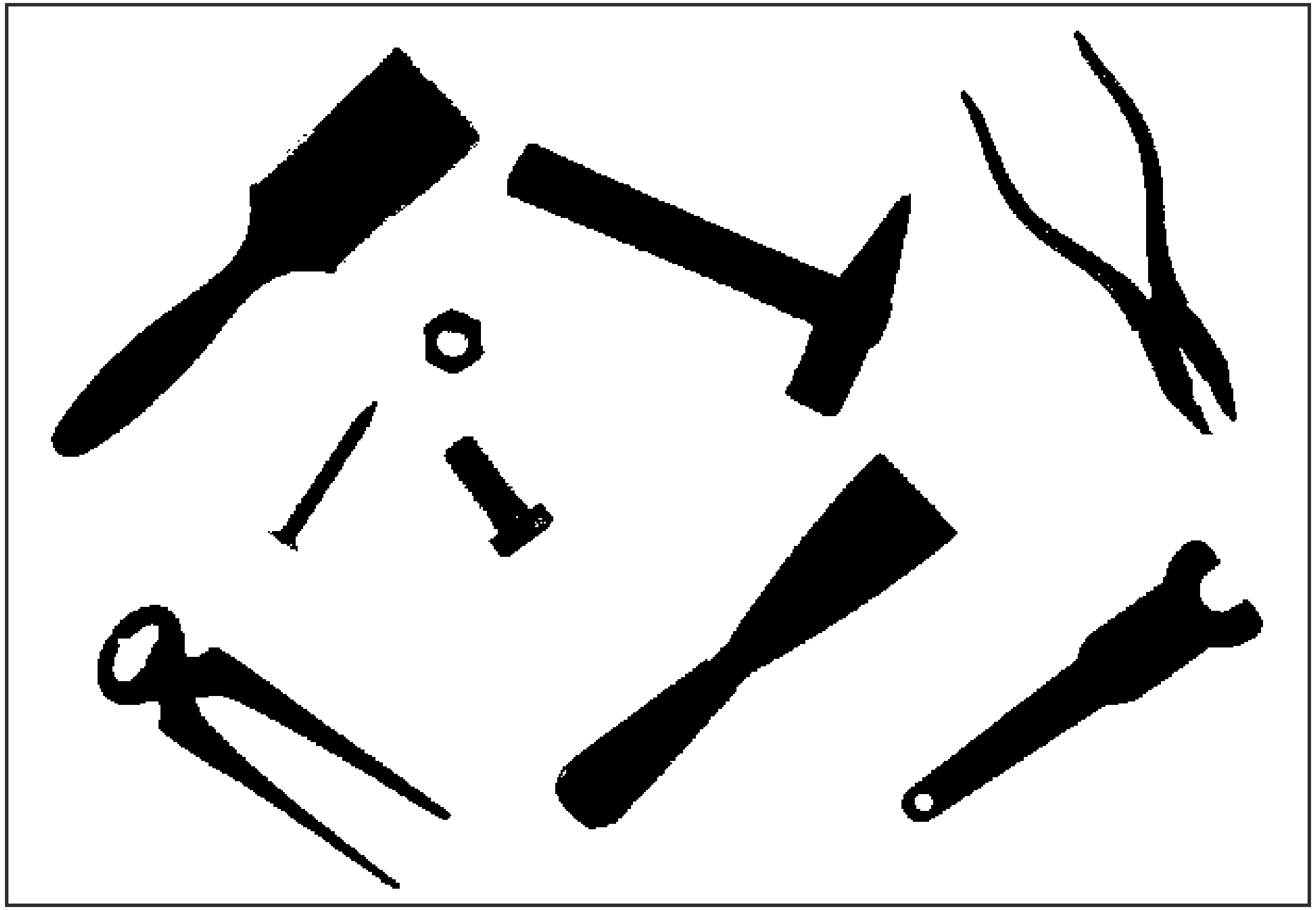
IM520/MC505
Computer Vision

SS 2017 – W. Burger

Part 4

2D Shape: moment/contour-based shape descriptors

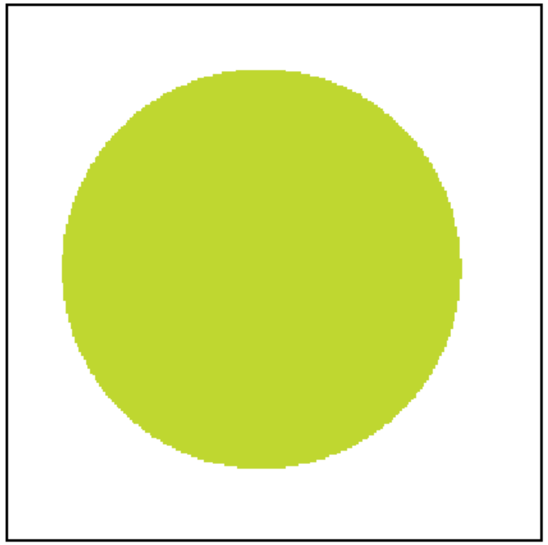
2D Shape Classification



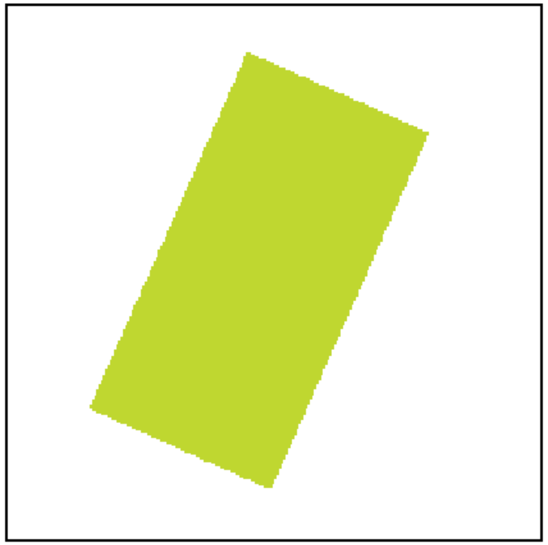
Geometric properties of binary regions – “circularity”

$$Circularity(\mathcal{R}) = \frac{4\pi \cdot Area(\mathcal{R})}{Perimeter^2(\mathcal{R})}$$

Shape measure that is independent of **position**, **orientierung** and **size** of the region!



1.001



0.672

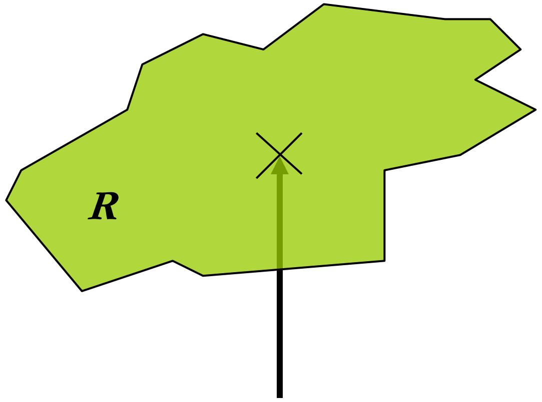
0.086

(results with corrected perimeter values)

Statistical shape properties of binary regions

Center of gravity (centroid)

$$\bar{x} = \frac{1}{|\mathcal{R}|} \cdot \sum_{(u,v)\in\mathcal{R}} u \qquad \bar{y} = \frac{1}{|\mathcal{R}|} \cdot \sum_{(u,v)\in\mathcal{R}} v$$



Region interpreted as a flat body.

Position of centroid is calculated with **sub-pixel** accuracy (floating-point values)!

Also applicable for non-connencted regions (eg., point clouds).

Statistical shape properties – “moments”

Ordinary moments of order *p*,*q*

$$m_{pq} = \sum_{(u,v)\in\mathcal{R}} I(u,v) \cdot u^p v^q$$

← grayscale image

$$m_{pq} = \sum_{(u,v)\in\mathcal{R}} u^p v^q$$

← binary image

Example: area of a region (can be expressed as the zero-order moment):

$$Area(\mathcal{R}) = |\mathcal{R}| = \sum_{(u,v)\in\mathcal{R}} 1 = \sum_{(u,v)\in\mathcal{R}} u^0 v^0 = m_{00}(\mathcal{R})$$

Central moments (translation invariance)

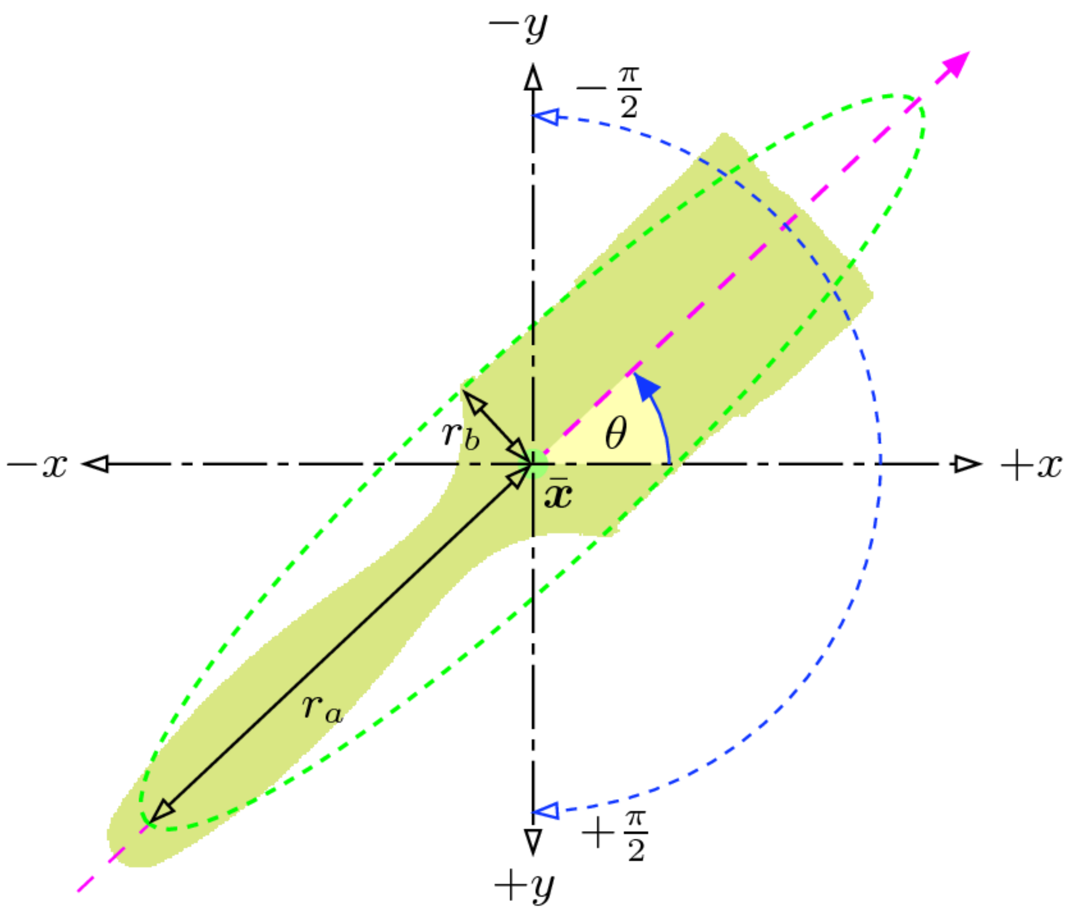
Central moment of order *p*,*q*

Grayscale images: $$\mu_{pq}(\mathcal{R}) = \sum_{(u,v)\in\mathcal{R}} I(u,v)\cdot(u-\bar{x})^p\cdot(v-\bar{y})^q$$

Binary images: $$\mu_{pq}(\mathcal{R}) = \sum_{(u,v)\in\mathcal{R}} (u-\bar{x})^p\cdot(v-\bar{y})^q$$

Distribution of pixel coordinates relative to the region's **centroid**.

Applications of region moments (1)



Calculating the **orientation** of a shape *R* (direction of the major axis through the centroid):

$$\theta(\mathcal{R}) = \frac{1}{2} \tan^{-1} \left(\frac{2 \cdot \mu_{11}(\mathcal{R})}{\mu_{20}(\mathcal{R}) - \mu_{02}(\mathcal{R})} \right)$$

Applications of region moments – *eccentricity* and *equivalent ellipse*

Eccentricity of a region (shape feature):

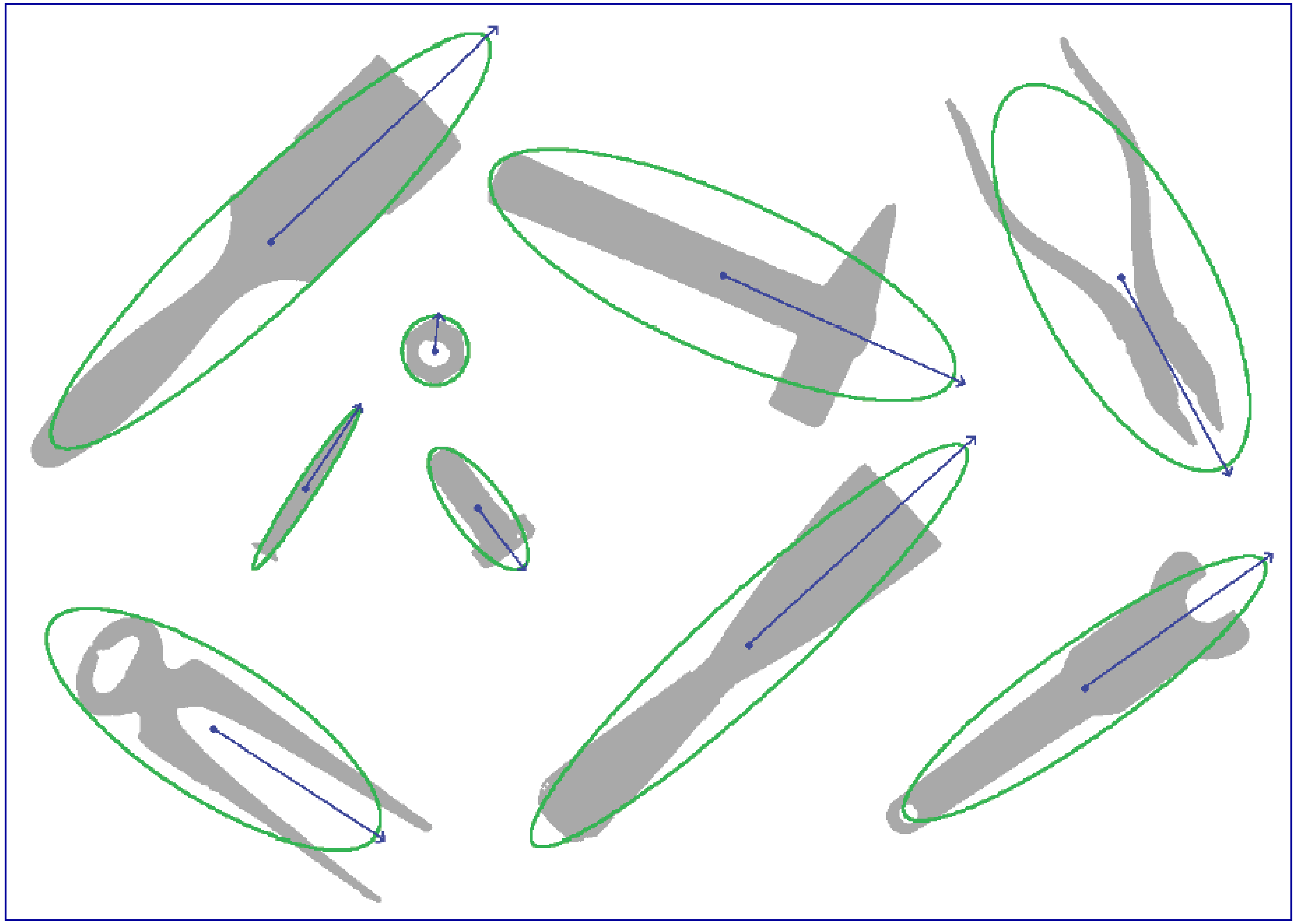
$$\mathcal{E}(\mathcal{R}) = \frac{a_1}{a_2} = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4 \cdot \mu_{11}^2}}{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4 \cdot \mu_{11}^2}}$$

Axis ratio of the **equivalent ellipse**: if **filled**, it would have the same 1st and 2nd order central moments as the region itself

Actual radii of the equivalent ellipse can be found through the region area:

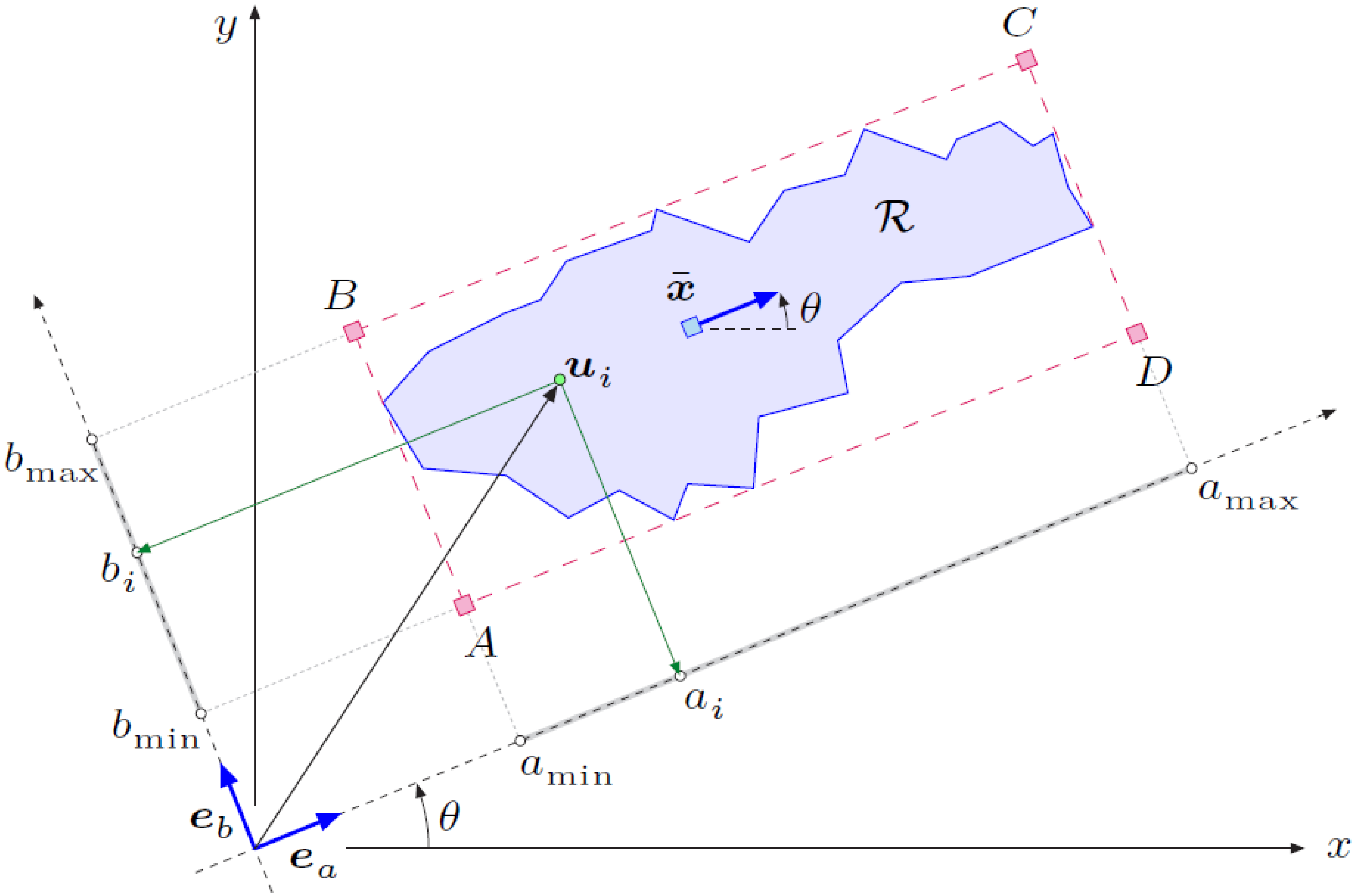
$$r_a = 2\sqrt{\frac{\lambda_1}{|\mathcal{R}|}} = \sqrt{\frac{2\,a_1}{|\mathcal{R}|}} \qquad r_b = 2\sqrt{\frac{\lambda_2}{|\mathcal{R}|}} = \sqrt{\frac{2\,a_2}{|\mathcal{R}|}}$$

Example: orientation, exccentricity, equiv. ellipse

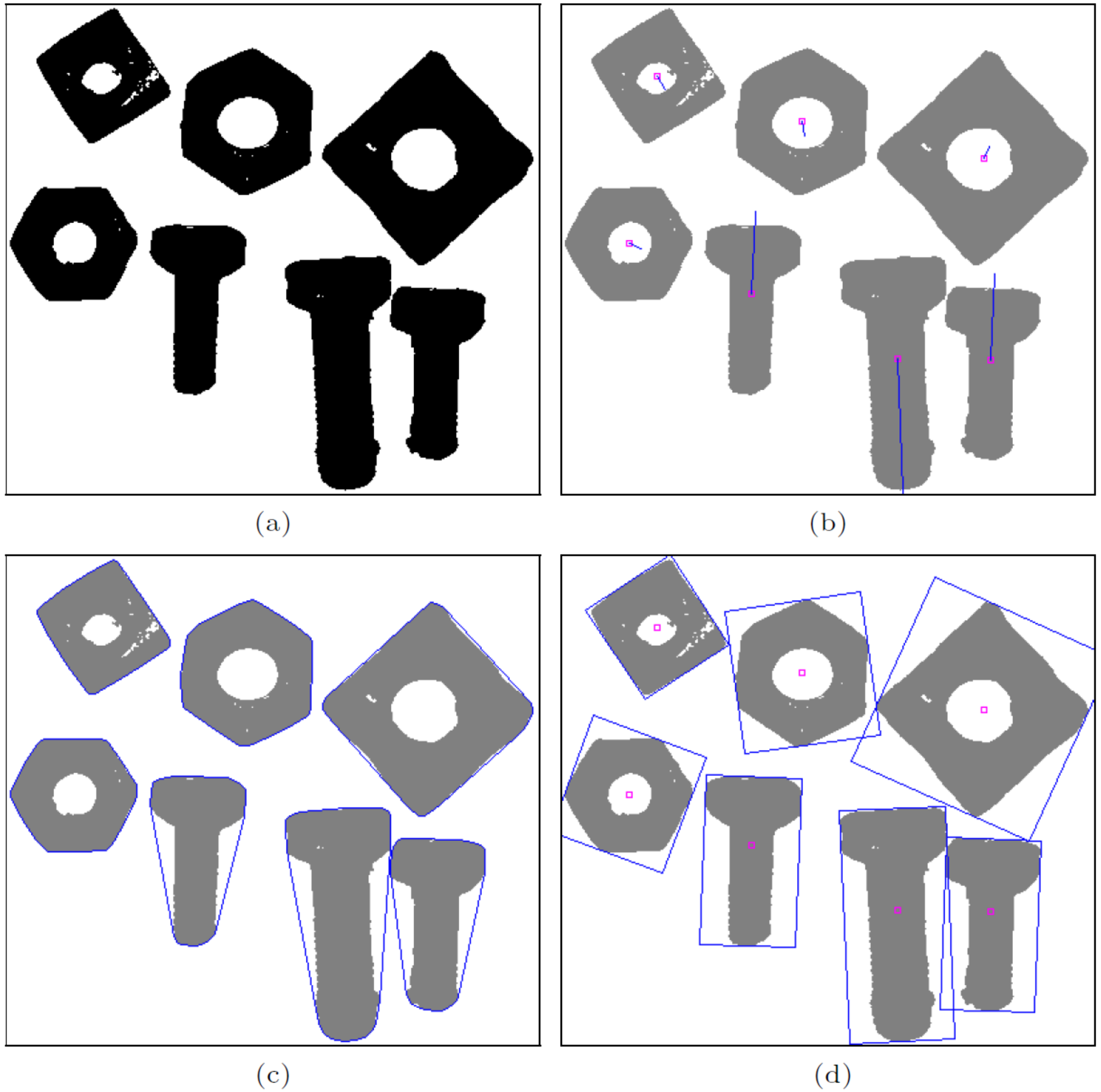


Major axis-aligned bounding box

Calculation of a region's major axis-aligned bounding box. The unit vector $\boldsymbol{e}_a$ is parallel to the region's major axis (oriented at angle θ); $\boldsymbol{e}_b$ is perpendicular to $\boldsymbol{e}_a$. The projection of a region point $\boldsymbol{u}_i$ onto the lines defined by $\boldsymbol{e}_a$ and $\boldsymbol{e}_b$ yields the lengths a_i and b_i, respectively (measured from the coordinate origin). The resulting quantities $a_{\min}$, $a_{\max}$, $b_{\min}$, $b_{\max}$ define the corner points (A, B, C, D) of the axis-aligned bounding box. Note that the position of the region's centroid ($\bar{\boldsymbol{x}}$) is not required in this calculation.



Geometric region properties



Geometric region properties. Original binary image (a), centroid and orientation vector (length determined by the region's eccentricity) of the major axis (b), convex hull (c), and major axis-aligned bounding box (d).

Normalized central moments

Central moment values of course depend on the absolute size of the region since the value depends directly on the distance of all region points to its centroid. So, if a 2D shape is scaled uniformly by some factor $s \in \mathbb{R}$, its central moments multiply by the factor

$$s^{(p+q+2)}. \tag{2.22}$$

Thus size-invariant "normalized" moments are obtained by scaling with the reciprocal of the $\boxed{\text{area } \mu_{00} = m_{00}}$ raised to the required power in the form

$$\boxed{\bar{\mu}_{pq}(\mathcal{R}) = \mu_{pq}(\mathcal{R}) \cdot \left(\frac{1}{\mu_{00}(\mathcal{R})} \right)^{(p+q+2)/2}} \tag{2.23}$$

for $(p+q) \geq 2$ [45, p. 529].

Hu's (7) Invariant Moments

These features are also known as moment invariants since they are **invariant under translation, rotation, and scaling**. While defined here for binary images, they are also applicable to grayscale images. In practice, the logarithm of the results is used since the raw values can have a very large range.

$$
\begin{aligned}
\phi_1 &= \bar{\mu}_{20} + \bar{\mu}_{02}, \\
\phi_2 &= (\bar{\mu}_{20} - \bar{\mu}_{02})^2 + 4\,\bar{\mu}_{11}^2, \\
\phi_3 &= (\bar{\mu}_{30} - 3\,\bar{\mu}_{12})^2 + (3\,\bar{\mu}_{21} - \bar{\mu}_{03})^2, \\
\phi_4 &= (\bar{\mu}_{30} + \bar{\mu}_{12})^2 + (\bar{\mu}_{21} + \bar{\mu}_{03})^2, \\
\phi_5 &= (\bar{\mu}_{30} - 3\,\bar{\mu}_{12}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3(\bar{\mu}_{21} + \bar{\mu}_{03})^2] + \\
&\quad (3\,\bar{\mu}_{21} - \bar{\mu}_{03}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot [3\,(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2], \\
\phi_6 &= (\bar{\mu}_{20} - \bar{\mu}_{02}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2] + \\
&\quad 4\,\bar{\mu}_{11} \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}), \\
\phi_7 &= (3\,\bar{\mu}_{21} - \bar{\mu}_{03}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3\,(\bar{\mu}_{21} + \bar{\mu}_{03})^2] + \\
&\quad (3\,\bar{\mu}_{12} - \bar{\mu}_{30}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot [3\,(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2].
\end{aligned}
\tag{10.47}
$$

Complex Invariant Moments by *Flusser*

Hu's moments have been shown to be *redundant* and *incomplete*.
Flusser defines a set of **11 scale and rotation-invariant features** based on complex-values moments:

Grayscale images:

$$c_{pq}(\mathcal{R}) = \sum_{(u,v)\in\mathcal{R}} I(u,v) \cdot (x + \mathrm{i}\cdot y)^p \cdot (x - \mathrm{i}\cdot y)^q$$

Binary images:

$$c_{pq}(\mathcal{R}) = \sum_{(u,v)\in\mathcal{R}} (x + \mathrm{i}\cdot y)^p \cdot (x - \mathrm{i}\cdot y)^q$$

Scale-normalized moments:

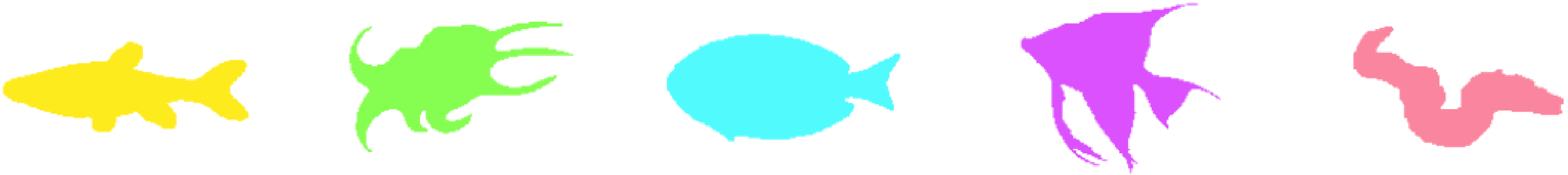
$$\hat{c}_{p,q}(\mathcal{R}) = \frac{1}{A^{(p+q+2)/2}} \cdot c_{p,q}(\mathcal{R})$$

region area *A*

Flusser's invariant shape moments

$$\begin{aligned}
\psi_1 &= \mathrm{Re}(\hat{c}_{1,1}), & \psi_2 &= \mathrm{Re}(\hat{c}_{2,1} \cdot \hat{c}_{1,2}), & \psi_3 &= \mathrm{Re}(\hat{c}_{2,0} \cdot \hat{c}_{1,2}^2), \\
\psi_4 &= \mathrm{Im}(\hat{c}_{2,0} \cdot \hat{c}_{1,2}^2), & \psi_5 &= \mathrm{Re}(\hat{c}_{3,0} \cdot \hat{c}_{1,2}^3), & \psi_6 &= \mathrm{Im}(\hat{c}_{3,0} \cdot \hat{c}_{1,2}^3), \\
\psi_7 &= \mathrm{Re}(\hat{c}_{2,2}), & \psi_8 &= \mathrm{Re}(\hat{c}_{3,1} \cdot \hat{c}_{1,2}^2), & \psi_9 &= \mathrm{Im}(\hat{c}_{3,1} \cdot \hat{c}_{1,2}^2), \\
\psi_{10} &= \mathrm{Re}(\hat{c}_{4,0} \cdot \hat{c}_{1,2}^4), & \psi_{11} &= \mathrm{Im}(\hat{c}_{4,0} \cdot \hat{c}_{1,2}^4). & & \qquad (10.51)
\end{aligned}$$

Flusser's moments example



ψ_1	0.3730017575	0.2545476083	0.2154034257	0.2124041195	0.3600613700
ψ_2	0.0012699373	0.0004247053	0.0002068089	0.0001089652	0.0017187073
ψ_3	0.0004041515	0.0000644829	0.0000274491	0.0000014248	-0.0003853999
ψ_4	0.0000097827	-0.0000076547	0.0000071688	-0.0000022103	-0.0001944121
ψ_5	0.0000012672	0.0000002327	0.0000000637	0.0000000083	-0.0000078073
ψ_6	0.0000001090	-0.0000000483	0.0000000041	0.0000000153	-0.0000061997
ψ_7	0.2687922057	0.1289708408	0.0814034374	0.0712567626	0.2340886626
ψ_8	0.0003192443	0.0000414818	0.0000134036	0.0000003020	-0.0002878997
ψ_9	0.0000053208	-0.0000032541	0.0000030880	-0.0000008365	-0.0001628669
ψ_{10}	0.0000103461	0.0000000091	0.0000000019	-0.0000000003	0.0000001922
ψ_{11}	0.0000000120	-0.0000000020	0.0000000008	-0.0000000000	0.0000003015

Shape matching with region moments

Given **two shapes** *A*,*B* with associated **moment vectors**

$$\left.\begin{aligned} \boldsymbol{f}_A &= (\psi_1(A), \ldots, \psi_{11}(A)) \\ \boldsymbol{f}_B &= \underbrace{(\psi_1(B), \ldots, \psi_{11}(B))}_{} \end{aligned}\right\}$$

How **similar** are shapes *A* and *B*?

