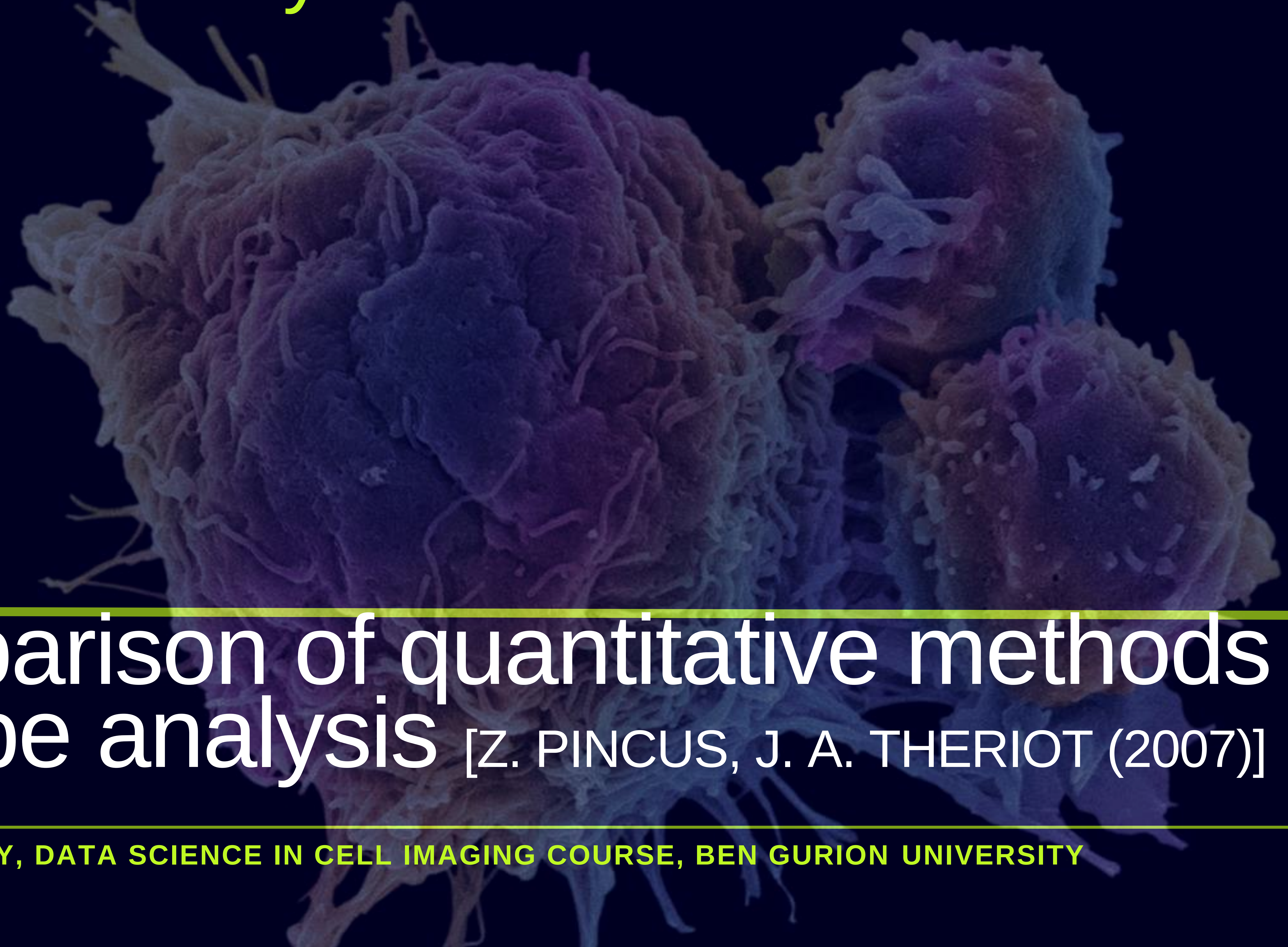
Paper Summary



Comparison of quantitative methods for cell - shape analysis [Z. PINCUS, J. A. THERIOT (2007)]

