# Scale Invariant Detection and Tracking of Elongated Structures

Amaury Nègre, James L. Crowley, Christian Laugier

# Scale Invariant Detection and Tracking of Elongated Structures

Amaury Nègre[1], James L. Crowley[1], and Christian Laugier[1]

INRIA Grenoble Rhones Alpes Research Centre
Montbonnot, France
firstname.lastname@inrialpes.fr

**Abstract.** This paper describes a method for the detection of tracking of elongated structures that is robust under changes of scale and orientation. This method is based on extending the concept of scale invariant natural interest points to include elongated ridge structures. An operator is proposed that directly detects ridge points and provides an estimation of their elongation and orientation. A tracking process is used to follow elongated features over time and to robustly observe changes in scale and orientation. Changes in scale are used to directly estimate time to contact. Experimental results demonstrate that the method works well in cluttered scenes that are typical of urban environments.

## 1 Introduction

Scale invariant image descriptions have been studied since the late 1970s. Burt [1] proposed a rapid algorithm to compute a multi-scale Laplacian pyramid. Crowley [2] proposed a scale invariant algorithm for the Laplacian pyramid and showed that such representations could be used to compute scale invariant image descriptions composed of peaks (scale invariant interest points) and ridges. Lowe adopted scale invariant peaks in the Laplacian Pyramid as the basis for a Scale Invariant Feature Transform (SIFT) [6]. This operation is now widely used in image matching, tracking and object categorization. Despite the widespread success of the SIFT operator, problems remain. For many real world objects, the detected feature points are not centered on the "physical" object they represent. More importantly in the presence of elongated shapes, multiple interest points will appear along the central axis of the objects and the position of such points along the object principle axis can be highly unstable.

In this article, we generalize the notion of scale invariant interest points to include elongated "ridge" lines. We develop a multi-scale version of the ridge-line detector proposed by [9], and show that such structures may be tracked with a particle filter to estimate the motion of an object. We demonstrate that the rate of change of scale of a ridge may be used to directly estimate the time to contact for an object in the scene. Such a method may be used for visual motion estimation and obstacle detection in cluttered urban environments.

## 2 Scale invariant ridge detector

### 2.1 Algorithm description

The SIFT operator uses the scale at which a peak in a Laplacian is a local maximum to determine the most appropriate scale for local image description. This maximum is referred to as the characteristic scale. We note in passing that such a characteristic scale is not limited to peaks in the Laplacian but can be determined at most points in an image [3]. A precise estimate of the characteristic scale can be obtained at any image point by interpolating the Laplacian values over a range of scale using a 3rd order interpolation function.

Two problems can arise with the use of scale invariant feature points

1. The Laplacian exhibits local extrema in the center of object but also on the edges, where intrinsic scale is not meaningful;
2. In the case of elongated object such as pedestrians or light poles, the position of detected points may be unstable along the axis of the object.

These phenomena can give rise to spurious or unstable feature points that degrade the reliability of a tracking or image matching.

To solve these problems, we extend the concept of natural interest points to include ridge lines. Such ridge lines naturally occur along the central axis of any elongated object, at a scale that corresponds to the visual width. Just as natural image points can be detected and tracked, so can natural ridge lines. Unlike interest points, ridge lines can be used to directly provide local orientation without the need for local histograms of gradients.

The detection algorithm operates as follows:

1. For each pixel of the Laplacian pyramid we first remove the edge response by eliminating pixels where the ratio between the Laplacian and gradient values is greater than a threshold.
2. We then determine the ridge direction using a local Hessian [9]. This direction is normal to the principal curvature direction given by the eigen-vector associated to the biggest eigen-value of the Hessian matrix $\mathcal{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial xy} \\ \frac{\partial^2 f}{\partial xy} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$ where $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial xy}$, $\frac{\partial^2 f}{\partial y^2}$ are the second derivatives of the Image.
3. For each pixel we search the length $l$ that maximizes the following score function:

$$\begin{aligned} S(X, l, \boldsymbol{u}) = \sum_{k=0..l} & (|L(X + k \cdot \boldsymbol{u_X})| + |L(X - k \cdot \boldsymbol{u_X})| \\ & - 2 \cdot |L(X + k \cdot \boldsymbol{u_X}) - L(X - k \cdot \boldsymbol{u_X})| \\ & - 2 \cdot L_{min}) \end{aligned} \tag{1}$$

