



Spracovanie obrazu apl.

Andrej FERKO

Comenius University Bratislava

PG1, 7. 12. 2020, FMFI UK

Príklad (Hearn-Baker, 2014, p. 376), logo UK

- Foto/sken, výpočet >> obraz, políčko, IFS
- Abstraktný obraz, funkcia
- $f(x, y)$,
- kde rovinné súradnice x, y ,
- Hodnota f výška/šedá
- Veta o reprezentácii, Fourier
- Digitalizácia, vzorkovanie, 1D

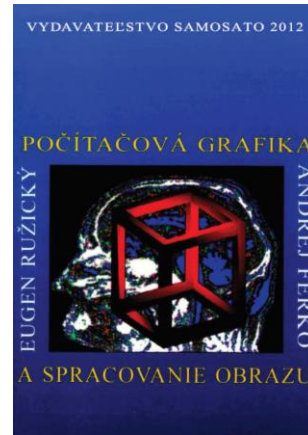
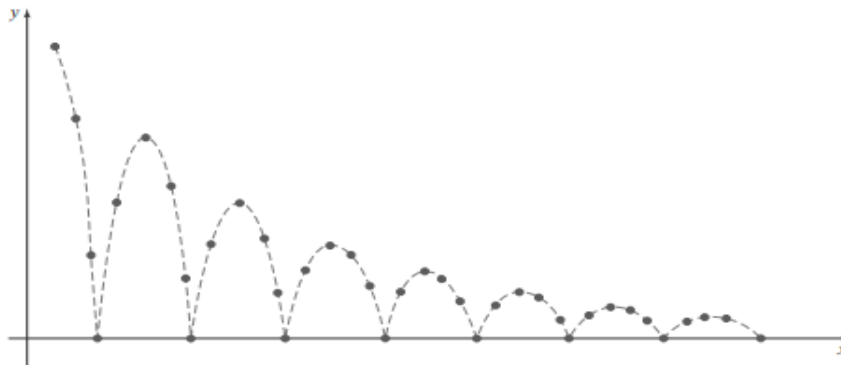
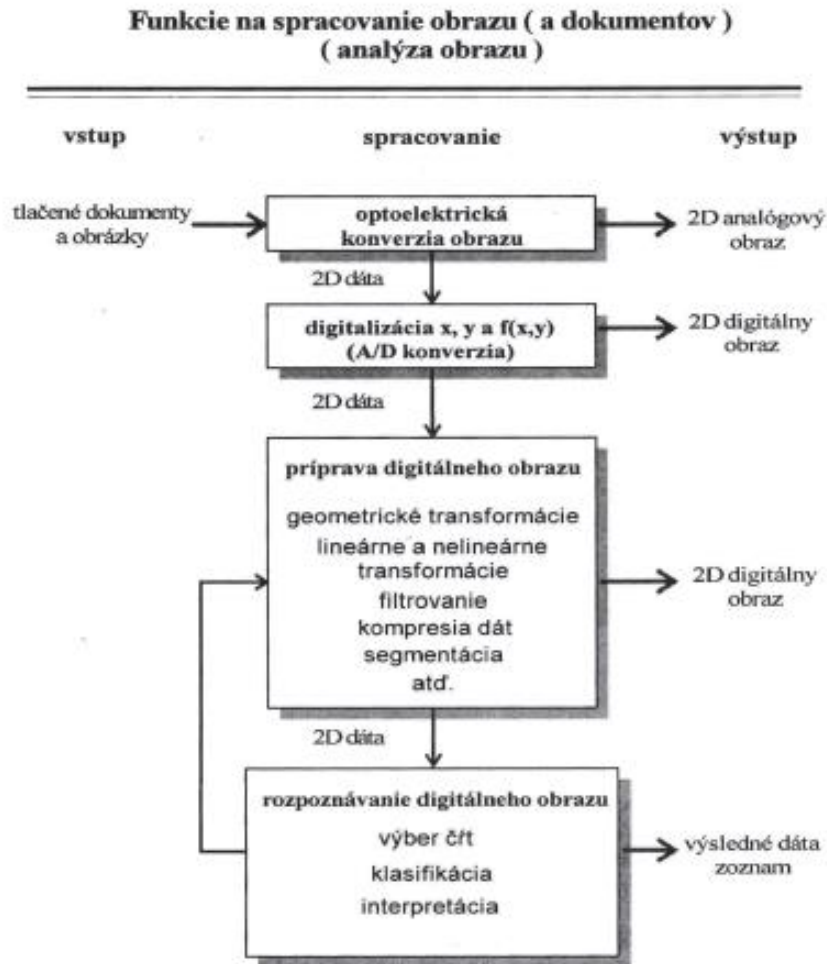


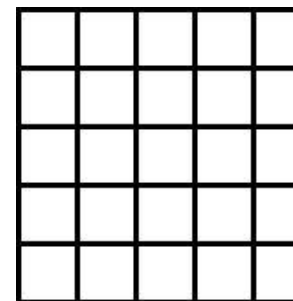
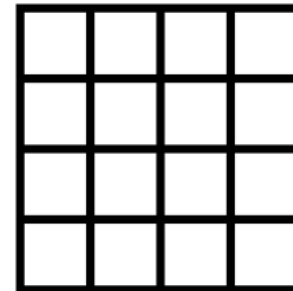
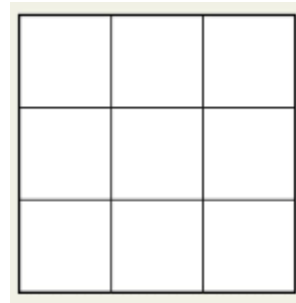
FIGURE 16
Approximating the motion of a
bouncing ball with a damped sine
function (Eq. 10).



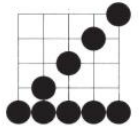
Operácie



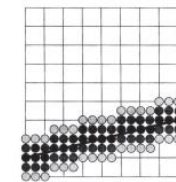
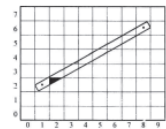
Obr. 1.6 Funkcie na spracovanie obrazu



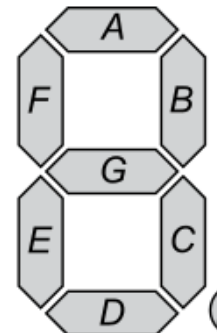
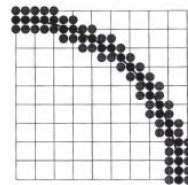
Obr. 5.4 Aliasing úsečiek



