Chapter 2
Review of Hough Transform for Line Detection

This chapter describes the basics of the Hough transform (HT). The terminology to be used in this text is defined in Sects. 2.1 and 2.2. The relationship of the HT (for lines) and the Radon and Fourier transforms is sketched out in Sect. 2.3. Section 2.4 reviews the most common existing line parameterizations used for line detection by the HT and gives a quick comparison of the important ones.

2.1 Hough Transform Basics

The HT [1] is sometimes understood not as a specific algorithm for object detection but as a wide class of algorithms that share a common structure. Princen et al. [2] formalized HT as a *hypothesis testing* process. The structure of HT when described as generically as possible is:

1. Some *evidence* is extracted from the input.
2. For each piece of the evidence, *accumulators* corresponding to the *hypotheses* that are supported by that evidence are incremented. Possible hypotheses are represented by an N-dimensional *parameter space* of accumulators.
3. Probable hypotheses are detected as peaks in the parameter space.

When the HT is used to detect objects in a raster image, the evidence can consist of edge or corner points, local features, and similar. The extracted evidence can be processed completely, or subsampled in a particular way.

The dimensionality of the parameter space is determined by the number of degrees of freedom (DoF) of the hypotheses. The parameter space can be—in the most straightforward manner—represented as an N-dimensional array. The size of the array is determined by the size of the interesting portion of the parameter space and the required precision. It is clear that this representation quickly becomes impractical, as the number of DoF increases. Alternatively, the parameter space can be represented by a linked list or another sparse representation [3].

A. Herout et al., *Real-Time Detection of Lines and Grids*,
SpringerBriefs in Computer Science, DOI: 10.1007/978-1-4471-4414-4_2,

The accumulation of the evidence can also be viewed as *voting*. Usually, one piece of evidence affects many hypotheses but only a small portion of the parameter space. The accumulator for each hypothesis is usually (and in the original Hough's form of the transformation) integer, i.e., each piece of the evidence either does or does not support a given hypothesis. However, several variations of HT use fractional accumulators, so that the hypotheses can be supported only partially. Such methods model the parameter space by fuzzy [4] or probabilistic [5] ways or perform some sort of antialiasing [6].

2.2 HT Formalization for 2D Curves

HT is typically used for detecting curves with an analytical description. In that case, the evidence are edge points detected in the input raster image. Such edge points can typically be detected by gradient operators such as Sobel or Prewitt. The hypotheses are the possible curves of a given class in the image. For example, a line has two and a circle has three DoF in a 2D space, but HT can be used for detection of objects such as hyperspheres or hyperplanes in spaces of arbitrary dimensionality.

If a family of (2D) curves is specified by an implicit function

$$f(x, y, p_1, \ldots, p_N) = 0, \tag{2.1}$$

where x and y are the image space coordinates and values $p_1, \ldots, p_N$ the parameter space coordinates, a point (x, y) that lies on a curve specifies a portion of the parameter space that describes all curves passing through this point. The parameter space coordinates and their mapping to the curve specify the *parameterization* of the curve. Curves typically have many possible parameterizations. It is always possible to use a different base if the curve parameters form a vector space, but often, many parameterizations fundamentally different in their nature are usable. Algorithm 1 describes the detection of a curve specified by Eq. (2.1).

In many cases, the edge points are detected by a detector that can estimate the edge orientation (e.g., the Sobel operator). The edge orientation can then be used to reduce the amount of curves that are plausible for the given edge point. Only the accumulators for curves whose tangent direction is close to the direction of the detected edge are incremented. This technique was used by O'Gorman and Clowes [7] to speed up line detection, but it can be used for a wider variety of curves. Various fast HT implementations described in Chaps. 3 and 6 utilize this technique for speeding up the accumulation process.

Shapes that do not have a simple analytical description can be detected by using the Generalized HT by Ballard [8]. In GHT, the object is not described by an equation but by a set of contour elements (edge points). Each contour element is described by its position with respect to the object reference point and the edge orientation. The parameter space has a dimension from two to four (object position, orientation,

Algorithm 1 Implicit curve detection by Hough Transform.

```
Input: Input image $I$, size of parameter space $H$
Output: Detected curves $C$
  $P_I = \{(x, y)|(x, y)$ are coordinates of a pixel in I$\}$
  $P_H = \{(p_1, \ldots, p_N)|(p_1, \ldots, p_N)$ are coordinates in H$\}$
  $H(x) \leftarrow 0, \forall x \in P_H$
  for all $x \in P_I$ do
    if at $x$ is an edge in $I$ then
      for all $\{p : p \in P_H, f(x, p) = 0\}$ do
        $H(p) \leftarrow H(p) + 1$
      end for
    end if
  end for
  $C = \{p|p \in P_H$, at $p$ is a high local maximum in $H\}$
```

and scale), but the representation of the detected object is complex even for simple shapes.

