientational preference.

4. Scale Invariant Descriptor Construction

Having laid out the ideas underlying our front-end processing, we now describe our method for computing SIDs, depicted as a block diagram in Fig. 3.

We initially band-pass filter the input image at multiple scales σ, and estimate at each scale the amplitude, phase, and orientation features $A(x,y,\sigma), \phi(x,y,\sigma), \theta(x,y,\sigma)$ with the Monogenic signal. We sample A, ϕ, θ around each point with a log-polar grid, taking measurements from larger scales as we move further from (x_0, y_0):

$$f_d(r,u) = f(x_0 - c\sigma_0^r \cos(u), y_0 - c\sigma_0^r \sin(u), \sigma_0^r), \quad (2)$$

where f_d is the sampled version of f, $f = A, \phi$ or θ and c is a constant, which we set to 1.

We use A_d, ϕ_d, θ_d to construct the following functions:

$$A_d \cos(\phi_d), A_d \sin(\phi_d), A_d \cos(2\theta_d^*), A_d \sin(2\theta_d^*). \quad (3)$$

Edges and ridges are indicated by $A_d \cos(\phi_d)$ and $A_d \sin(\psi_d)$ respectively. To remove the dependence of the orientation estimate θ_d on image rotations we form $\theta_d^*(r,u) = \theta_d(r,u) - u$ and then multiply it by two to make orientations identical modulo π; we then use the functions $A_d \cos(2\theta_d^*), A_d \sin(2\theta_d^*)$ to capture image orientation.

Our sampling strategy guarantees that a scaling/rotation of the image becomes a vertical/horizontal translation of A_d, ϕ_d, θ_d, so this carries over to the computed features, as shown in Fig. 4. We eliminate variations due to these translations with the Fourier Transform Modulus (FTM) [5]: if $F(\omega_r, \omega_u)$ is the Fourier transform of $f(r,u)$, we have

$$f(r - r_0, u - u_0) \overset{\mathcal{F}}{\leftrightarrow} F(\omega_r, \omega_u)\exp\left(-j(\omega_r r_0 + \omega_u u_0)\right).$$

Taking the magnitude of the Fourier transform eliminates variation due to translation, as $|\exp(jx)| = 1$. Further, we eliminate multiplicative changes due to lightning by normalizing each FTM to have unit L_1 norm.

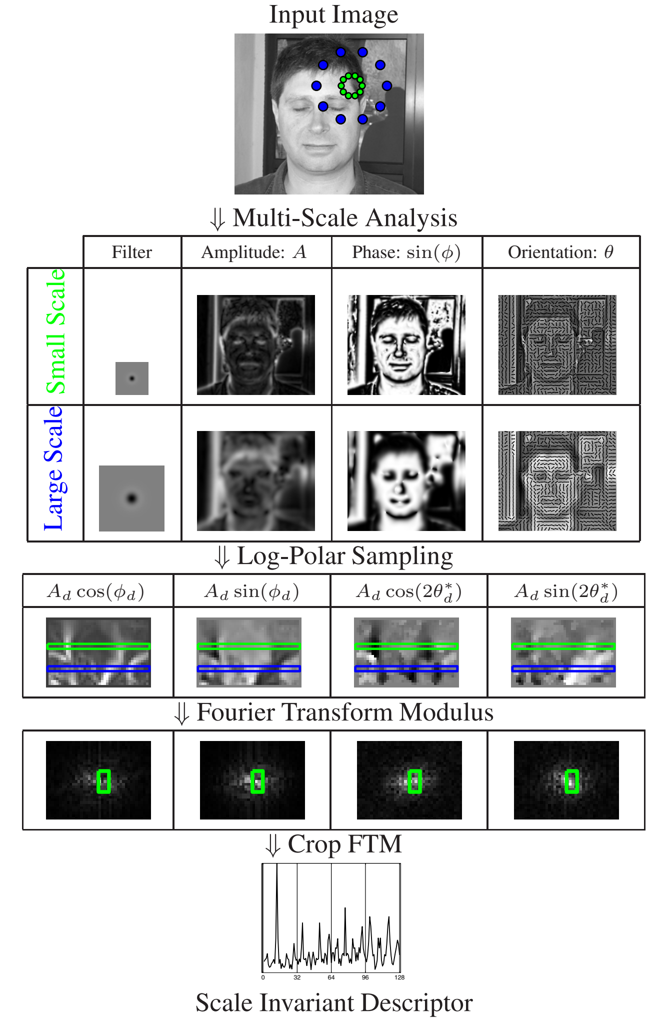
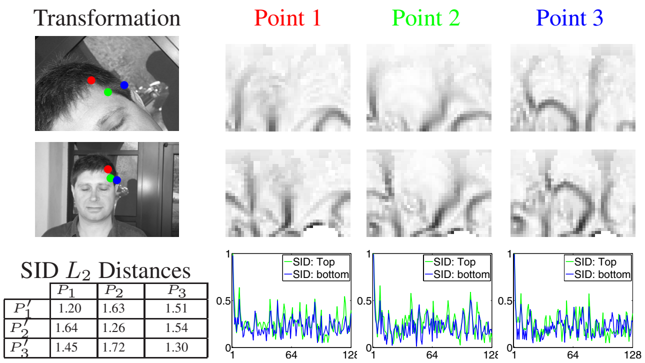


