Dense Hough transforms on gray level images using multi-scale derivatives

Antoine Manzanera
ENSTA-ParisTech - U2IS
828, Boulevard des Maréchaux
91762 Palaiseau CEDEX
http://www.ensta-paristech.fr/~manzaner/

Abstract

The Hough transform for detecting parameterised shapes in images is still today mostly applied on binary images of contours or connected sets, which implies pre-processing of the images that may be costly and fragile. However the simple estimation of the spatial derivatives provides in every pixel the local geometry that can be used for dense voting processes, directly applied on the gray scale image. For lines and circles, the local information even allows to perform a direct 1-to-1 projection from the image to the parameter space, which greatly accelerates the accumulation process. In this paper we advocate the use of direct detection on gray scale images by combining Hough transform and multi-scale derivatives. We present the algorithms and discuss their results in the case of analytical shapes for order one (lines), and two (circles), and then we present the generalised Hough transform based on quantised derivatives for detecting arbitrary (non-analytical) shapes.

1 Introduction

Since its introduction in one of the first applications of computer vision [1], the Hough transform has rapidly turned into a classical tool for detecting parameterised shapes in images [2, 3]. In its original form, it is applied on binary images of contours, which implies pre-processing of the images. Then, the transform on binary images is performed using either of the two classical dual approaches: (i) the many-to-1 projection, which picks n-tuples of points from the binary image and select the unique corresponding points in the n-dimensional parameter space, and (ii) the 1-to-many back-projection, which, for every point of the binary image, draws the corresponding (n-1)-manifold in the parameter space.

Remarkably, the interest for Hough transforms remained strong and many variations have been proposed until recently [4, 5]. But it is also remarkable that the proposed algorithms generally follow the original form in the sense that the projection -or voting process- is performed sparsely on contour portions or salient points, using 1-to-many projection (most often), or (generally decimated) many-to-1 projection. We believe that there is a fundamental interest in performing a dense projection, i.e. allowing every pixel to vote, and this can be done directly on the gray level image by estimating the (multi-scale) spatial derivatives. Although more pixels are voting, these methods must be much faster, first because there is no pre-processing (other than the computation of the derivatives), and second because -in the case of parameterised shapes- one can perform a direct 1-to-1 projection from the image to the parameter space.

The idea of using the local derivatives to accelerate the Hough transform is not new, it has been proposed for lines by O'Gorman and Clowes [6] and for differentiable curves by Shapiro [7]. But those approaches were still used on curves and, to our knowledge, have not been densely applied on gray level images, probably because at the time they were proposed, the techniques for estimating the derivatives on discrete 2d functions, based on finite differences, were not considered precise enough. More recently, Valenti and Gevers [8] have proposed an efficient eye centre location algorithm based on a voting scheme using the scale space curvature estimation. However, they still reduced the voting pixels to a thin contour previously calculated.

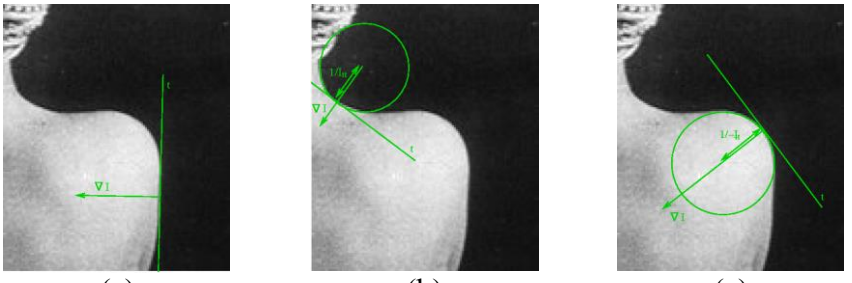
In this paper we advocate the use of dense Hough transform directly on the gray level signal using multi-scale derivatives, and weighting the votes by the strength of the derivative (gradient magnitude and Frobenius norm of the Hessian matrix typically). We present the complete algorithms and discuss their results in the case of analytical shapes for order one (lines), and two (circles), and we present the generalised Hough transform based on quantised derivatives for detecting arbitrary (non-analytical) shapes.