The shapes for the GHT can also be expressed by random forests [9]. This representation improves the performance for some kinds of objects. The Hough forests and the HT in general are the alternative of the object detection by a scanning window classifier.

2.3 Hough Transform for Lines and its Relationship to the Radon and Fourier Transform

The HT was originally developed for detecting lines and it is still popular in this particular area. Therefore, a large amount of different line parameterizations and various algorithmic modifications exist. Because 2D lines are rank 1 polynomial functions and therefore have two DoF, all line parameterizations are two dimensional. For now, a line will be specified by its normal vector $\mathbf{n} = (n_x, n_y)$ and the distance from the origin ϱ. Every point $\mathbf{p} = (p_x, p_y)$ that lies on the line fulfills

$$\mathbf{p} \cdot \mathbf{n} = \varrho. \tag{2.2}$$

HT for line detection is closely related to the *Radon Transform* [10, 11]. In the continuous case, these two transforms are identical. They differ in the discrete case, but some of the properties of the Radon transform are still applicable and useful. S.R. Deans [11] examined the properties of the Radon transform of a line segment, a pixel, and a generic curve.

The Radon Transform in an N-dimensional space transforms a function $f : \mathbb{R}^N \rightarrow \mathbb{R}$ onto its integrals over hyperplanes. A point $\mathbf{p}$ lies on a hyperplane

$(\mathbf{n}, \varrho)$ iff (if and only if) $\mathbf{p} \cdot \mathbf{n} = \varrho$. The function f must vanish to zero outside of some area around the origin. Otherwise, the value of the line integral would be infinite.

Equation (2.3) differs from the HT structure (Sect. 2.1) in the order of the iteration. The HT iterates over the affected hypotheses for every piece of the evidence so it is building the whole parameter space at once. The RT finds the amount of evidence for a given hypothesis by iterating over the evidence, so it tests every hypothesis separately.

$$\begin{aligned} \mathcal{R}^N[f](\mathbf{n}, \varrho) &= \int\limits_{\mathbb{R}^N} f(\mathbf{x})\delta(\mathbf{x} \cdot \mathbf{n} - \varrho)\mathrm{d}\mathbf{x} \\ &= \int\limits_{\mathbf{x}\cdot\mathbf{n}=\varrho} f(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \int\limits_{\mathbf{n}^{\perp}} f(\varrho\mathbf{n} + \mathbf{x})\mathrm{d}\mathbf{x} \end{aligned} \tag{2.3}$$

Through the relation to the Radon Transform, the HT is also related to the Fourier Transform. One- and N-dimensional versions of the Fourier transform are

$$\mathcal{F}^1[f](\xi) = \int\limits_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}\mathrm{d}x, \tag{2.4}$$

$$\mathcal{F}^N[f](\mathbf{w}) = \int\limits_{\mathbb{R}^N} f(\mathbf{x})e^{-2\pi i(\mathbf{x}\cdot\mathbf{w})}\mathrm{d}\mathbf{x}. \tag{2.5}$$

The transforms are related via the Projection-Slice theorem [10], where the Radon transform is the projection part. One-dimensional Fourier transform of the Radon transform is equal to a slice of the N-dimensional Fourier transform along the direction specified by the hyperplane's normal vector. Notation

$$\mathcal{R}^N[f](\mathbf{n}, \varrho) = \mathcal{R}^N_{\mathbf{n}}[f](\varrho) \tag{2.6}$$

will be used. The relation between Radon and Fourier transform is then

$$\mathcal{F}^1[\mathcal{R}^N_{\mathbf{n}}[f]](\xi) = \mathcal{F}^N[f](\xi\mathbf{n}). \tag{2.7}$$

Due to this relation, 2D convolution in the image space can be transformed to 1D convolution in the parameter space as

$$\mathcal{R}^N_{\mathbf{n}}[f * g](\varrho) = (\mathcal{R}^N_{\mathbf{n}}[f] * \mathcal{R}^N_{\mathbf{n}}[g])(\varrho). \tag{2.8}$$

The proof can be found in the work of Natterer [10]. This feature can be used not only for image filtering, but it also generalizes the technique Han et al. used for calculation of the α-cut in the Fuzzy HT [4].

Moving the filtering from the image to the parameter space can be beneficial not only because of the lower computational cost. For example, some filtering methods can interfere with the edge detection and moving the filtering after the voting step (to the voting space) allows for filtering without disturbing the edge detection phase.