Figure 3. Descriptor Construction: The image is processed at multiple scales to obtain phase, amplitude and orientation estimates. These are sampled with a log-polar grid, using larger scale estimates as we move outwards - the points on the green/blue rings use estimates from the 'green/blue' scales. The sampled functions A_d, ψ_d, θ_d and the features built from these are defined in log-polar coordinates, which turn image scalings and rotations into translations. These are then eliminated by taking the Fourier transform modulus (FTM). We pick 32 high-energy components from each FTM and form a 128-dimensional descriptor.

We exploit the Fourier Transform's concentration of energy to keep a small set of high energy components lying at low frequencies; these contain most of the information about the signal, and are robust to noise. Further, as the FTM is symmetric, i.e. $|F(\omega_1,\omega_2)| = |F(-\omega_1,-\omega_2)|$ we only need to consider two quadrants of the FTM domain. We therefore ignore coefficients with negative horizontal frequency, and keep the FTM coefficients lying within a 4×8 box in the frequency domain, as shown in Fig. 3.



	P_1	P_2	P_3
P_1'	1.20	1.63	1.51
P_2'	1.64	1.26	1.54
P_3'	1.45	1.72	1.30

Figure 4. Demonstration of robustness to image scalings and rotations, with two images differing by a scale factor of 3 and an angle of $\pi/4$ (hardest angle). Top row: the $A_d \cos(\phi_d)$ functions computed on three different points are shifted versions of each other (the horizontal borders, $0, 2\pi$ are identical). Bottom row: the SIDs remain mostly intact by scaling, and thresholding their L_2 distances can give all correct matches without false positives.

Combining the 32-dimensional features extracted from the four signals of (3) we obtain our 128-dimensional *scale invariant descriptor* (SID).

In practice our descriptor is not perfectly invariant to scale changes, due to the limited number of scales considered. Therefore a change in image scale, e.g. a zoom, introduces new observations at fine scales and removes others at the coarse scales, as the top row of Fig. 4 shows. This can distort the FTM, and hence the SID. However, as Fig. 4 shows, for a scale change of size 3 our descriptor is robust to such variations. The three points considered can easily be confused: The green and blue points are on edges, so they look locally similar. The green and red points are on the same side of the face, so they have similar contexts. We observe that the distortions introduced by a scale change are not large enough to confound the point descriptors; empirically we observed that our descriptor is reliable for changes in image scale up to an order of 4.

