

# Kapitel 5c

## Berechnung von Invarianten für diskrete Objekte

# Invariants for Discrete Structures – An Extension of Haar Integrals over Transformation Groups to Dirac Delta Functions

Hans Burkhardt <sup>1</sup>, Marco Reisert <sup>1</sup>, and Hongdong Li <sup>2</sup>

<sup>1</sup>University of Freiburg, Computer Science Department, Germany

<sup>2</sup>National ICT Australia (NICTA), Australian National University, Canberra  
ACT, Australia

<http://lmb.informatik.uni-freiburg.de/>

In C. E. Rasmussen and H. H. Bühlhoff and M. A. Giese and B. Schölkopf, editors *Proceedings of the 26th DAGM Symposium*,  
Tübingen, Germany, Aug./Sep. 2004.

# Summary

1. Introduction
- ~~2. Invariants for continuous objects~~
3. Invariants for discrete objects
  - Invariants for polygons
  - 3D-meshes
  - Discrimination performance and completeness
4. Experiments: Object classification in a Tangram database
5. Conclusions

# Introduction

- Increased interest in 3D models and 3D sensors induce a growing need to support e.g. the automatic search in such databases
- As the description of 3D objects is not canonical  
→ use invariants for their description

# Invariant integration over Euclidean group

For (cyclic) image translation and rotation:

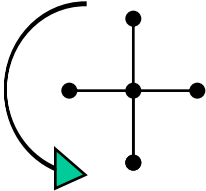
$$(g\mathbf{X})[i, j] = \mathbf{X}[k, l]$$

$$\begin{pmatrix} k \\ l \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} t_0 \\ t_1 \end{pmatrix}$$

all indices to be understood modulo the image dimensions.

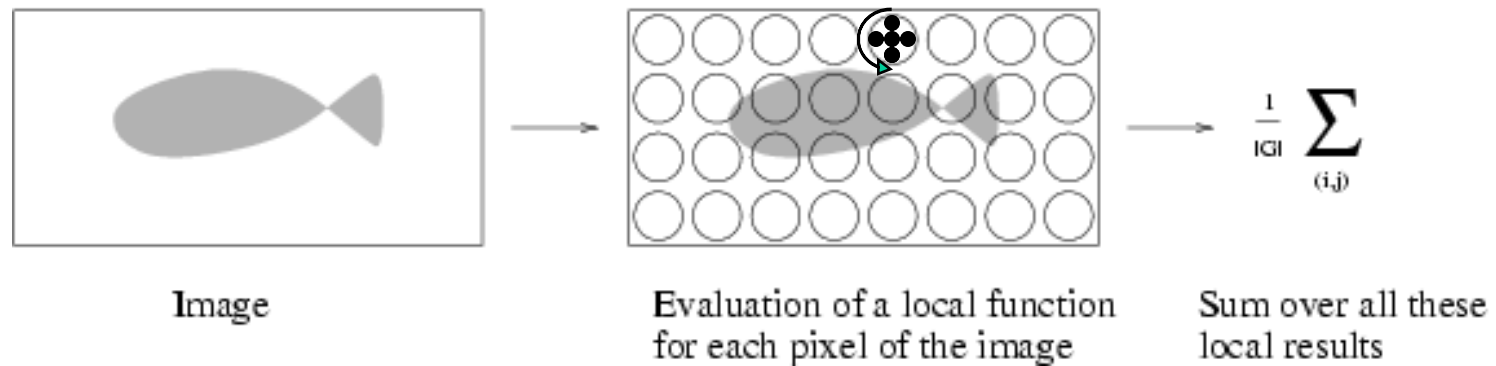
$$A[f](\mathbf{X}) = \frac{1}{2\pi NM} \int_{t_0=0}^N \int_{t_1=0}^M \int_{\varphi=0}^{2\pi} f(g\mathbf{X}) d\varphi dt_1 dt_0$$

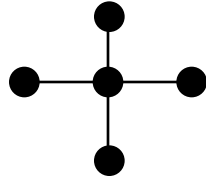
Use as kernel functions monomials of pixels of local support and integrate over the Euclidean motion:



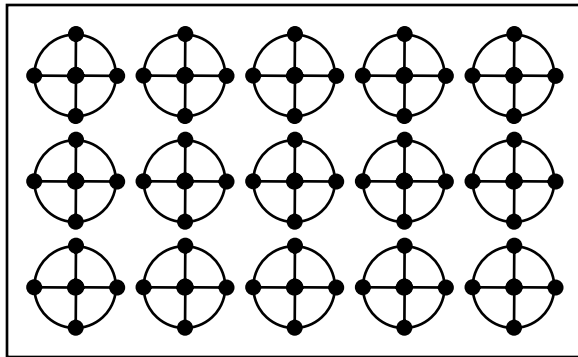
A diagram showing a 3x3 grid of points with a central point. A curved arrow indicates a counter-clockwise rotation around the center. A green arrow points to the right from the bottom-left point.