YISHAIA ZABARY, DATA SCIENCE IN CELL IMAGING COURSE, BEN GURION UNIVERSITY

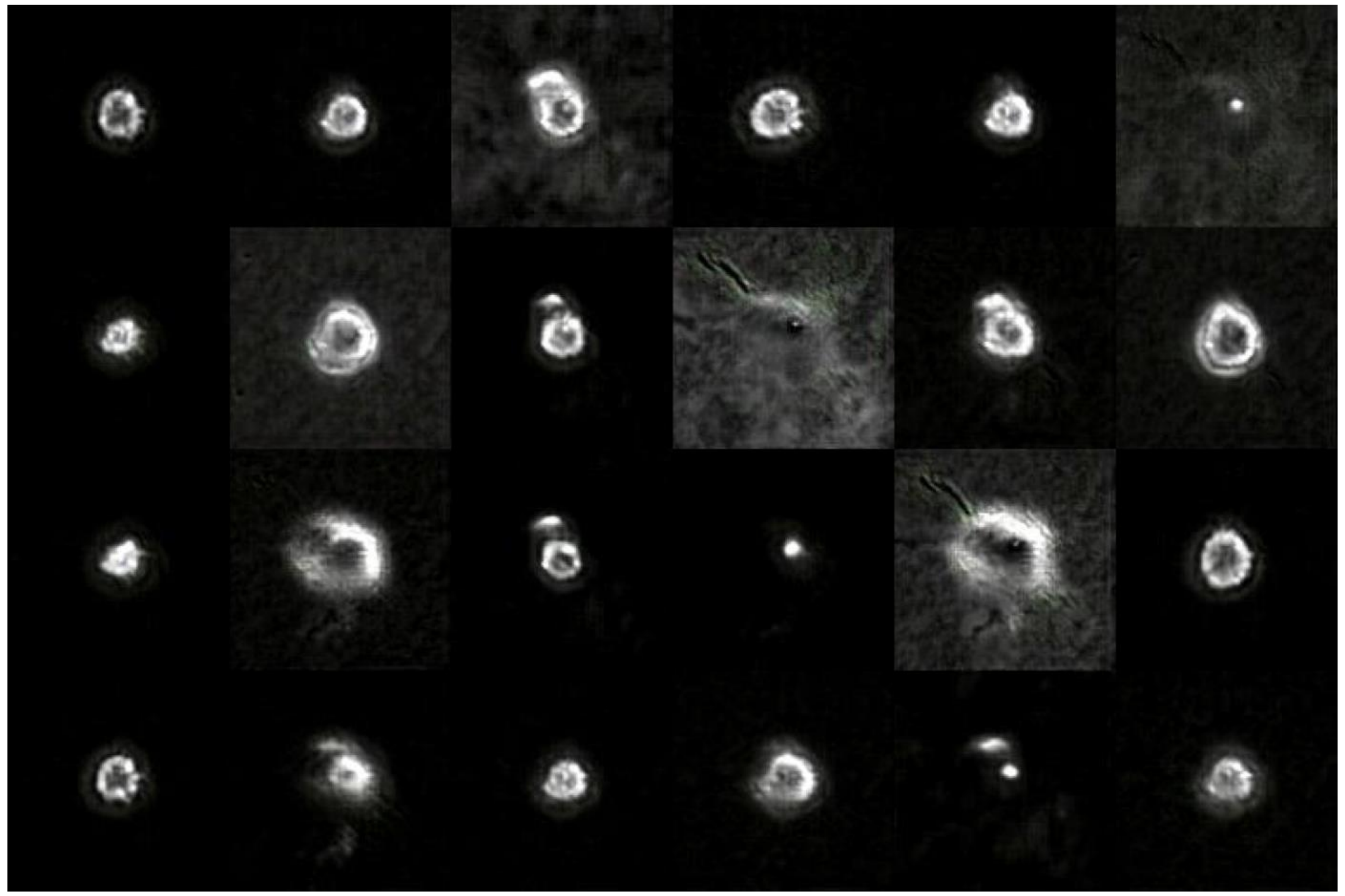
Previous Paper Summaries

- Extract ***data points*** from microscopy image stacks about the cells.
- Shallow/Deep learning model.
- Results!

Wait….what data?

Why cell shape is important?

- Cell morphology is often functionality related.
- Some cancer cells even change shape between different stages.