Obr. 5.5 Antialiasing úsečky



Obr. 5.6 Hrubé čiary pre vykreslenie úsečky a hranice



DP

1665 >> Logo UK, metadáta, paradáta

File:Rembrandt, Old Man, possibly a portrait of Jan Amos Comenius 1665.jpg



Creation of a Digital Image

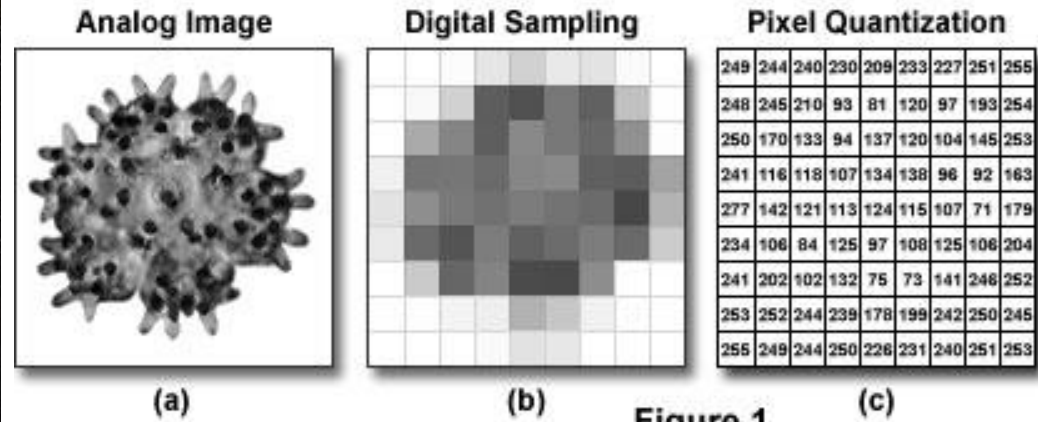


Figure 1

Spatial Resolution Effect on Pixelation in Digital Images

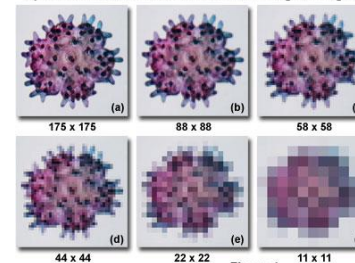
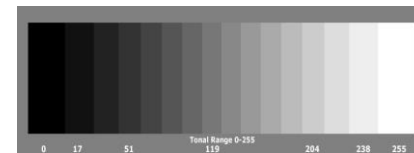
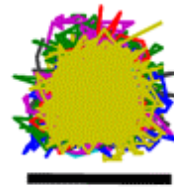
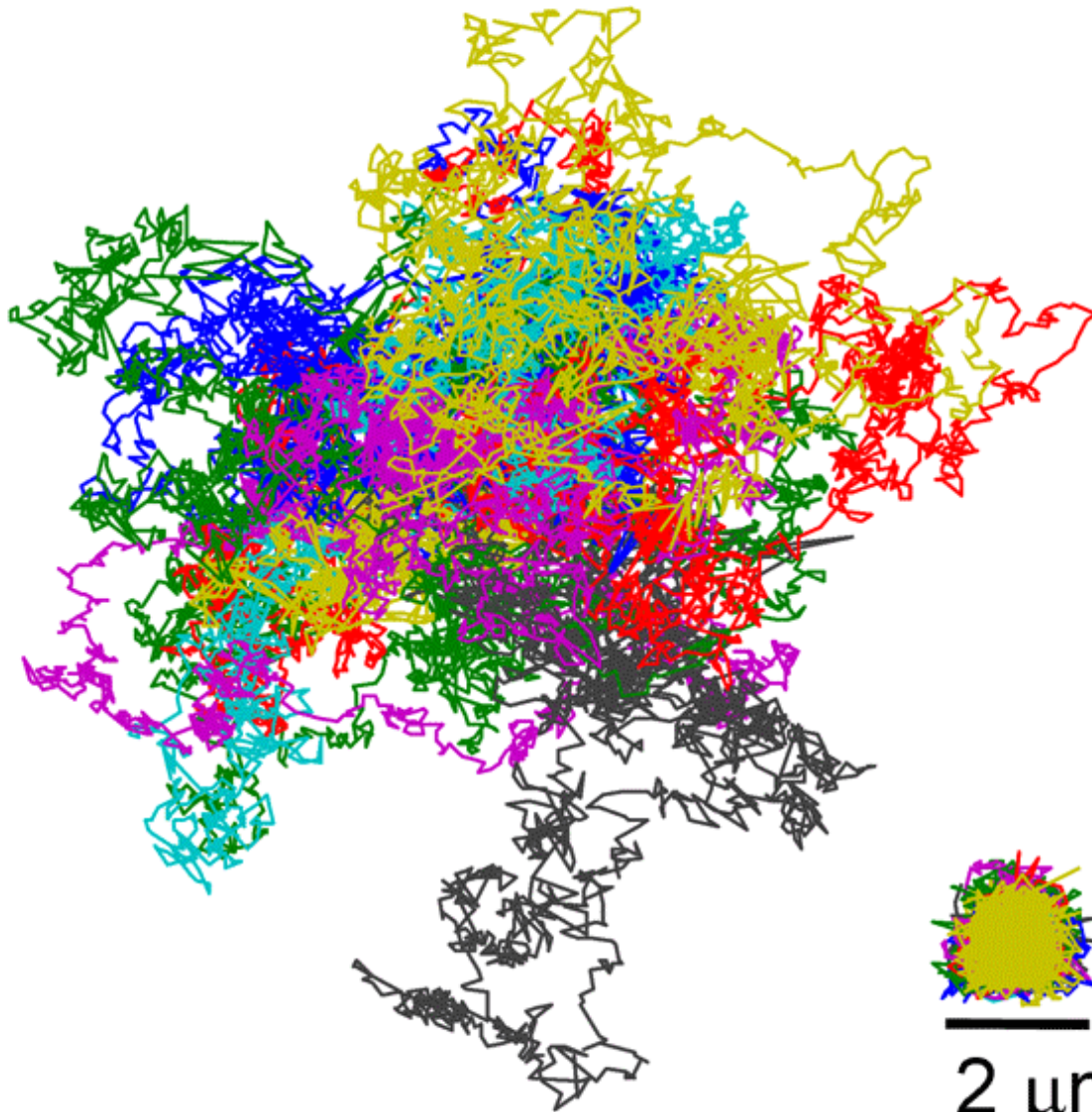


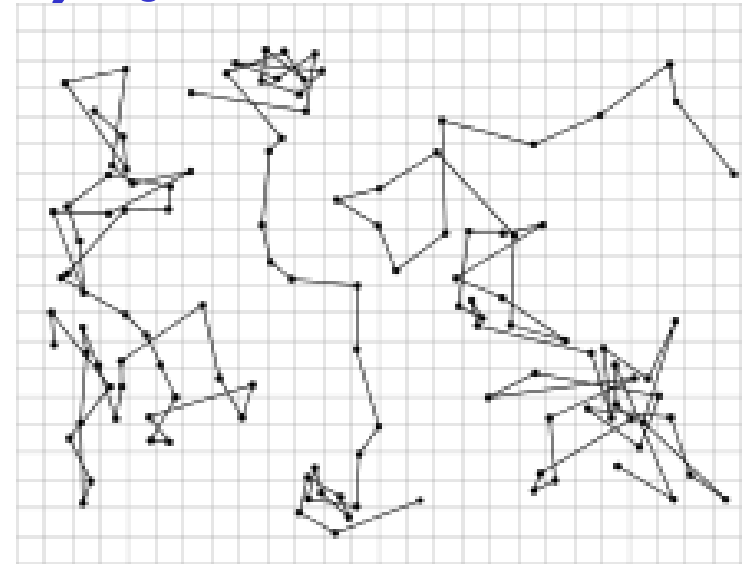
Figure 4



Brown, 1827, AE 1905, Page Rank Random Walk...