$$f(\mathbf{X}) = m_{00}^3 m_{01}^1 m_{0-1}^5 m_{10}^2 m_{-10}^3$$

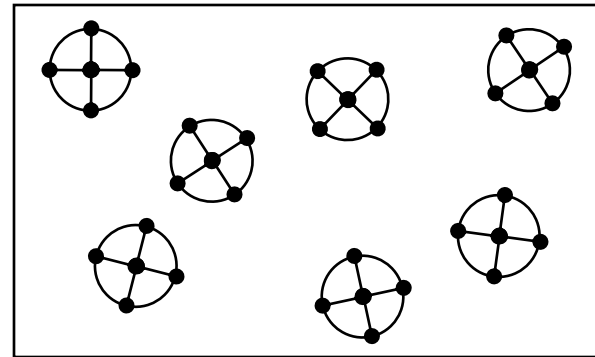




Monomial

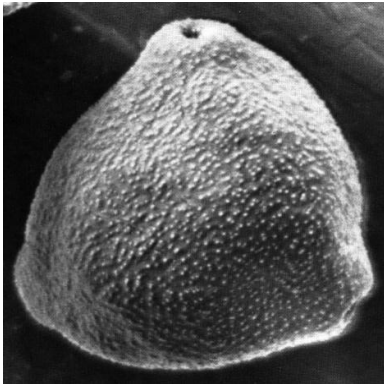


Deterministic integral  
over the planar  
Euclidean motion

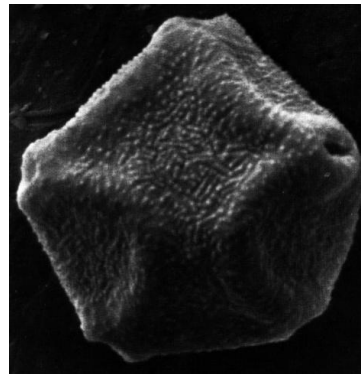


Monte-Carlo-Integration

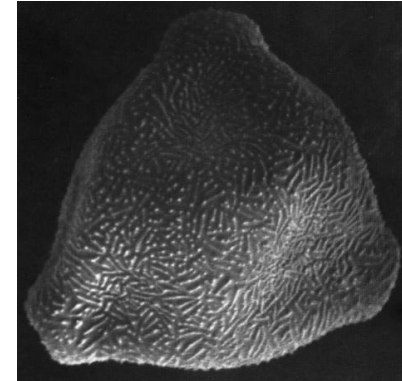
# Pollen examples



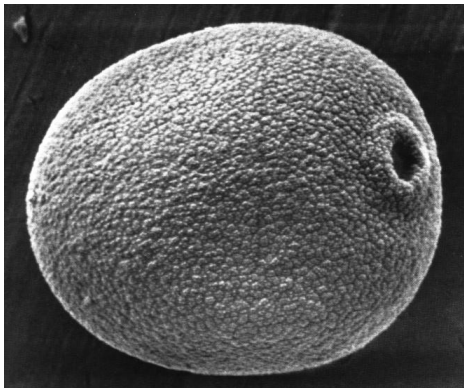
Hasel



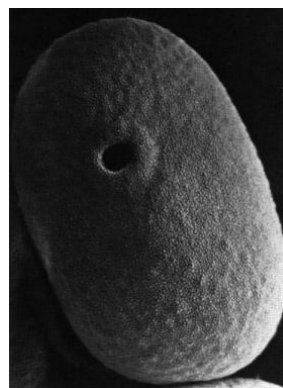
Erle



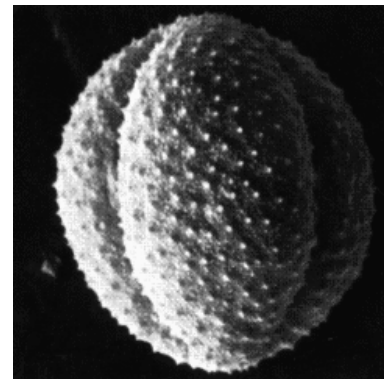
Birke



Gräser



Roggen

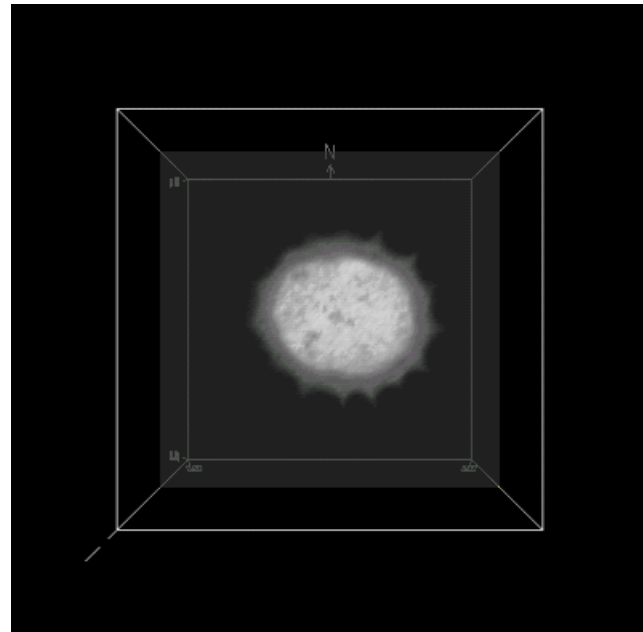


Beifuß

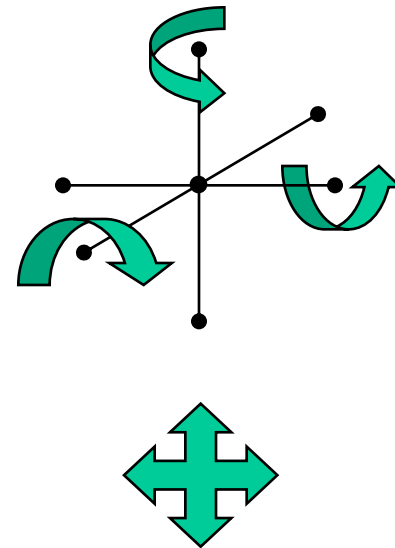
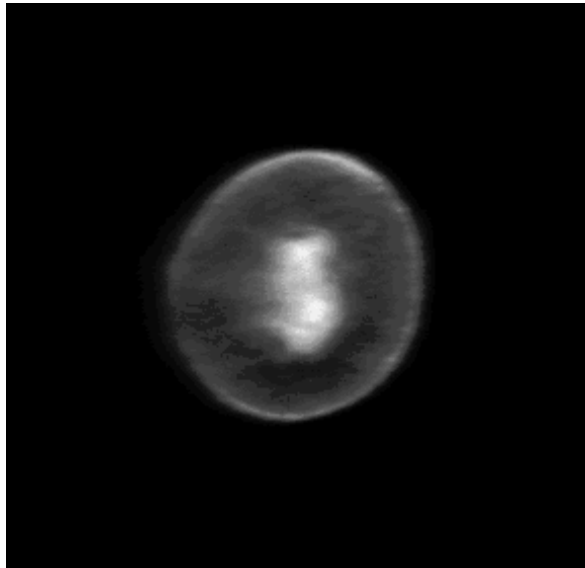
+ 33 further species (not relevant for allergies)



# Gänseblümchen/daisy pollen grain



# Eibe/Taxus



Integrate over Euclidean Motion

# Extension of Haar-Integrals to Discrete Structures

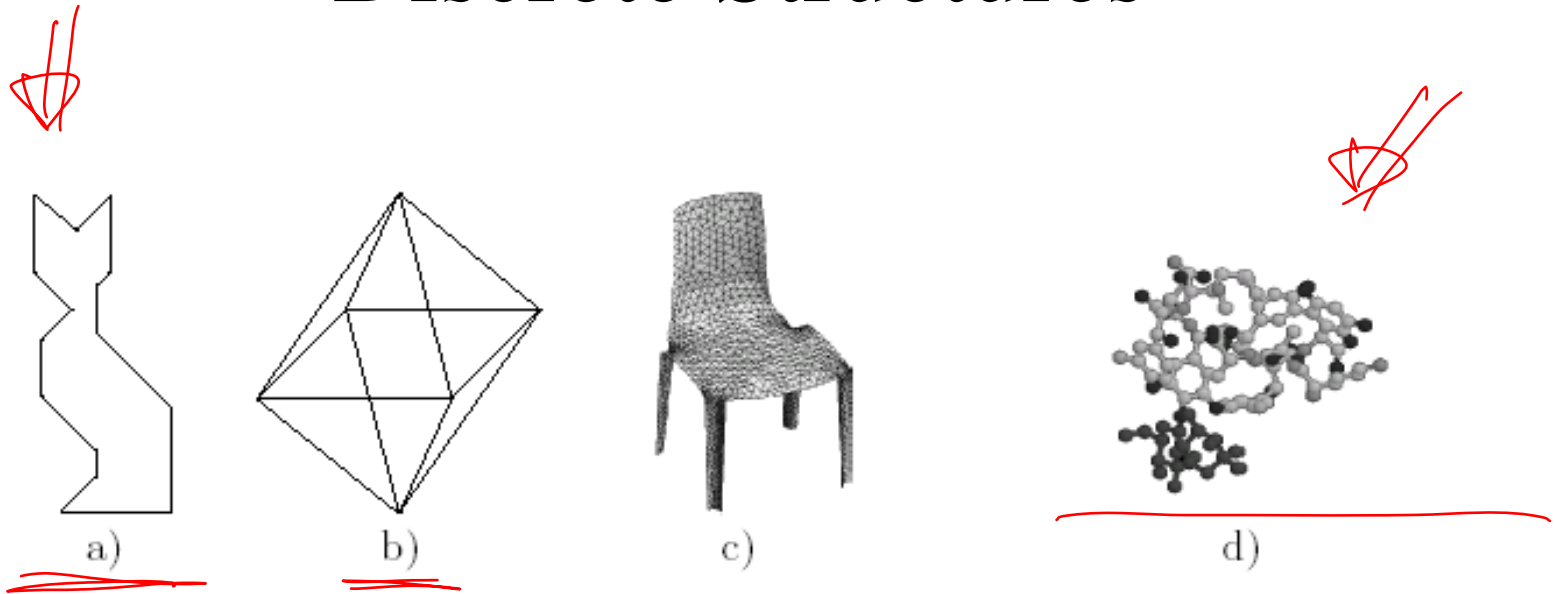


Fig. 1. Discrete structures in 2D and 3D: (a) closed contour described by a polygon (b) wireframe object (c) 3D triangulated surface mesh (d) molecule.

Describe discrete structures with Dirac delta functions!

# The Five Platonic Solids

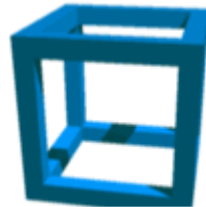
Octahedron



Icosahedron.gif



Dodecahedron



Hexahedron



Tetrahedron

A platonic solid is a polyhedron (Polyeder) all of whose faces are congruent (they differ only in a Euclidean motion) regular polygons, and where the same number of faces meet at every vertex. The best know example is a *cube* (or *hexahedron* ) whose faces are six congruent squares.

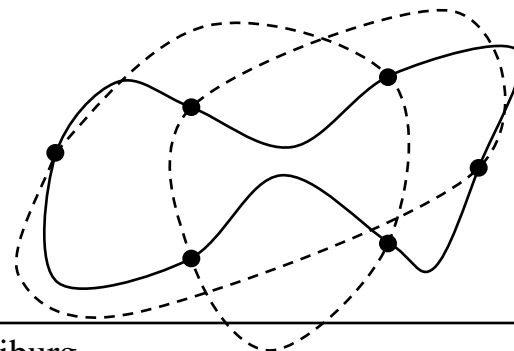
# Invariants on discrete Structures

(topologically equivalent structures)

Chose proper kernel functions on distributions

$$T f \mathbf{X} = \int_G f(g\mathbf{X}) dg$$

1. Chose kernel functions which are different from zero only at the vertices and which act only on neighborhoods of degree  $m$ .
2. As each vertex can be visited in an arbitrary permutation of all points by a continuous Euclidean motion the integral is changed into a invariant sum over all vertices with Euclidean-invariant local discrete features !!
3. Use principle of rigidity to reach completeness: use a basis of features which are locally rigid and which can be pieced together in a unique way to the global object (see invariants for triangle!).



# Invariants for discrete objects

1. For a discrete object  $\Delta$  and a kernel function  $f(\Delta)$  it is possible to construct an invariant feature  $T[f](\Delta)$  by integrating  $f(g\Delta)$  over the Euclidean transformation group  $g \in G$ .
2. The kernel function is properly designed, such that it delivers a value dependent on the discrete features of a local neighborhood, when a vertex of the object moved by the continuous Euclidean motion  $g$  hits the origin and has one specific orientation.
3. Let us assume that our discrete object is different from zero only at its vertices. A rotation and translation invariant local discrete kernel function  $h$  takes care for the algebraic relations to the neighboring vertices and we can write:

$$f \Delta = \sum_{i \in \mathbb{V}} h(\Delta, \mathbf{x}_i) \delta(\mathbf{x} - \mathbf{x}_i)$$

where  $\mathbb{V}$  is the set of vertices and  $\mathbf{x}_i$  the vector representing vertex  $i$ .

4. In order to get finite values from the distributions it is necessary to introduce under the Haar integral another integration over the spatial domain  $\mathbf{X}$ .
5. By choosing an arbitrary integration path in the continuous group  $G$  we can visit each vertex in an arbitrary order the integral is transformed into a sum over all local discrete functions allowing all possible permutations of the contributions of the vertices.

# Extension of Haar-Integrals to Discrete Structures

$$\begin{aligned} T f \Delta &:= \int_G \int_{\mathbf{X}} f(g\Delta) d\mathbf{x} dg = \int_G \left[ \int_{\mathbf{X}} \sum_{i \in \mathbb{V}} h(g\Delta, g\mathbf{x}_i) \delta(g\mathbf{x} - g\mathbf{x}_i) d\mathbf{x} \right] dg \\ &= \int_G \left[ \sum_{i \in \mathbb{V}} h(\Delta, \mathbf{x}_i) \right] dg = \sum_{i \in \mathbb{V}} h(\Delta, \mathbf{x}_i) \end{aligned}$$

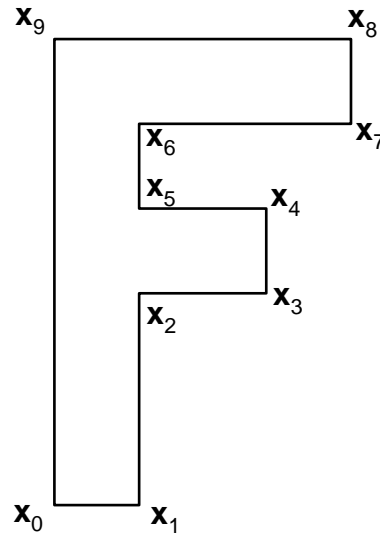
Intuitive result: get *global* Euclidean invariants by summation over discrete *local* Euclidean invariants  $h(\Delta, \mathbf{x}_i)$  !