Where $L(X)$ is the Laplacian value at pixel $X$ and $\boldsymbol{u}$ is the previously computed direction. This score is high when many pixels along the line have a high

Laplacian ($|L(X+k\cdot \boldsymbol{u_X})|+|L(X-k\cdot \boldsymbol{u_X})|$), and is maximal at the center of the object (as the term $|L(X+k\cdot \boldsymbol{u_X})-L(X-k\cdot \boldsymbol{u_X})|$ increases near contrast boundaries). The term $L_{min}$ makes the score decrease when $k$ is greater than the object's characteristic length. It should be noted that the cost of the score function is only proportional to the maximal searched length.

4. We then search local maxima of the previous score in the pyramid to obtain best segments.

When the feature is not a perfect line, the local direction of the principal curvature is not strictly aligned with the feature. The score can then be improved by scanning in neighboring directions.

A typical example of ridge segments detected in an image of an urban environment is shown in Figure 2.1. The detected segments are represented by red ellipses, the main axis of the ellipses represents the segment's half-edges, and the width of the ellipses represents the object scales. The interesting point is that all detected segments are centered on the object they represent and the detected scales fit very well object's size.

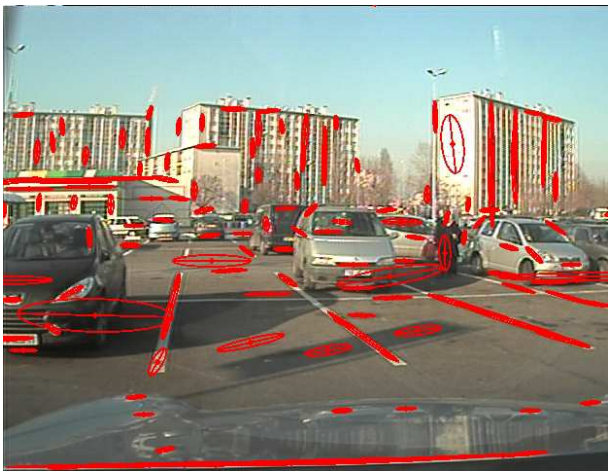


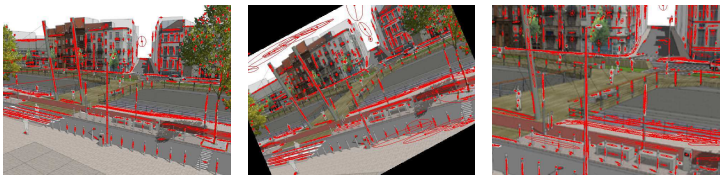
**Fig. 1.** Scale invariant segment detector in an urban environment, the detected segments are represented by red ellipses.

### 2.2 Performance evaluation

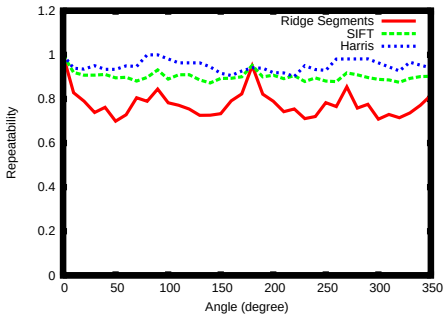
We used the repeatability criterion as described in [8] to evaluate the stability of this method for ridge segments detection to various image transformations. The repeatability criterion is determined as the proportion of image features that can be matched when an image is subjected to a known transformation. For ridge segments,

correspondence is determined using the the center position, orientation and scale of the ridge segment. The maximal accepted matching distance (the validation gate) is proportional to the scale of the ridge segment.