2.4 Line Parameterizations

Usage of the HT for line parameterization requires a point transformation to the Hough space. The motivation for introducing new parameterizations is to find the optimal trade-off between requirements for the transformation. These requirements include preference of bounding Hough space, discretization with minimal aliasing errors, and uniform distribution of discretization error or intuitive mapping from original system to the Hough space. In general, the image of a point can be a curve of different shapes; for example circle, sinusoid curve, straight line, etc.

A subset favoring intuitive mapping and very fast line rasterization is the set of *point-to-line mappings* (PTLM). It contains such parameterizations where a point in the source image corresponds to a line in the Hough space and—naturally for the HT—a point in the Hough space represents a line in the x-y image space. PTLM were studied by Bhattacharya et al. [12], who proved that for a single PTLM, the Hough space must be infinite. However, for many PTLMs, a complementary PTLM can be found, so that the two mappings define two finite Hough spaces containing all lines possible in the bounded image. An example is the original HT (Eq. 2.14) which uses one Hough space for the vertical lines and the second one for the horizontal lines.

The following text will first present transformations based on 'classic' line parameterizations and the representations of these parameters. The second half belongs to the parameterizations using values of intersections of the line and a bounding object. At the end of this section, a comparison of the mentioned parameterizations is made, focusing on utilization of the Hough space and its main characteristics.

Slope–Intercept Parameterization

The first HT, introduced and patented by Paul Hough in 1962 [1], was based on the line equation in the slope-intercept form. However, the exact line equation is not mentioned in the patent itself. Commonly, the slope-intercept line equation has the form:

$$\ell : y = xm + b, \tag{2.9}$$

but the method used in the Hough's patent corresponds to the parameterization:

$$\ell : x = ym + b. \tag{2.10}$$

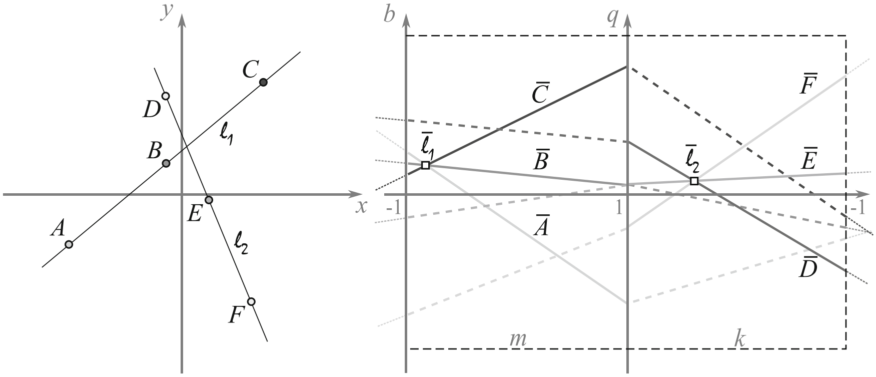

Fig. 2.1: Hough transform using *m–b* and *k–q* parameterizations of a line. *Left* input image; *right* corresponding Hough space

Using parameters m and b, all lines passing through a single point form a line in the Hough space, so it is a PTLM. As in every PTLM, the parameter space of all possible lines in a bounded input image is infinite [12]. Using a bounded parameter space requires at least two complementary spaces of parameters. In the case of the slope–intercept line equation, these spaces are, for example, the two based on these equations (Fig. 2.1):

$$\begin{aligned} y &= xm + b, \\ x &= yk + q. \end{aligned} \tag{2.11}$$

The slope–intercept parameterization is one of the parameterizations that allows for moving the image convolution to the parameter space as shown in Sect. 2.3. Contrary to Eq. (2.8), some scaling is necessary.

$$\mathcal{H}_m[f * g](b) = \frac{1}{\sqrt{1+m^2}}(\mathcal{H}_m[f] * \mathcal{H}_m[g])(b). \tag{2.12}$$

Cascaded Hough Transform

Tuytelaars et al. [13] added a third space and used this three-fold parameterization to detect the vanishing points and the horizon. This modification, named *Cascaded Hough Transform*, uses three pairs of parameters based on Eq. (2.13).

$$ax + b + y = 0 \tag{2.13}$$

The three subspaces are created as shown in Fig. 2.2. The first one has coordinates $a - b$, the second $(1/a) - (b/a)$, and the third $(1/b) - (a/b)$. All spaces are restricted to the interval $[-1, 1]$ in both directions. Each point (x, y) in the original space is transformed into a line in each of the three subspaces. Moreover, even in an