Remember: The delta function has the following selection property:

$$\int_{-\infty}^{+\infty} f(x) \delta(x-a) dx = f(a)$$

# Euclidean Invariants for Polygons

We assume e.g. to have given a polygon with 10 vertices, e.g.

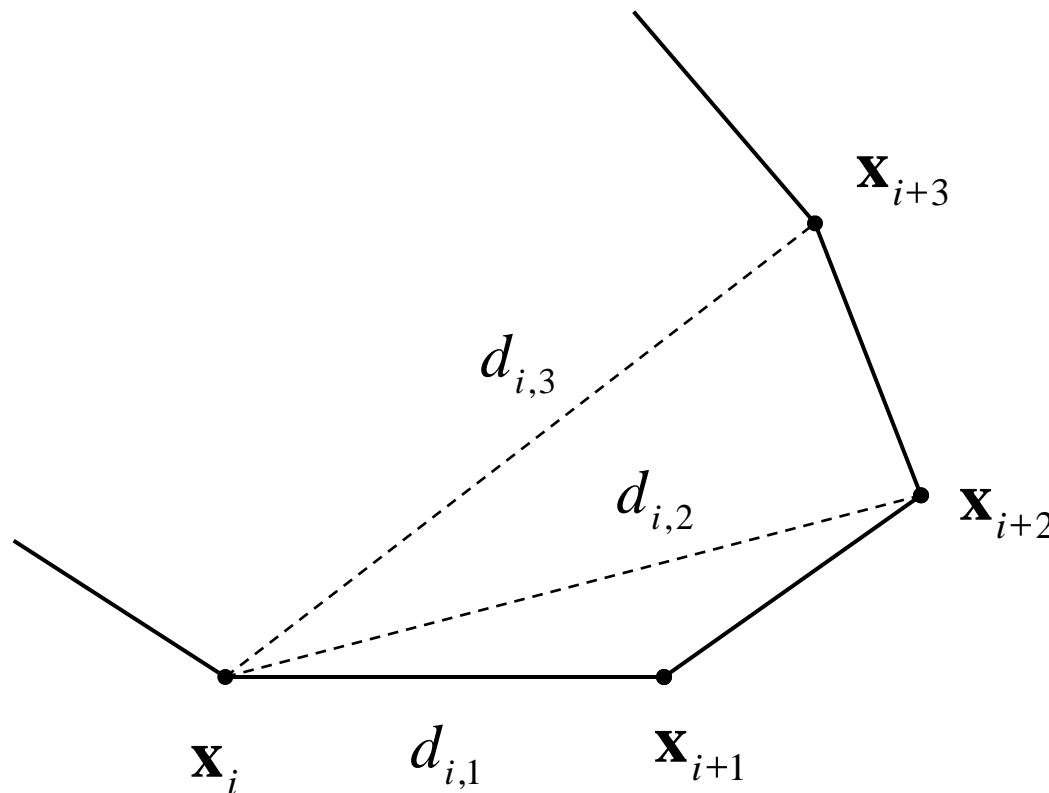


$$\mathbf{x}_i = \begin{bmatrix} 0 & 1 & 1 & 2.5 & 2.5 & 1 & 1 & 3.5 & 3.5 & 0 \\ 0 & 0 & 2.5 & 2.5 & 3.5 & 3.5 & 4.5 & 4.5 & 5.5 & 5.5 \end{bmatrix}$$



Choose as local Euclidean invariants distances of vertex  $i$  and its  $k$ -th righthand neighbours:

$$d_{i,k} = \|\mathbf{x}_i - \mathbf{x}_{\langle i+k \rangle}\|$$

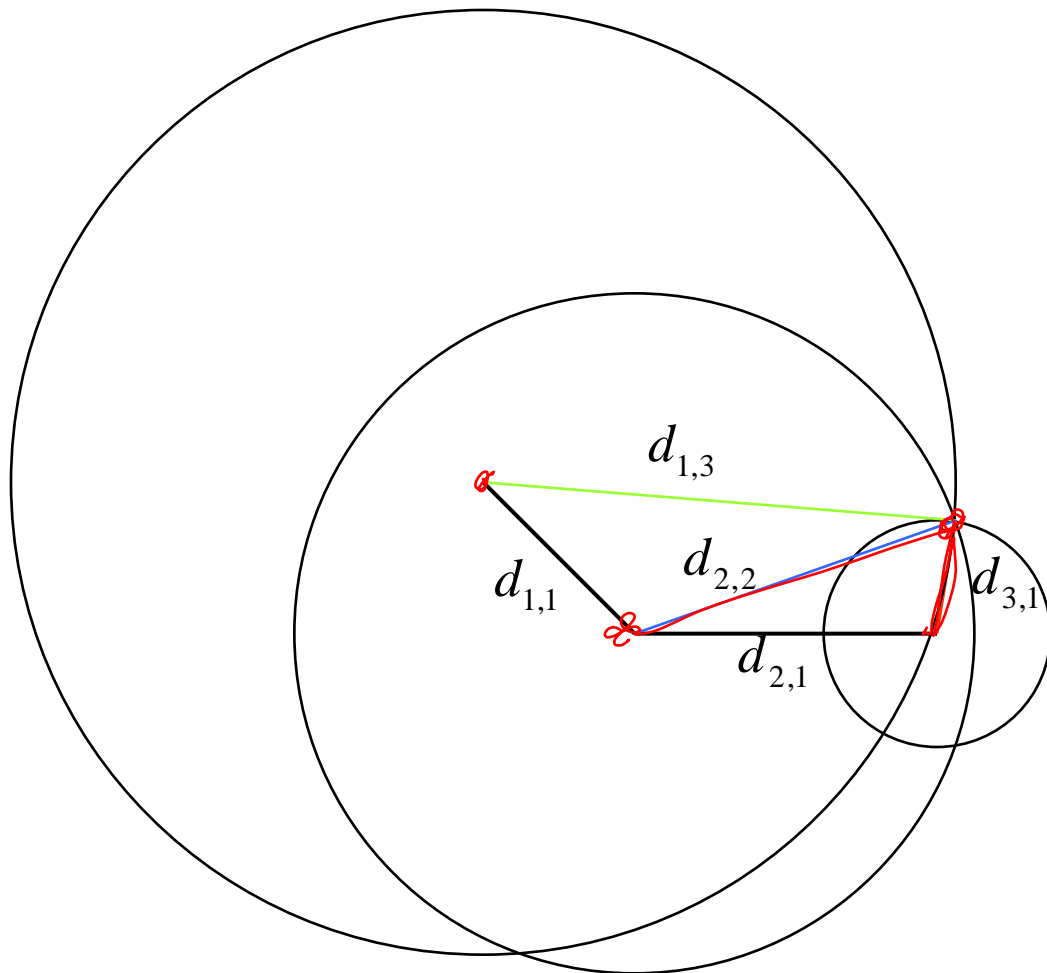


The elements:

$$d_{i,1}, d_{i,2}, d_{i,3}$$

form a basis for a polygon, because they uniquely define a polygon (up to a mirror-polygon) !

**Principle of rigidity**



- $d_{i,3}$
- $d_{i,2}$
- $d_{i,1}$

Given two edges  $d_{1,1}$  and  $d_{2,1}$ . Then the third vertex is uniquely defined by the set:

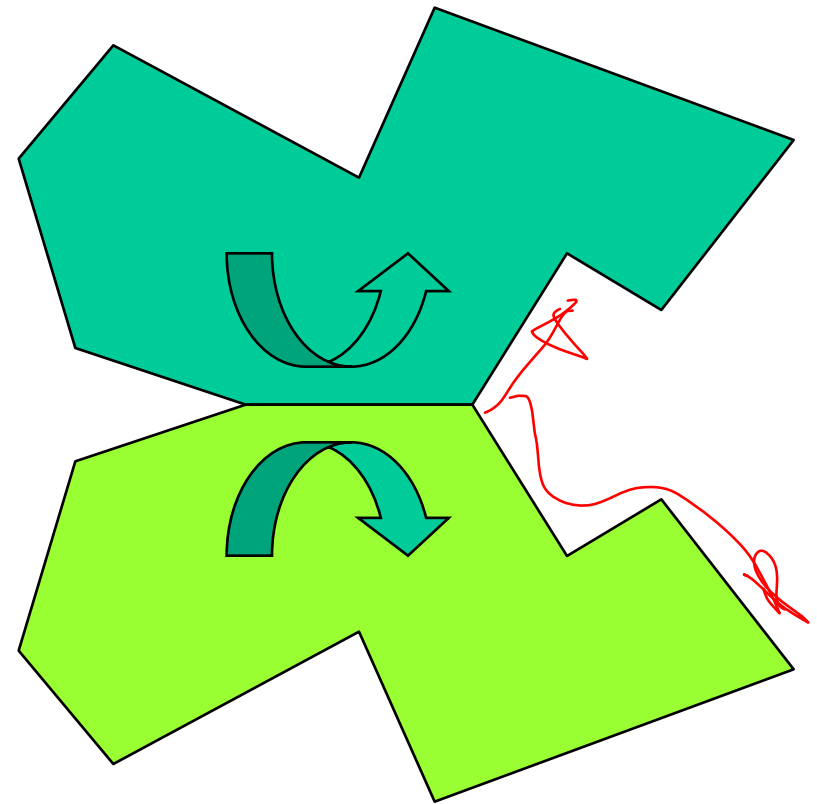
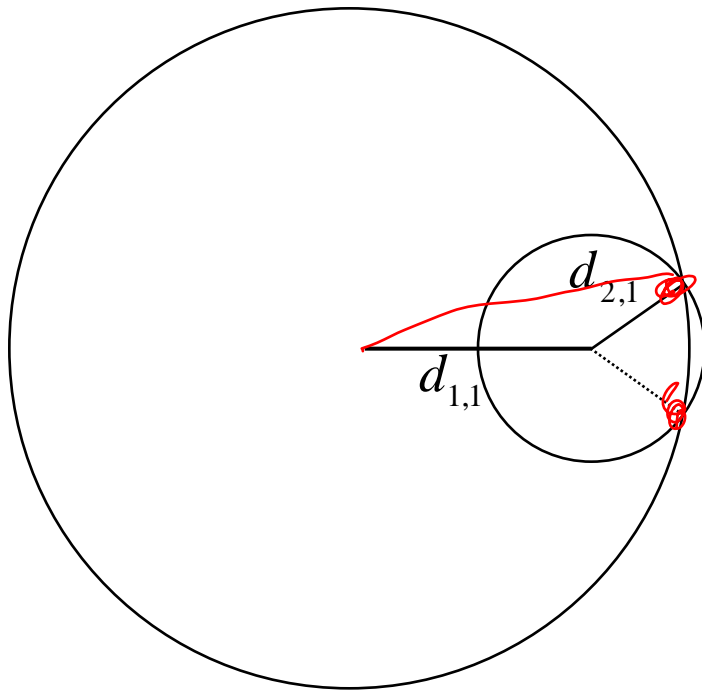
$$d_{i,1}, d_{i,2}, d_{i,3}$$

Because there is a unique intersection point of three circles.

This means, that a whole polygon can be uniquely generated by this basis elements iteratively.

With the first two distances we get two initial configurations. Then all further vertices will be unique.

The two initial configurations give two possible polygons, where one is just the mirror image of the other along the first edge as its axis.



As discrete functions of local support we derive monomials from distances between neighbouring vertices and hence we get invariants by summing these discrete functions of local support (DFLS) over all vertices:

$$\tilde{x}_{n_1, n_2, n_3, n_4} = \sum_{i \in \mathbb{V}} h(\Delta, \mathbf{x}_i) = \sum_{i \in \mathbb{V}} d_{i,1}^{n_1} d_{i,2}^{n_2} d_{i,3}^{n_3} d_{i,4}^{n_4}$$

Choosing the following 8 values for the exponents we would end up with a corresponding invariant feature vector and a set of 8 invariants:

$i$	$n_1$	$n_2$	$n_3$
$\tilde{x}_0$	1	0	0
$\tilde{x}_1$	1	1	0
$\tilde{x}_2$	1	0	1
$\tilde{x}_3$	1	1	1
$\tilde{x}_4$	2	0	0
$\tilde{x}_5$	2	1	0
$\tilde{x}_6$	2	0	1
$\tilde{x}_7$	2	1	1

We clearly recognize e.g.

$\tilde{x}_0$

as the circumference of the polygon as an invariant.

For the above example of the letter F we get the following invariants:

$$\tilde{\mathbf{x}} = 21 \quad 44 \quad 83.82 \quad 68.6 \quad 184.6 \quad 665.3 \quad 149.5 \quad 751.6^T$$

**How complete is this set of invariants?**

a b c  
c b a

# Discrimination Performance, question of completeness

We expect a more and more complete feature space by summing over an increasing number of monomials of this basis elements

Looking at a triangle as the most simplest polygon one can show that the following three features derived from the three sides  $\{a, b, c\}$  form a complete set of invariants:

$\tilde{x}_0 = a + b + c, \quad \tilde{x}_1 = a^2 + b^2 + c^2, \quad \tilde{x}_2 = a^3 + b^3 + c^3$

oder auch:

$\tilde{x}_0 = a + b + c, \quad \tilde{x}_1 = ab + bc + ca, \quad \tilde{x}_2 = abc$

The last features are equivalent to the elementary symmetrical polynomials in 3 variables which are a complete set of invariants with respect to all permutations.

# Elementarsymmetrisches Polynom

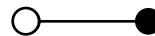
Ein Polynom kann eindeutig durch seine Wurzeln oder auch durch seine Koeffizienten definiert werden (vollständige Invarianten):

$$(x-a)(x-b)(x-c) = x^3 - \underbrace{(a+b+c)}_{p_2} x^2 + \underbrace{(ab+bc+ca)}_{p_1} x - \underbrace{abc}_{p_0}$$

Das Polynom (und damit auch seine Koeffizienten) ist invariant gegenüber einer beliebigen Permutation der Wurzeln!

Somit ergibt sich folgende Korrespondenz:

Alle Permutationen der 3  
Wurzeln  
a,b,c  
(symmetr. Gruppe)



Koeffizienten des  
elementarsymmetrischen  
Polynoms:

$$p_0 = -abc$$

$$p_1 = ab + bc + ca$$

$$p_2 = -(a + b + c)$$

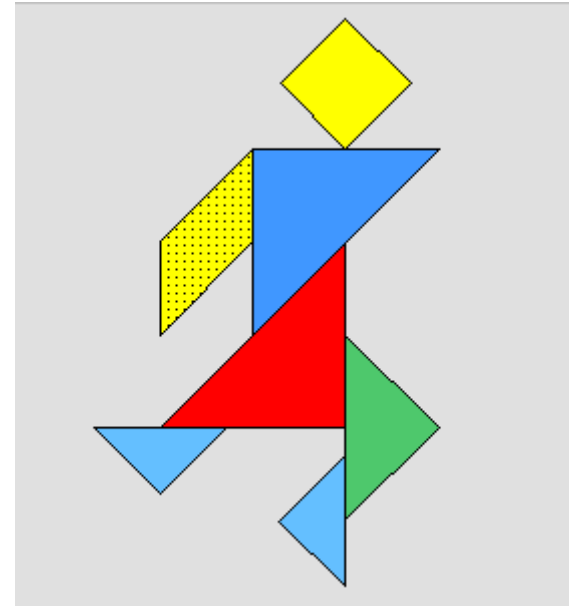
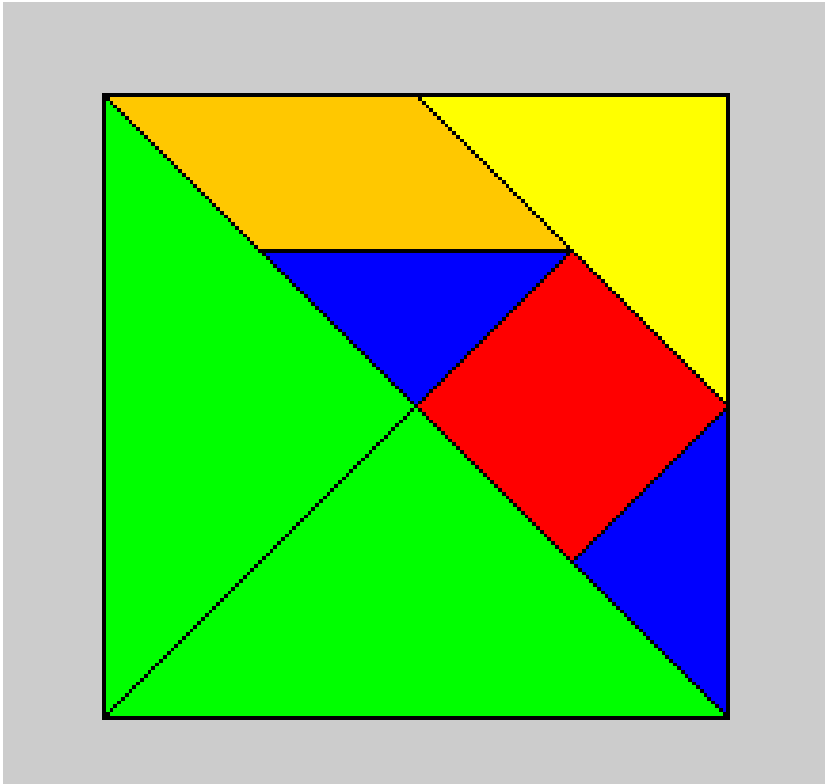
# Completeness for finite Groups

(Emmy Noether, 1916)