2 μm

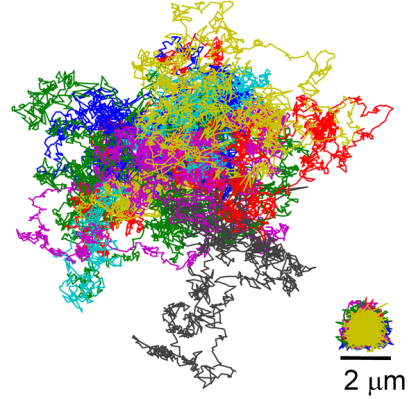


Particle System

Technical Graphics

Applications of Geometry and Problem Solving

Project Maths



Particle System



Q Všetko **Obrázky** Videá Správy Mapy Viac Nastavenia Nástroje

Zberky Bezpečné vyhľadávanie

unity

primordial particle

spark ar

unity particle

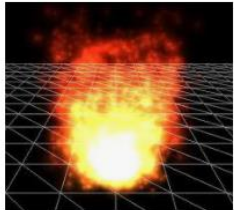
emitter

particle emitter

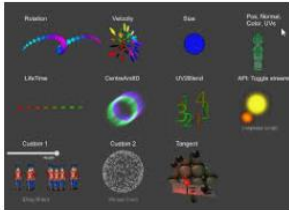
unity3d

animation

unity 3d



Particle system - Wikipedia
en.wikipedia.org



Unity - The Particle System - Tutorialspoint
tutorialspoint.com



Flexible and performant Particle Systems in Unity...
unity3d.com



Self-feeding Particle System by David Torno - Pro...
providecoalition.com



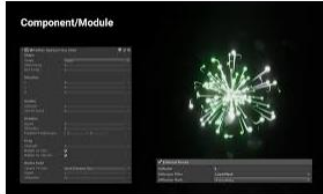
Particle Systems
www2.cs.uregina.ca



Particle System Overview - Valve Developer Commu...
developer.valvesoftware.com



Blender Particle System Overview - ParticleSy...
particlessystems.net



Flexible and performant Particle Systems in Uni...
unity3d.com



PRODUCE DAZZLING PARTICLE EFFECTS
magzter.com



Particle System Basic, by ...
modelingcommons.org



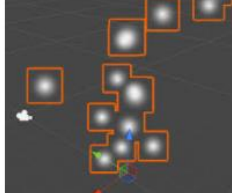
GitHub - skeeto/webgl-particles: Web...
github.com



Particle System | Animation | Dust Simulatio...
m.youtube.com



Unity Particle System Overview - ParticleSystems.Net
particlessystems.net



Introduction to Particle Systems ...
learn.unity.com



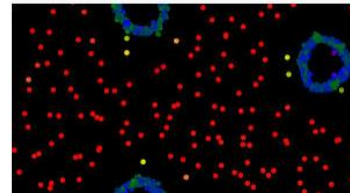
Boris FX | BCC Particle System
borisfx.com



Introduction To Unity: Particle...
raywenderlich.com



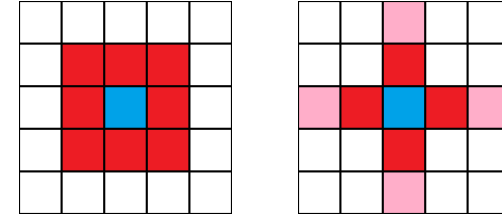
Light 'Boids' Particle System - Finding Essence
ivanrinaldo.weebly.com



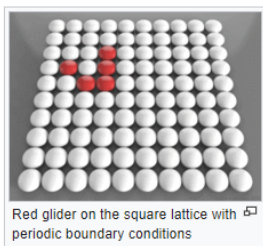
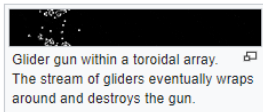
Primordial Particle Systems | Softology's Blog
softologyblog.wordpress.com

Cellular Automaton, LIFE

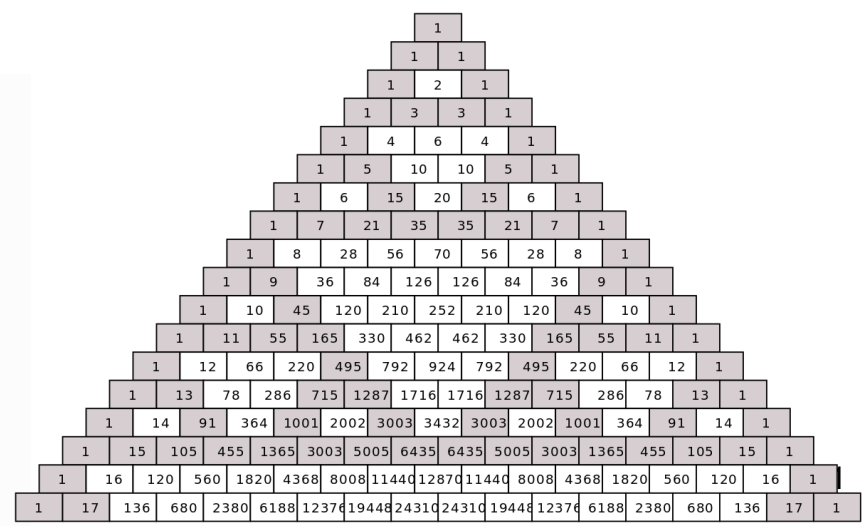
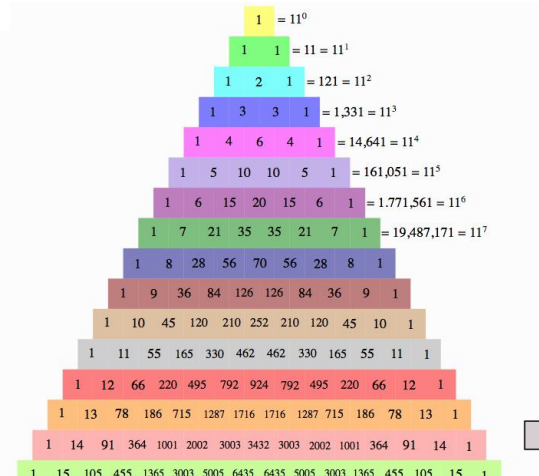
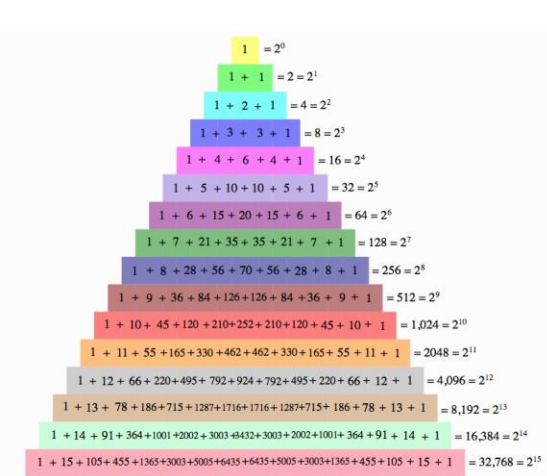
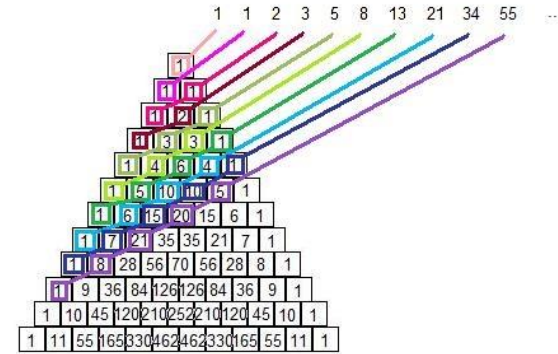
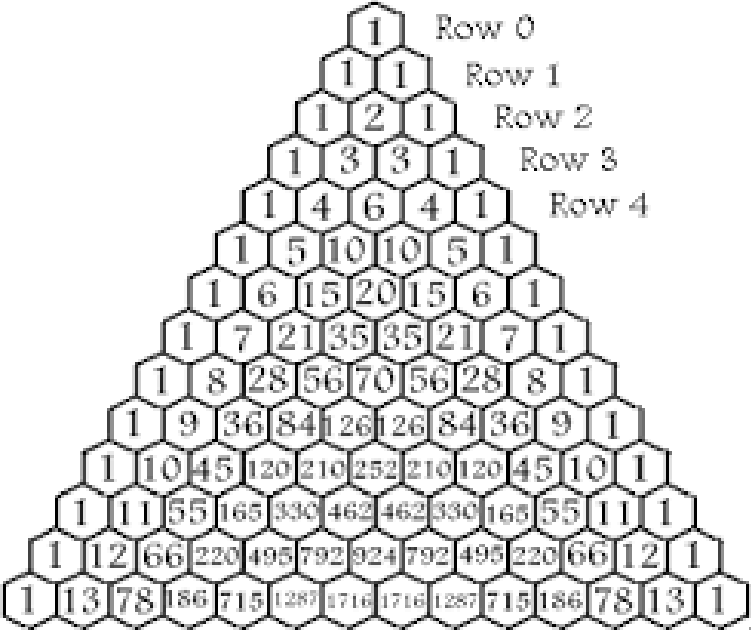
1. Any live cell with two or three live neighbours survives.
 2. Any dead cell with three live neighbours becomes a live cell.
 3. All other live cells die in the next generation. Similarly, all other dead cells stay dead.
- The initial pattern constitutes the *seed* of the system