11-dimensional **feature vector**
(*m*-dimensional in general)

Some form of **similarity or distance measure** is needed …

Euclidean feature distance (example)

Euclidean distance (in feature space) as a similarity measure between shapes A,B:

$$\mathrm{d_E}(A, B) = \|\boldsymbol{f}_A - \boldsymbol{f}_B\| = \Big[\sum_{i=1}^{11} |\psi_i(A) - \psi_i(B)|^2\Big]^{1/2}$$

	0.000	0.183	0.245	0.255	0.037
	0.183	0.000	0.062	0.071	0.149
	0.245	0.062	0.000	0.011	0.210
	0.255	0.071	0.011	0.000	0.220
	0.037	0.149	0.210	0.220	0.000

Inter-class distances

Problem: magnitude of feature elements may vary over a *large numeric range*

- Feature **components with large magnitude dominate** the distance measure – smaller features contribute nothing.
- Thus moment-based shape comparison based on simple Euclidean distance is not very selective!
- Alternatives: (a) use logarithmic values, (b) **weighted distance**

$$\mathrm{d}'_{\mathrm{E}}(A,B) = \Big[\sum_{i=1}^{11} w_i \cdot |\psi_i(A) - \psi_i(B)|^2\Big]^{1/2}$$

different weight
for each feature
dimension *i*

- What are the **proper weights**?
- Can they be calculated automatically?

Mahalanobis distance

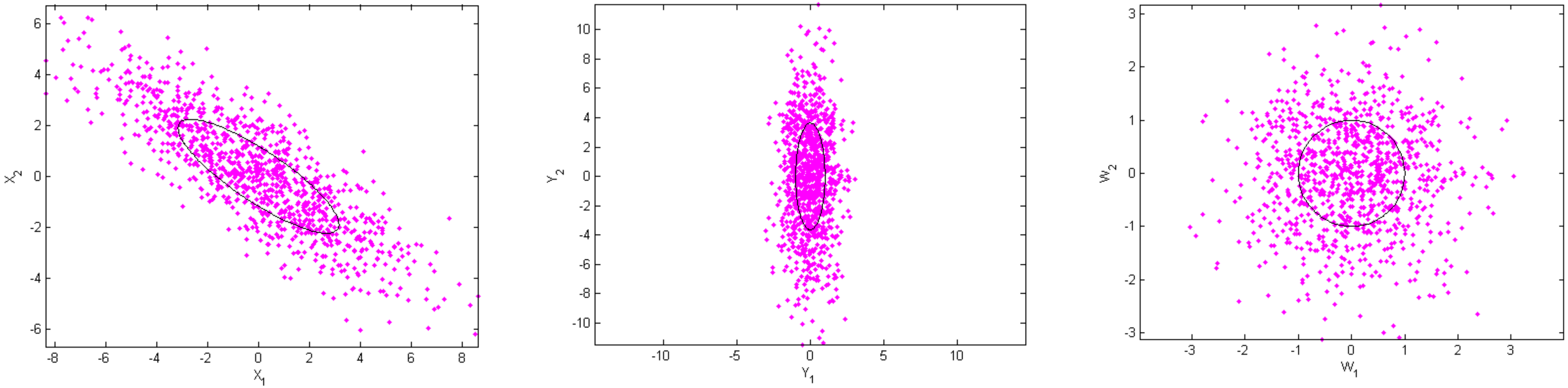
- Mahalanobis distance accounts for the **statistical distribution** of each vector component.
- All **components are transformed** such that the complete feature set fits a *N*-dimensional **Gaussian** distribution (sphere) with $\mu = 0$ and $\sigma = 1$ in all dimensions.
- Avoids large magnitude components dominating the smaller ones.

Definition:

$$\boxed{\mathrm{d_M}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \|\boldsymbol{x}_a - \boldsymbol{x}_b\|_\mathrm{M} = \sqrt{(\boldsymbol{x}_a - \boldsymbol{x}_b)^\mathsf{T} \cdot \boldsymbol{\Sigma}^{-1} \cdot (\boldsymbol{x}_a - \boldsymbol{x}_b)},}$$

inverse of the m x *m*
covariance matrix
of the feature set

Mahalanobis distance “whitening” example

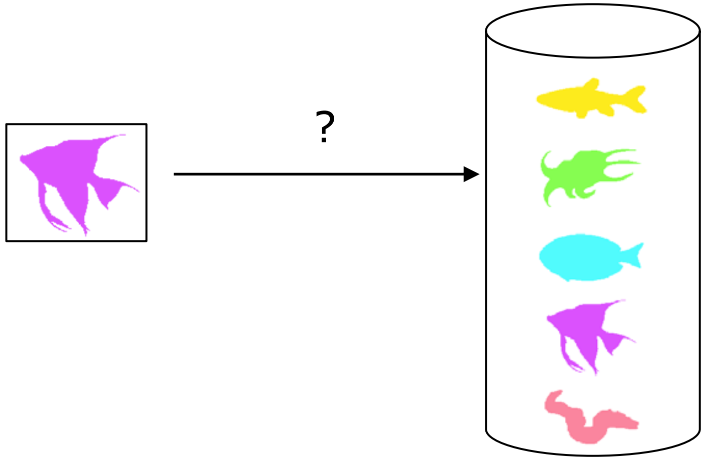


All contours are at Mahalanobis distance

Image source: “Whitening and Coloring Transformations for Multivariate Gaussian Data”,ECE 662 Student Lecture by Maliha Hossain, 2014.

Matching against many candidates

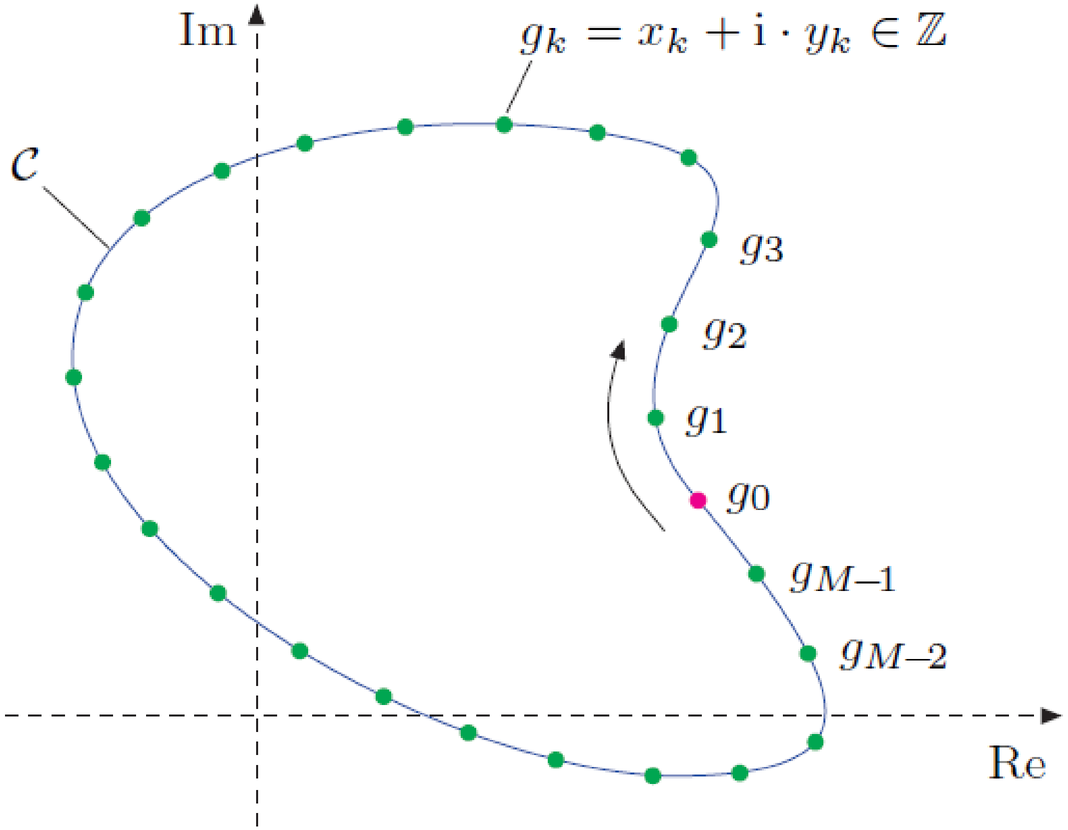
- Given is a (possibly large) database of **reference shapes**.
- How to structure (**index**) the database for faster matching?
- How to **avoid repeated calculations** (Mahalanobis mapping)?
- How to tell if a fit is “good enough”?



More on this in the lab …

Contour-Based Shape Descriptors

Fourier Shape Descriptors – boundaries in the complex plane

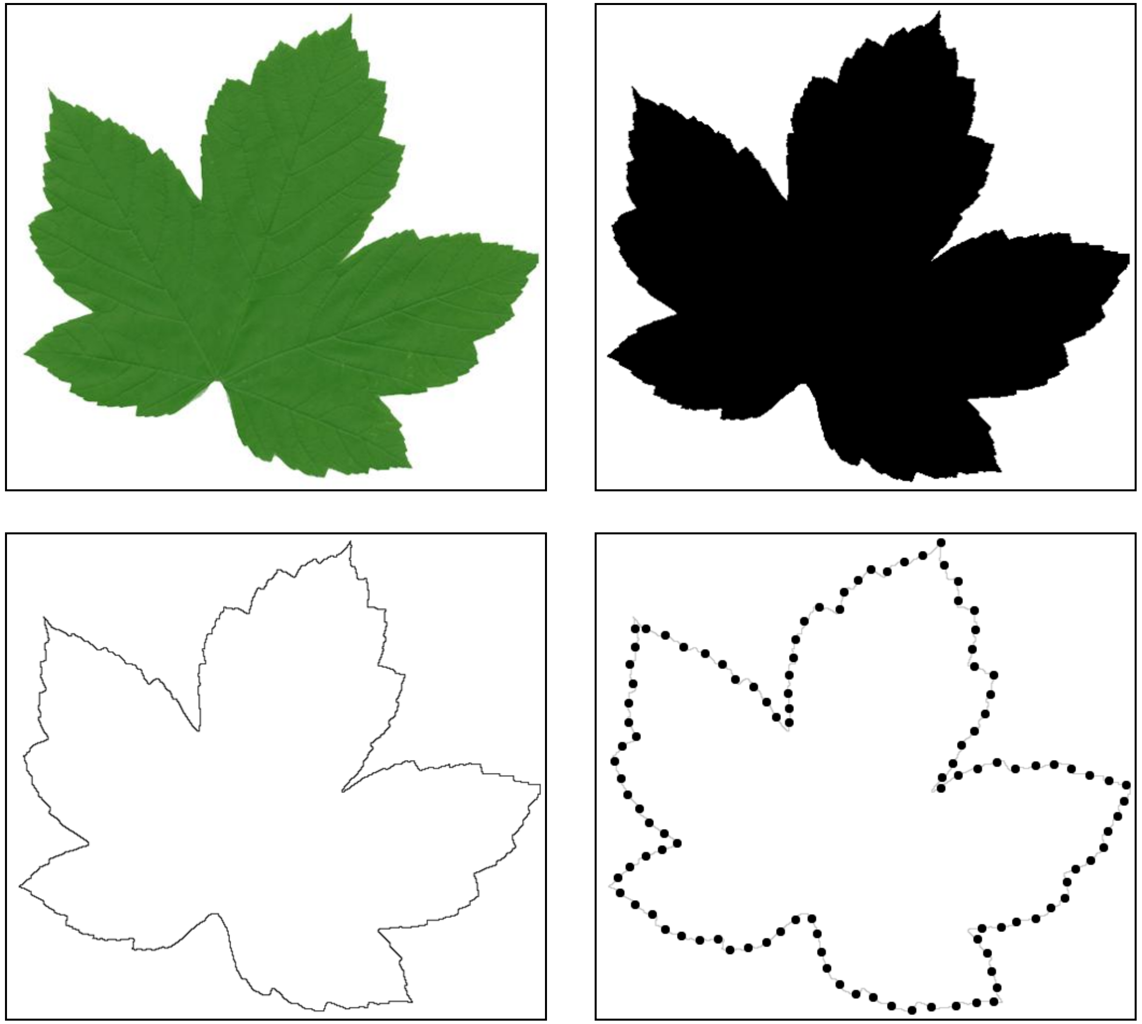


2D contour samples are interpreted as a periodic sequence of **complex values**.

Example

Original

Connected region



Closed contour

Sub-sampled contour

Discrete Fourier Transform (DFT)

$$G_m = \frac{1}{P} \sum_{k=0}^{P-1} z_k \cdot e^{-\mathrm{i}2\pi \frac{mk}{P}}$$

$$z_k = \sum_{m=0}^{P-1} G_m \cdot e^{\mathrm{i}2\pi \frac{mk}{P}}$$

Forward transformation:

$z(k) \rightarrow G(m)$

Inverse transformation:

$G(m) \rightarrow z(k)$

Euler's notation of complex values:

$$\begin{aligned} z &= a + \mathrm{i}b \\ &= |z| \cdot (\cos\psi + \mathrm{i} \cdot \sin\psi) \\ &= |z| \cdot e^{\mathrm{i}\psi} \end{aligned}$$

Calculating the DFT

$$G_m = \frac{1}{P}\sum_{k=0}^{P-1}\Big[\underbrace{x_k + \mathrm{i}\cdot y_k}_{z_k}\Big]\cdot\Big[\cos\big(2\pi\tfrac{mk}{P}\big) - \mathrm{i}\cdot\sin\big(2\pi\tfrac{mk}{P}\big)\Big]$$

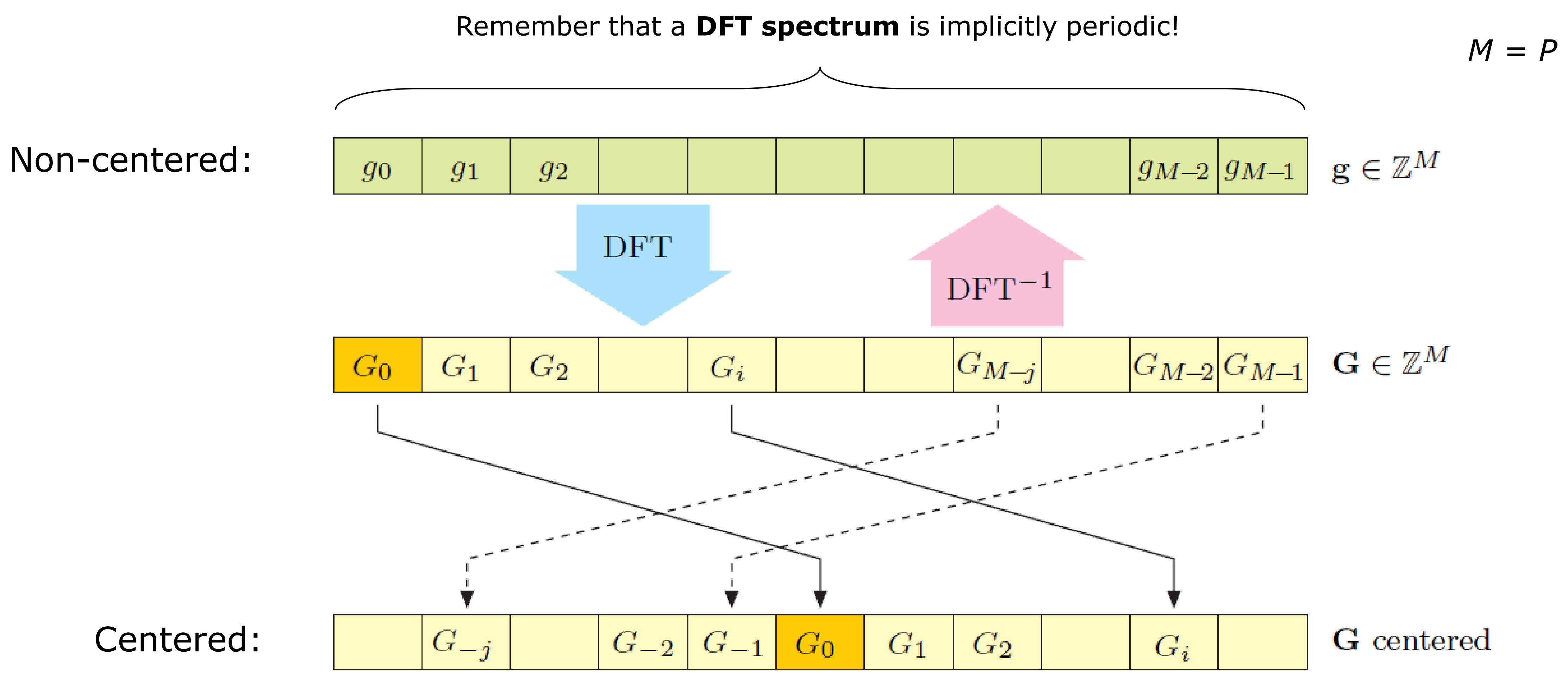
m = 0…*P*-1

$$\boxed{\begin{aligned}\mathrm{Re}(G_m) &= \frac{1}{P}\sum_{k=0}^{P-1}\Big[x_k\cdot\cos\big(2\pi\tfrac{mk}{P}\big) + y_k\cdot\sin\big(2\pi\tfrac{mk}{P}\big)\Big]\\ \mathrm{Im}(G_m) &= \frac{1}{P}\sum_{k=0}^{P-1}\Big[y_k\cdot\cos\big(2\pi\tfrac{mk}{P}\big) - x_k\cdot\sin\big(2\pi\tfrac{mk}{P}\big)\Big]\end{aligned}}$$