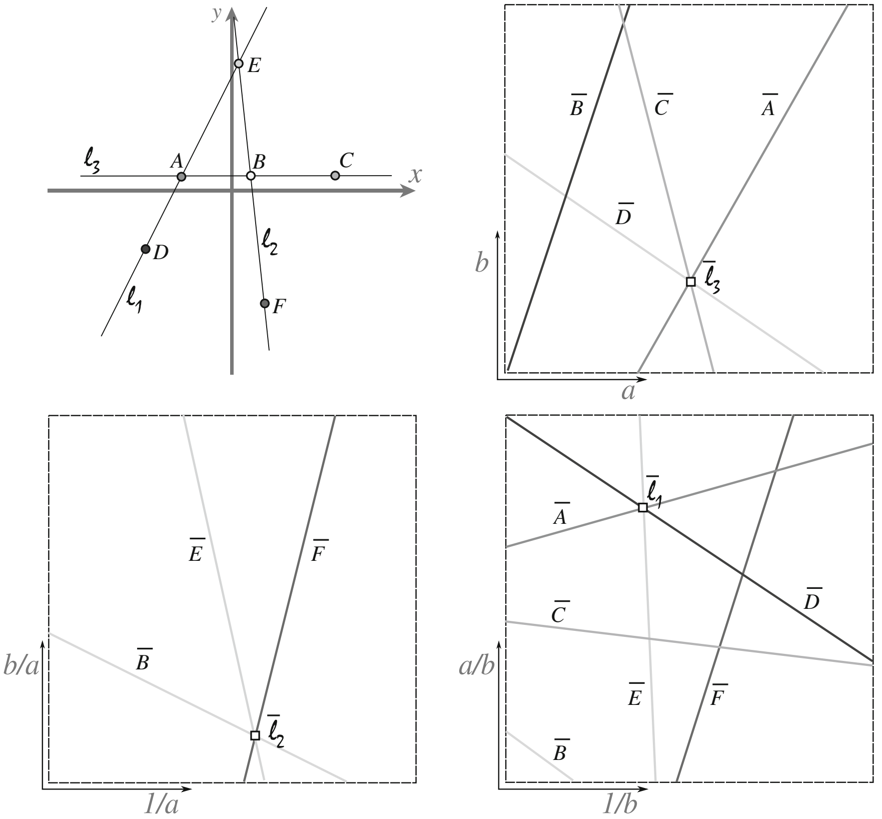

Fig. 2.2: Cascaded Hough transform using three spaces. *Left-top* input image; *Right* and *Bottom* corresponding Hough spaces

unbounded image plane, every line corresponds to a point with coordinates within area $[-1, 1] \times [-1, 1]$ for one of the spaces (see Fig. 2.2).

Consider the input image scaled to get image boundaries ± 1. The significant part of the third subspace is without any vote and just the first two subspaces are needed to represent all lines from the image. However, CHT is mainly used for vanishing points and line detection, where the third space is indispensable. For more details, please see [13, 14].

$\theta - \varrho$ **parameterization**

In 1972, Duda and Hart [15] introduced a very popular parameterization denoted as $\theta - \varrho$ which is very important for its inherently bounded parameter space. It is based on the line equation in the normal form (2.14)

$$y \sin\theta + x \cos\theta = \varrho, \tag{2.14}$$

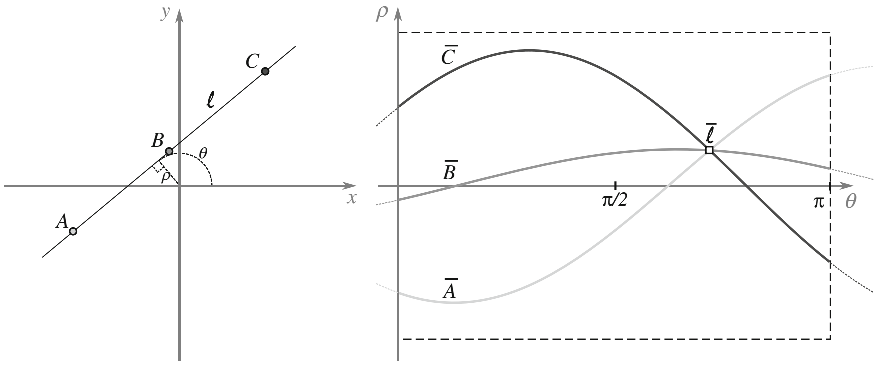

Fig. 2.3: Hough transform using $\theta - \varrho$ parameterization of line. *Left* input space; *right* corresponding Hough space

where parameter θ represents the angle of inclination and ϱ is the length of the shortest chord between the line and the origin of the image coordinate system (Fig. 2.3 left). In this case, images of all lines passing through a single point form a sinusoid curve in the parameter space (Fig. 2.3 right). Hence, $\theta - \varrho$ is not a PTLM and for a bounded input image it has a bounded parameter space.