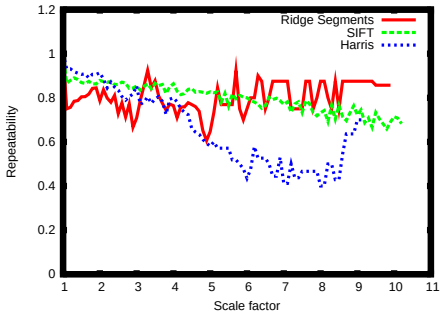
We compared the the repeatability of ridge segments to interest points detected using a Harris detector [5] computed at different scales as well as the SIFT detector [6]. The results are shown in Figure 2. In this experiment we observe that the ridge segments are not as much robust as rotation change, but actually the low performance comes from the localization error that can be high along the ridge direction. As expected, the change of scale doesn't affect the detector's performances.



(a) Original test image and detected segments. (b) Ridges in rotated test image. (c) Ridges in Scaled test image.




(d) Repeatability with respect to rotation angle.




(e) Repeatability with respect to scale factor.

**Fig. 2.** Repeatability of Harris, SIFT and Ridge-Segment detector with respect to rotation and scale change. We can see that the segment detector obtain moderate score for rotation changes but gives similar results than the SIFT detector with change of scale.

# 3 Estimating time-to-contact using the change of scale in ridge segments

## 3.1 Introduction of the Time-to-conctact

The time to contact "$\tau$" (also called time-to-collision or time-to-crash) can provide important information for the detection and avoidance of obstacles. Time to contact can be seen as directly determining the distance between two objects in a termporal space. It has been shown in [7] that the time to contact between the camera and a visible obstacle can be computed in the image space by measuring the variation of the characteristic scale.

Let "$s$" represent the characteristic scale of an image feature. In this case, $\tau$ can be approximated by :

$$\tau = \frac{s}{\frac{\partial s}{\partial t}} \tag{2}$$

The difficulty with this method is to identify and measure the change of scale of an image feature in the video sequence. The characteristic scale of a ridge segment provides exactly such a measure. Tracking ridge segments makes it possible to estimate both the visual motion of a feature and the time to contact.

## 3.2 Tracking Ridge Segments

We use the score function, described in 1, as a measure function for tracking ridge segments. Because this function is non-Gaussian and non linear, we have employed a particle filter for the tracking process [4]. The particle filter uses a set of samples (particles) to represent a probability distribution resulting for a Bayesian filtering process. Formally, if we let $X_t$ be the state of the target at time $t$, $Z_t$ be the observation of the target at time $t$, then the goal of the particle filter is to estimate :

$$P(X_t|Z_{t_1}...Z_{t_n})$$

To this end, the filter relies on two models : (1) the observation model $P(Z_t \mid X_t)$ that predicts a target observation, and (2) a displacement model $P(X_{t_k} \mid X_{t_{k-1}})$ that predicts the motion of the particles.

Target tracking is implemented as follows: we use a set of particles to characterize the target's position and the target's speed in the image coded by three vectors, each vith 3 dimensions :

- $\boldsymbol{c}$ : the segment's center position in the 3D Scale-Space;
- $\boldsymbol{v}$ : the segment's center speed ;
- $\boldsymbol{r}$ : the 3D vector between the center and an extremity (called "half-edge" in the following).

$$X = \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{v} \\ \boldsymbol{r} \end{pmatrix}$$

When a target is identified in an image for the first time, all particles are initialized around the detected position and with a zero speed. Next, each camera's image ($Z_t$) is used as an observation to update the particle filter. The observation model takes into account the score function described in Equation 1 :

$$P(Z_t|X_t)) \propto S(c_t, ||\boldsymbol{r}_t||, \frac{\boldsymbol{r}}{||\boldsymbol{r}_t||})$$

In between two images, we consider that the target's center is subjected to a Gaussian acceleration $\boldsymbol{a}$ and that the half-edge can also be affected by a Gaussian noise $\boldsymbol{n}$. Each particle is then updated using the following model :

$$X_{t+\Delta t} = \begin{pmatrix} \boldsymbol{c_{t+\Delta t}} \\ \boldsymbol{v_{t+\Delta t}} \\ \boldsymbol{r_{t+\Delta t}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c_t} + (\boldsymbol{v_t} + \boldsymbol{a} \cdot \Delta t) \cdot \Delta t \\ \boldsymbol{v_t} + \boldsymbol{a} \cdot \Delta t \\ \boldsymbol{r_t} + \boldsymbol{n} \cdot \Delta t \end{pmatrix}$$

The output of the particle filter is a probabilistic estimation of the target state. In practice, to obtain a single state of the target, we compute the average pose of all particles weighted by their probabilities. It can be noted that the tracking is automatically ended either when the target get out the image frame or when the observation probability will decrease. As we track many segment at same time, we also add a mechanism to fuse differents segments that converge.