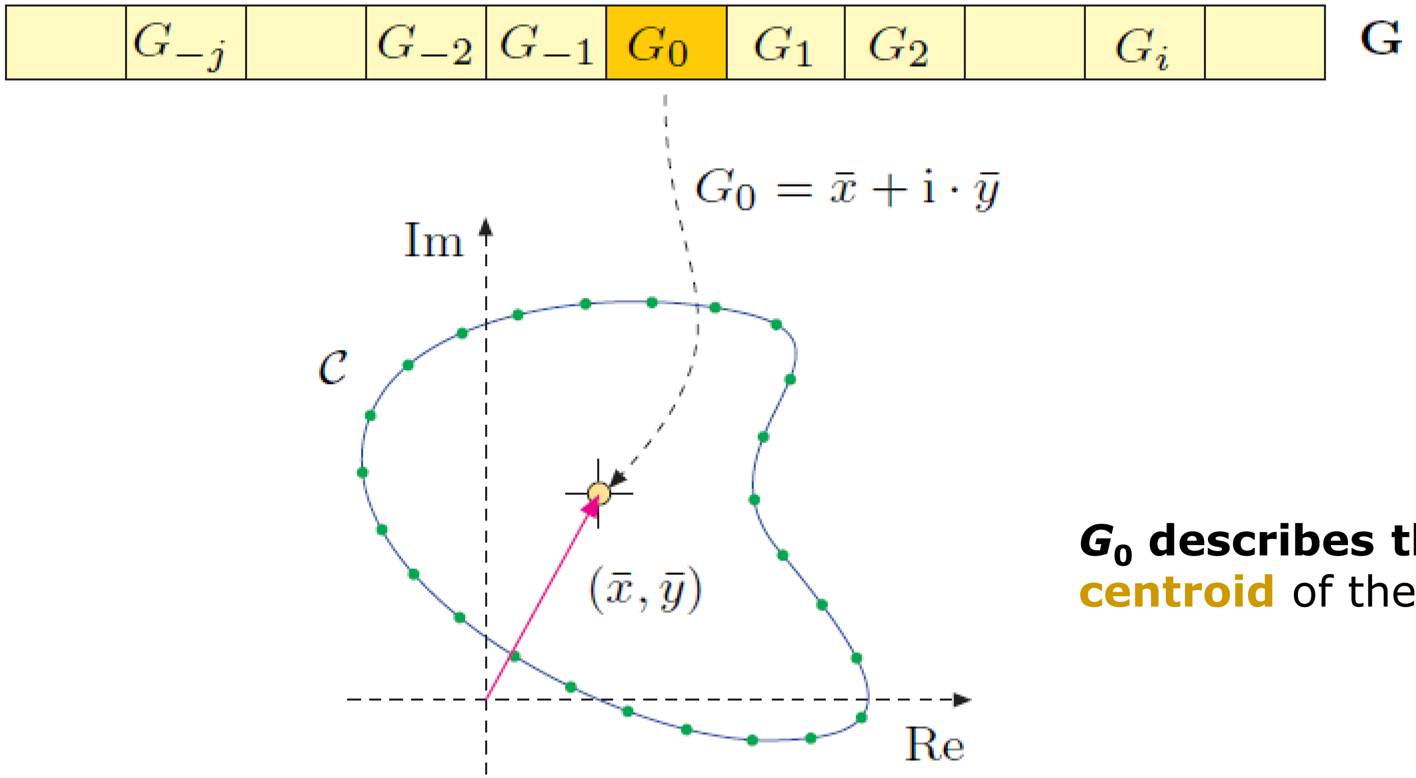
Nothing *complex* any more ;-)

FFT (fast Fourier transform) can be used if $P=2^n$

Fourier Descriptor Coefficients – centered representation

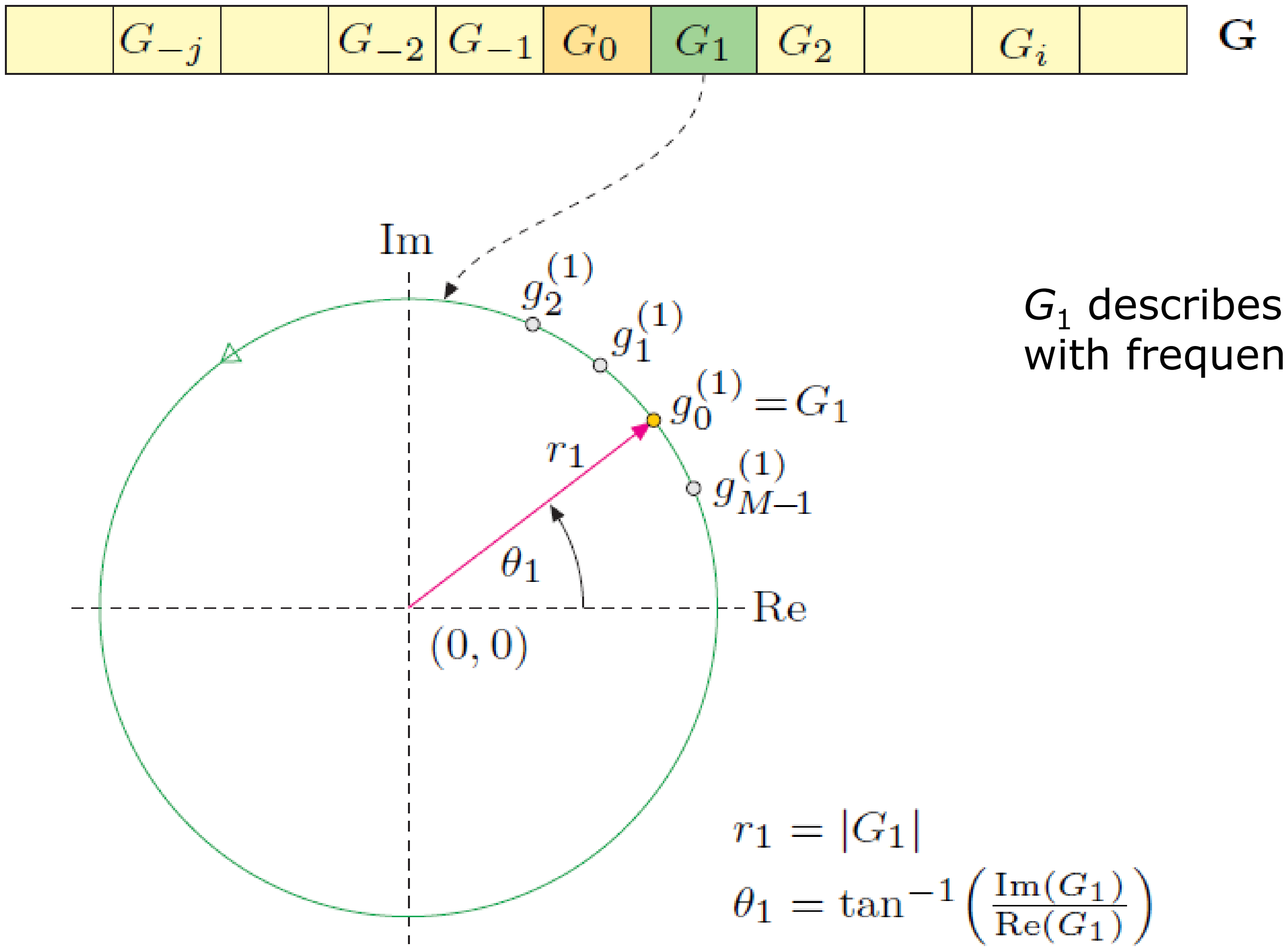


Single Fourier coefficients – geometric meaning



***G_0* describes the**
centroid of the boundary points

Single Fourier coefficients – geometric meaning



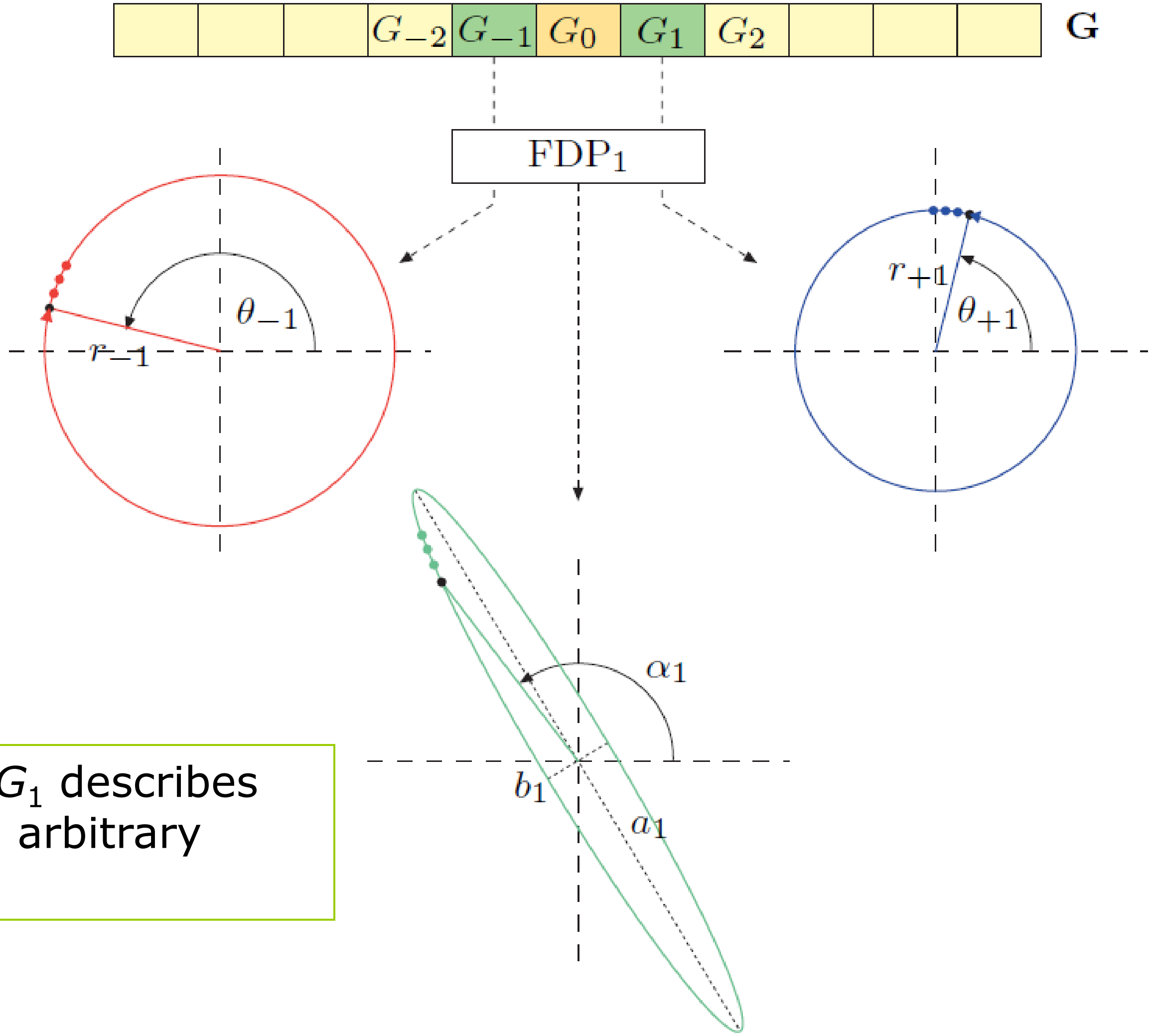
$$r_1 = |G_1|$$

$$\theta_1 = \tan^{-1}\left(\frac{\mathrm{Im}(G_1)}{\mathrm{Re}(G_1)}\right)$$

G_1 describes a circle with frequency 1

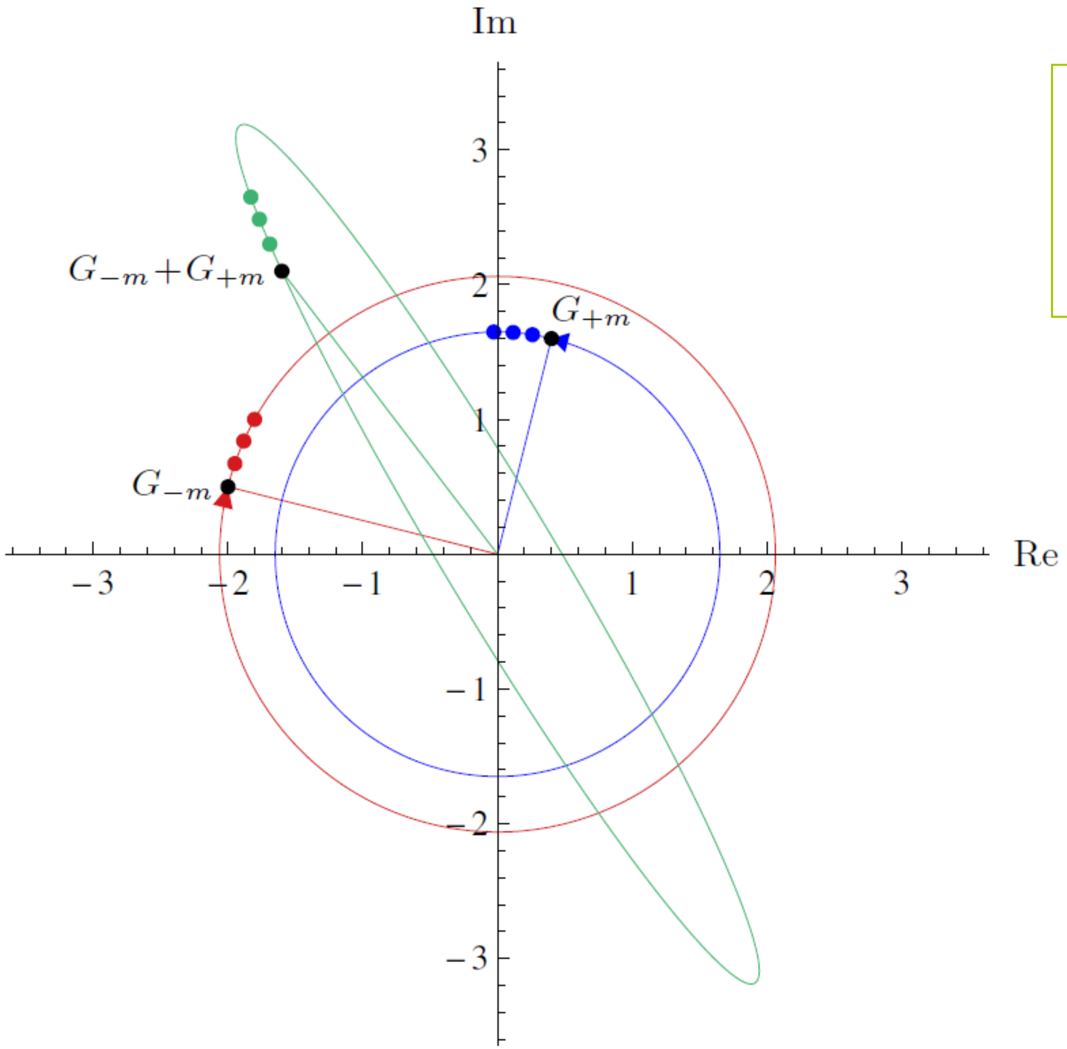
Fourier coefficient (descriptor) pair

G_{-1} also describes a circle rotating in the **opposite** direction (with frequ. 1)



The pair G_{-1} + G_1 describes an **ellipse** with arbitrary orientation.

Fourier coefficient (descriptor) pair



In general, a FD-pair (G_{-m},G_m) describes an ellipse which oscillates *m* times as we go along the shape's boundary length.