Still lifes	Oscillators	Spaceships
Block	Blinker (period 2)	Glider
Beehive	Toad (period 2)	Light-weight spaceship (LWSS)
Loaf	Beacon (period 2)	Middle-weight spaceship (MWSS)
Boat	Pulsar (period 3)	Heavy-weight spaceship (HWSS)
Tub	Pentadecathlon (period 15)	

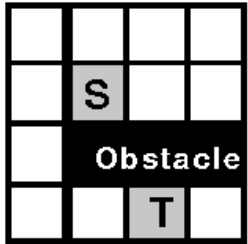


Pascal, 1D CA

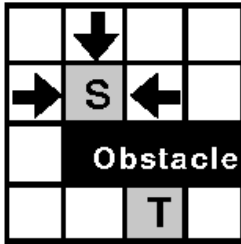


Maze router, Lee

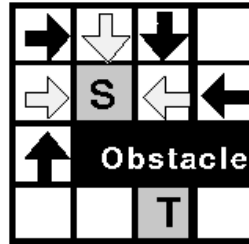
Initialization



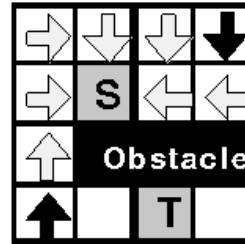
Expand 1



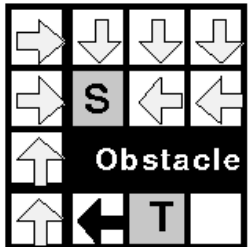
Expand 2



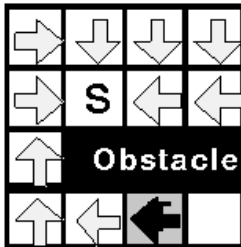
Expand 3



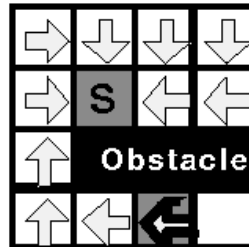
Expand 4



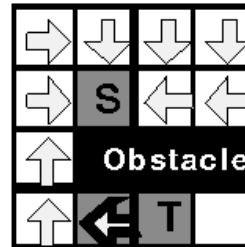
Expand 5



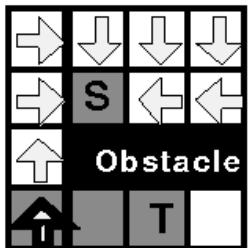
Backtrace 1



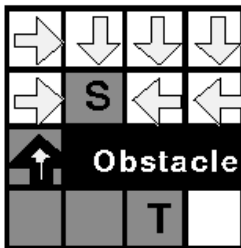
Backtrace 2



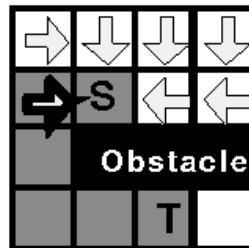
Backtrace 3



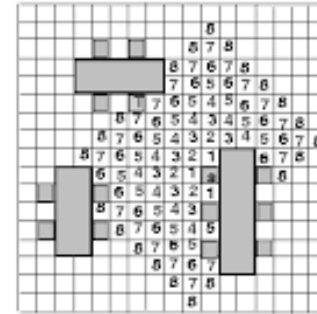
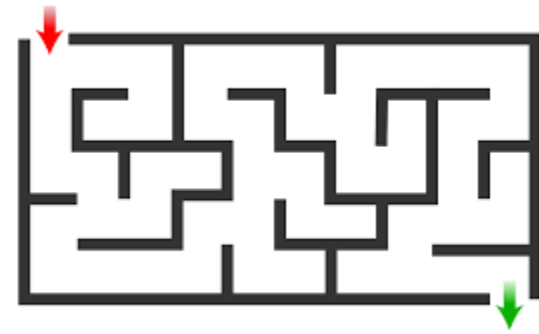
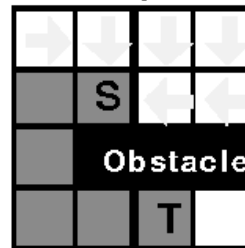
Backtrace 4



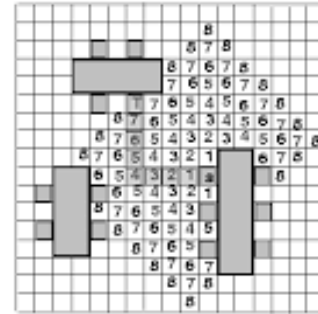
Backtrace 5



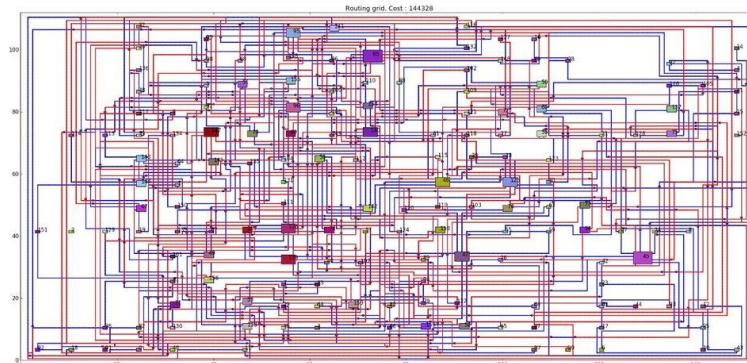
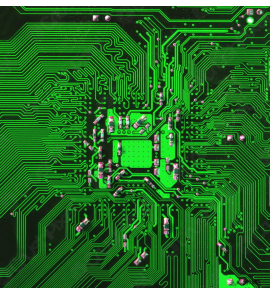
Cleanup