For the $\theta - \varrho$ parameterization, Eq. (2.8) can be used without any modification. Therefore, an image convolution (filtering) can be done by a 1D convolution of the Hough space of image f with kernel g as

$$\mathcal{H}_\theta[f * g](\varrho) = (\mathcal{H}_\theta[f] * \mathcal{H}_\theta[g])(\varrho). \tag{2.15}$$

Circle Transform

Similar to the $\theta - \rho$ parameterization, the *circle transform* [16] uses the normal equation of the line (2.14). However, instead of the θ and ρ parameters it uses the intersection of the line and its normal passing through the origin O. This point fully characterizes a line. From the other side, points corresponding to all possible lines passing through an arbitrary point create a circle (see Fig. 2.4), i.e., point P is by the CT transformed into circle $\mathbf{c}$. The center of the circle $\mathbf{c}$ is the midpoint between O and P and the radius is equal to one-half of the distance $|OP|$ (Eq. 2.16).

$$\begin{aligned} P &= (a, b)_{\mathbb{E}^2} = (\rho\cos\theta, \rho\sin\theta) \\ \mathbf{c} : x &= \frac{1}{2}(\rho\cos\theta + O_x) \\ y &= \frac{1}{2}(\rho\sin\theta + O_y) \end{aligned} \tag{2.16}$$

The problem of such accumulation is that at the origin and very close to it, the number of votes is much higher than in the rest of the accumulator space. It is because

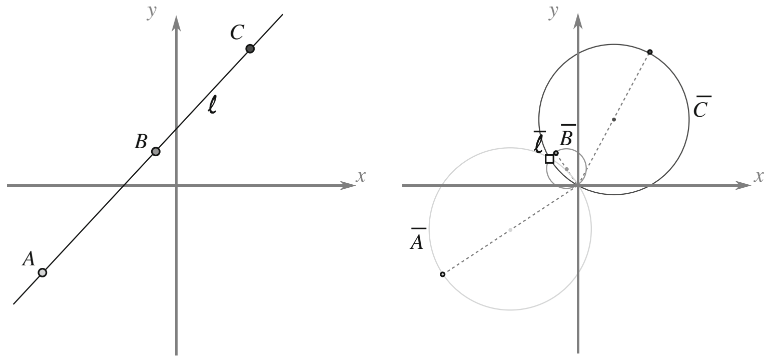

Fig. 2.4: Circle transform using θ–ϱ parameterization of line. *Left* input space; *right* corresponding Hough space

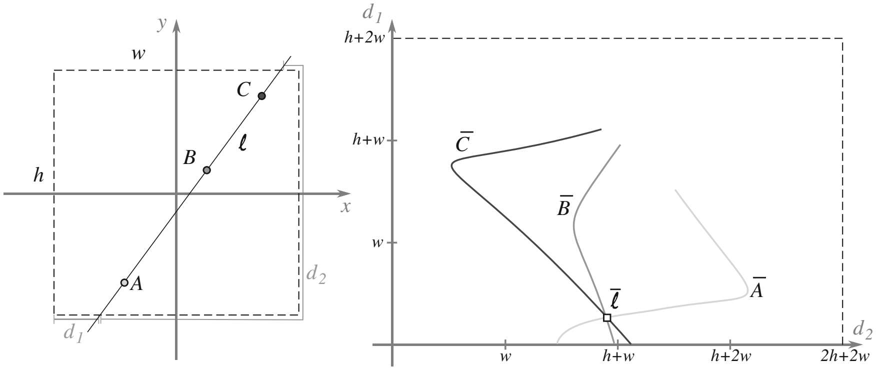

Fig. 2.5: Muff Transform. A line is parameterized by its intersections with the bounding rectangle

all circles pass through the origin by definition. The possible solution is to put the origin outside of the input image. However, the usage of the second space is needed because some line representations can now also lie outside of the input image.

Muff Transform

Several research groups invested effort to find other bounded parameterizations suitable for the HT. One of them is the *Muff-transform* introduced by Wallace in 1985 [17] (Fig. 2.5). As the basis for this parameterization, a bounding rectangle around the image is used. Each line intersects the bounding box at exactly two points. The distance of the first intersection (i.e. the nearest intersection from the origin along the perimeter) on the perimeter from the origin, and the distance between the first and the second intersections are used as the parameters.