Implementation Details: We use a band-pass filter with frequency response

$$F(\omega_x, \omega_y) = c_0 \sqrt{\omega_x^2 + \omega_y^2} \exp(-\sigma^2 \frac{\omega_x^2 + \omega_y^2}{c}), \quad c = 2,$$

whose scale in the time-domain is proportional to σ. c_0 is a numerically estimated constant, so that each filter has unit L_1 norm in the time domain. We use filters with $\sigma = 2(1.14)^n, n = 0, \ldots, 30$ and sample their features at points lying at distances from the center equal to $r = 2(1.14)^n, n = 0, \ldots 30$.

5. Descriptor Evaluation

We use the datasets provided in [23] to evaluate the performance of our descriptors. Using the code provided in [1] we compare to the SIFT [21] and GLOH descriptors which were reported in [23] to outperform most descriptors.

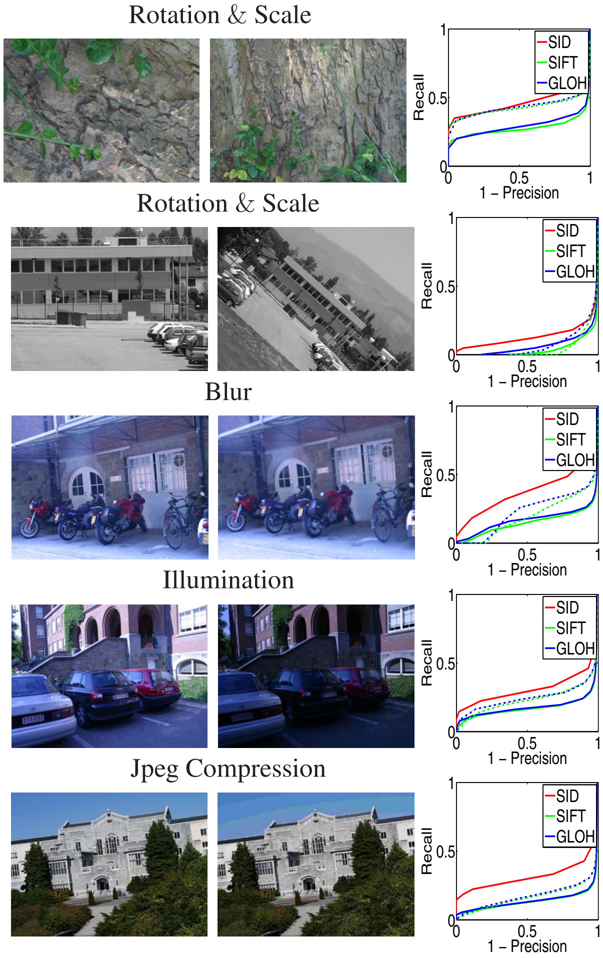


Figure 5. Precision-Recall curves for descriptors extracted around Hessian-Laplace interest points. The dashed lines are the performances of SIFT/GLOH after making up for misses due to orientation (see text for details). In most cases SIDs outperform SIFT/GLOH, even after this correction. The last three rows demonstrate robustness to other transformations.

The descriptors are computed around interest points extracted from two images of an identical scene; these images differ either by camera pose or a degrading transformation, like blurring, or jpeg compression. Ground truth correspondences are then used to evaluate the correspondences established based on thresholding the descriptor similiarities.

We observe that our descriptor systematically outperforms the SIFT/GLOH descriptors; we obtained similar results on most of the 40 images we experimented with using both the Harris- and Hessian- Laplace detectors, and considering both the similarity and nearest-neighbor criteria of [23]; we do not work with the affine-invariant detectors as we do not cover affine invariance. We have been concerned about whether this difference is an artifact of the different treatment of scale and orientation by SID and SIFT/GLOH;

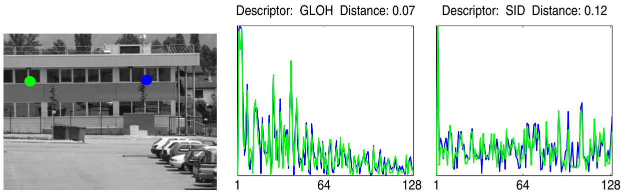


Figure 6. Using context to disambiguate appearance: At the scale at which GLOH descriptors are computed, the two corners are almost identical (we omit SIFT as it is worse). By using context, the SID tells the two corners apart, as is shown by its normalized distance, $\sum(D_1 - D_2)^2/\sum D_1^2$ that is almost twice that of GLOH.

we detail our evaluation settings below, but note that with the original code of [23] we obtain similar results.