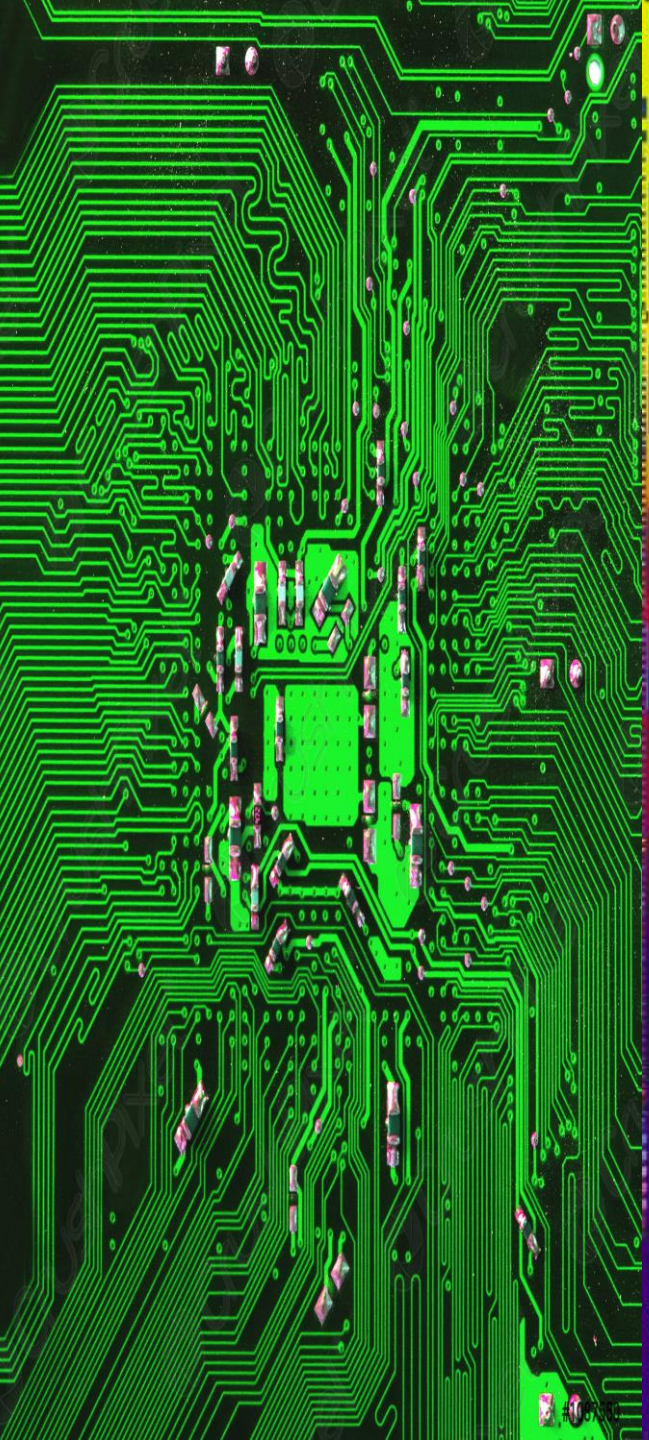


Filing

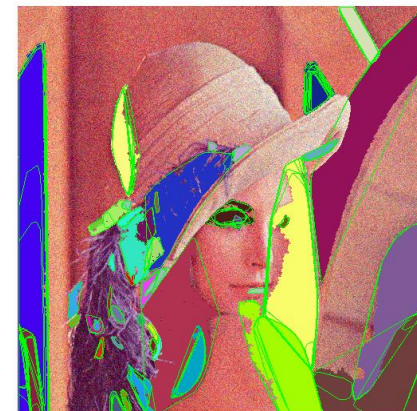
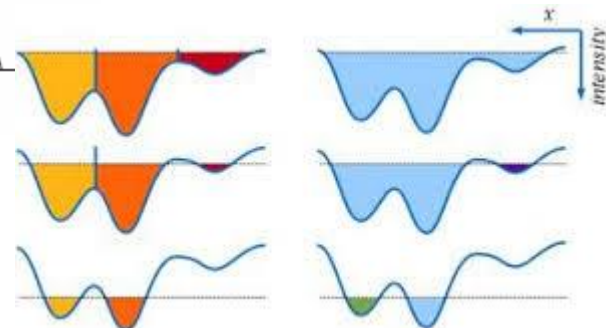
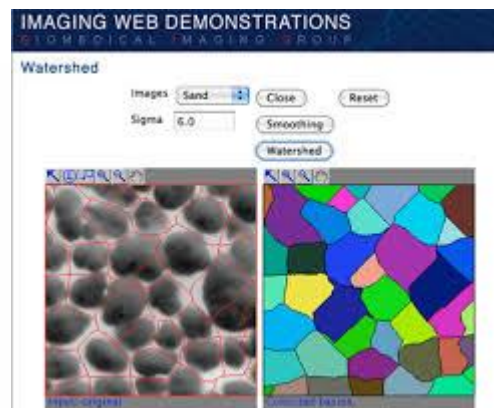
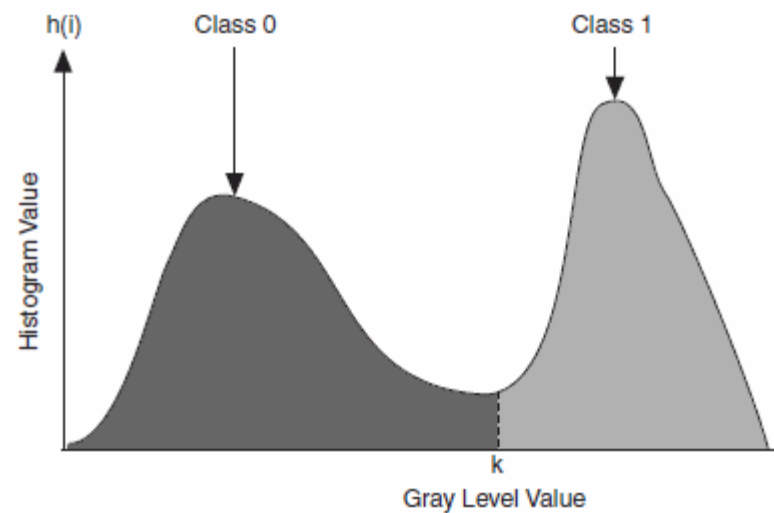
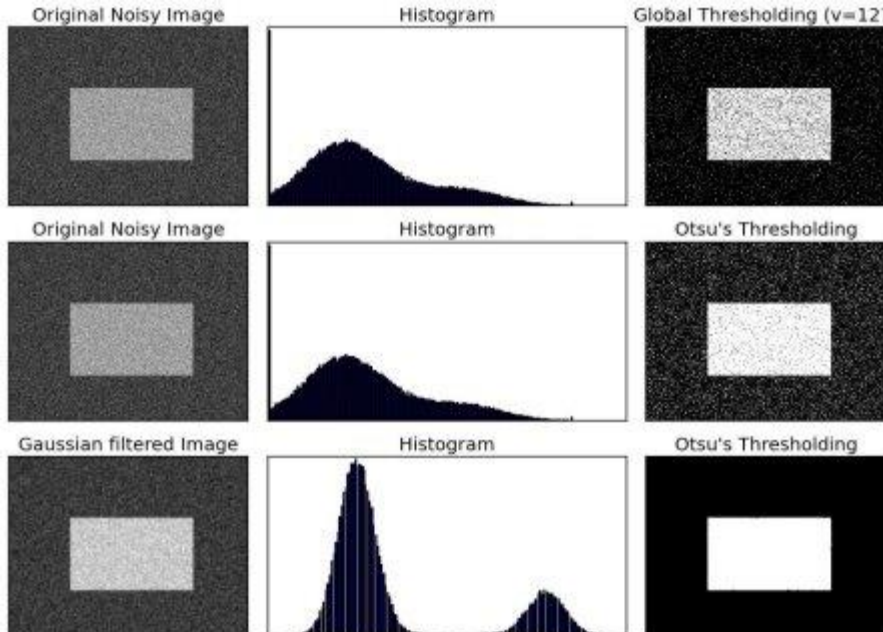