2 Analytical shapes

According to the scale space framework [9], the spatial derivatives are estimated in a digital image I relatively to a certain scale σ which represents the level of regularity, explicitly enforced by Gaussian smoothing:

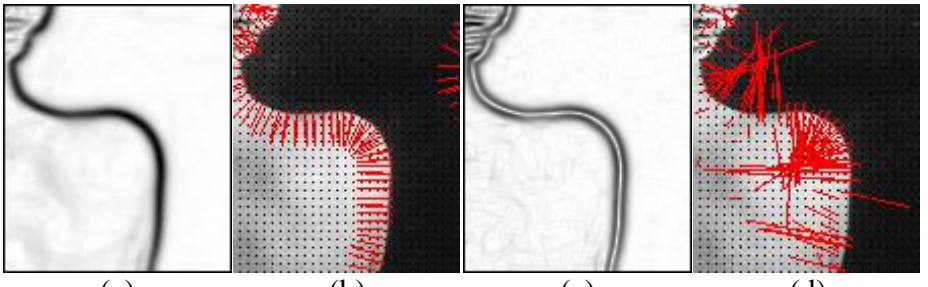
$$I^{\sigma}_{x^i y^j} = I \star \frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j} \ , \tag{1}$$

where $\star$ is the convolution, and G_σ the 2d Gaussian function of standard deviation σ. When working at a given scale, we will omit the σ superscript, and denote $\{I_x, I_y, I_{xx}, I_{xy}, I_{yy}\}$ the first and second order derivatives.



(a) (b) (c)

Figure 1: Direct detection from the spatial derivatives: Line from the gradient (a), and circle from the isophote positive (b) or negative (c) curvature.



(a) (b) (c) (d)

Figure 2: Voting weight at order 1: the norm of the gradient (a). Estimating the gradient direction (b), for pixels with weight over 10.0. Voting weight at order 2: the Frobenius norm of the Hessian matrix (c). Estimating the position of centre of osculating circle (d), for pixels with weight over 1.0. The scale estimation here is $\sigma = 2.0$.

2.1 First order: lines

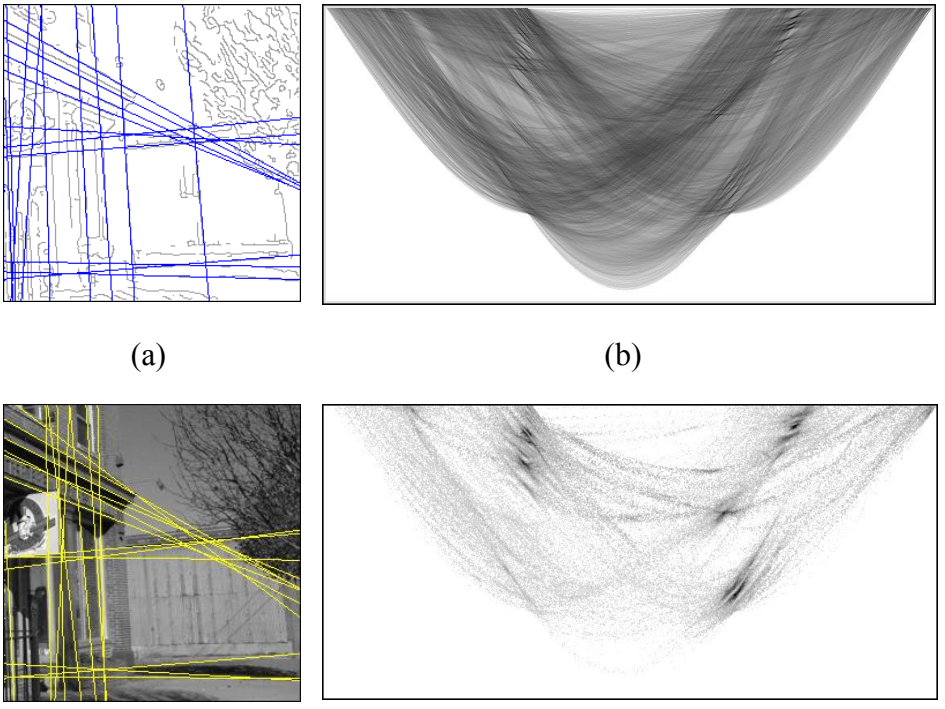
If $\nabla I = (I_x, I_y)$ is the estimated gradient vector, the value of the first derivative along any direction represented by unit vector $\mathbf{v}$ can also be estimated as $\mathbf{v}^T.\nabla I$. Thus the derivative along the direction orthogonal to the gradient is zero (isophote direction $\mathbf{t}$), and so if there is a line at this location, its orientation must the same as $\mathbf{t}$ (See Figure 1(a)). Now, to evaluate the significance of the location with respect to the presence of line, it is natural to use the strength of the first derivative, i.e. the norm of the gradient $||\nabla I|| = \sqrt{I_x^2 + I_y^2}$ (See Figure 2(a)), which must be normalised according to the scale σ [10], if this evaluation is done at different scales. The multi-scale voting weight at order 1 is then $\sigma||\nabla I||$. Following the classical (θ, ρ) parameterisation, where ρ is the distance between the line and the origin, and θ is the angle made by the normal to the line with the x axis, the complete algorithm is shown on table 1.