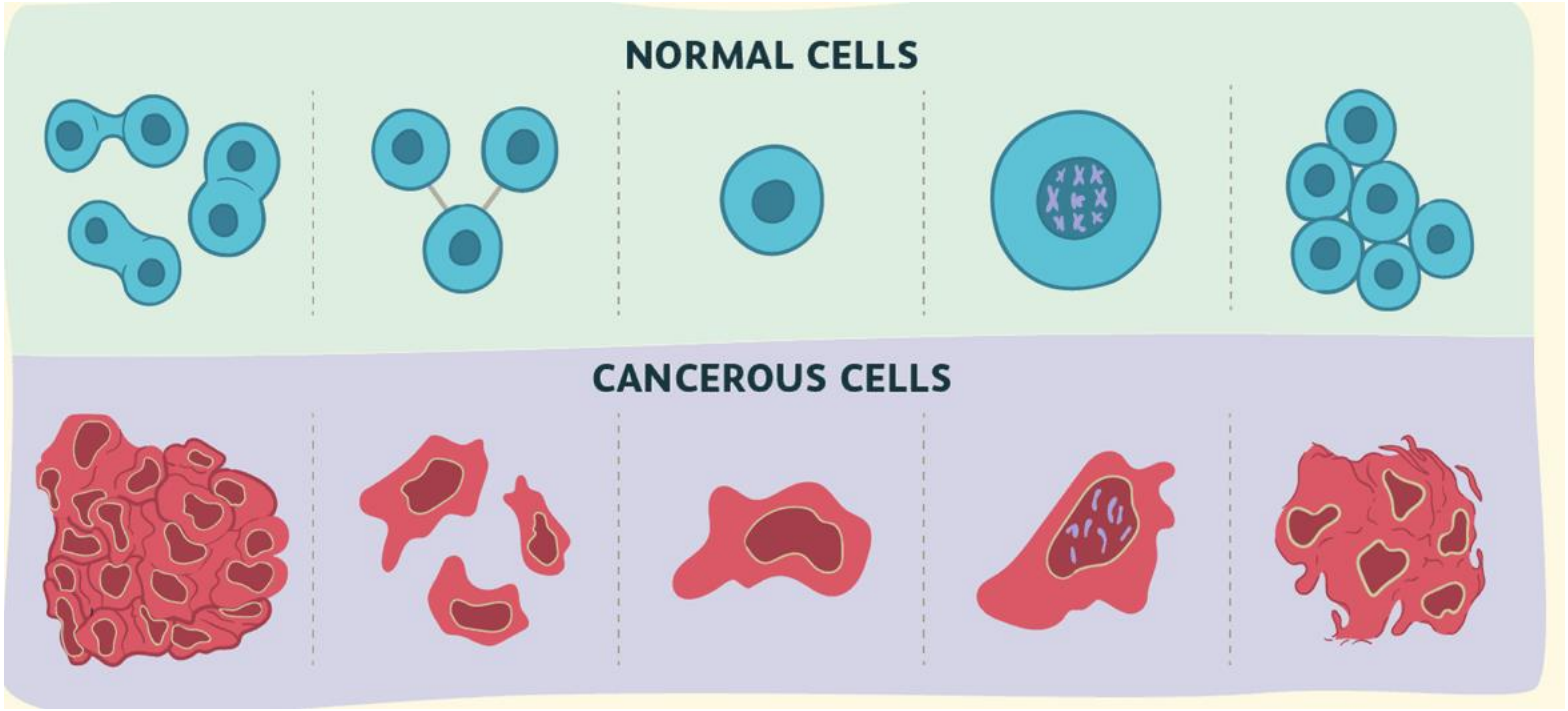


Image taken from verywellhealth.com

Quantifying a shape

- We need features that encode the shape into numbers (a vector).
- Features pre-requisites (i.e. criteria for good features):
  1. Fidelity
  2. Meaningful
  3. Interpretable

Feature Criterion No. 1 - Fidelity

- Encoded data must faithfully encode the shape, no added information.
- Good methods will discard a lot of the shape data, as it is un-relevant.