The global schema of the detection and tracking of scale invariant segments is shown on Figure 3.

## 4 Experimental results

### 4.1 Simple tracking

To demonstrate the utility of this method for tracking, we consider a simple video sequence with a black rectangle printed on a white sheet. The tracking result is shown on Figure 4(a). We can see that the estimated segment (red ellipse) is centered on the black rectangle, but that the length does not precisely fit the rectangle's length. None the less, the rectangle scale (height) is precisely approximated. To evaluate this precision, the height of the rectangle has been measured by hand in each of these images and used to estimate the actual time-to-contact. The results are plotted in Figure 4(b)(c). As expected the two curves fit very well.

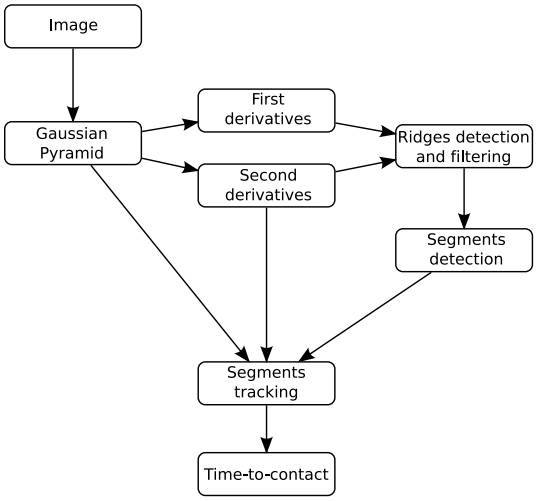
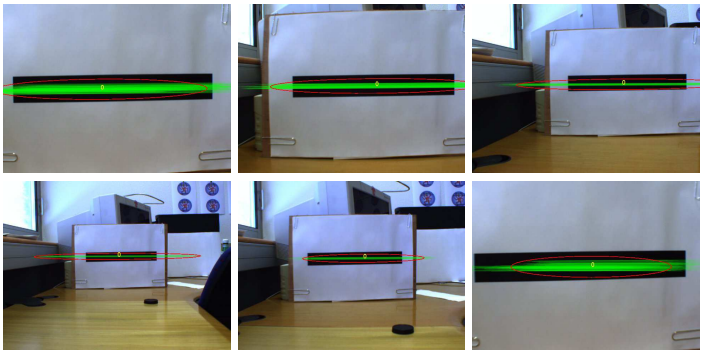
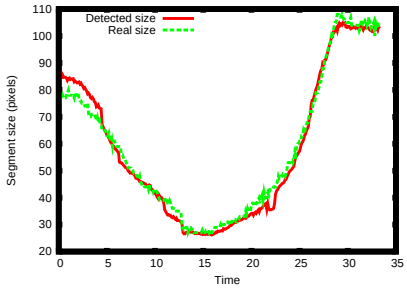


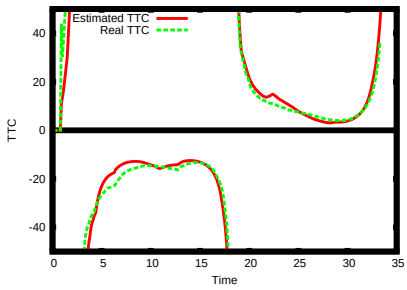

**Fig. 3.** Global schema of scale invariant segment detection and tracking.



(a) Image sequences.