Partial reconstruction from a single Fourier descriptor pair

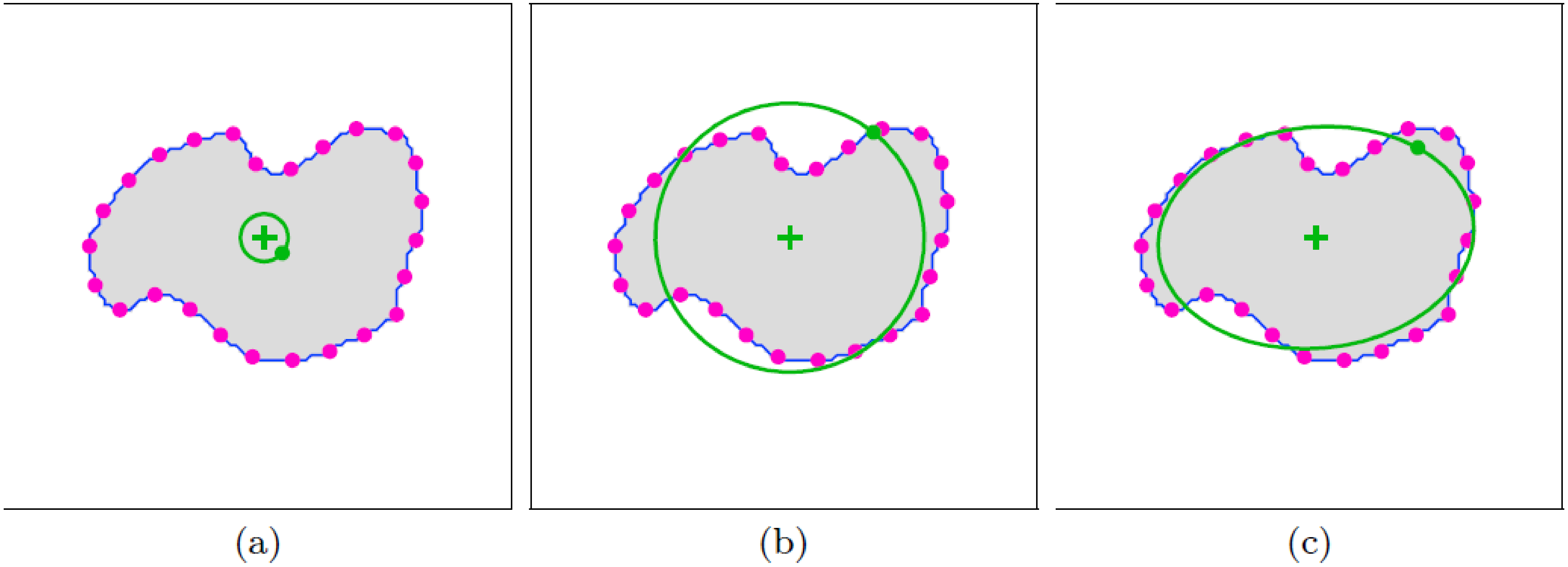
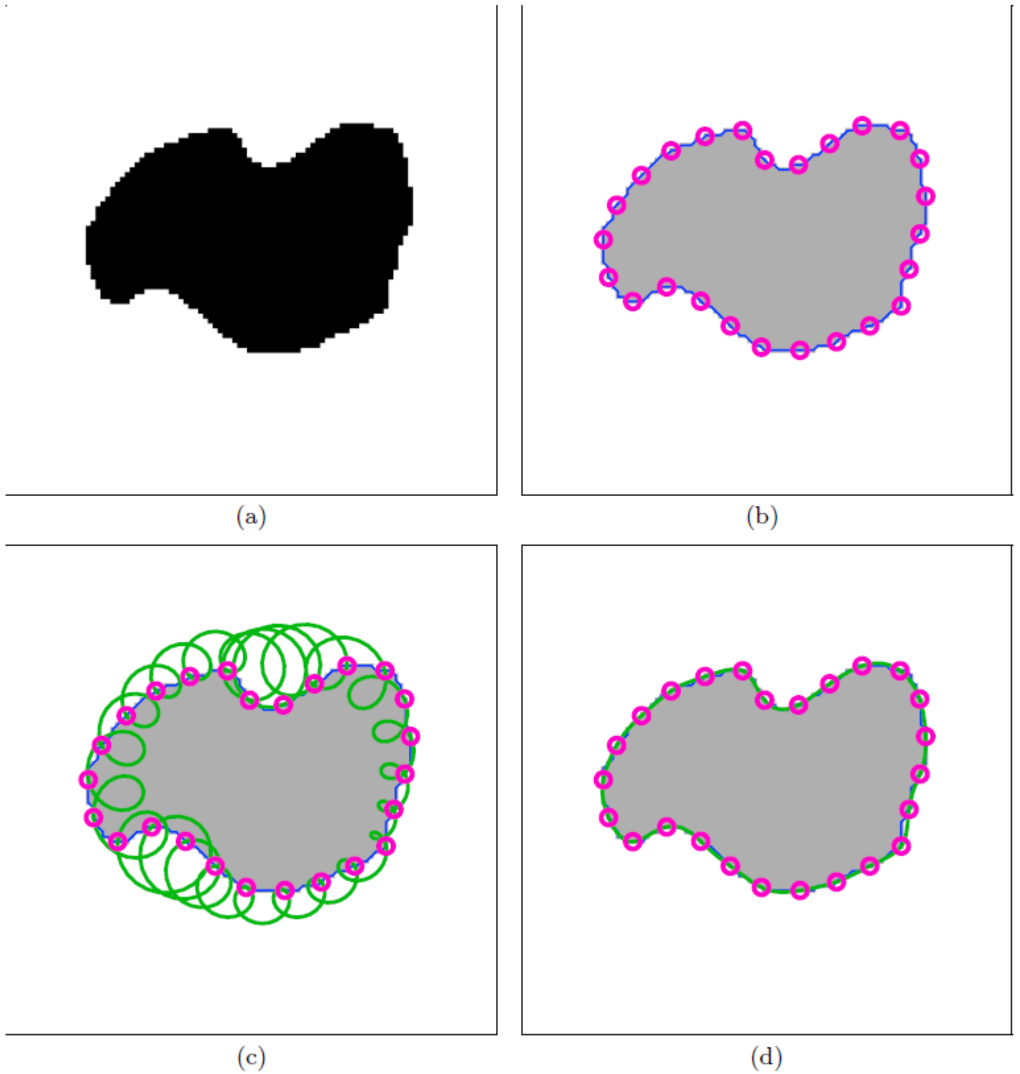


Figure 7.8 Partial reconstruction from single coefficients and descriptor pair. Circles reconstructed from DFT coefficient G_{-1} (a) and coefficient G_{+1} (b), placed at the centroid of the contour (G_0). The combined reconstruction for G_{-1}, G_{+1} gives the ellipse in (c). The dots on the green outlines show the path position for $t = 0$.

Full shape reconstruction from (all) Fourier descriptors

Original shape

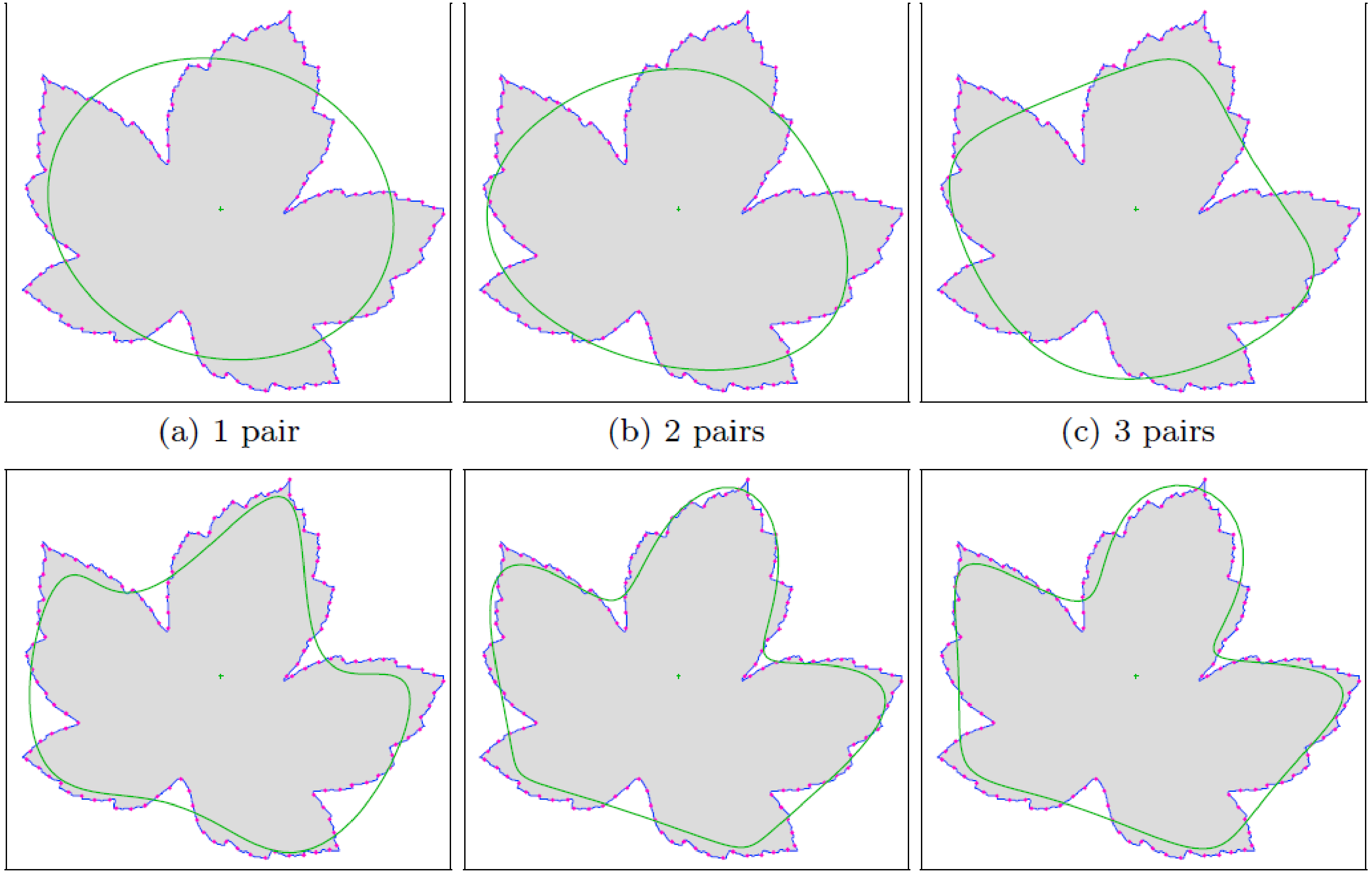


Original sample points

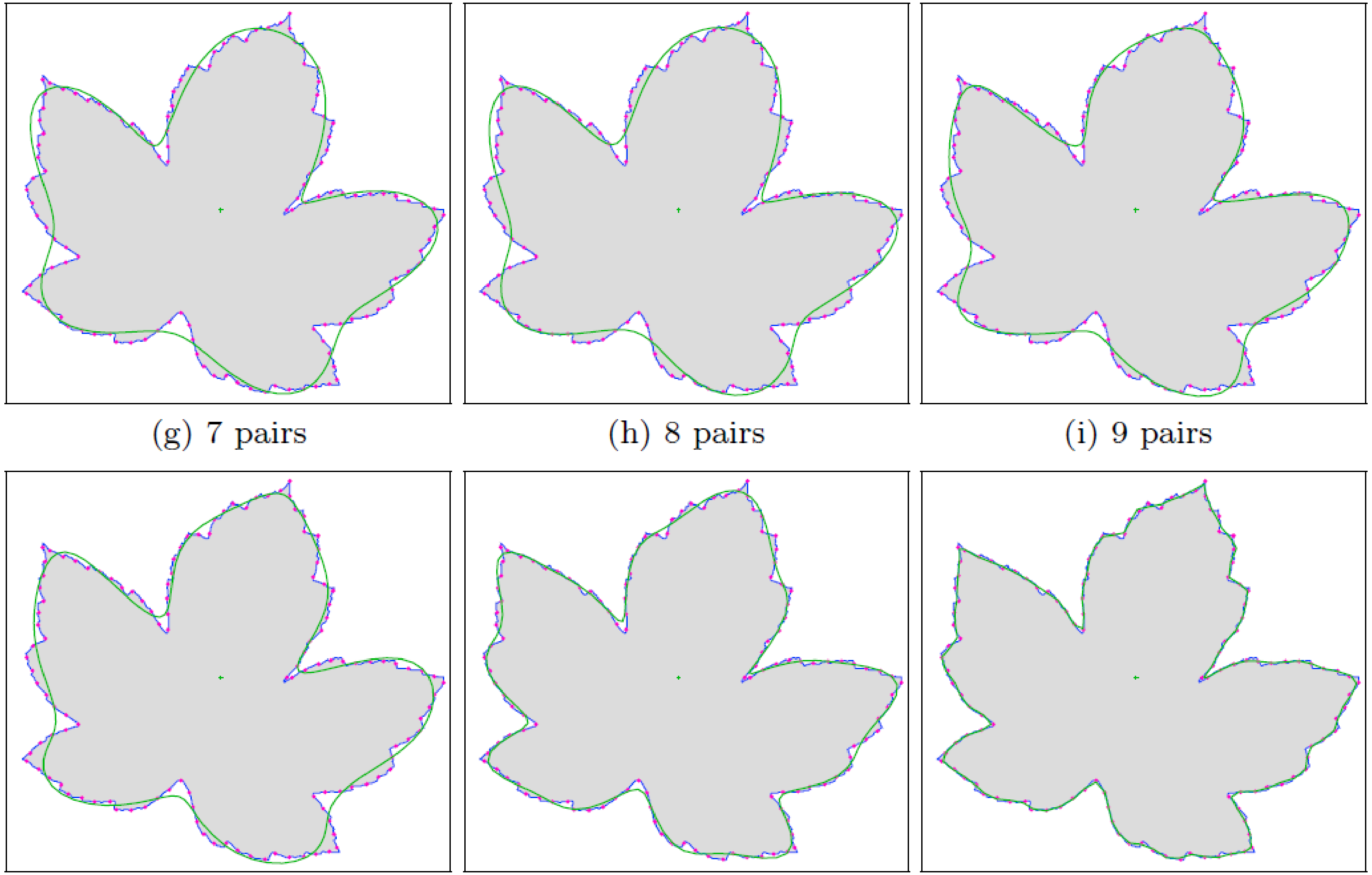
Reconstruction from **wrong** set of frequencies 0, …, *M*-1

Reconstruction from **proper** frequencies: -*M*/2, … +*M*/2

Reconstruction from a limited number of Fourier descriptor pairs (1)


(a) 1 pair (b) 2 pairs (c) 3 pairs (d) 4 pairs (e) 5 pairs (f) 6 pairs

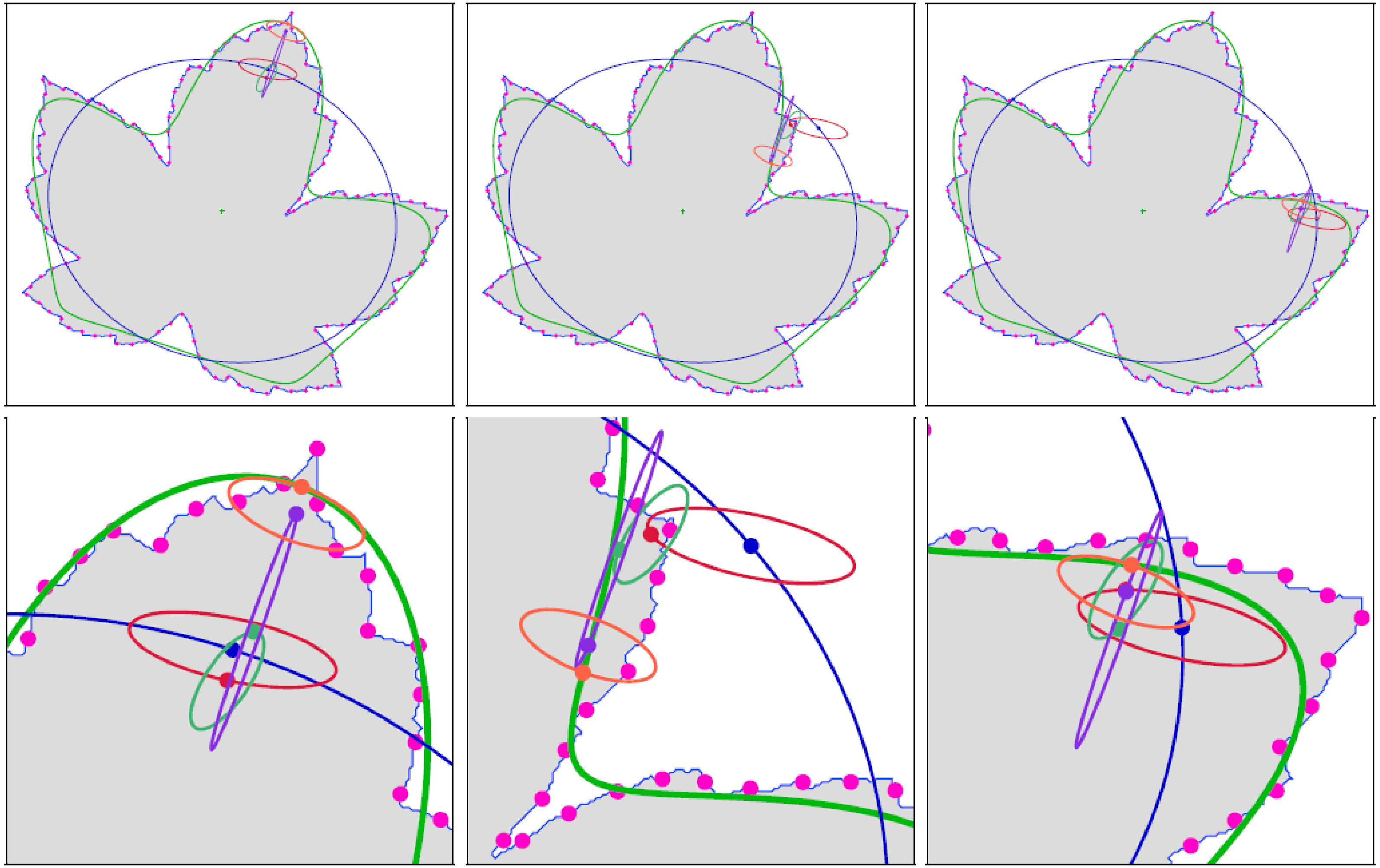
Reconstruction from a limited number of Fourier descriptor pairs (2)