Retrace

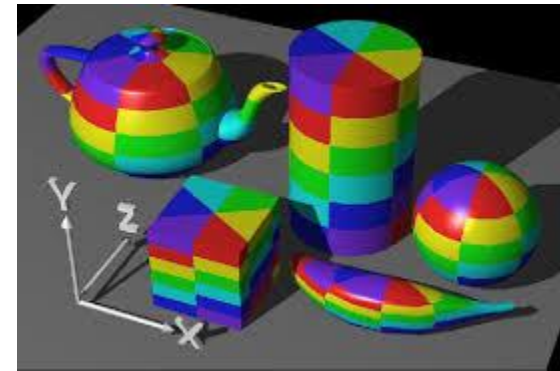
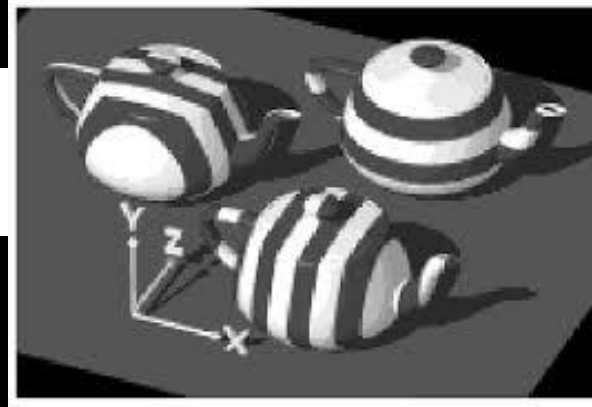
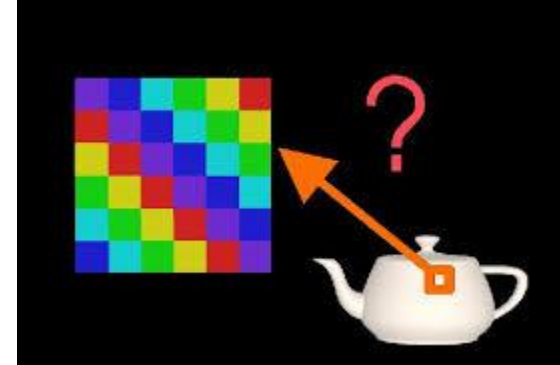
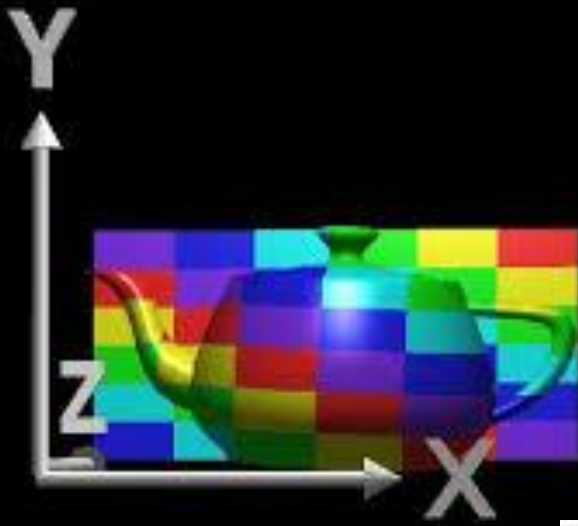




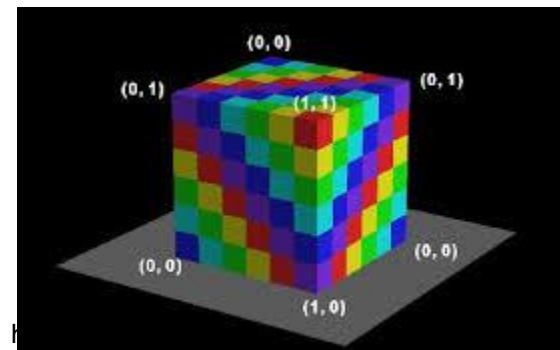
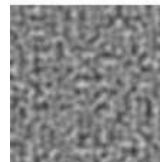
Segmentation



Wolfe, textures, Perlin



$(u+v) \bmod 3$
in $\{R, G, B\}$



1D texture, $t_1 + t_2 + t_3 \bmod 7$, “melody” CA

1. Select 3 values t_{1-3} of “Color” = {c, d, e, f, g, a, h} =_{def} {0, 1, 2, 3, 4, 5, 6},
2. $t_4 = (t_1 + t_2 + t_3) \bmod 7$, resp. $t_i = (t_{i-3} + t_{i-2} + t_{i-1}) \bmod 7$

“The initial pattern constitutes the *seed* of the system”

- e.g. {0, 0, 0} generates {0, 0, ... }
{0, 1, 2} generates {0, 1, 2, 3, 6... }

In total, $343 = 7^3$, folksongs in Central Europe



E. Sikudová
© 1994

Beauty of Computer Graphics - L-systems
by Andrej Ferko, 2003

TU Graz, WS 2003/04
ICG Seminarraum

The image shows a waveform on the left and musical notation on the right. The waveform is a simple line graph with a peak and a valley. The musical notation consists of several staves with notes and rests, illustrating a melody.



C - dur

c d e f g a h c

2 2 1 2 2 2 1

alamy stock photo

The image shows a piano keyboard with the notes C, D, E, F, G, A, H, C labeled. Below the keyboard, the sequence of notes c d e f g a h c is written. A red bracket underlines the sequence 2 2 1 2 2 2 1. Below this, a musical staff shows the notes c, d, e, f, g, a, h, c. The image is watermarked with 'alamy stock photo'.

A. Lindenmayer (1968)

- Use Formal Grammar for Organism Modeling
- 0L-system: triplet $\langle V, w, P \rangle$,
- where V is an alphabet of symbols,
- w (axiom) is the initial string, and
- P is the set of (context free) production rules
- Example: axiom F (draw the line segment)
- Alphabet: $F, +, -$ ($+$ & $-$ mean turn up or down)
- Rules: $+ \rightarrow +, - \rightarrow -, F \rightarrow F+F--F+F$

David G. Green (1993)

- **Set of rules and symbols:**
- **1. VARIABLES - can be replaced**
- **2. CONSTANTS - fixed elements**
- **3. RULES (“syntax”)**
- **4. START**

David G. Green (1993)

- Example: Fibonacci numbers
- *Variables: A B*
- *Constants: none*
- *Start: A*
- *Rules: A* \longrightarrow *B*
- *B* \longrightarrow *AB*

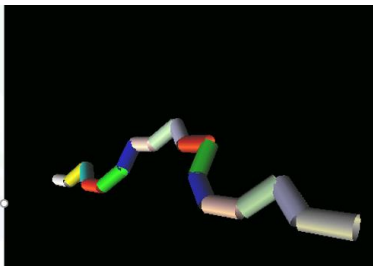
David G. Green (1993)

- Example: *Fibonacci numbers*
- *Stage 0 : A*
- *Stage 1 : B*
- *Stage 2 : AB*
- *Stage 3 : BAB*
- *Stage 4 : ABBAB*
- ...