We attribute this difference in performance to two reasons: first, SID captures contrast, orientation and symmetry in a continuous manner, without using histograms like SIFT/GLOH. This allows SID to put more information in a descriptor of the same dimensionality. Second, SID uses image context, which allows it to disambiguate among points and find more accurate matches. As shown in Fig. 6, the SID can tell apart two corners whose appearance is locally identical: as soon as we 'zoom out' structures like the tree make it possible to tell them apart.

Even in this setting, where scale estimation is reliable, we typically obtained better performance compared to scale dependent descriptors on most images that we experimented with. This suggests using SIDs also for other image areas where no scale estimates can be obtained.

On the downside, computing SIDs for an 512×765 image having 2414 points takes 35 sec.s on a 1.6 Ghz PC, which is slower than SIFTs. We can however use e.g. a pyramidal implementation to speed-up computation.

Evaluation Settings: We need to deal with points that are close in space but lie at different scales, e.g. corners where the Harris-Laplace detector responds at different scales to points that lie close. Such points are considered as different by the software of [1], but are found to be identical by SIDs which do not rely on the detector's scale. This eronneously penalizes SIDs as providing false positives, since they are actually matching the same point.

The inverse happens with orientation: points at the same location but with different orientations are considered as identical by [1], but result in different SIFT/GLOH descriptors. This penalizes these descriptors for not matching points that could not possibly be matched by construction. However, our descriptor can match such points, as it does not rely on the detector's orientation.

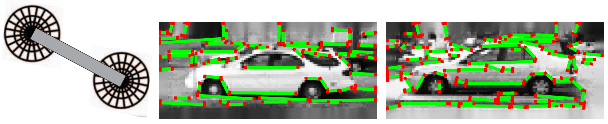
As our goal is to compare the information carried by the different descriptors on equal grounds, we resolve this situation as follows: First, we remove from our evaluation points that are less than 10 pixels apart, but are declared as different by [1] due to differing scales. For this we add a large constant to their descriptor distances, making sure they will not get matched. This reduces the number of false positives for SIDs, that will otherwise match points irrespective of scale. Second, if the SIFT/GLOH descriptors of two points are matched for a certain threshold, we force the descriptors computed for other orientations at these points to be matched as well. This typically increases the recall rate of SIFT/GLOH by an order of two, as is seen be comparing the solid and dashed lines in Fig. 5. Finally, we do not consider points lying within 30 pixels from the image boundary, as the descriptors there are distorted by missing data.

6. Token-based Object Representation

We now turn to combining our descriptors with contour segments, which has been our initial motivation for computing SIDs. Contour segments have been used for object category detection e.g. in [28, 4, 9] and were shown to compare favorably to systems using interest points. However, it is hard to estimate the image scale at contour locations, so associating scale-invariant appearance information to contours has been problematic.

This is a problem which we address with SIDs, as they do not require scale estimation. Specifically, we represent the image by a set of 'sketch tokens' $\{\mathcal{T}_i : i = 1, ..., N\}$. These are computed from the image by the Lindeberg primal sketch [18], followed by a line breaking algorithm. Each token is a straight edge/ridge segment with a SID computed at both ends, as shown in Fig. 7(a). We describe a token by the locations $\mathbf{x}_S, \mathbf{x}_E$ of its start- and end- points and the SIDs $\mathcal{A}_S, \mathcal{A}_E$ extracted around them.

We use this representation to both learn the object model, and perform inference. The main problem that we encountered in doing this has been the contour fragmentation problem, shown in Fig. 7(b,c). Even though the images are similar, there is variability in the tokens due to a combination of factors such as line breaking, thresholding, sensor noise etc. As we describe below, our object representation allows us to deal with this problem.