(g) 7 pairs (h) 8 pairs (i) 9 pairs

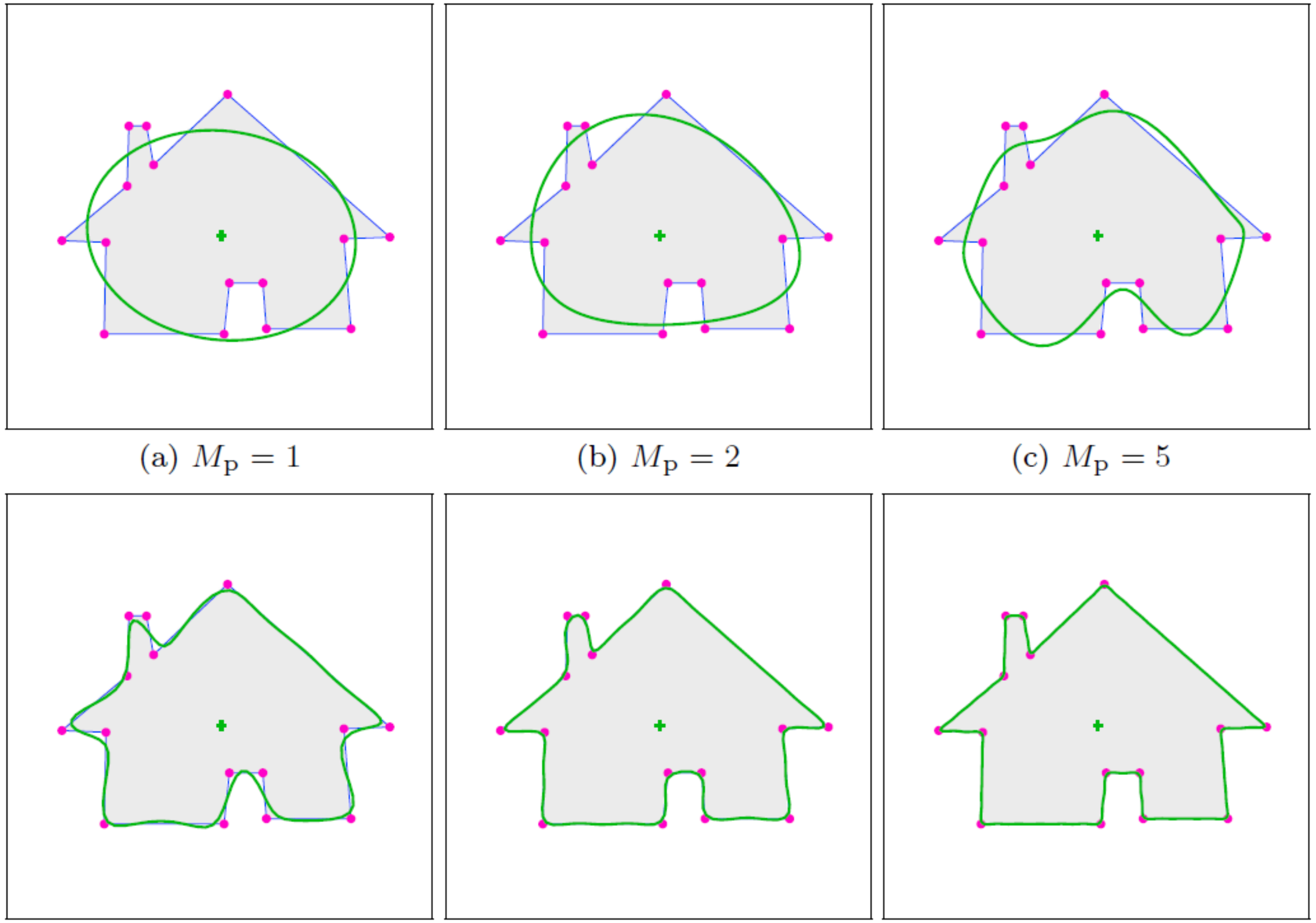
(j) 10 pairs (k) 15 pairs (l) 40 pairs

Reconstruction as superposition of ellipses (5 FD pairs)



(a) $t = 0.0$ (b) $t = 0.1$ (c) $t = 0.2$

Fourier desciptors from trigonometric data (polygon vertices)

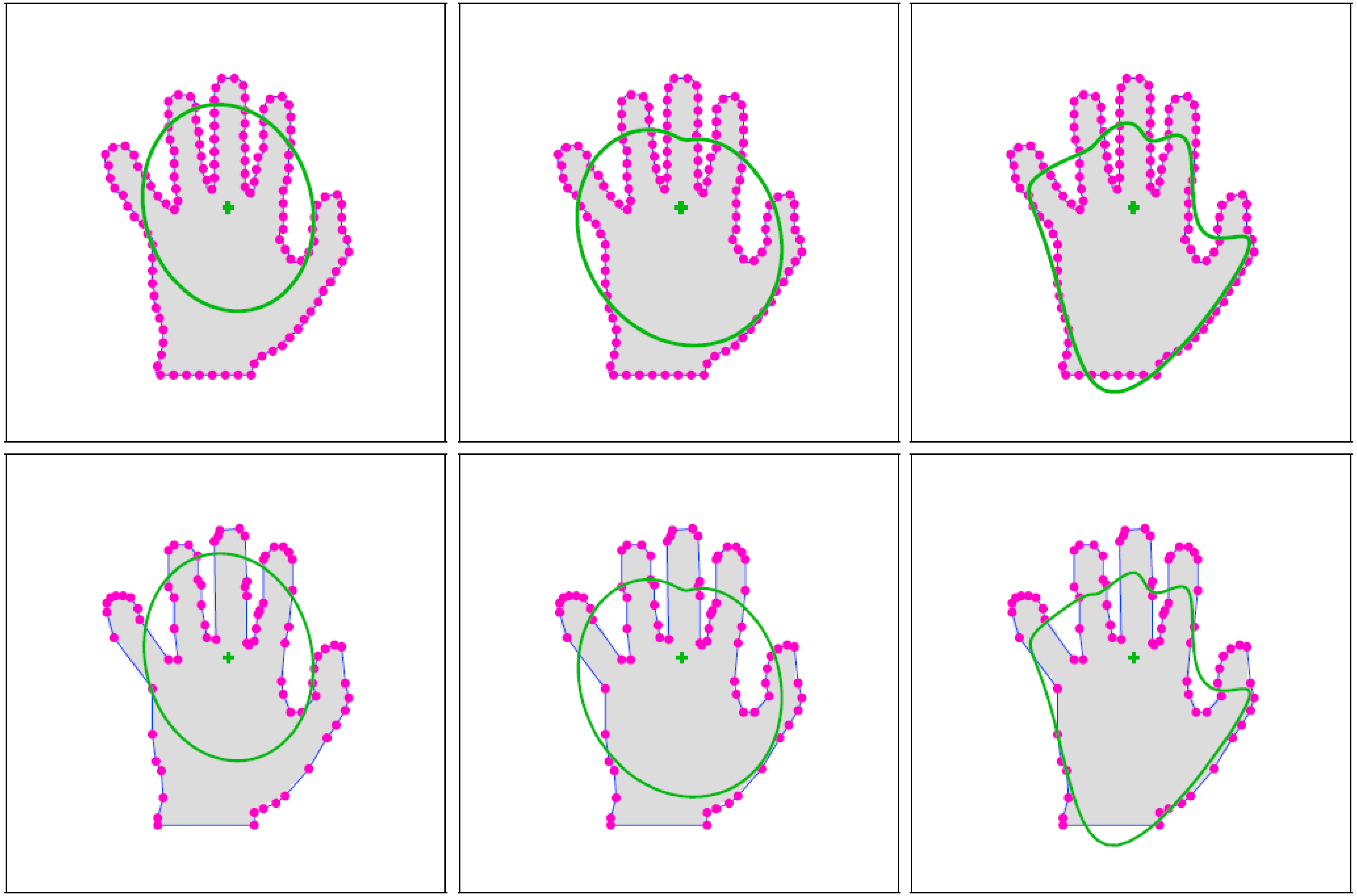


(a) $M_{\mathrm{p}} = 1$ (b) $M_{\mathrm{p}} = 2$ (c) $M_{\mathrm{p}} = 5$

(d) $M_{\mathrm{p}} = 10$ (e) $M_{\mathrm{p}} = 20$ (f) $M_{\mathrm{p}} = 50$

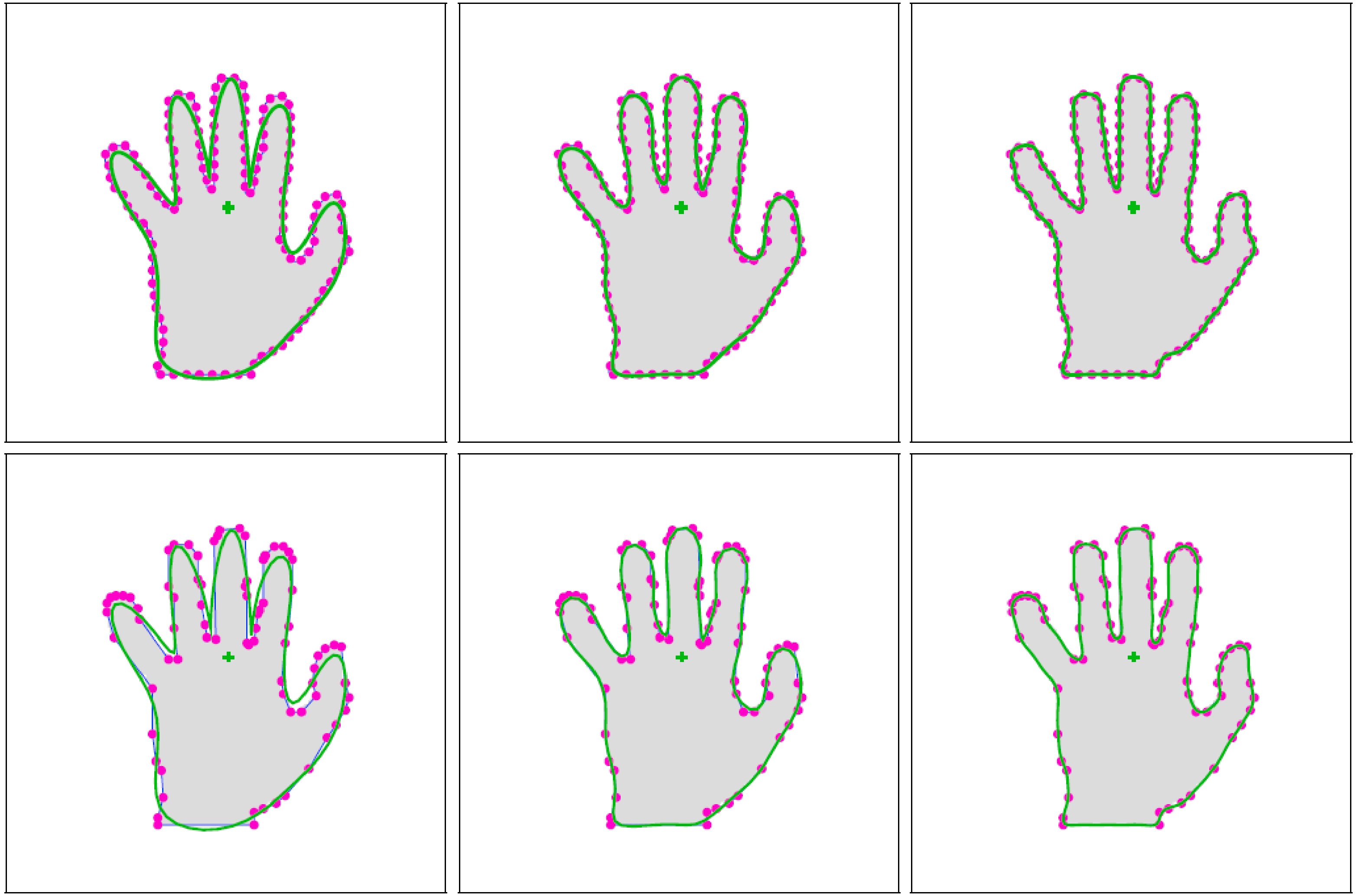
This technique requires **no regular sampling** of the contour path. FDs are calculated directly from the polygon vertices!

Fourier desciptors from trigonometric data (polygon vertices)



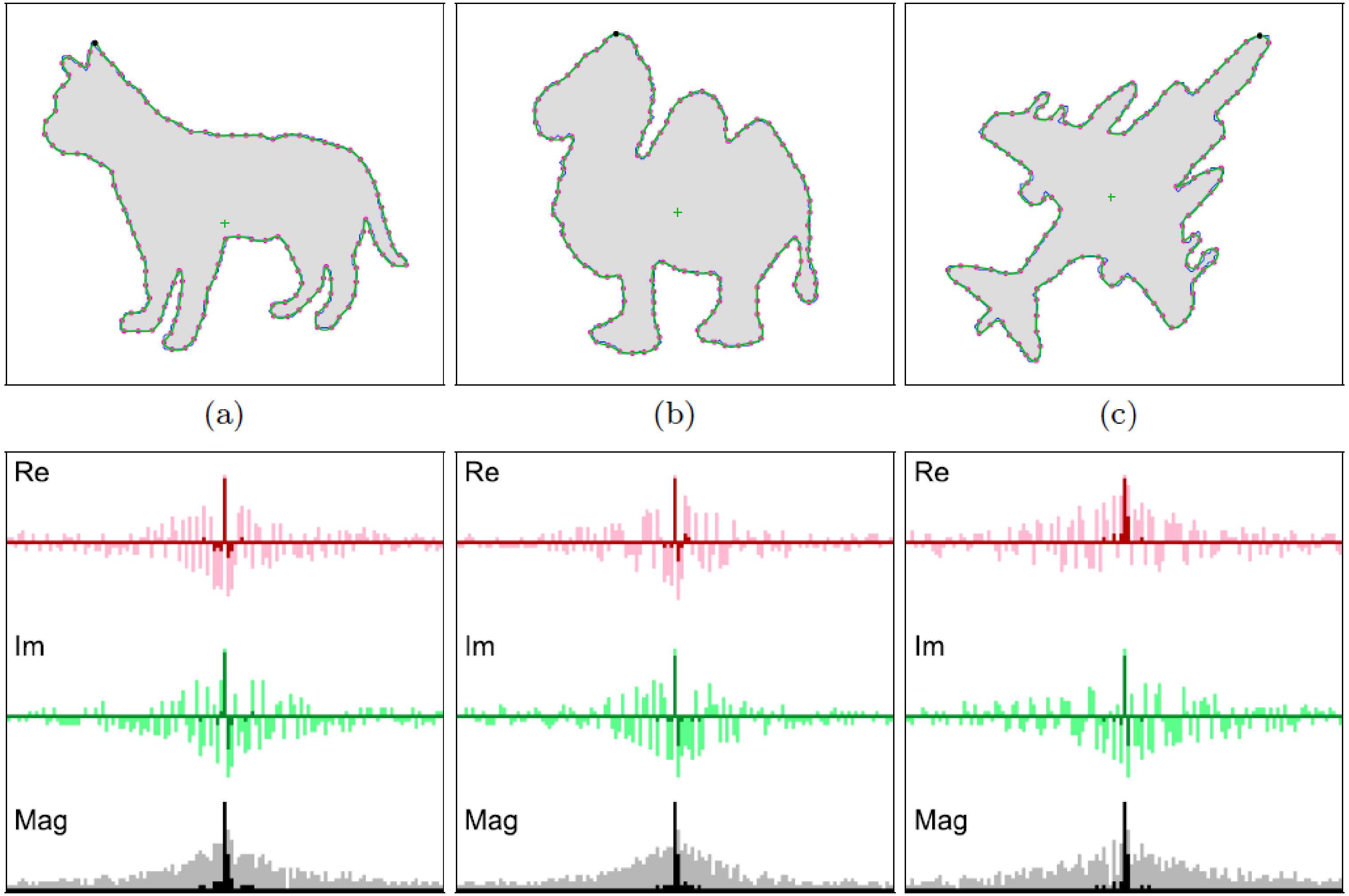
(a) $M_{\mathrm{p}} = 1$ (b) $M_{\mathrm{p}} = 2$ (c) $M_{\mathrm{p}} = 5$