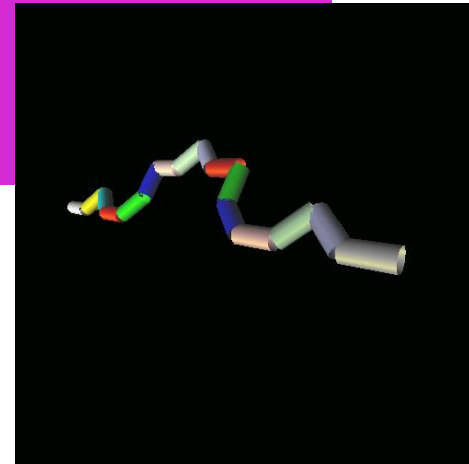
The power of L-systems

- We have to assign a meaning to the symbols and rules.
- The standard use is Turtle Graphics (Seymour Papert, LOGO)
- TURTLE is 2D or 3D cursor, having position and orientation, translation and rotation, draw and move modes, in device coordinates

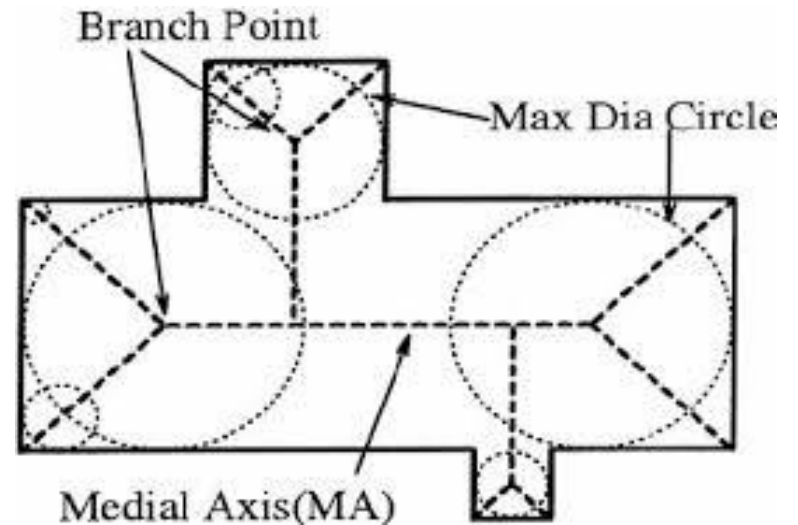
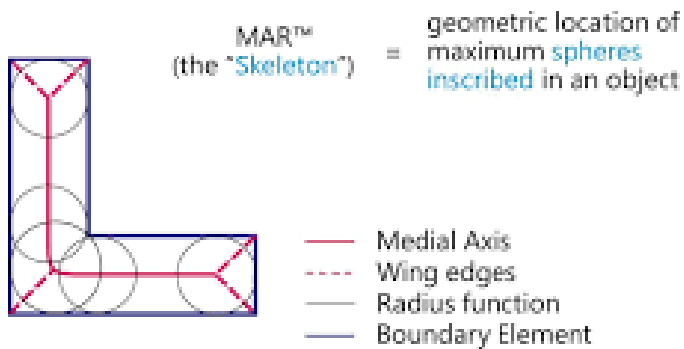
- By Armin Taschwer, TUG 2001
- Depth and grammar:
- Koch_simple:
- D: 3
- G: Start: B
- B = A-A++A-A
- A = ++cF-cF++cF-cF--

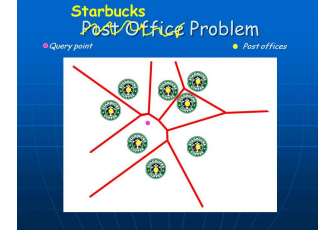


Armin Taschwer, TU Graz 2001



Roof, medial axis, skeleton, prairie fire





1D, 2D, many seeds, Voronoi, post office

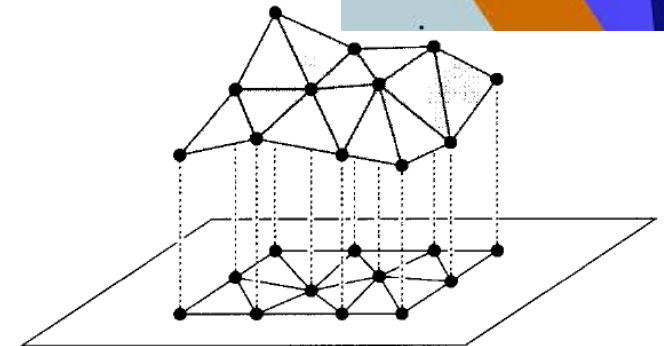
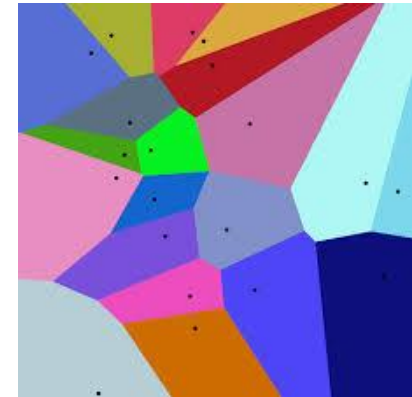
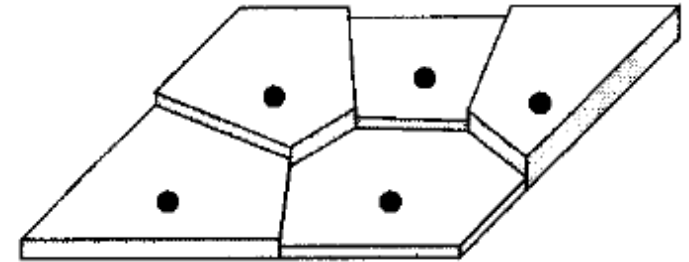
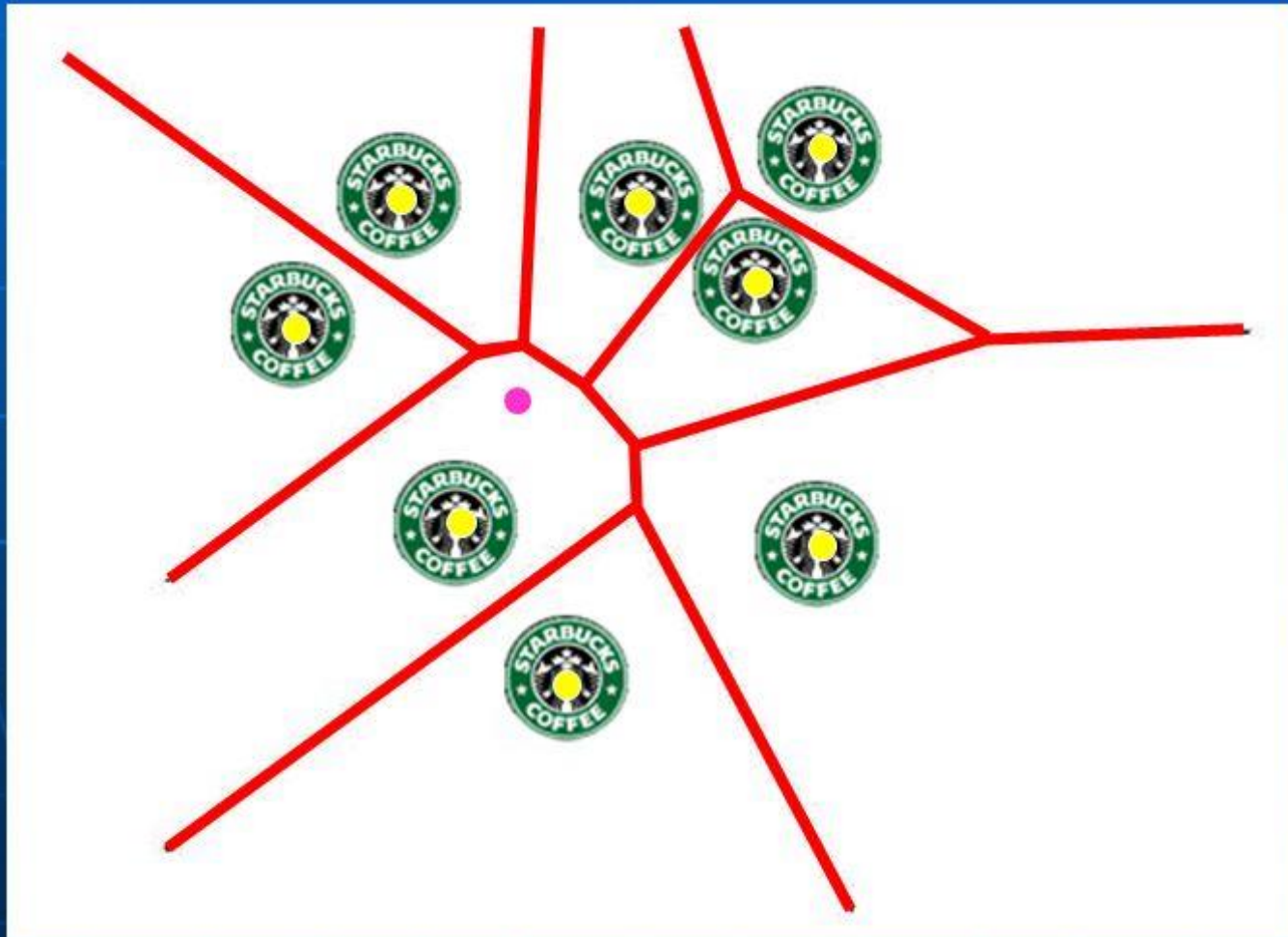