Table 1: 1-to-1 line Hough transform based on multi-scale gradient.

```
Gamma = function Hough_Lines (Image I)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            if ||∇I|| > 0:
                d ← p_x I_x + p_y I_y
                ρ ← |d| / ||∇I||
                if (I_x I_y < 0) and (I_y d > 0)
                    θ = π + arctan(I_y / I_x)
                else
                    θ = arctan(I_y / I_x)
                endif
                Gamma(ρ, θ) ← Gamma(ρ, θ) + σ||∇I||
            endif
        endfor
    endfor
end
```

Figure 3 shows the output of the transform, compared to the 1-to-many transform computed on a contour of the same image. The computation time of the 1-to-1 transform is in fact smaller than the

computation time of the contour image (which involves non local maxima suppression and hysteresis thresholding), while the complexity of the 1-to-many transform is one order of magnitude larger for each voting points, because every vote draws a sine curve in the parameter space. The 1-to-1 transform is naturally sparser, because, even if the number of voting pixels is significantly greater (all the pixels vote), only a few of them have significant vote, and more importantly, every pixel vote into one single point.



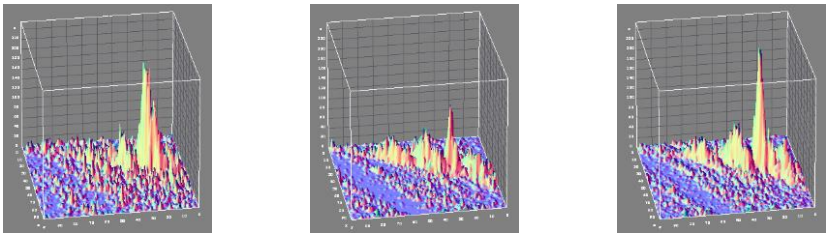
(a) (b)

(c) (d)

Figure 3: Top: 1-to-many line Hough transform (b) computed on the contour image (a): Canny algorithm with $\sigma = 1.5$, hysteresis threshold with $t_1 = 1.0$ and $t_2 = 6.0$. The 20 best lines are overlaid in blue. Bottom: 1-to-1 line Hough transform (d) computed on the grey level image (c) using multi-scale gradient ($\sigma \in \{1.0, 2.0, 4.0\}$).The 20 best lines are overlaid in yellow.

The sparsity, which can be a difficulty in the detection of the local maxima used to select the best lines, can be moderated by interpolating the vote over neighbouring cells according to the quantisation or by

explicitly smoothing the transform. But more interestingly, the concurrent use of multiple scales is a benefit for the detection, as illustrated on Figure 4: the finer scales improve the localisation of the main peaks while the coarser scales reduce the influence of the spurious structures. For the results shown on Figure 3, a non-local-maxima deletion was performed in the parameter space, and an exclusion distance of $(\pm\frac{2\pi}{100}, \pm 8)$ was applied to find the best (θ, ρ).