For finite Groups  $G$  with  $|G|$  elements and patterns of dimensionality  $N$  the group averages over all monomials of degree  $\leq |G|$  are complete and form a basis of the pattern space. The number of monomials is given by

$$\binom{N + |G|}{N}$$

# Experiment: Object classification in a Tangram database

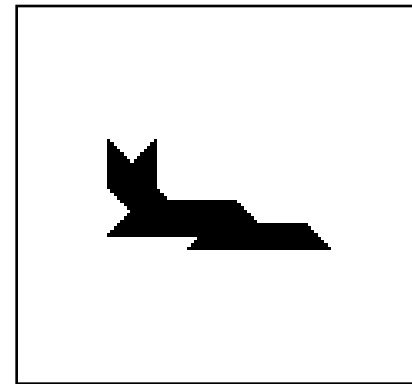
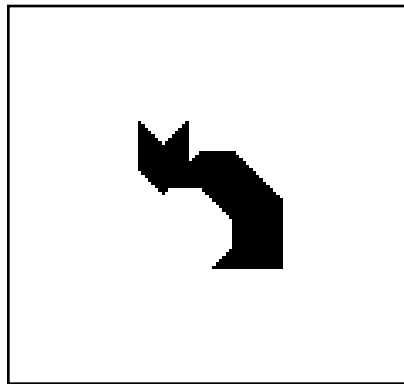
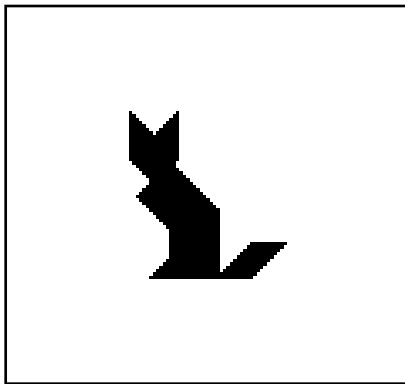
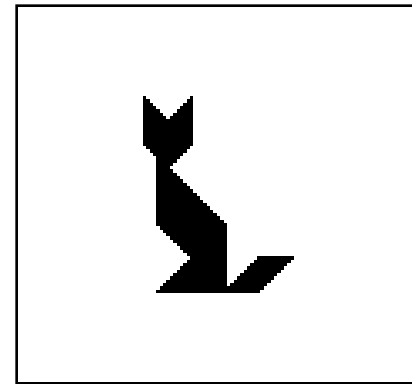
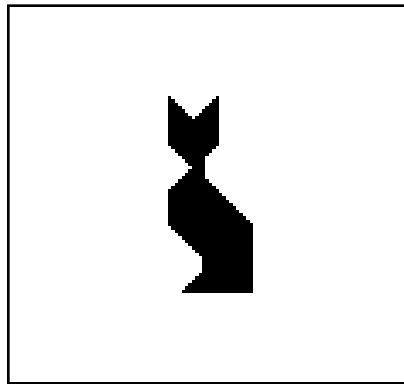
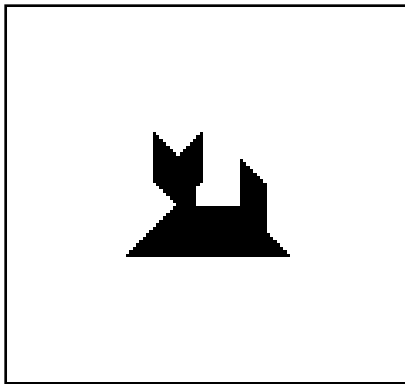


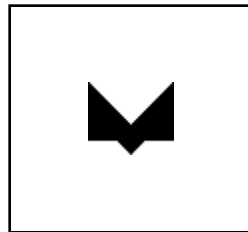
Take the outer contour as feature

Objects can not easily be  
discriminated with trivial  
geometric features!