(a) (b) (c)

Figure 7. (a) the sketch token obtained by a contour segment with an descriptor at either end. (b,c): Contour fragmentation problem: Similar images give different tokens.

6.1. Learning Procedure

Here we consider learning a model for the primal sketch tokens coming from an object category, e.g. cars or faces. We start by automatically registering 50 images belonging to the training sets of [2, 8] using the procedure reported in [15]. This deals with shape variation by aligning all images to a common, 'template' grid.

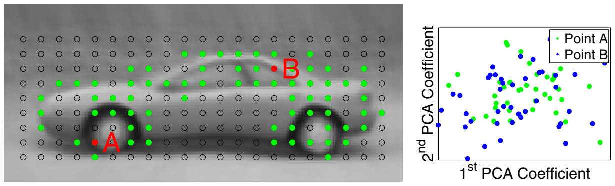


Figure 8. Left: Grid imposed on the template (black) and selected points combining frequency of appearance and consistency of the primal sketch (green/red). Right: Scatter plot of the first two PCA coefficients for the appearance descriptors extracted at the grid points labeled with red letters on the left. Please see in color.

To model tokens we first determine points on the car which can serve as their start and end points. For this, inspired by [30], we first define a regular grid, i.e. a discrete subset of locations. Next we quantize onto this grid the start- and end-point locations of the primal sketch tokens that we extract from the registered training images. We gather statistics for the transition probabilities between each pair of grid points, as shown in Fig. 9, by counting how often these grid points get linked by primal sketch tokens.

The resolution of the grid is important. A fine-scale grid will result in high complexity for detection and require much training data, while a coarse-scale grid will give a 'diffuse' model. Cross validation could be used to select an optimal grid size, but we use instead a solution which practically gave good results: we select a medium-scale grid, with points being 10 pixels apart in the horizontal and vertical dimensions and then prune out the grid points that rarely correspond to primal sketch tokens.

For pruning we use a criterion similar to the 'Google' criterion of [29]. At each grid point we estimate the quantity $m_i = p_i \sum_j p_{i,j} \log(p_{i,j})$ which combines: (a) the probability p_i of the grid point i being hit by the start-/end- point of a primal sketch token, and (b) the predictability (negative entropy) of its transition probability to another grid point. By thresholding this criterion we obtain the grid points shown in Fig. 8.

The object representation consists of the set of Grid-Point-Pairs (GPPs), $\mathcal{G}_p = (i_S, i_E)$ that start and end at the points that survive this pruning. Each such GPP has:

- Two distributions $P_{i_S}(.)$ and $P_{i_E}(.)$ for the descriptor features $\mathcal{A}_S$ and $\mathcal{A}_E$ at its end points. Instead of the full descriptor, we use its 20-dimensional projection on a PCA basis, constructed from background images.
- A distribution $P_\mathcal{G}(\mathbf{x}_S, \mathbf{x}_E|\mathbf{X}) = P_\mathcal{G}(\mathbf{x}_S|\mathbf{X})P_\mathcal{G}(\mathbf{x}_E|\mathbf{X})$ for the positions $\mathbf{x}_S, \mathbf{x}_E$ of its end points conditioned on the pose $\mathbf{X}$ of the object. These are estimated based on the deformation statistics, and are modeled by Gaussian distributions for computational convenience.

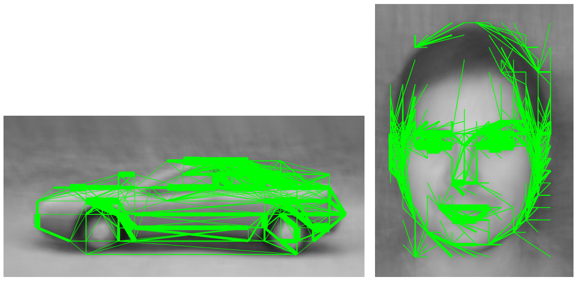


Figure 9. Sketch transition probabilities capturing the statistics of edge tokens for faces and cars. The width of the line connecting two points is proportional to the probability of a sketch token transiting from one grid point to another.