(a) (b) (c)

Figure 4: Influence of the multi-scale: Topographic close-up around a maximum of the 1-to-1 line Hough transform, using 1 single fine scale (a): $\sigma = 1.0$, 1 single coarse scale (b): $\sigma = 4.0$ and 3 scales (c): $\sigma \in \{1.0, 2.0, 4.0\}$.

2.2 Second order: circles

If $H_I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$ is the estimated Hessian matrix, the value of the second derivative along any couples of direction represented by unit vectors $\mathbf{u}$ and $\mathbf{v}$ can be estimated as $\mathbf{u}^T H_I \mathbf{v}$. When $\mathbf{u} = \mathbf{v} = \mathbf{t}$, with $\mathbf{t}$ in the isophote direction, we get:

$$I_{tt} = \frac{I_{xx} I_y^2 - 2 I_{xy} I_x I_y + I_{yy} I_x^2}{||\nabla I||^3} \tag{2}$$

This is the second derivative in the direction of the isophote, that is, the estimation of the curvature, which represents the inverse radius of the osculating circle to the isophote curve. Then, if there is a circle at location $(\mathbf{p}_x, \mathbf{p}_y)$ with gradient ∇I and curvature $I_{tt} \neq 0$, the radius of this circle must be $r = \frac{1}{|I_{tt}|}$ and its centre must be $(\mathbf{p}_x, \mathbf{p}_y) - \frac{\nabla I}{I_{tt} ||\nabla I||}$ (See Figure 1 (b) and (c)). Again, we can evaluate the significance of the location with respect to the presence of circle by using the strength of the second derivative, i.e. the Frobenius norm of the Hessian matrix $||H_I||_F = \sqrt{I_{xx}^2 + 2 I_{xy}^2 + I_{yy}^2}$ (See Figure 2 (c) and (d)), which must

be normalised by σ^2 if using different scales. See Table 2 for the complete algorithm.

Table 2: 1-to-1 circle Hough transform based on multi-scale gradient and curvature.

```
Gamma = function Hough_Circles (Image I)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            H_I = ( I_xx^σ(p)  I_xy^σ(p) )
                  ( I_xy^σ(p)  I_yy^σ(p) )
            if ||H_I||_F > 0:
                κ ← I_xx I_y^2 − 2 I_xy I_x I_y + I_yy I_x^2
                r ← ||∇I||^3 / κ
                (c_x, c_y) = (p_x, p_y) − ∇I||∇I||^2 / κ
                Gamma(r, c_x, c_y) ← Gamma(r, c_x, c_y) + σ^2 ||H_I||_F
            endif
        endfor
    endfor
end
```

Figure 5 shows results of this algorithm, compared to the 1-to-many Hough transform performed on the contour of the same image. The computation time of the 1-to-1 transform is still smaller than the computation time of the contour image, whereas the computation complexity of the 1-to-many transform is 2 order of magnitude more per voting points because every point draws a surface (cone) in the 3d parameter space. The sparsity of the 1-to-1 transform is still more visible than for the lines, because of a larger parameter space. The detection of maxima is then more challenging. For the results shown in Figure 5, we have applied a 3d recursive exponential smoothing filter ($\gamma = 2.0$) in the Hough space, followed by a non-local-maxima deletion and an exclusion distance of $(\pm 1, \pm 3, \pm 3)$ for the detection of the best $(r, \mathbf{c}_x, \mathbf{c}_y)$.