Feature Criterion No. 2 - Meaningful

- Encoded information must be **biologically** meaningful.
- Often if not always task specific.

Feature Criterion No. 3 - Interpretable

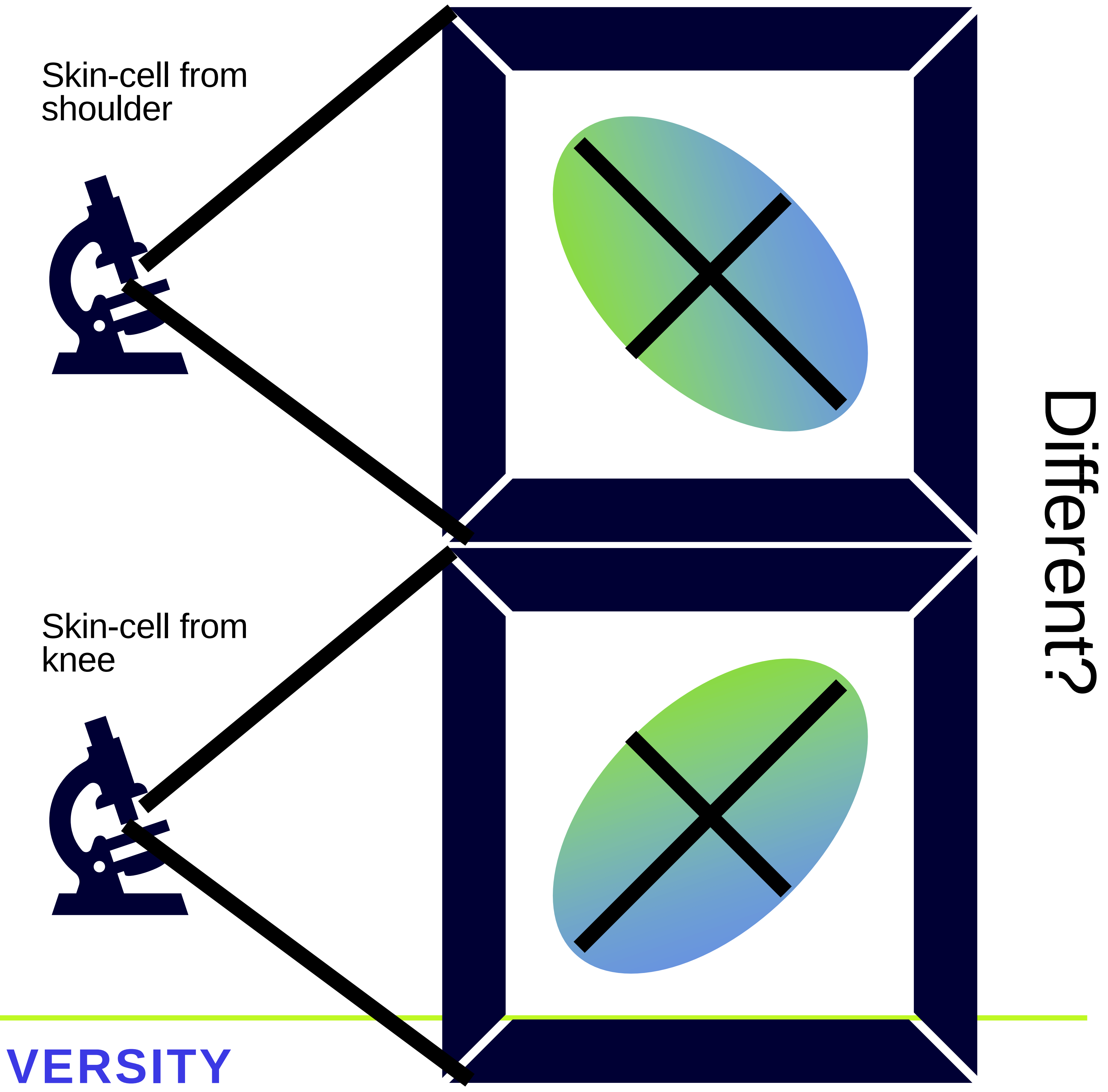
- Extension of the second criterion.
- Intuitive measurements.
- Example - for self-propelled movement, a good feature might be the ratio between the radius and circumference of the cell.