Fourier desciptors from trigonometric data (polygon vertices)



(d) $M_{\mathrm{p}} = 10$ (e) $M_{\mathrm{p}} = 20$ (f) $M_{\mathrm{p}} = 50$

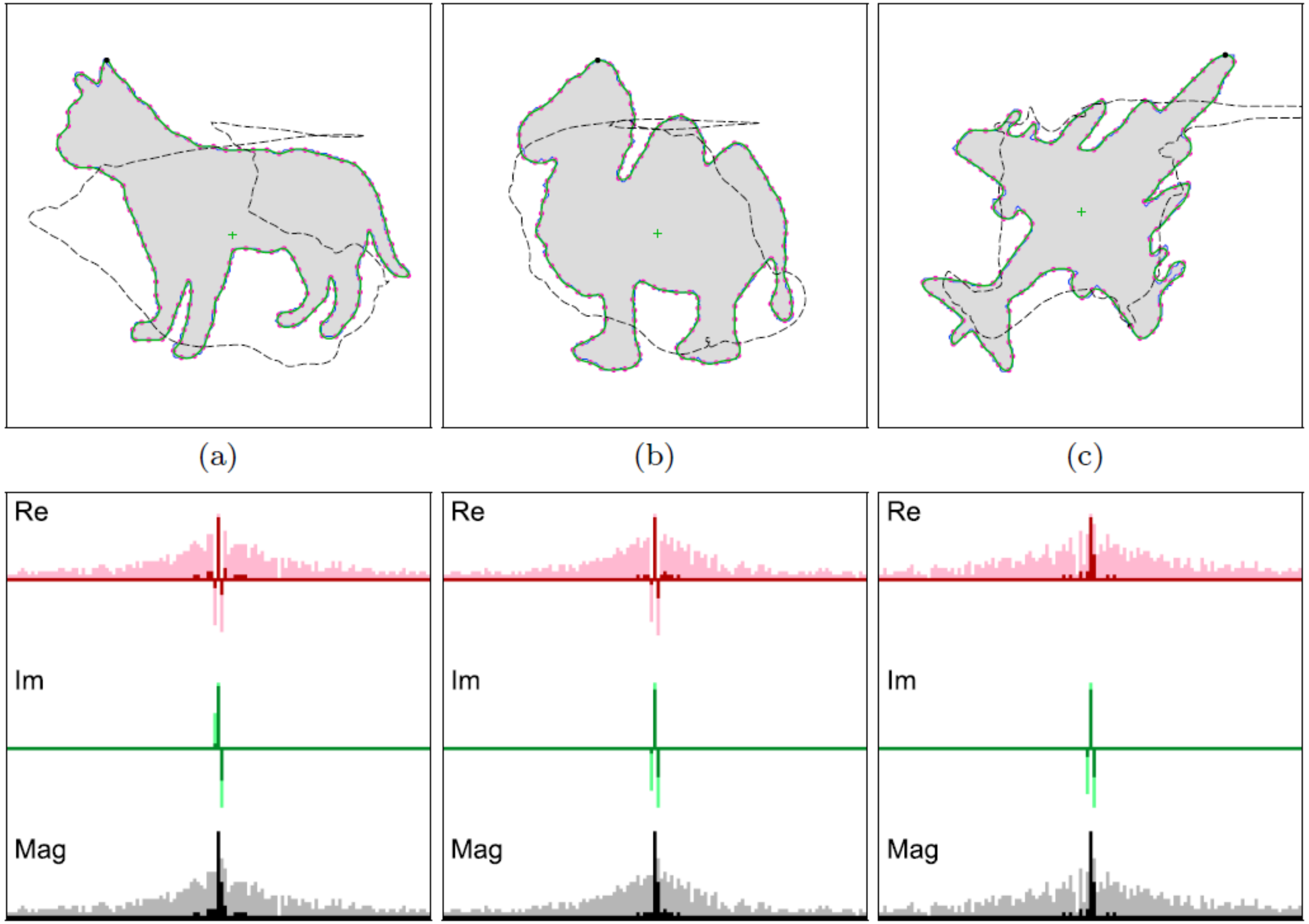
Making FDs invariant against geometric shape transformations



(a) (b) (c)

(d) (e) (f)

Effects of removing Fourier phase information



(a) (b) (c)
(d) (e) (f)

The phase of all coefficients (except 1st FD pair) is set to zero.

Effects of spatial transformations

Table 7.1 Effects of spatial transformations upon the corresponding DFT spectrum. The original boundary samples are denoted g_k, the DFT coefficients G_m.

Operation	Boundary samples	DFT coefficients
Forward transformation	g_k, for $k=0,\ldots,M-1$	$G_m = \frac{1}{M}\cdot\sum_{k=0}^{M-1} g_k \cdot e^{-\mathrm{i}2\pi m \frac{k}{M}}$
Backward transformation	$g_k = \sum_{m=0}^{M-1} G_m \cdot e^{\mathrm{i}2\pi m \frac{k}{M}}$	G_m, for $m=0,\ldots,M-1$
Translation (by $z \in \mathbb{C}$)	$g'_k = g_k + z$	$G'_m = \begin{cases} G_m + z & \text{for } m = 0 \\ G_m & \text{otherwise} \end{cases}$
Uniform scaling (by $s \in R$)	$g'_k = s \cdot g_k$	$G'_m = s \cdot G_m$
Rotation about the origin (by β)	$g'_k = e^{\mathrm{i}\cdot\beta} \cdot g_k$	$G'_m = e^{\mathrm{i}\cdot\beta} \cdot G_m$
Shift of start position (by k_s)	$g'_k = g_{(k+k_s) \bmod M}$	$G'_m = e^{\mathrm{i}\cdot m \cdot \frac{2\pi k_s}{M}} \cdot G_m$
Direction of contour traversal	$g'_k = g_{-k \bmod M}$	$G'_m = G_{-m \bmod M}$
Reflexion about the x-axis	$g'_k = g^*_k$	$G'_m = G^*_{-m \bmod M}$

Making Fourier Descriptors Invariant – Translation

- Translation-Invariance: **ignore** coefficient G_0 (or set to zero) – all other coefficients are independent of x/y-position.

Encoding / Comparing Shapes using Fourier Descriptors

- **Translation:** G_0 describes **absolute position** and can be ignored for shape comparison (set to zero).
- **Scale:** Normalize all coefficients with respect to the **magnitude** of G_1.
- **Rotation:** Normalize for shape rotation and boundary start position.
- **Two shapes** can then be **compared** using a limited number of Fourier coefficients (N), e.g. by …

Encoding / Comparing Shapes using Fourier Descriptors

1. **Magnitude only distance** (simple, no phase normalization required)

$$\mathrm{dist}_{\mathrm{M}}(\boldsymbol{G}_1, \boldsymbol{G}_2) = \Bigg(\sum_{\substack{m=-M_{\mathrm{p}},\\ m\neq 0}}^{M_{\mathrm{p}}} \big(|\boldsymbol{G}_1(m)| - |\boldsymbol{G}_2(m)| \big)^2 \Bigg)^{1/2}$$
$$= \Bigg(\sum_{m=1}^{M_{\mathrm{p}}} \big(|\boldsymbol{G}_1(-m)| - |\boldsymbol{G}_2(-m)| \big)^2 + \big(|\boldsymbol{G}_1(m)| - |\boldsymbol{G}_2(m)| \big)^2 \Bigg)^{1/2},$$

2. **Complex valued distance** (phase normalization is required!)

$$\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1, \boldsymbol{G}_2) = \Bigg(\sum_{\substack{m=-M_{\mathrm{p}},\\ m\neq 0}}^{M_{\mathrm{p}}} \big| \boldsymbol{G}_1(m) - \boldsymbol{G}_2(m) \big|^2 \Bigg)^{1/2}$$
$$= \Bigg(\sum_{m=1}^{M_{\mathrm{p}}} \big| \boldsymbol{G}_1(-m) - \boldsymbol{G}_2(-m) \big|^2 + \big| \boldsymbol{G}_1(m) - \boldsymbol{G}_2(m) \big|^2 \Bigg)^{1/2}$$

Use of *phase* in Fourier shape matching

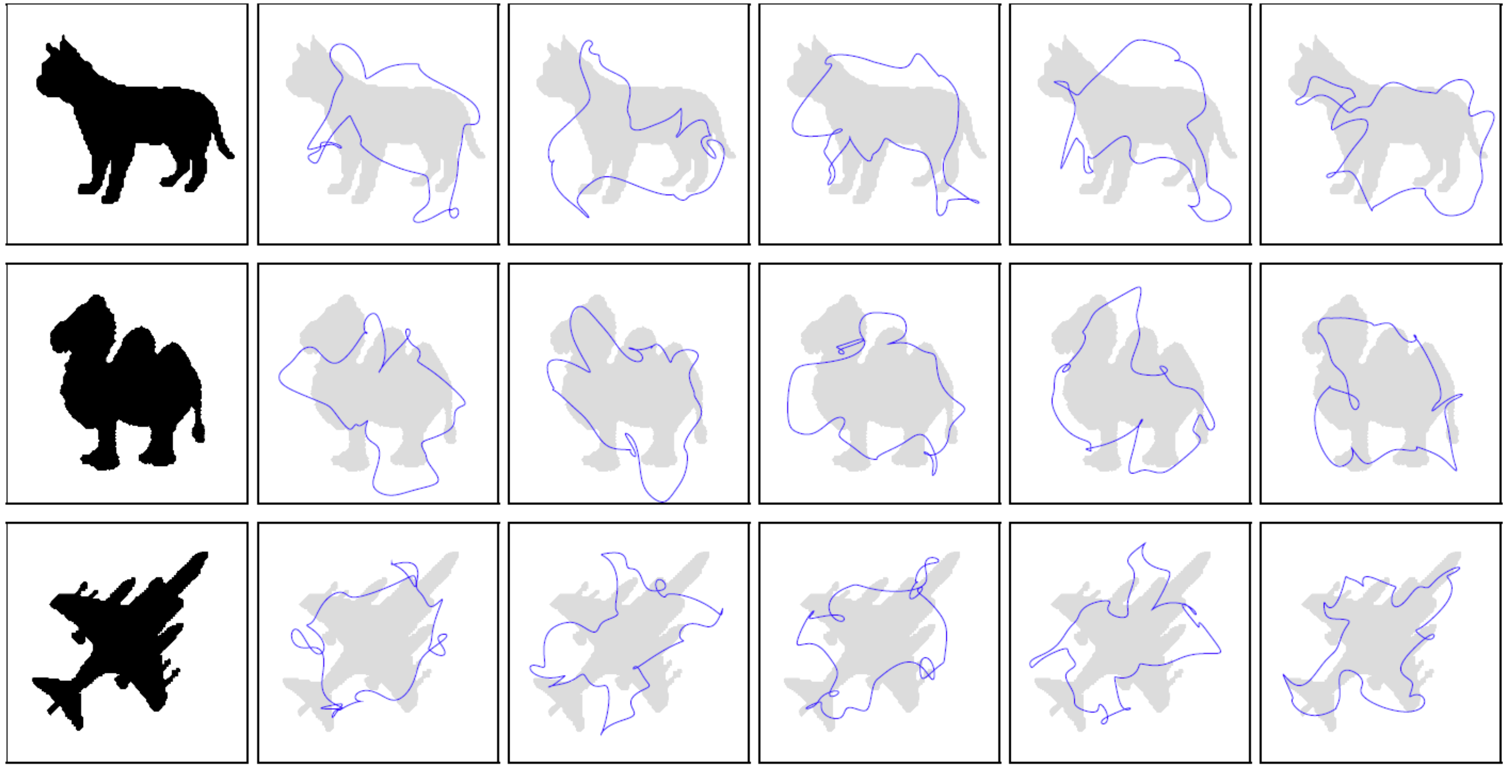
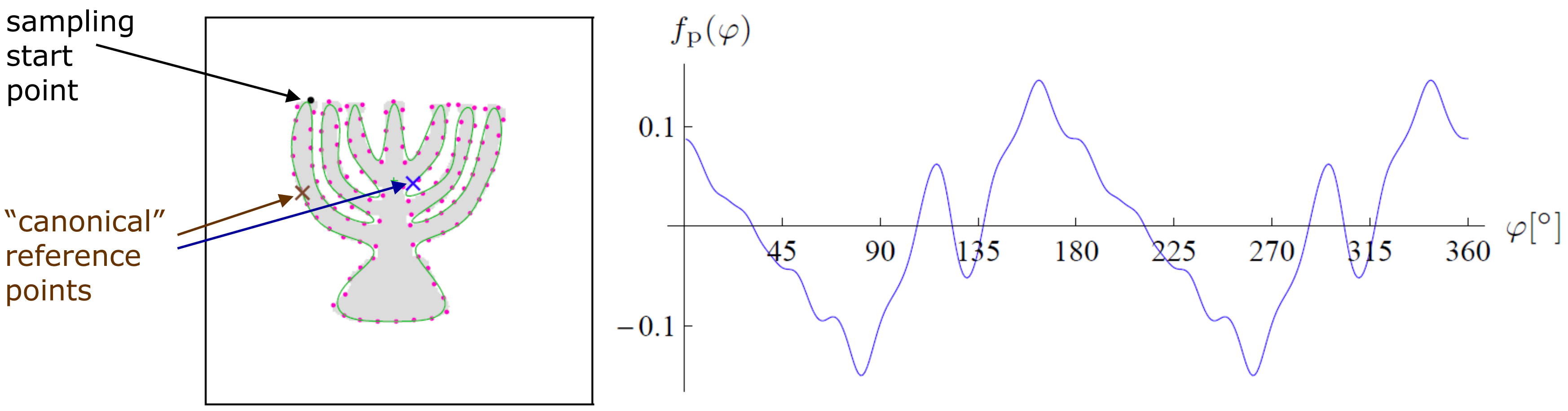
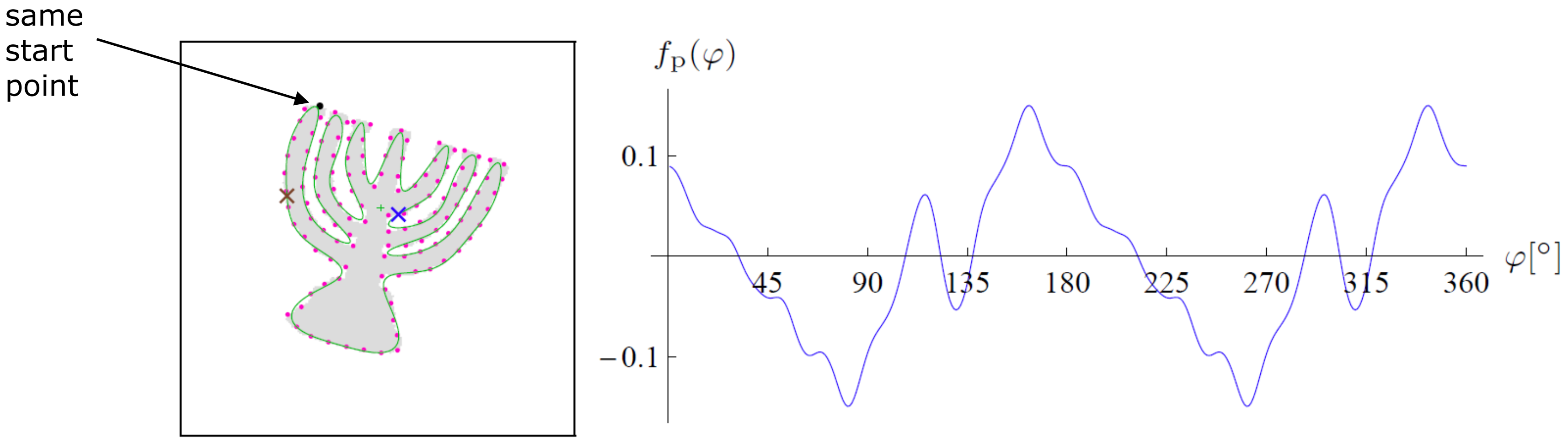


Figure 6.20 Randomized phase. Reconstruction of shapes from Fourier descriptors with the phase of all coefficients (except G_{-1}, G_0 and G_{+1}) individually randomized. Note that the magnitude of the coefficients is exactly the same for each shape category, so all blue shapes would be considered "equivalent" to the original shape at the left by a magnitude-only matcher.