- A distribution for whether there is a primal sketch token in the image that connects i_S with i_E. This is a Bernoulli distribution, with probability of success $\pi_\mathcal{G}$ equal to the transition probabilities among nodes.

These distributions give a likelihood estimate for a primal sketch token $\mathcal{T}_i$ conditioned on a GPP $\mathcal{G}_p = (i_S, i_E)$ and the object pose, $\mathbf{X}$:

$$P(\mathcal{T}_i|\mathcal{G}_p, \mathbf{X}) = P_{i_S}(\mathcal{A}_S)P_{i_E}(\mathcal{A}_E)P_\mathcal{G}(\mathbf{x}_S, \mathbf{x}_E|\mathbf{X})\pi_\mathcal{G} \quad (4)$$

Finally, we learn separate background distributions $P_B(\mathcal{T}_i)$ for edge and ridge tokens using the background images of [8]. We fit the distribution of token lengths with an exponential distribution, $P(S) = a\exp(-aS)$ and a non-parametric distribution for token orientation is built using the embedding $\theta \rightarrow (\cos(\theta), \sin(\theta))$ of θ to R^2. The fore- and background distributions are not commensurate, since they model different aspects of the tokens. In principle we should use the Jacobian of the mapping from one representation to the other, but as we do not have analytic expressions we multiply the background distribution with a manually determined correcting factor.

Our grid-based model decouples three major sources of variation: bottom-up detection artifacts due to fragmentation, appearance and shape variation. This makes it easier to both gather statistics in order to train the model, and utilize it during object detection. For example, the appearance distributions are learned independently of the sketch tokens, using all 50 training images.

7. Scale Invariant Detection

We now describe how we use the sketch tokens for object detection (localization) in conjunction with our object representation.

7.1. Sketch Pruning

We first reduce the number of sketch tokens that are potential matches to the object, by pruning tokens whose appearances are unlikely to have been generated by the object.

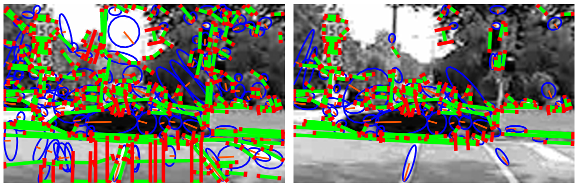


Figure 10. Left: Edge (green) and Ridge (blue) tokens extracted from our bottom-up system. Right: pruned features based on their descriptor likelihoods under the model hypothesis.

Specifically, for each token we compute the likelihoods of its endpoint SIDs, $\mathcal{A}_S, \mathcal{A}_E$ under the appearance distributions $P_i(\mathcal{A})$ constructed at each object grid point i during training. If $\max_i P_i(\mathcal{A})$ falls below a conservative threshold for either $\mathcal{A}_S$ or $\mathcal{A}_E$, then we reject this token. Such tokens typically appear in the background clutter, and we find empirically that this stage has few false negatives, as shown in Fig. 10.

7.2. Voting-based Object Detection

We perform object detection by relating the primal sketch tokens in the image to the model GPPs. For this we use a voting method similar to [17, 14] and estimate the object pose $\mathbf{X}$ by gathering evidence from all possible correspondences between tokens and GPPs. This evidence for an object at location X conditioned on matching token $\mathcal{T}_i$ to the GPP $\mathcal{G}_p$ equals the log-likelihood ratio:

$$R(\mathbf{X}|\mathcal{T}_i, \mathcal{G}_p) = \log \frac{P(\mathcal{T}_i|\mathcal{G}_p, \mathbf{X})}{P_B(\mathcal{T}_i)}. \tag{5}$$

Apart from an additive term that depends on the descriptor likelihoods, this expression can be written as a sum of two terms, obtained by breaking the pose likelihood, $\log(P(X_S, X_E|\mathbf{x})) = \log(P(X_S|\mathbf{X})) + \log(P(X_E|\mathbf{X}))$. These terms are pre-computed for 7 object scales spanning the range between $[.3, 3]$ and are used during detection to efficiently vote for objects. Apart from feature extraction, the detection algorithm thereby typically takes less than 10 seconds.