Intuitive, simple, shape features

- height, width, alignment, circularity.
- All are low on fidelity.
- Most low on meaningfulness as well (But very task specific!).

Shape numeric representation

1. Segmentation.
2. Dimensions re-representation.



Segmentation, Binary Masks & Distance Heat Maps

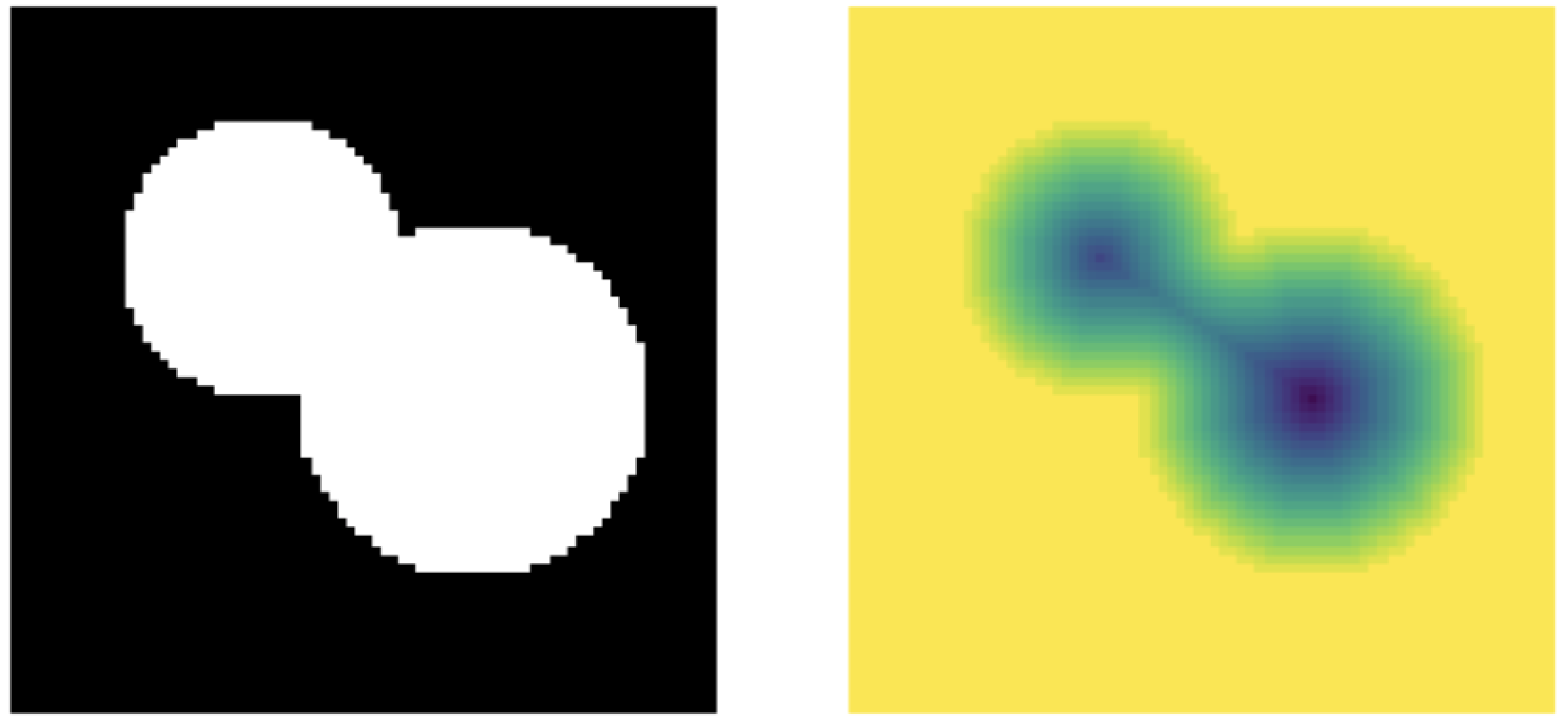


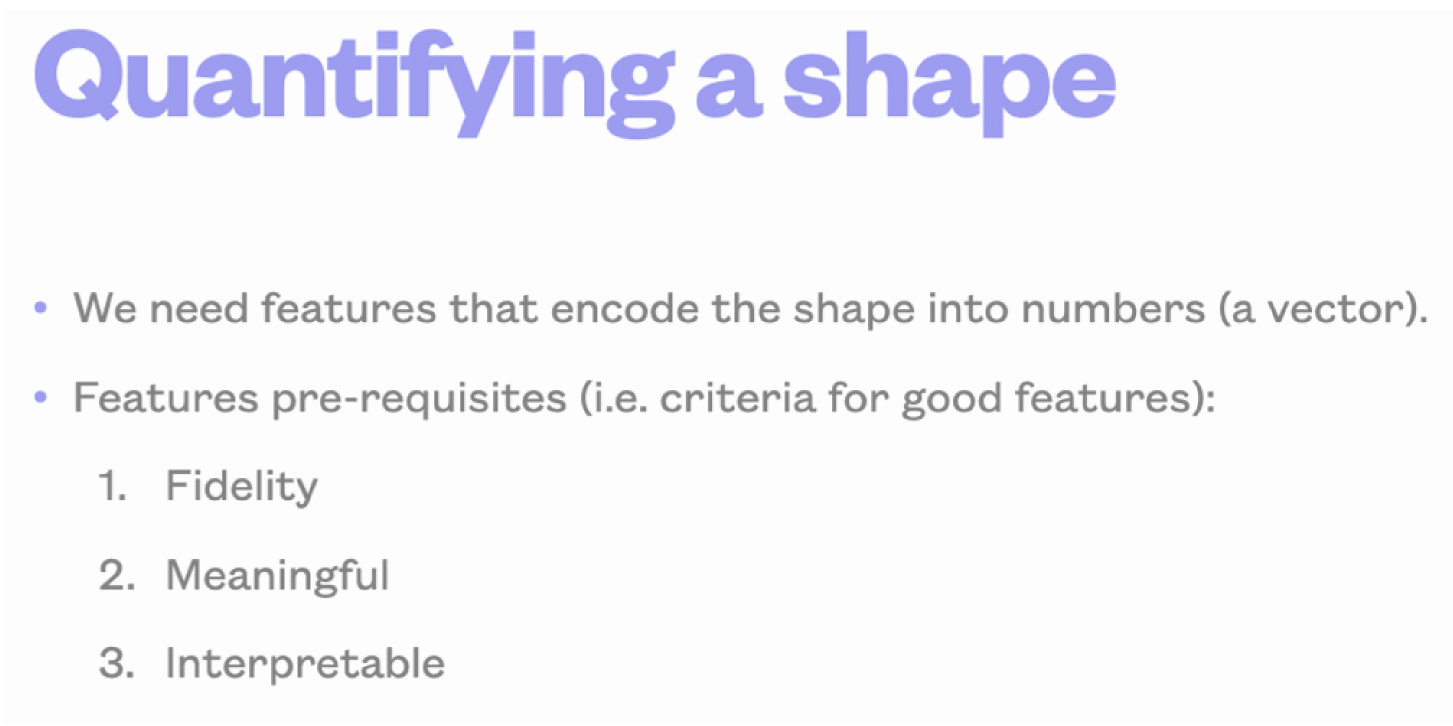
Image taken from scikit-image documentation

Binary masks	Distance heat maps
Explicit boundary	Implicit boundary
Contains a lot of redundant data	Contains a lot of redundant data
Does not use intensity.	Considers intensity as a significant feature.
Demands same topology and can cause many errors	Prevalent in the computer vision community and easy to implement

Re-representation of shape data

General notion:

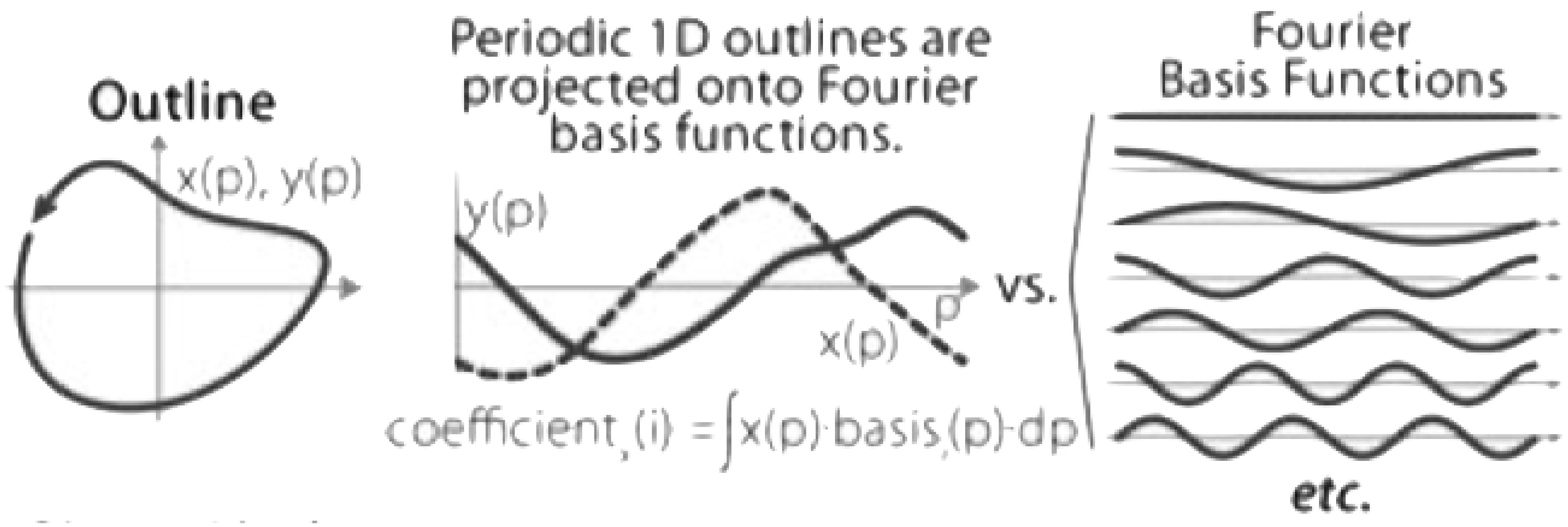
- Transform the representation to a higher dimensional space.
- Choose the best dimensions that balance fidelity and compactness.


Quantifying a shape

- We need features that encode the shape into numbers (a vector).
- Features pre-requisites (i.e. criteria for good features):
  1. Fidelity
  2. Meaningful
  3. Interpretable

Fourier & Linear Projections

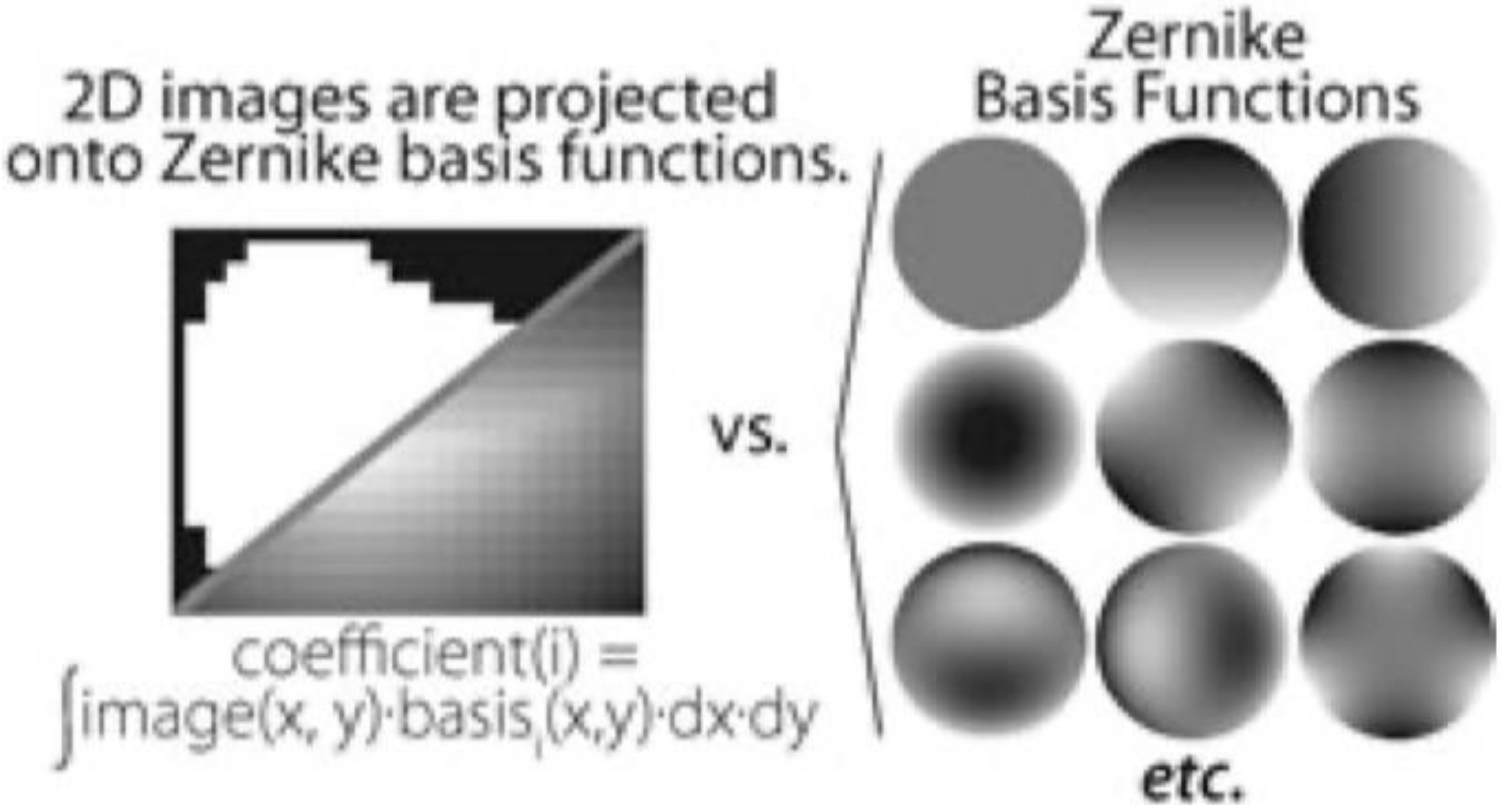
- Data must be periodic (cell periphery is a closed loop).
- Doesn’t necessarily corresponds to biological phenomena.
- Usually used when shape is explicit.