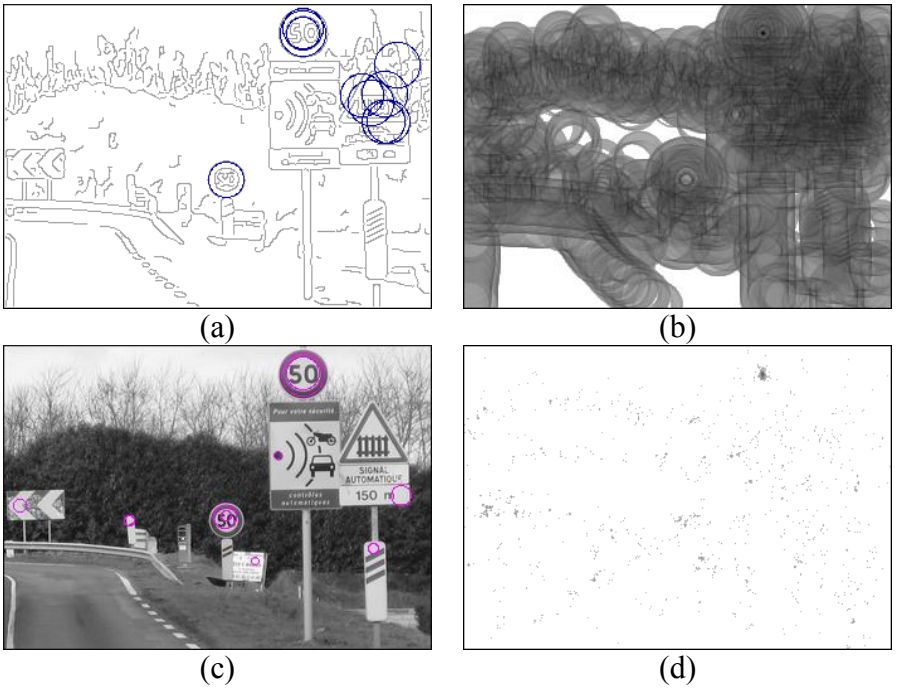


Figure 5: Top: One plane ($\rho = 19$) from the 1-to-many circle Hough transform (b) computed on the contour image (a): same parameters as Figure 3. The 7 best circles are overlaid in blue. Bottom: The same plane from the 1-to-1 line Hough transform (d) computed on the grey level image (c) using multi-scale curvature ($\sigma \in \{1.0, 2.0, 4.0\}$). The 7 best circles are overlaid in magenta.

3 Non analytical shapes

The direct calculation of the Hough transform on the gray scale image using multi-scale derivatives can also be used for the generalised Hough transform to detect arbitrary objects. In the classical approach [3], the arbitrary shape is a closed contour indexed by the local orientation. Since then, many variations on implicit shape models have been proposed. For example Leibe *et al* [4] use a collection of interest points instead of a contour, and index every point using visual codebook obtained by clustering.

In voting based representation, it is clearly important to have a significant number of voting points. In this sense, we believe that the use of the whole image instead of a collection of contour or interest points is a benefit. Table 3 describes the algorithm used to create the representation (the R-table) of a shape template T which is simply a

gray scale image. Every pixel of the template is indexed by a contrast invariant derivative of order 1: the argument of the gradient, and of order 2: the curvature. These derivatives are quantised to limit the size of the R-table, and the curvature is bounded to $[-1, 1]$ (the high curvatures are merged). Every new entry in the R-table add a new element to the list corresponding to the calculated index, which contains the relative coordinates of the voting point with respect to the centre of the template, and also, in conformity with the previous section, the voting weight of the point, corresponding to the magnitude of the gradient or to the Frobenius norm of the Hessian matrix according to the order of the index. Likewise, several scales of estimation can be used, at the cost of multiplying the number of R-tables. Obviously, the size of the R-tables can be reduced by eliminating the entries whose weight is considered too small.

Table 3: R-Tables at order 2 calculated on a gray scale template T.