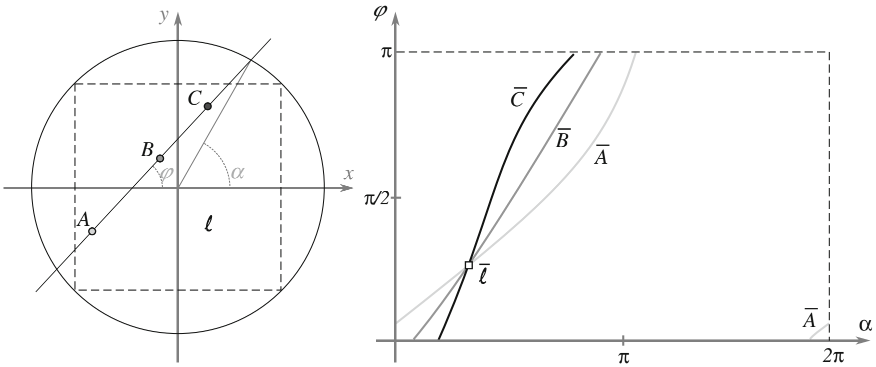

Fig. 2.6: Fan-Beam Transform. Line is parametrized by its orientation and intersection with the bounding circle

The main advantages are the bounded Hough space and the discretization error connected directly to the discretization of the input image; thus, it is possible to represent all necessary values as integers. The problem is the very sparse usage of the Hough space. However, this can be eliminated by rearrangement of the components of the accumulator space. Also, the curve accumulated to the space has discontinuities caused by corners of the bounding rectangle.

Fan–Beam Parameterization

Using a circle instead of a rectangle defines another bounded parameterization, called the *fan–beam* parameterization [10] (Fig. 2.6). Again, a line and a circle intersect at exactly two points. Angles defined by these two points were used for a line parameterization for the first time by Eckhardt and Maderlechner [18].

Similar to Muff transform, the parameters belong to the original image plane which makes the discretization intuitive. In contrast, the accumulated line is smooth, and the aliasing error is thus minimized. The accumulator, again, needs rearrangement for optimal utilization of the memory.

Forman Transform

The Muff transform is also a basis for a parameterization introduced by Forman [19], who combined it with $\theta - \varrho$ and represented lines by the first intersection point and the line's orientation θ (Fig. 2.7).

2.4.1 A Quick Comparison of Selected Line Parameterizations

Different line parameterizations offer advantages and each one of them has its costs. One important aspect of the parameterizations' properties is the uniformity of sampling of the voting space. The behavior of different parameterizations is illustrated by images in Fig. 2.8.

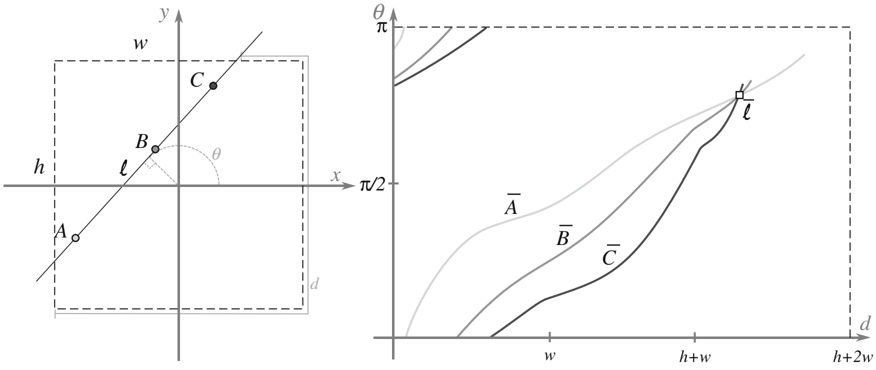

Fig. 2.7: Forman's parameterization based on θ–ϱ and Muff Transform

Figure 2.8 shows the utilization of the Hough space for different parameterizations in a manner similar to the work of Mejdani et al. [20]. The Hough space is accumulated for a 128 × 128 square image with each point considered as an edge point for accumulation (left column) and for an image with four lines (right column). The more the votes are accumulated, the darker the color is used. Always, the point with a maximal number of votes has absolutely black color and the point with the lowest value is white. Between these values, the color is linearly interpolated depending on the number of votes.

It should be noted that each parameterization can have an accumulator space a little bit different for different implementations, for example when the origin is in the center of the input image or in its corner. Some of the parameterizations have the accumulator space composed from several subspaces, which also enables a better arrangement for optimal utilization of computer memory. However, the arrangement used in Fig. 2.8 is sufficient for the illustration and corresponds to Figs. 2.1–2.7.