We allow each token $\mathcal{T}_i$ to vote for all GPP's for which: (i) the difference between the GPP orientation and the token falls below a threshold ($\pi/8$) and (ii) the likelihood of both token descriptors with respect to the GPP distribution is above a conservative threshold. These conditions reduce the number of correspondences between tokens and GPP's by over two orders of magnitude. For example, they reduce 6400 possible correspondences between GPP's and tokens to approximately 30 possible correspondence per token.

Finally, by considering correspondences for a range of relative scales between GPPs and tokens spanning $[.3, 3]$ we are able to deal with scale variation. Depending on the token/GPP relative scales, we vote for different object scales using different expressions for $\log(P(X_S, X_E|\mathbf{x}))$, as mentioned above.

The total votes for a pose $\mathbf{X}$ are computed by:

$$V(\mathbf{X}) = \sum \max(R(\mathbf{X}|\mathcal{T}_i, \mathcal{G}_p), 0), \tag{6}$$

where the sum is over all allowed correspondences between the GPP's and the primal sketch tokens. We obtain a small set of poses by nonmaximum suppression, as in [2].

7.3. Detection Results

We explore the importance of the appearance cue on the Caltech face database [8] and the UIUC car dataset [2]. Specifically, we examine the merit of the information carried by SIDs by removing the appearance-related part of the expression in (5). As is shown in Fig. 11 this has a direct impact on detection performance. The detection system that does not use appearance information behaves similarly to the system of [2], while by introducing appearance it gets closer to the current state-of-the-art. Similar results hold for multi-scale cars, where the EER (67.8) may be below the current state of the art [24], but the introduction of appearance information has resulted in a clear improvement in performance. As these results provide clear proof-of-concept for the usefulness of SIDs for contour-based detection, we are currently working on incorporating SIDs in more elaborate detection systems.

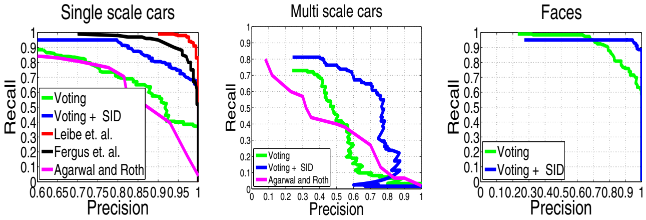


Figure 11. Precision-Recall curves for object detection: Introducing appearance information from SIDs systematically improves the performance of a baseline detector, and makes it comparable to appearance-based detection systems.

8. Conclusion

This paper describes a method to construct scale invariant descriptors without requiring scale selection. Our experimental results demonstrate that these descriptors compare favorably to current state-of-the-art alternatives when evaluated on standard datasets, while at the same time being applicable to image structures such as edges.

This has allowed us to introduce scale-invariant appearance information in contour-based detection, by using a SID to describe the image appearance at the start and end points of edge/ridge segments. We have explored the usefulness of this information for an object detection task, where we observed systematic improvements in performance compared to a voting scheme using only contour information.

In future work we intend to explore the merits of introducing appearance in more elaborate detection systems and develop efficient algorithms for feature extraction.