$$
\begin{array}{l}
\mathrm{RT} = \textbf{function}\ \mathrm{Create\_R{-}Table}\ (\mathrm{Template\ T}, \mathrm{scale}\ \sigma) \\
\quad \textbf{forall}\ \mathrm{pixel}\ (\mathbf{p}_x, \mathbf{p}_y) \in \{0, w_T\} \times \{0, h_T\} \\
\qquad \nabla I \leftarrow (I_x^\sigma(\mathbf{p}), I_y^\sigma(\mathbf{p})) \\
\qquad H_I = \left(\begin{smallmatrix} I_{xx}^\sigma(\mathbf{p}) & I_{xy}^\sigma(\mathbf{p}) \\ I_{xy}^\sigma(\mathbf{p}) & I_{yy}^\sigma(\mathbf{p}) \end{smallmatrix} \right) \\
\qquad \textbf{if}\ \ ||\nabla I|| > 0 \\
\qquad\quad \alpha = \arctan \frac{I_y}{I_x} \\
\qquad\quad \xi \leftarrow \frac{I_{xx} I_y^2 - 2 I_{xy} I_x I_y + I_{yy} I_x^2}{||\nabla I||^3} \\
\qquad\quad \textbf{if}\ \ \xi < -1\ \textbf{then}\ \xi \leftarrow -1 \\
\qquad\quad \textbf{else if}\ \ \xi > 1\ \textbf{then}\ \xi \leftarrow 1 \\
\qquad\quad \mathrm{RT}(1, \sigma, \alpha) \leftarrow \mathrm{RT}(1, \sigma, \alpha) \cup (\frac{w_T}{2} - \mathbf{p}_x, \frac{h_T}{2} - \mathbf{p}_y, ||\nabla I||) \\
\qquad\quad \mathrm{RT}(2, \sigma, \xi) \leftarrow \mathrm{RT}(2, \sigma, \xi) \cup (\frac{w_T}{2} - \mathbf{p}_x, \frac{h_T}{2} - \mathbf{p}_y, ||H_I||_F) \\
\qquad \textbf{endif} \\
\quad \textbf{endfor} \\
\textbf{end}
\end{array}
$$

The off-line calculated R-Table is then used for online detection using the generalised Hough transform shown in Table 4, which is basically the same as the classical algorithm, except that every pixel vote according to its orientation and curvature indexes, and its votes are weighted as indicated in the R-table.

Table 4: Generalised Hough transform on gray scale image at order 2.

```
Gamma = function Hough_General (Image I,R–Table RT)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            H_I = ( I_xx^σ(p)  I_xy^σ(p) )
                  ( I_xy^σ(p)  I_yy^σ(p) )
            if ||∇I|| > 0
                α = arctan I_y / I_x
                ξ ← (I_xx I_y^2 − 2 I_xy I_x I_y + I_yy I_x^2) / ||∇I||^3
                if ξ < −1 then ξ ← −1
                else if ξ > 1 then ξ ← 1
                forall (δ_x, δ_y, ω) ∈ RT(1, σ, α)
                    Gamma(p_x + δ_x, p_y + δ_y) ← Gamma(p_x + δ_x, p_y + δ_y) + ω
                endfor
                forall (δ_x, δ_y, ω) ∈ RT(2, σ, ξ)
                    Gamma(p_x + δ_x, p_y + δ_y) ← Gamma(p_x + δ_x, p_y + δ_y) + ω
                endfor
            endif
        endfor
    endfor
end
```

One example of construction of the prototype is shown on Figure 6. In this example, one single image template is used, and 4 R-Tables are constructed, corresponding to 2 orders and 2 scales of estimation. The corresponding labels (gradient orientation and curvature) are quantised to 30 values which form the number of indexes of the R-tables. The calculation of the general Hough transform is shown on Figure 7 on a composite image of side viewed cars (All images are taken from the image database of UIUC for car detection [11]). For selecting the best detections, an exclusion distance corresponding to the quarter of the template sizes was used. What can be seen from this experiments is that the general Hough transform calculated from gray scale derivatives keeps the good properties of the transform on contours: invariance to contrast changes, robustness to occlusions, while being faster to compute and less sensitive to poor contrast in the detection, because every pixel is voting according to the significance of its counterpart in the template.