Invariance against varying start point:
fixed start point under shape rotation

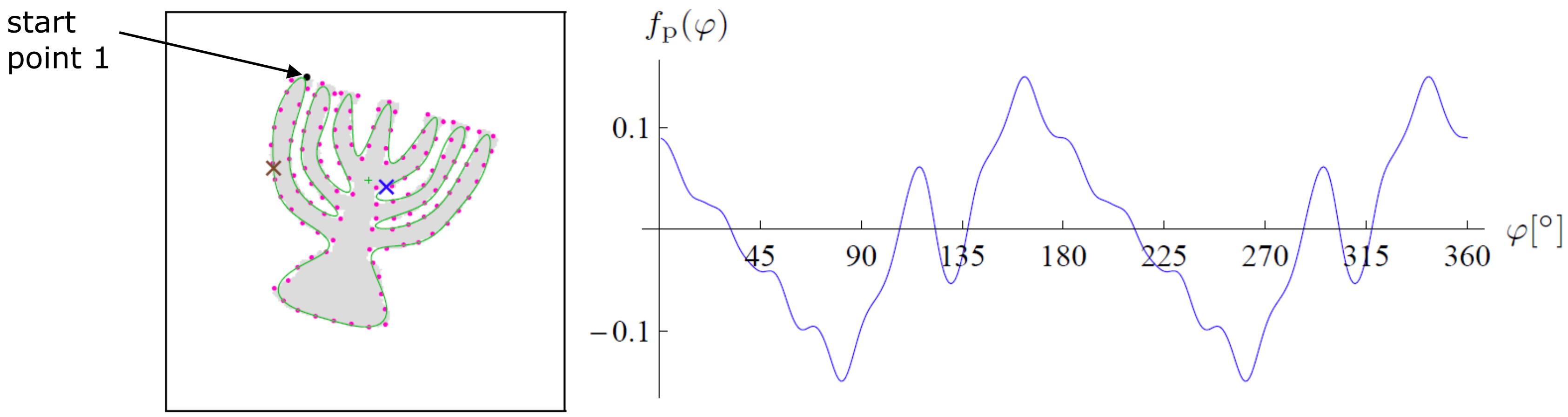


(a) rotation $\theta = 0°$, start point phase $\varphi_s = 0°$

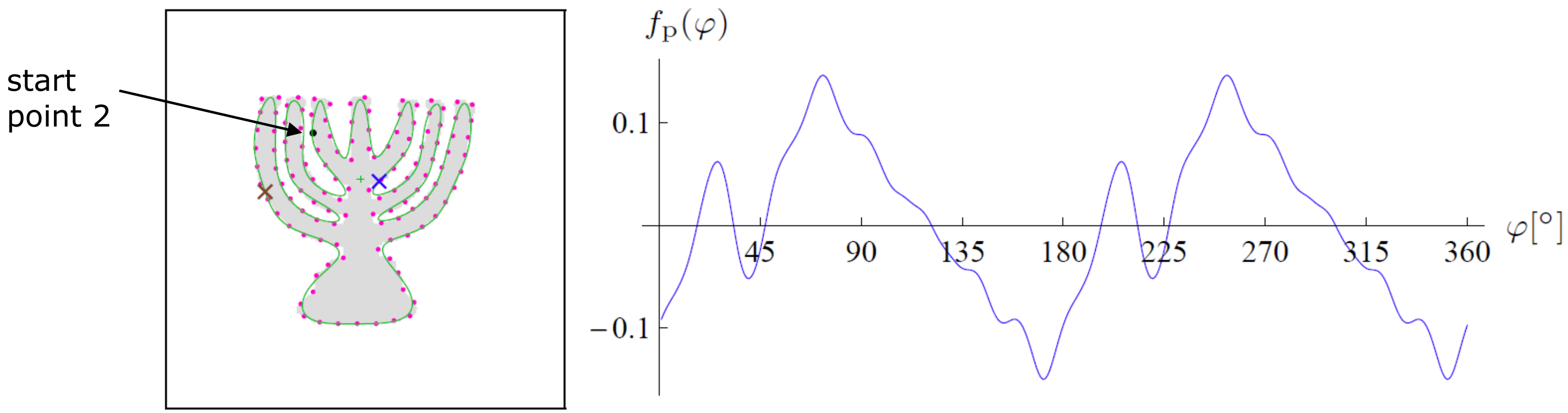


(b) rotation $\theta = 15°$, start point phase $\varphi_s = 0°$

Invariance against varying start point:
fixed start point under shape rotation

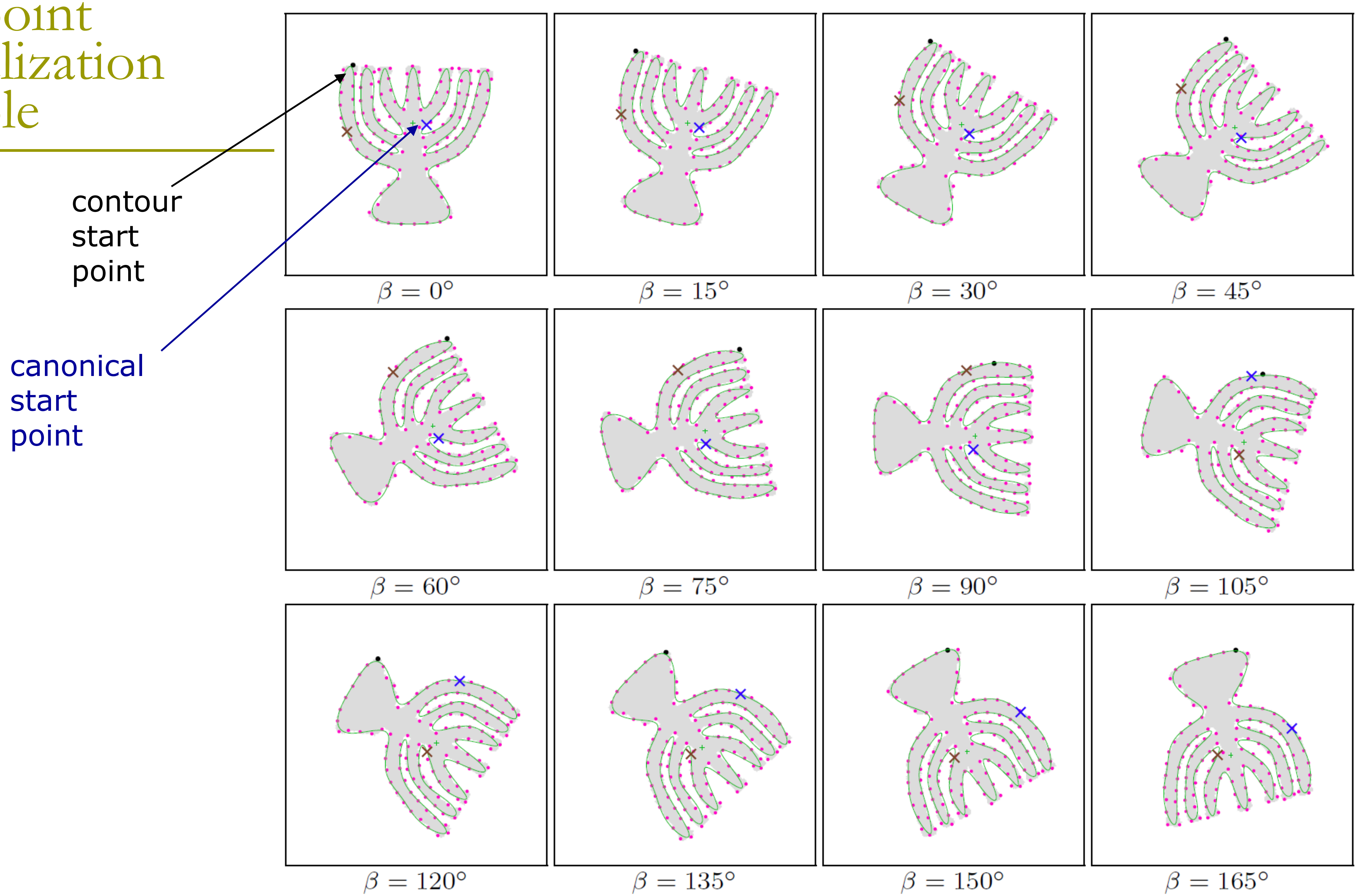


(b) rotation $\theta = 15°$, start point phase $\varphi_s = 0°$



(c) rotation $\theta = 0°$, start point phase $\varphi_s = 90°$

Start point normalization example



Orientation is not unique – we actually need 2 reference descriptors

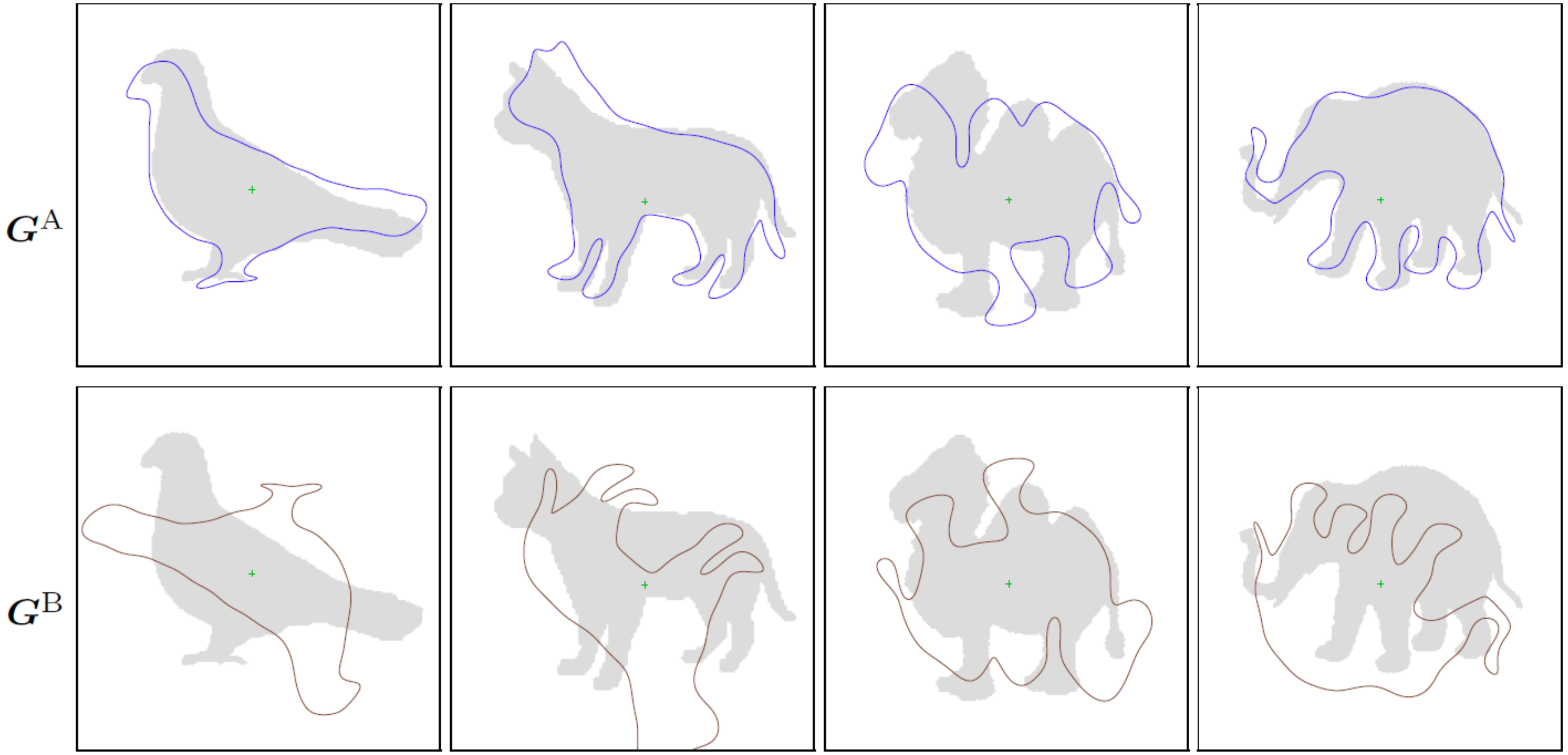


Figure 6.19 Reconstruction of various shapes from Fourier descriptors normalized for start point shift and shape rotation. The blue shapes (rows 1, 3) correspond to the normalized Fourier descriptors G^A with start point phase φ_A. The brown shapes (rows 2, 4) correspond to the normalized Fourier descriptors G^B with start point phase $\varphi_B = \varphi_A + \pi$. No scale normalization was applied for better visualization.

Inter-class distances

complex-valued (**with** phase)

magnitude-only (**without** phase)

Distances should be large between shapes of different categories (off-diagonal).

	bird	cat	camel	elephant	hand	harrier	menora	piano	creature
	0.000	4.529	4.482	5.007	5.525	4.314	7.554	5.174	7.076
	3.156	0.000	5.788	4.708	5.711	5.701	7.181	5.543	7.677
	2.648	3.005	0.000	4.429	5.573	3.726	7.014	4.013	8.480
	3.487	1.933	2.549	0.000	6.100	4.618	5.338	4.369	8.743
	4.627	3.146	3.132	2.372	0.000	6.079	8.540	5.580	7.136
	3.712	3.707	2.687	3.553	4.294	0.000	6.818	4.958	8.284
	5.835	4.893	4.563	4.162	3.788	5.775	0.000	6.826	11.072
	4.037	2.426	2.610	1.876	1.848	3.405	4.315	0.000	7.666
	6.030	6.261	5.554	5.492	5.955	5.914	5.190	6.049	0.000

$\mathrm{dist}_{\mathrm{M}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$ $\mathrm{dist}_{\mathrm{C}}(\boldsymbol{G}_1, \boldsymbol{G}_2)$

Intra-class distances

$\alpha =$	0°	17°	34°	51°	68°	85°	102°	119°	136°	153°	170°	187°	204°
													
dist_M	0.000	0.070	0.126	0.151	0.103	0.058	0.143	0.107	0.195	0.190	0.105	0.078	0.053
dist_C	0.000	0.141	0.222	0.299	0.198	0.111	0.274	0.159	0.313	0.400	0.142	0.162	0.092
													
dist_M	0.000	0.134	0.144	0.176	0.167	0.055	0.104	0.206	0.227	0.135	0.164	0.083	0.174
dist_C	0.000	0.222	0.214	0.252	0.244	0.081	0.141	0.310	0.339	0.197	0.231	0.157	0.281
													
dist_M	0.000	0.117	0.346	0.147	0.142	0.141	0.109	0.100	0.125	0.163	0.099	0.147	0.106
dist_C	0.000	0.229	0.728	0.367	0.310	0.386	0.161	0.186	0.202	0.252	0.141	0.191	0.271
													
dist_M	0.000	0.121	0.195	0.272	0.170	0.057	0.135	0.175	0.216	0.176	0.092	0.112	0.160
dist_C	0.000	0.180	0.317	0.392	0.278	0.080	0.218	0.257	0.307	0.266	0.160	0.198	0.248
													
dist_M	0.000	0.127	0.138	0.179	0.130	0.048	0.131	0.115	0.329	0.173	0.202	0.109	0.132
dist_C	0.000	0.179	0.186	0.361	0.180	0.085	0.234	0.188	0.496	0.263	0.313	0.182	0.195

Distances should be **small** between shapes of the **same** category.

magnitude-only (**without** phase)

complex-valued (**with** phase)