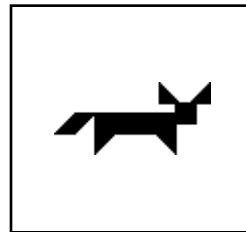


# Cats

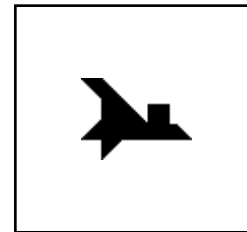




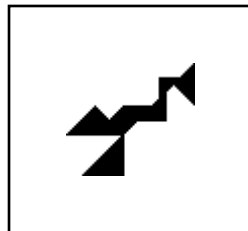
bat.png  
n1029923508



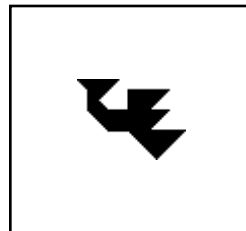
cow.png  
n1029410674



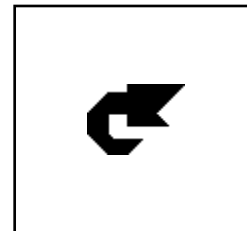
seal.png  
n1033206320



crab1.png  
n1029696706



crab2.png  
n1029696603



crab3.png  
n1029696531





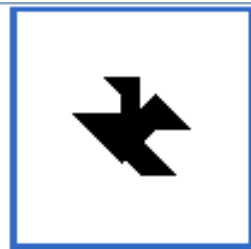
bird01.png

abc



bird02.png

abc



bird03.png

abc



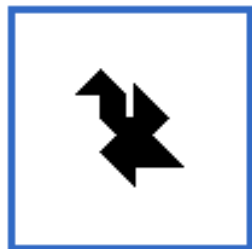
bird04.png

abc



bird05.png

abc



bird06.png

abc



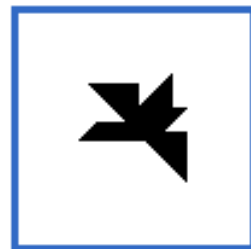
bird07.png

abc



bird08.png

abc



bird09.png

abc



bird10.png

abc



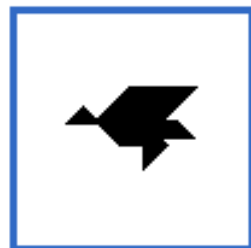
bird11.png

abc



bird12.png

n1029274381



bird13.png

abc



bird14.png

n1031134075



bird15.png

abc



bird16.png

abc



bird17.png

abc



bird18.png

abc



bird19.png

abc



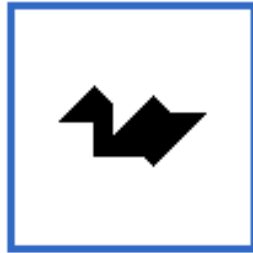
bird20.png

abc



bird21.png

n1029274303



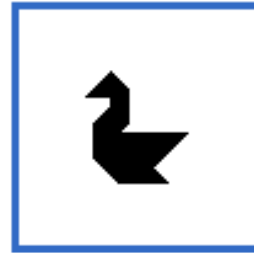
bird22.png

n1029274076



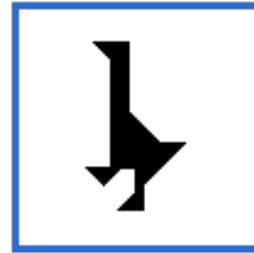
bird23.png

n1029274221



bird24.png

n1029273873



bird25.png

n1029257517



bird26.png

n1029257609



bird27.png

n1029273740



bird28.png

n1029257014



bird29.png

n1029273642



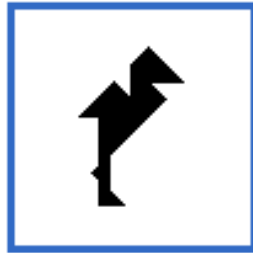
bird30.png

n1029273957



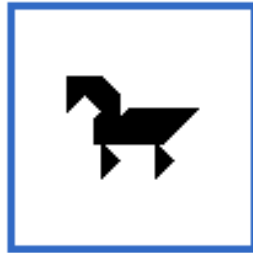
bird31.png

n1029256353



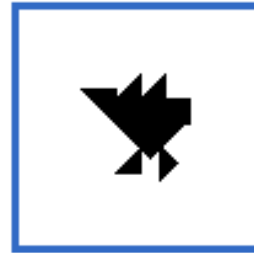
bird32.png

n1029247568



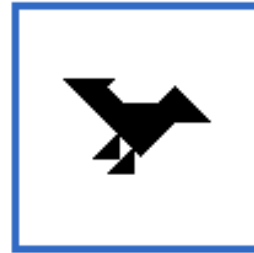
bird33.png

n1029248013



bird34.png

n1029256623



bird35.png

n1029256825



bird36.png

n1029256908



bird37.png

n1029256719



bird38.png

n1029256850



bird39.png

n1029247693



bird40.png

n1029256450

n1030059490



cat1.png

n1029271362



cat2.png

n1029271289



cat3.png



cat4.png

n1029271232



cat5.png

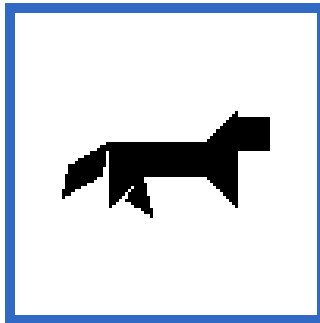
n1029271177



cat6.png

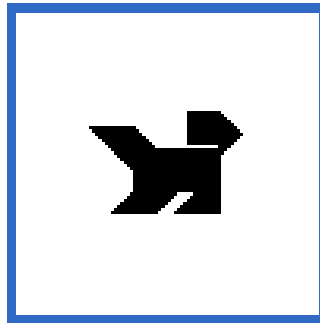
n1029271100

n1029679376



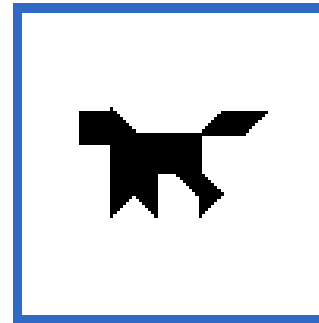
dog1.png

n1029679294

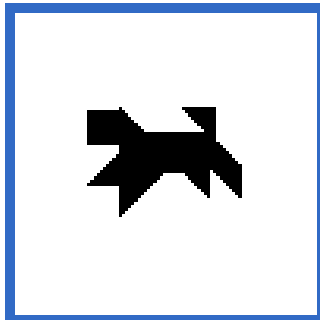


dog2.png

n1029679234

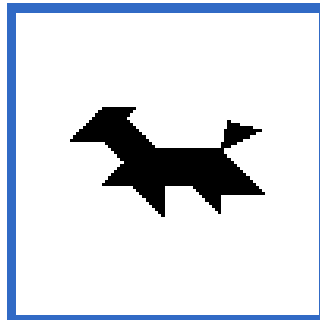


dog3.png



dog4.png

n1029679142



dog5.png

n1029679056



fish01.png

n1029698383



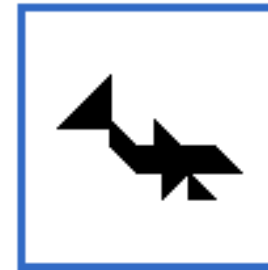
fish02.png

n1029697938



fish03.png

n1029698015



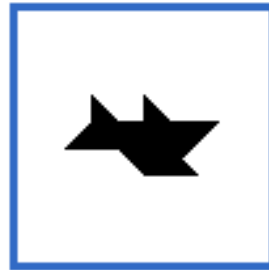
fish04.png

n1029698468



fish05.png

n1029697659



fish06.png

n1029698270



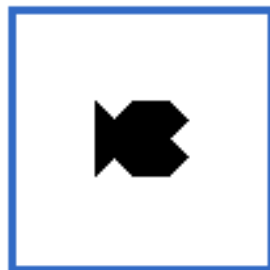
fish07.png

n1029697733



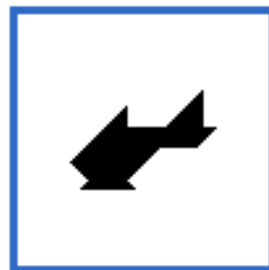
fish08.png

n1029698189



fish09.png

n1029924996



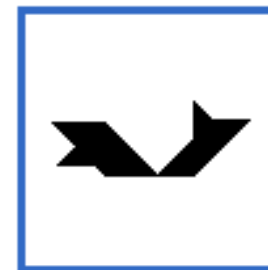
fish10.png

n1029698080



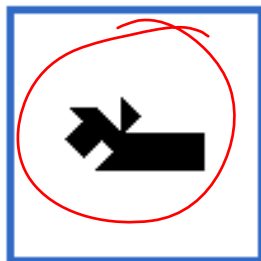
fish11.png

n1029697529



fish12.png

n1029696327



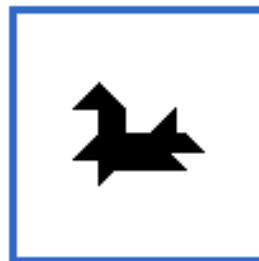
horse01.png

n1029673849



horse02.png

n1029412951



horse03.png

n1029673301



horse04.png

n1029673699



horse05.png

n1029410571



horse06.png

n1029673233



horse07.png

n1029413836



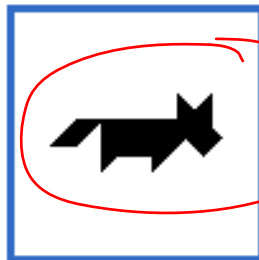
horse08.png

n1029414596



horse09.png

n1029673607



horse10.png

n1029673466





rabbit1.png

n1029281087



rabbit2.png

n1029013797



rabbit3.png

n1029280188



rabbit4.png

n1029280344



rabbit5.png

n1029280468



rabbit6.png

n1029280610



rabbit7.png

n1029280728



snake.1.png

n1029696405



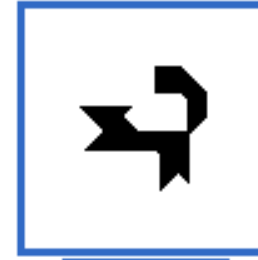
snake.2.png

n1029696963



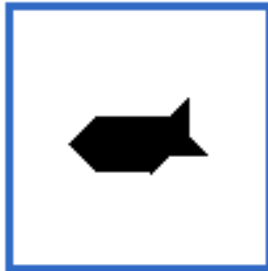
squirrel.1.png

n1029678741



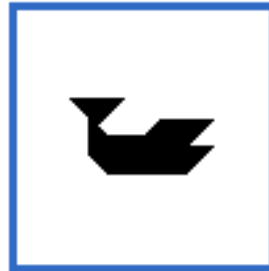
squirrel.2.png

n1029678885



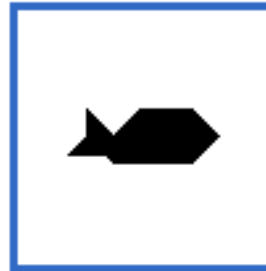
whale.1.png

n1029698609



whale.2.png

n1029697597



whale.3.png

n1029697454



whale.4.png

n1029696811

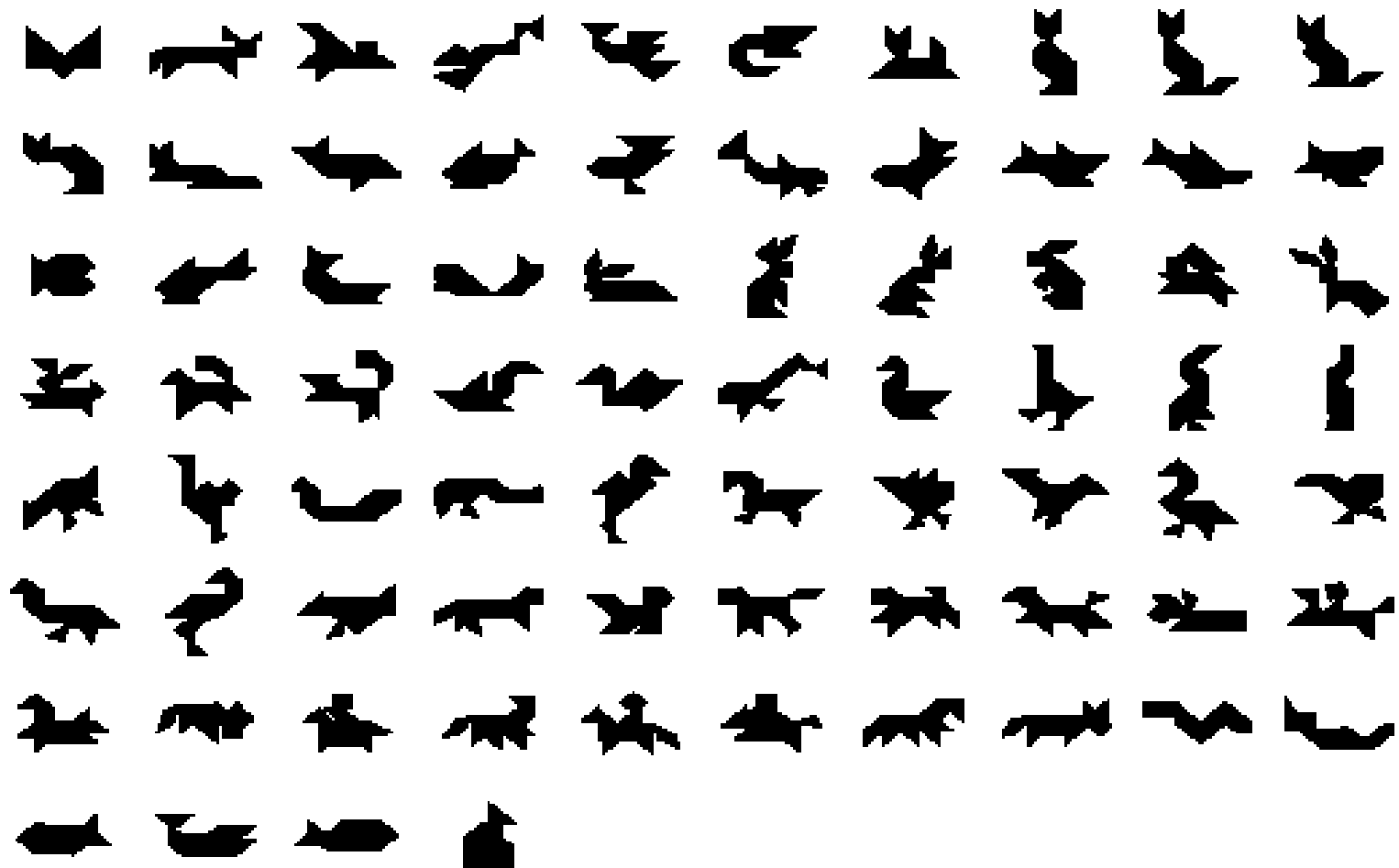


Fig. 6. The 74 tangrams used in the experiment.

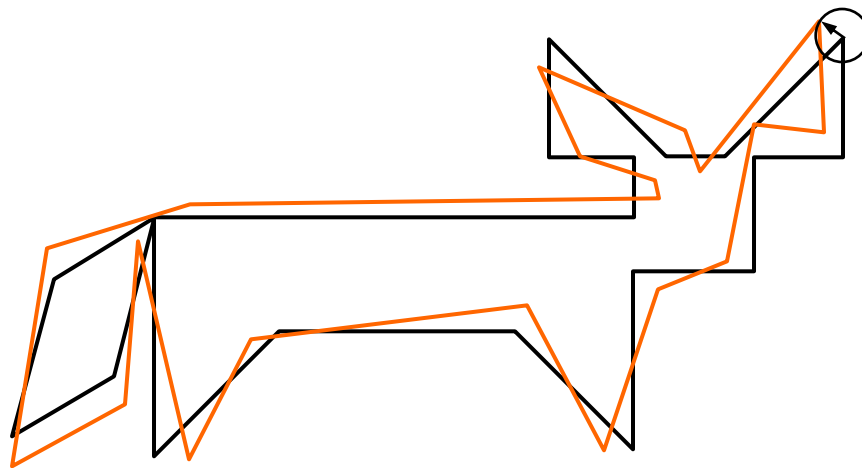
The experiments were conducted with three sets of 6, 10 and 14 invariants respectively using the following exponent table:

$i$	$\tilde{x}_0$	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$	$\tilde{x}_7$	$\tilde{x}_8$	$\tilde{x}_9$	$\tilde{x}_{10}$	$\tilde{x}_{11}$	$\tilde{x}_{12}$	$\tilde{x}_{13}$
$n_1$	1	0	0	1	1	0	0	1	0	0	2	0	0	0
$n_2$	0	1	0	1	0	1	0	0	1	0	0	2	0	0
$n_3$	0	0	1	0	1	1	0	0	0	1	0	0	2	0
$n_4$	0	0	0	0	0	0	1	1	1	1	0	0	0	2

$$\tilde{x}_{n_1, n_2, n_3, n_4} = \sum_{i \in \mathbb{V}} h(\Delta, \mathbf{x}_i) = \sum_{i \in \mathbb{V}} d_{i,1}^{n_1} d_{i,2}^{n_2} d_{i,3}^{n_3} d_{i,4}^{n_4}$$

The classification performance was measured against additive noise of 5%, 10% and 20%.