From the rasterized Hough spaces, several characteristic aspects can be observed. The main is the utilization of the accumulator (Table 2.1). The best utilization has the PClines parameterization (Chap. 4); on the other hand, the largest unused parts has the Fan beam transform. The second is the distribution of the votes and the number of bins with the maximal value. In an ideal case, one bin in the accumulator space corresponds to one line in the input image. This implies that the maximum of the votes has to be equal to the length (number of pixels) of the longest line in the image. Such a line is the diagonal and it occurs two times in the rectangular shape. That means two bins with maximal votes. A higher number of maximal bins, for example in the slope-intercept parameterization, indicates the presence of aliasing errors caused by discretization.

Different transformations also vary in the rasterized and accumulated shape in the Hough space. From computational aspect, the fastest is accumulation of lines. However, as mentioned at the beginning of this section and proved in [12], all PTLM need at least two twin spaces. This causes discontinuities and (dis)favors lines with

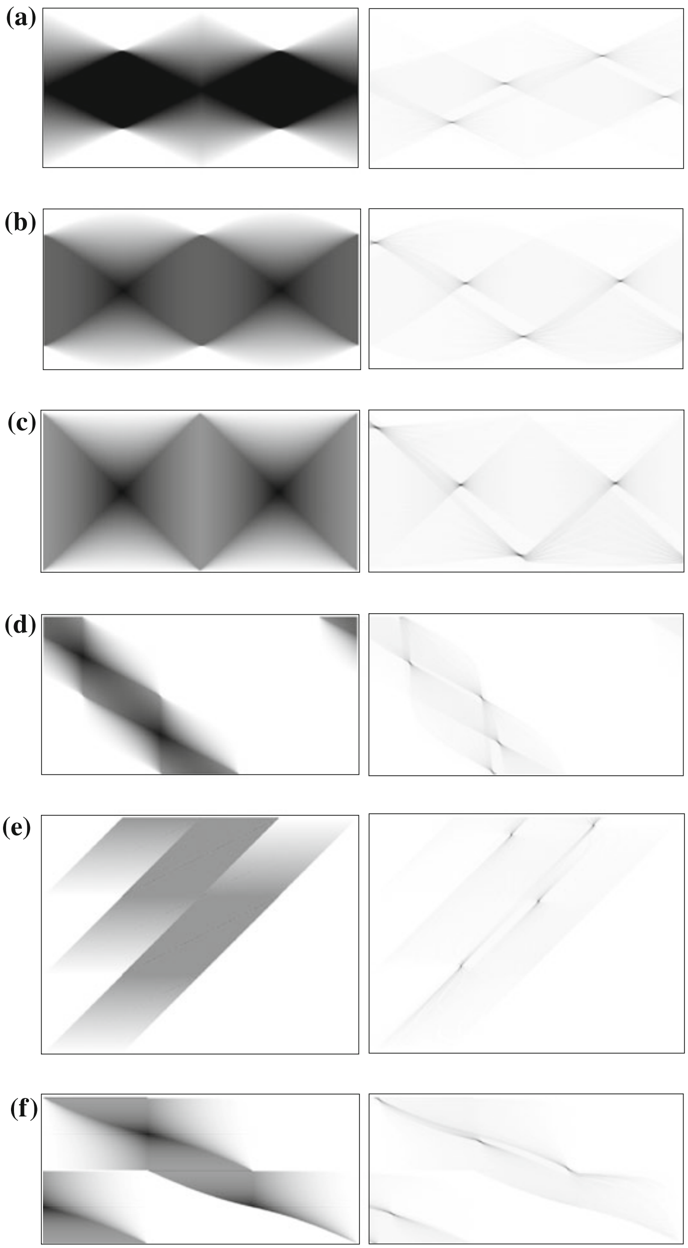

Fig. 2.8: **a** Slope-intercept parameterization; **b** θ–ϱ parameterization; **c** PClines (see Chap. 4); **d** fan-beam parameterization; **e** Muff transform; **f** Forman transform

Table 2.1: Basic characteristics of the selected transformations

	Parameterization	Utilization (%)	Image of a point	Space components
slope–intercept	Slope–intercept	75	Lines	2
$\theta - \varrho$	Inclination–distance	89	Sinusoid curve	1
Cascaded	Slope–intercept	–	Lines	3
Circle	Inclination–distance	–	Circle	2
PClines	xy coordinate	100	Lines	1
Muff	Bounding rectangle	50	Curve	6
Fan–beam	Bounding circle	36	Curve	1
Forman	Inclination–distance Bounding box	63	Curve	1

The first column reflects the used parameters/values; the second is the fraction of the bins from the Hough space where at least one vote is present; the third is the shape of the image of a point after mapping; and the last is the number of components in which the Hough space can be divided canonically

specific orientations. For example, when using the $\theta - \varrho$ transformation, a point is mapped to a smooth sinusoid curve and the discretization error is distributed uniformly through the whole Hough space. On the other hand, a PTLM slope–intercept parameterization prefers diagonal lines over horizontal and vertical (for more information about the discretization error, please see Sect. 4.3). The types of point images for different parameterizations are concluded in Table 2.1.