Zernike Polynomials

OR…FOURIER 2.0

- Originally developed for optical-engineering.
- Represents best circularly smooth shapes.
- Zernike base functions are orthogonal.
- Stacking together Zernike polynomial representations (as separate features) increases fidelity for irregular shapes.



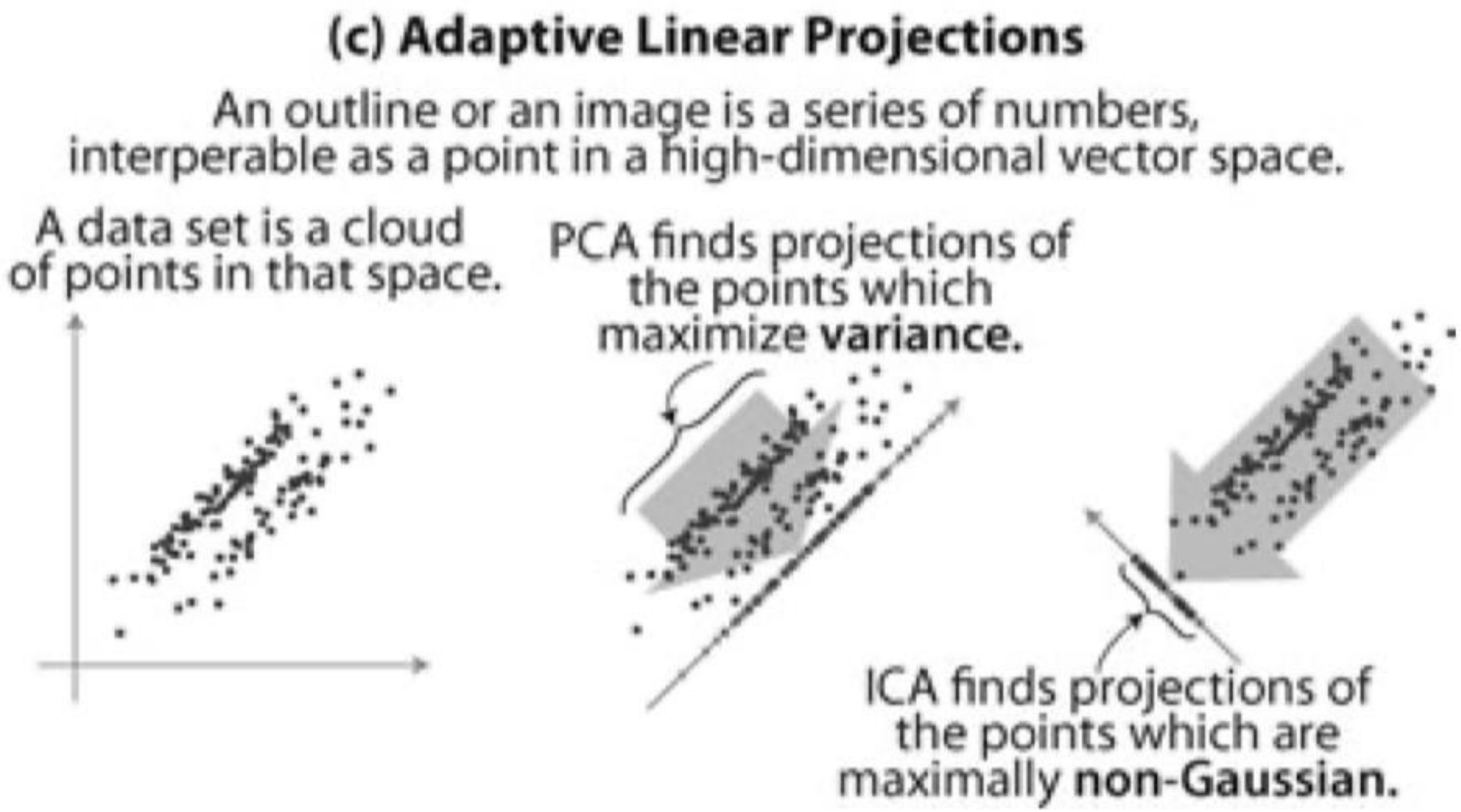
PCA & LDA Principal Component Analysis, Linear Discriminant analysis.