# 10% Noise

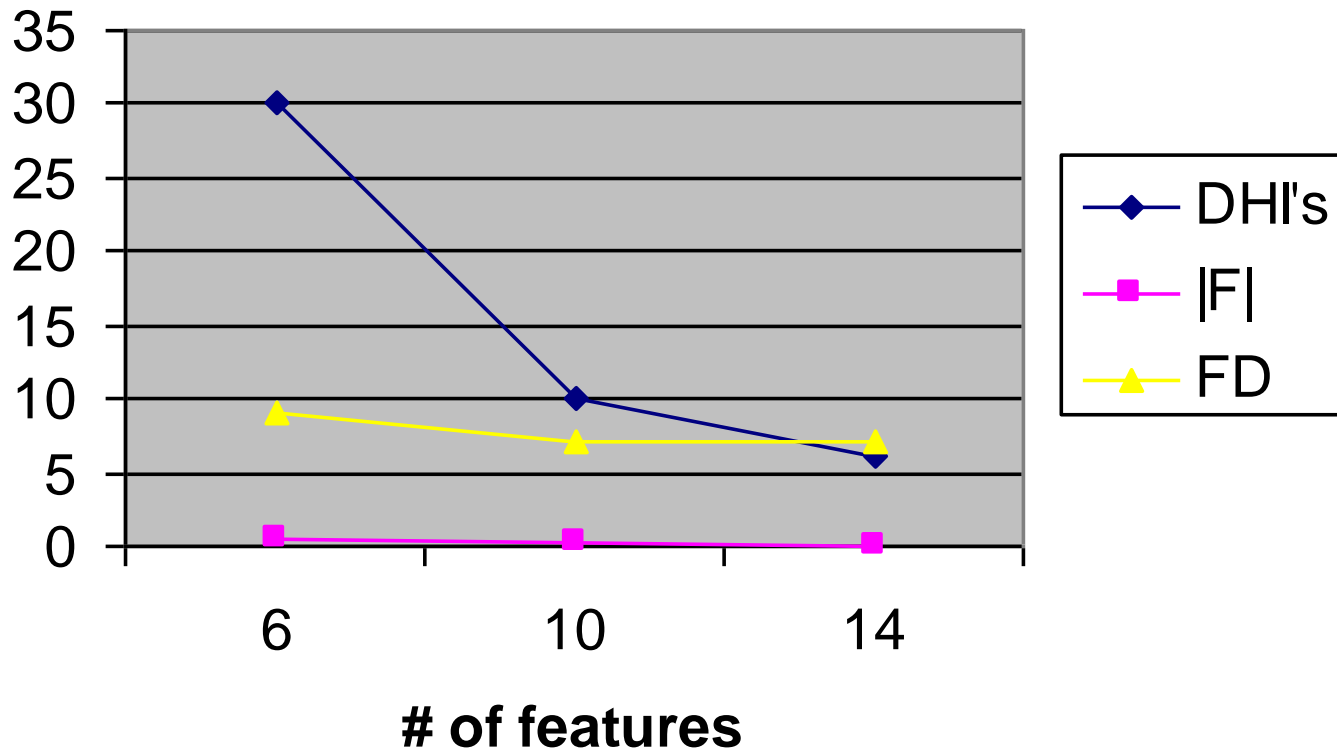


Classification error (in percent) for 5%, 10% and 20% noise for 74 tangrams with a Euclidean (E) and a Mahalanobis (M) Classifier

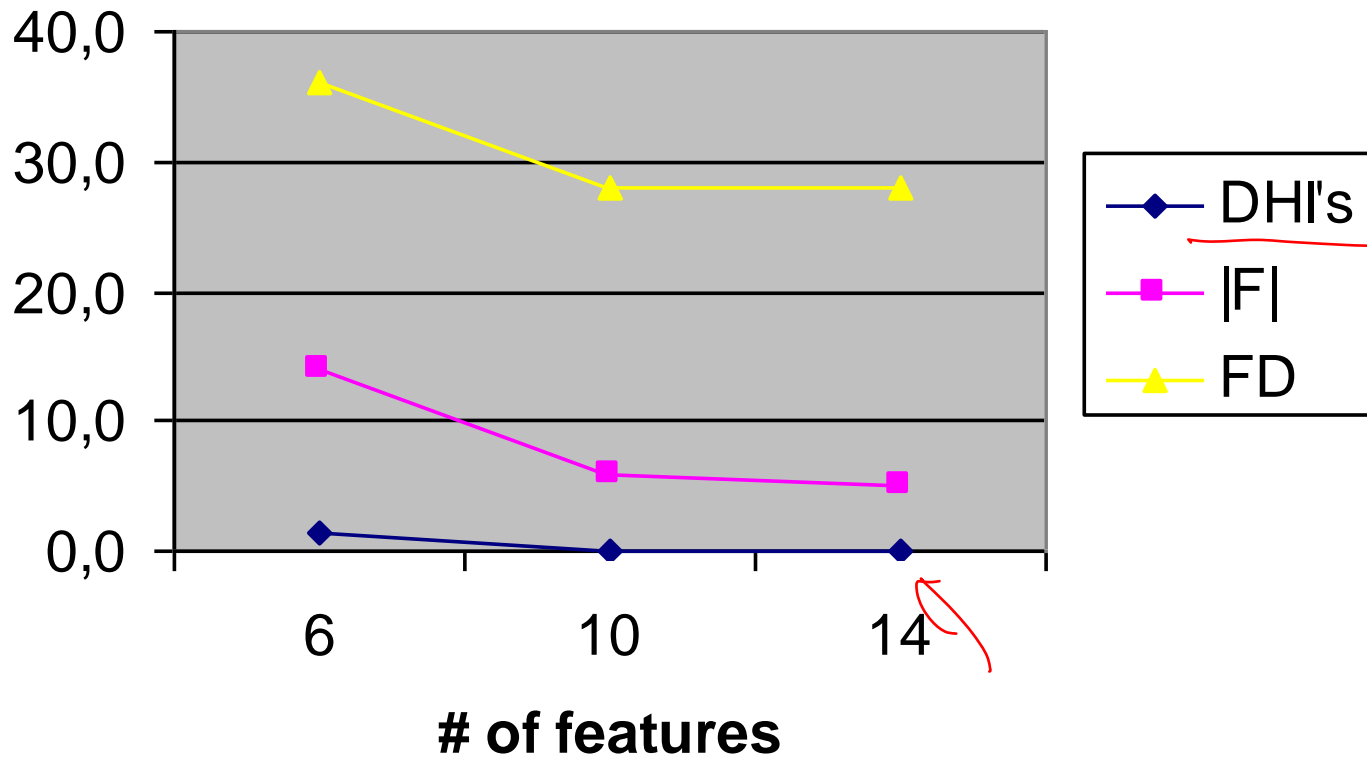
noise (in percent)	# of invariants	metric	DHI's class. error in %	FD error in %	F  error in %
5	6	E	30	9	0.5
5	10	E	10	7	0.2
5	14	E	6	7	0
10	6	M	1.5	36	14
10	10	M	0	28	6
10	14	M	0	28	5
20	6	M	25	75	59
20	10	M	7	70	50
20	14	M	3	70	46

Empirical evaluation for the degree of completeness!