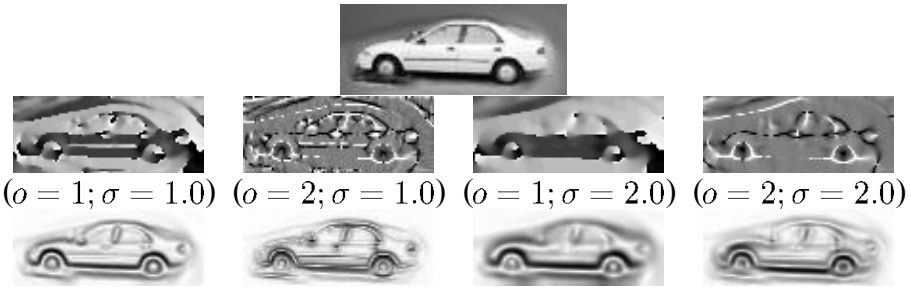


Figure 6: Object template (top). Labels used as indexes of the R-tables (middle line), at order 1 and 2, for two scales, and their associated weights (bottom line).

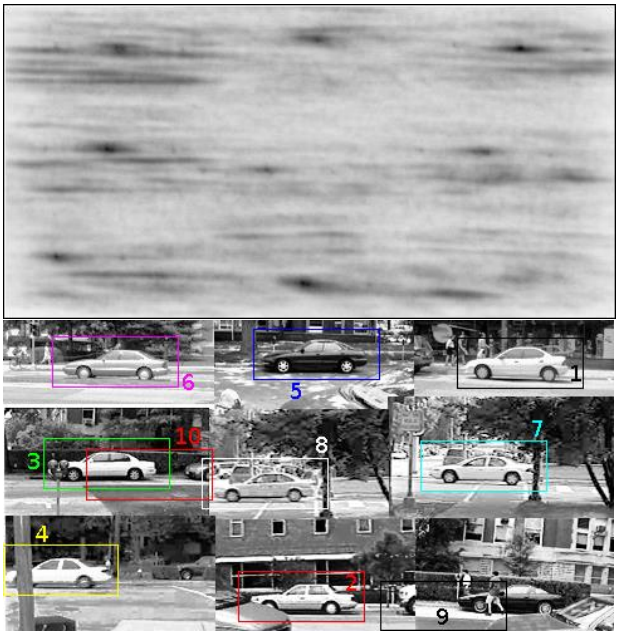


Figure 7: General Hough transform calculated on a composite image of side viewed cars, using the R-tables obtained from the template of Figure 6. The 10 best detections are shown as overlaid rectangles, numbered by order of detection.

4 Concluding remarks

Our purpose in this paper was to convince that the dense Hough transform on gray level images using multi-scale derivatives is interesting both in terms of robustness (thanks to a higher number of

weighted votes), and computational efficiency (thanks to lighter pre-processing and 1-to-1 vote). It seems hard to design a systematic evaluation to decide more objectively when dense derivatives should be preferred to sparse contours, because it hardly relies on the quality of the contour. Naturally, it can be said that contours will work better for finding curves made of discontinuous structures (e.g. finding alignments in plant fields). In the general case, a more thorough validation is needed to evaluate the influence of the chosen scales and weights. We are also planning to design a more general framework to apply the dense Hough transform for object detection based on multiple derivatives.

References

[1] Hough, P.: Machine analysis of bubble chamber pictures. In: Int. Conf. High Energy Accelerators and Instrumentation. (1959)

[2] Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Com. of the Association for Computing Machinery **15** (1972) 11–15

[3] Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition **13** (1981) 111–122

[4] Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision. (2004)

[5] Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. International Journal of Computer Vision **87** (2010)

[6] O'Gorman, F., Clowes, B.: Finding picture edges through collinearity of feature points. IEEE Trans. on Computers **C-25** (1976) 449–456

[7] Shapiro, S.: Feature space transforms for curve detection. Pattern Recognition **10** (1978) 129–143

[8] Valenti, R., Gevers, T.: Accurate eye center location and tracking using isophote curvature. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska (2008)

[9] Florack, L., Ter Haar Romeny, B., Viergever, M., Koenderink, J.: The Gaussian scale-space paradigm and the multiscale local jet. Int. J. of Computer Vision **18** (1996) 61–75

[10] Lindeberg, T.: Feature detection with automatic scale selection. Int. J. of Computer Vision **30** (1998) 77–116

[11] Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 1475–1490