The last property shown in Table 2.1 is the number of components of the Hough space. This reflects the number of subspaces required, for example, from the definition of the PTLMs. The value also serves as information of discontinuity of the whole space. The higher the number is, the more the joints are in the space which implies more discontinuities.

References

1. Hough, P.V.C.: Method and means for recognizing complex patterns. U.S. Patent 3,069,654, 1962
2. Princen, J., Illingowrth, J., Kittler, J.: Hypothesis testing: a framework for analyzing and optimizing Hough transform performance. IEEE Trans. Pattern Anal. Mach. Intell. **16**(4), 329–341 (1994). http://dx.doi.org/10.1109/34.277588
3. Kälviäinen, H., Hirvonen, P., Xu, L., Oja, E.: Probabilistic and non-probabilistic Hough transforms: overview and comparisons. Image Vis. Comput. **13**(4), 239–252 (1995). doi:10.1016/0262-8856(95)99713-B, http://www.sciencedirect.com/science/article/pii/026288569599713B
4. Han, J.H., Kóczy, L.T., Poston, T.: Fuzzy Hough transform. Pattern Recogn. Lett. **15**, 649–658 (1994). doi:10.1016/0167-8655(94)90068-X, http://dl.acm.org/citation.cfm?id=189757.189759
5. Stephens, R.: Probabilistic approach to the Hough transform. Image Vis. Comput. **9**(1), 66–71 (1991). doi:10.1016/0262-8856(91)90051-P, http://www.sciencedirect.com/science/article/pii/026288569190051P

6. Kiryati, N., Bruckstein, A.: Antialiasing the Hough transform. CVGIP Graph. Models Image Process. **53**(3), 213–222 (1991). doi:10.1016/1049-9652(91)90043-J, http://www.sciencedirect.com/science/article/pii/104996529190043J
7. O'Gorman, F., Clowes, M.B.: Finding picture edges through collinearity of feature points. IEEE Trans. Comput. **25**(4), 449–456 (1976)
8. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn. **13**(2), 111–122 (1981)
9. Gall, J., Yao, A., Razavi, N., Gool, L.V., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 2188–2202 (2011). http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.70
10. Natterer, F.: The Mathematics of Computerized Tomography. Wiley, New York (1986)
11. Deans, S.R.: Hough transform from the radon transform. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-3**(2), 185–188 (1981). doi:10.1109/TPAMI.1981.4767076
12. Bhattacharya, P., Rosenfeld, A., Weiss, I.: Point-to-line mappings as Hough transforms. Pattern Recogn. Lett. **23**(14), 1705–1710 (2002). http://dx.doi.org/10.1016/S0167-8655(02)00133-2
13. Tuytelaars, T., Proesmans, M., Gool, L.V., Mi, E.: The cascaded hough transform. In: Proceedings of ICIP, pp. 736–739 (1997)
14. Tuytelaars, T., Proesmans, M., Gool, L.V.: The cascaded Hough transform as support for grouping and finding vanishing points and lines. In: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle, AFPAC '97, pp. 278–289. Springer, London, UK (1997). http://dl.acm.org/citation.cfm?id=646049.677093
15. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972). http://doi.acm.org/10.1145/361237.361242
16. Sewisy, A.A.: Graphical techniques for detecting lines with the hough transform. Int. J. Comput. Math. **79**(1), 49–64 (2002). doi:10.1080/00207160211911, http://www.tandfonline.com/doi/abs/10.1080/00207160211911
17. Wallace, R.: A modified Hough transform for lines. In: Proceedings of CVPR 1985, pp. 665–667 (1985)
18. Eckhardt, U., Maderlechner, G.: Application of the projected Hough transform in picture processing. In: Proceedings of the 4th International Conference on Pattern Recognition, pp. 370–379. Springer, London, UK (1988)
19. Forman, A.V.: A modified Hough transform for detecting lines in digital imagery. Appl. Artif. Intell. **III**, 151–160 (1986)
20. El Mejdani, S., Egli, R., Dubeau, F.: Old and new straight-line detectors: description and comparison. Pattern Recogn. **41**, 1845–1866 (2008). doi:10.1016/j.patcog.2007.11.013, http://dl.acm.org/citation.cfm?id=1343128.1343451