### CI. Error for 5% Noise, Euclidean-CI.

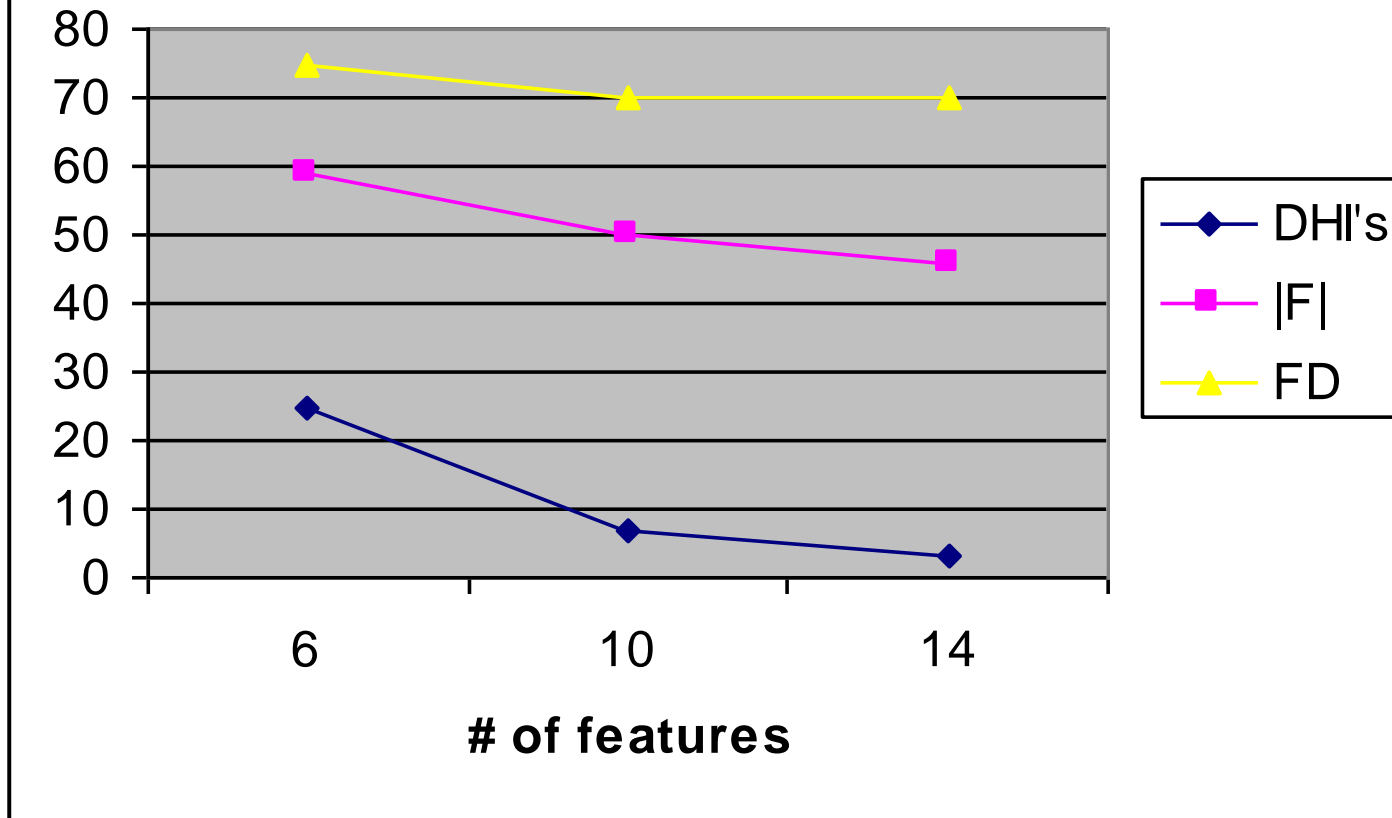


## CI. Error for 10% Noise, Mahalanobis-CI.

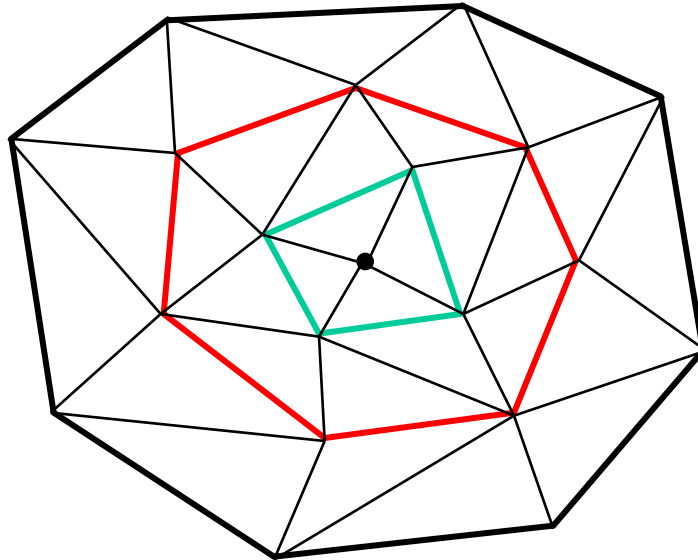




## 20% Noise, Mahalanobis-CI.

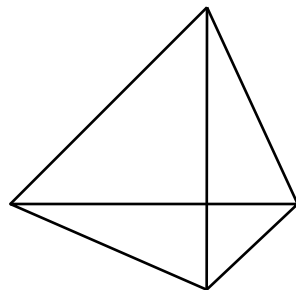


# 3D-Meshes



neighbourhood of  
degree one

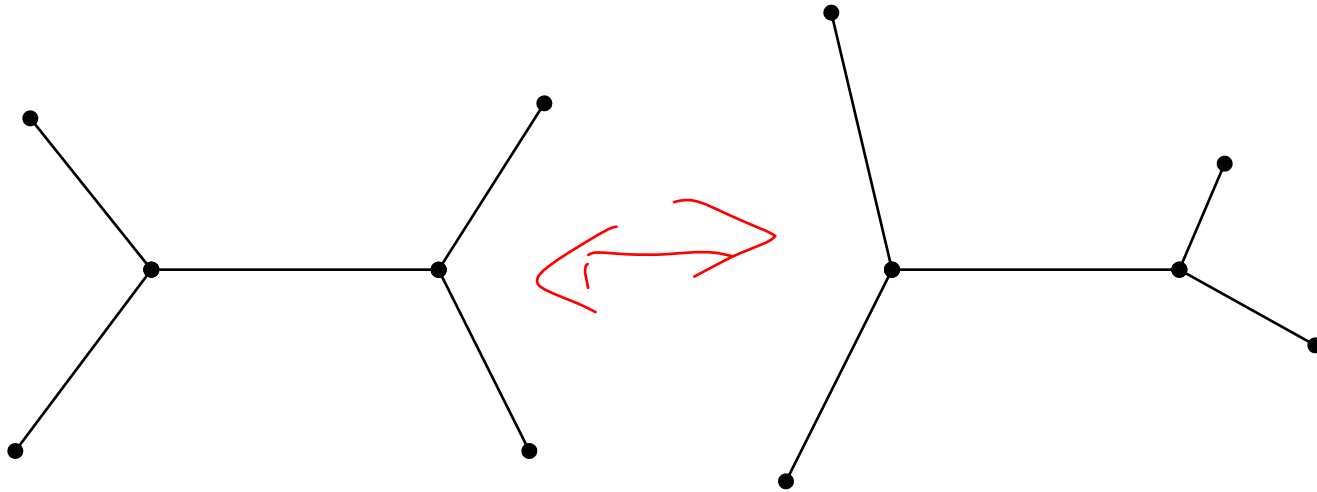
neighbourhood of  
degree two



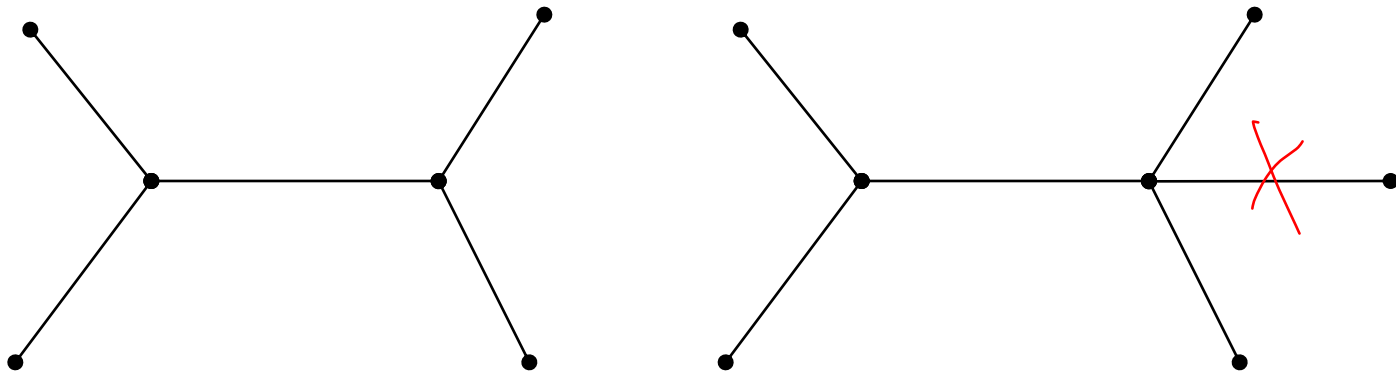
tetrahedron as the basic  
building block for a  
polyhedron

# Topologically equivalent structures and (TE) non topologically equivalent structures (NTE)

TE



NTE



# Properties

- If we constrain our calculation to a finite number of invariants we end up with a simple linear complexity in the number of vertices. This holds if the local neighborhood of vertices is resolved already by the given data structure; otherwise the cost for resolving local neighborhoods must be added.
- In contrast to *graph matching* algorithms we apply here ~~algebraic techniques to solve the problem~~. This has the advantage that we can apply hierarchical searches for retrieval tasks, namely, to start only with one feature and hopefully eliminate already a large number of objects and then continue with an increasing number of features etc.

# Conclusion

- In this paper we have introduced a novel set of invariants for discrete structures in 2D and 3D.
- The construction is a rigorous extension of Haar integrals over transformation groups to Dirac Delta Functions.
- The resulting invariants can easily be calculated with linear complexity in the number of vertices.
- The proposed approach has the potential to be extended to other discrete structures and even to the more general case of weighted graphs.

## Literature: (<http://imb.informatik.uni-freiburg.de>)

- (1) S. Siggelkow and H. Burkhardt. Image retrieval based on local invariant features. In Proceedings of the IASTED International Conference on Signal and Image Processing (SIP) 1998, pages 369-373, Las Vegas, Nevada, USA, October 1998. IASTED.
- (2) M. Schael and H. Burkhardt. Automatic detection of errors on textures using invariant grey scale features and polynomial classifiers. In M. K. Pietikäinen, editor, Texture Analysis in Machine Vision, volume 40 of Machine Perception and Artificial Intelligence, pages 219-230. World Scientific, 2000.
- (3) O. Ronneberger, U. Heimann, E. Schultz, V. Dietze, H. Burkhardt and R. Gehrig. Automated pollen recognition using gray scale invariants on 3D volume image data. Second European Symposium on Aerobiology, Vienna/Austria, Sept. 5-9, 2000.
- (4) H. Burkhardt and S. Siggelkow. Invariant features in pattern recognition - fundamentals and applications. In C. Kotropoulos and I. Pitas, editors, Nonlinear Model-Based Image/Video Processing and Analysis, pages 269-307. John Wiley & Sons, 2001.