A. Space Variant Filtering for Scale Invariance

Consider a one-dimensional signal $I(x)$, and a feature function $F(x)$ obtained by filtering $I(x)$ with a kernel g_σ, where $\sigma = ax$:

$$F(x) = \int_t I(t) g_{ax}(x-t)dt = \int_t I(t)\frac{1}{ax} g_1(\frac{x-t}{ax})dt \qquad (7)$$

$g_1(x)$ is a unit-norm L_1 kernel at scale 1 and $\frac{1}{ax}$ guarantees that g_{ax} has unit norm. If I' is a scaled version of I, i.e. $I'(t\sigma_0) = I(t)$, its feature function, F' at $x\sigma_0$ will be:

$$\begin{aligned} F'(x\sigma_0) &= \int_t I'(t)\frac{1}{ax\sigma_0} g_1(\frac{x\sigma_0 - t}{ax\sigma_0})dt = \\ &\overset{t'\sigma_0=t}{=} \int I'(t'\sigma_0)\frac{1}{ax\sigma_0} g_1(\frac{x\sigma_0 - t'\sigma_0}{ax\sigma_0})\sigma_0 dt' = \\ &= \int I(t')\frac{1}{ax} g_1(\frac{x-t'}{ax})dt' = F(x) \qquad (8) \end{aligned}$$

which proves that the features F' of the transformed image are scaled version of the features F of the original image. By a logarithmic sampling we can thus turn image scalings into translations. We can show in the same way in 2D that $F'(x\sigma_0, y\sigma_0) = F(x, y)$.

References

[1] Descriptor and Detection Evaluation Software, Visual Geometry Group web-page. Available from: <http://www.robots.ox.ac.uk/∼vgg/software/>. 4, 5

[2] S. Agrawal and D. Roth. Learning a Sparse Representation for Object Detection. In *ECCV*, volume 4, pages 113–130, 2002. 6, 7

[3] A. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, 2001. 2

[4] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005. 2, 5

[5] D. Casasent and D. Psaltis. Position, rotation, and scale invariant optical correlation. *Applied Optics*, 15(7):258–261, 1976. 3

[6] G. Dorkó and C. Schmid. Maximally stable local description for scale selection. In *ECCV*, 2006. 2

[7] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Trans. on Signal Processing*, 2001. 2

[8] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *CVPR*, 2003. 6, 7

[9] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. Technical report, INRIA, 2006. 5

[10] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publisher, 1995. 3

[11] S. Hahn. *Hilbert Transforms in Signal Processing*. Artech House, 1996. 2

[12] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *CVPR*, 2004. 2

[13] I. Kokkinos, G. Evangelopoulos, and P. Maragos. Texture Analysis and Segmentation Using Modulation Features, Generative Models and Weighted Curve Evolution. *IEEE Trans. PAMI*, 2008. To appear. 3

[14] I. Kokkinos, P. Maragos, and A. Yuille. Bottom-Up and Top-Down Object Detection Using Primal Sketch Features and Graphical Models. In *CVPR*, 2006. 7

[15] I. Kokkinos and A. Yuille. Unsupervised Learning of Object Deformation Models. In *ICCV*, 2007. 6

[16] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 3

[17] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *ECCV*, 2004. SLCV workshop. 7

[18] T. Lindeberg. Edge Detection and Ridge Detection with Automatic Scale Selection. *IJCV*, 30(2), 1998. 1, 2, 5

[19] T. Lindeberg. Feature Detection with Automatic Scale Selection. *IJCV*, 30(2), 1998. 2

[20] T. Lindeberg and L. Florack. Foveal scale-space and the linear increase of receptive feld size as a function of eccentricity. *Comp. Vis. and Im. Understanding*, 1996. 2

[21] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004. 1, 4

[22] P. Maragos and A. Bovik. Image Demodulation Using Multidimensional Energy Separation. *J. Opt. Soc. Amer. (A)*, 12(9):1867–1876, 1995. 3

[23] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 2005. 2, 4, 5

[24] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. 2006. 3, 7

[25] P. Perona and J. Malik. Detecting and Localizing Edges Composed of Steps, Peaks and Roofs. In *ICCV*, pages 52–57, 1990. 3

[26] S. Belongie and J. Malik and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 2002. 1

[27] E. L. Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977. 2

[28] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning. In *ICCV*, 2005. 5

[29] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 6

[30] L. Wiskott and C. Malsburg. A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes. *IJPRAI*, 7(4):934–948, 1993. 6

[31] S. Zokai and G. Wolberg. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. 14(10), 2005. 2