(b) Scale of the tracked segment




(c) Time-To-Contact of the tracked segment

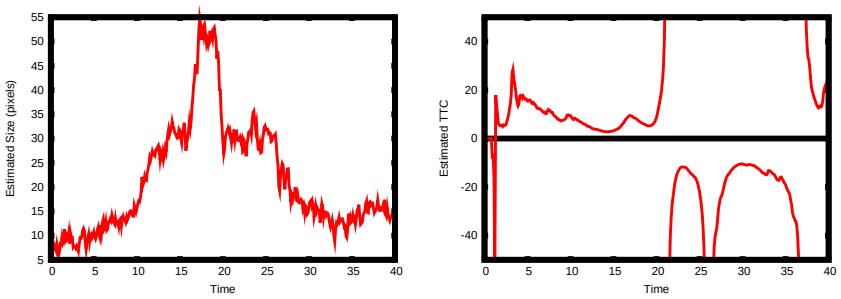
**Fig. 4.** Tracking of a simple black rectangle on a white sheet. In (a) we can see the set of particles particles (green lines) and the estimated state (red ellipse). The plot (b) represents the estimated scale of the tracked segment and a ground truth obtained by measuring by hand the rectangle height for each image. We can see in (c) that the estimated time-To-Contact fits very well the ground truth.

### 4.2 Tracking Ridge Segments in Urban Scenes

To demonstrate the efficiency of the scale invariant segment detector and tracking, we have tested our algorithm on real world sequences. To obtain real-time performance, we implemented the algorithm on a Graphic Processing Unit (GPU), which made it possible to detect and track simultaneously 64 segments in 640x480 images at 17 frames per second.

**Fig. 5.** Segment detection (right) and tracking (left) in an urban context. Red ellipses with blue axes represent the tracked segments, black curves represents segment's trajectories in the image.

Some results are shown in Figure 5, tracked segments are represented by red ellipse and each segment's trajectory in the image is drawn with a black curve. In this sequence, we noted that the segment's tracker locks well on to road lines, components of the cars, trees and poles. For a more detailed analysis, we plotted the evolution of scale of the pole located in the center right part of the image (Figure 6(a)). As the camera moves forward and backward, the object's size increases and decreases which is approved by the tracker. The approximated time-to-contact (Figure 6(b)) is also appreciated as it begins positive and decreases (which mean the obstacle is approaching) and next it swaps negative, which mean the obstacle is getting away).




(a) Evolution of scale for the segments localized in the white pole.

(b) Estimated Time-To-Contact of the white pole.

**Fig. 6.** (a) Evolution of the scale of the tracked segments localized on the white pole (center right of images). As the car moves forward and backward, the scale increases and decreased. The estimated Time-To-Contact (b) is well approximated : at the beginning it is positive and it decreases (which mean the obstacle is approaching) and next the TTC swaps negative (which means the obstacle is getting away).

## 5 Conclusion

Ridge segments are a natural complement to interest points. Ridge segments are particularly useful in scenes, such as urban environments that contain many elongated objects. In this paper, we have have described a method for detecting ridge segments and shown that such a method can provide good repeatability under changes in orientation and scale. We have demonstrated a tracking algorithm using a particle filter for tracking ridge segments, and have demonstrated that such tracking can be used to directly estimate time to contact for elongated objects observed from a moving camera. Because this detector uses the center of an object, (as opposed to an edge or corner) it can provide improved stability over large changes in viewing angle for elongated objects.

## References

1. P. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 9(4):532–540, 1983.
2. J. L. Crowley and A. C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–170, 1984.
3. J.L. Crowley D. Hall and V. Colin de Verdière. View invariant object recognition using coloured receptive fields. *Machine GRAPHICS and VISION*, 9(2):341–352, 2000.
4. A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer, 2001.
5. C. Harris and M. Stephens. A combined corner and edge detector. pages 189–192, Manchester, 1988.
6. D. G. Lowe. Object recognition from local scale-invariant feature. In *International Conference on Computer Vision*, pages 1150–1157, 1999.
7. A. Negre, C. Braillon, J.L. Crowley, and C. Laugier. Real-time time-to-collision from variation of intrinsic scale. In *Proc. of the Int. Symp. on Experimental Robotics*, Rio de Janeiro, Brazil, 2006.
8. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. pages 230–235, 1998.
9. T. T. H. Tran and A. Lux. A method for ridge extraction. *Asian Conference on Computer Vision*, 2004.