- Input: multi-dimensional dataset.
- Output: description of dataset about linear, orthogonal axis using the most accountable dimensions for variations in the data (by order of accountability).
  LDA attempts to also maximise distance between labeled groups.
- Choose to -n- most significant (‘accountable’) axes for less complex shape models (SVM).
- Both were widely used in face-recognition(prior to deep-learning methods).

ICA Independent Component Analysis

- Same input as PCA & LDA.
- Main idea: seek dimensions in which the data representation is not normal.
- ‘Unmix’ data to original (unknown) sources (i.e. cocktail party problem).
- Extracted component are not orthogonal.
- Also applied successfully to face recognition task.

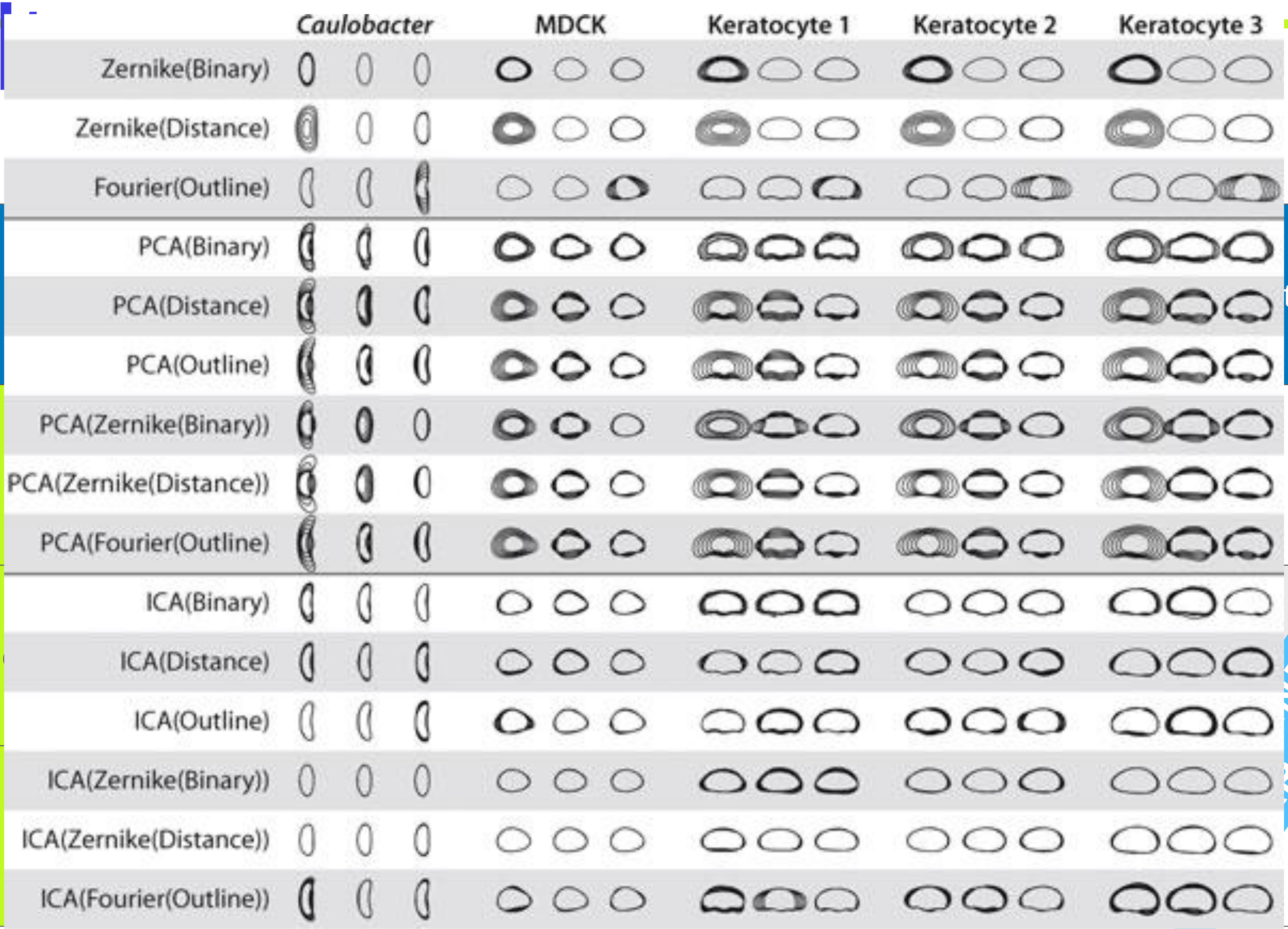
ICA & PCA differences



Comparing methods

- Reconstruction Fidelity.
- Groups separation.
- Features embedded in obtained signal.

Resul



	teroperable
PCA	-
Zernike Polyn	
ICA	-

NDS ON THE
RCHER & THE DATA

See for yourself

- PCA - https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- LDA - https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html
- ICA - https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html
- Zernike polynomials - https://pypi.org/project/opticspy/