Figure 9.2
Obtaining a polyhedral terrain from a set of sample points

Starbucks Post Office Problem

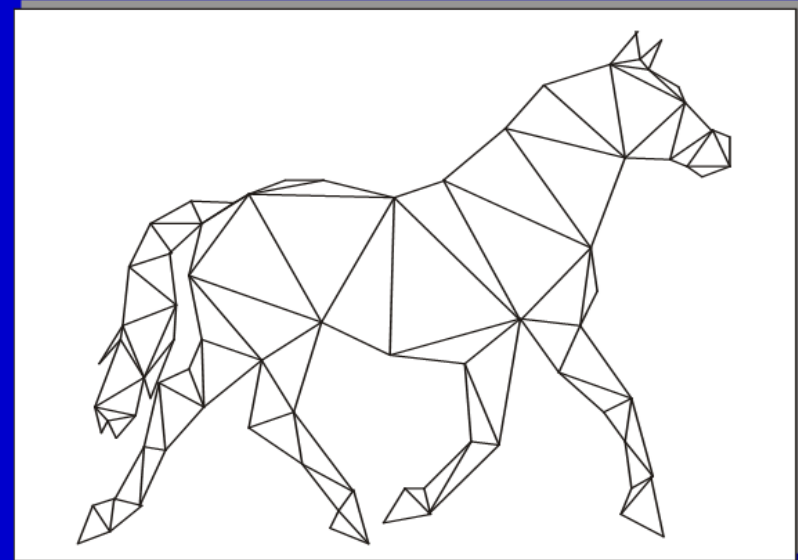
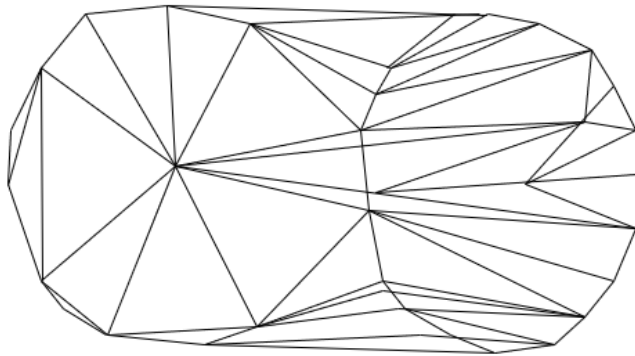
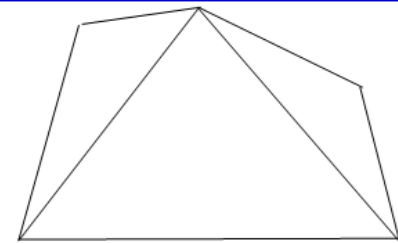
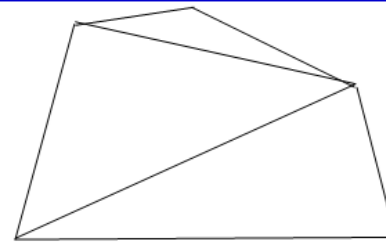
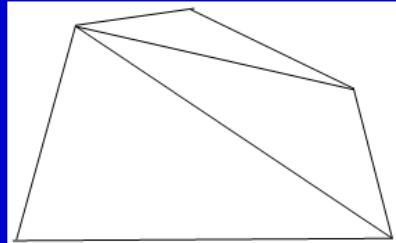
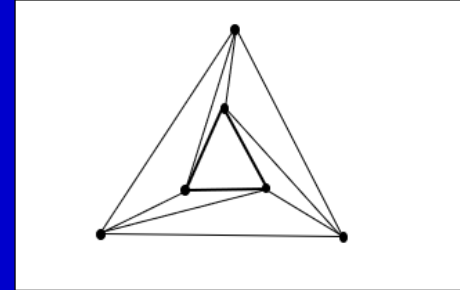
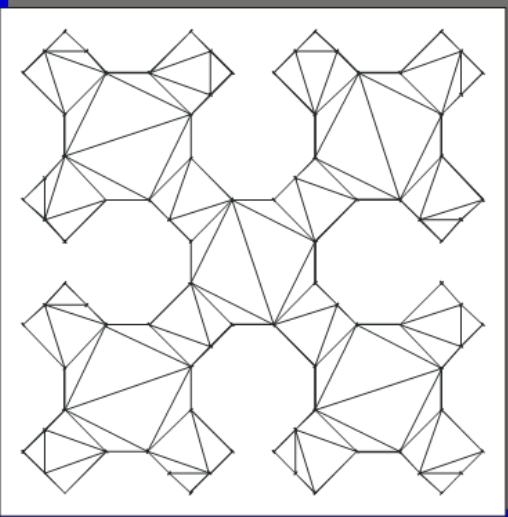
● Query point

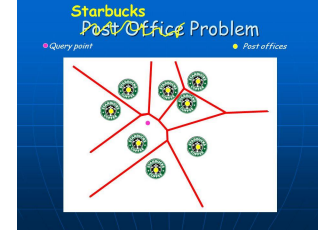
● Post offices



Planar Triangulations

Optimisation criteria, triangle ordering, art gallery...





1D, 2D, many seeds, Voronoi, post office

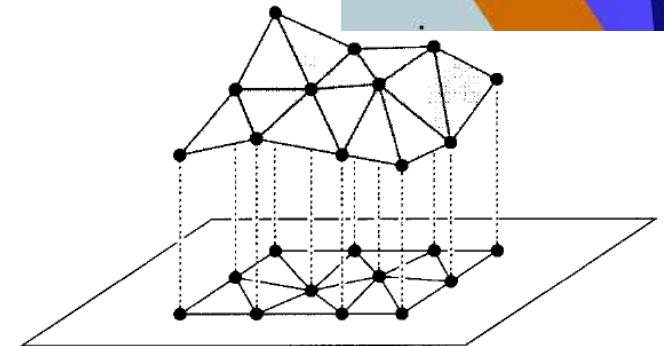
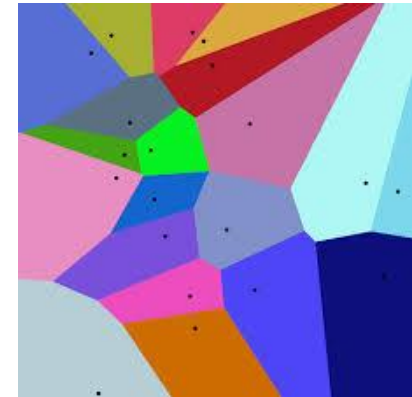
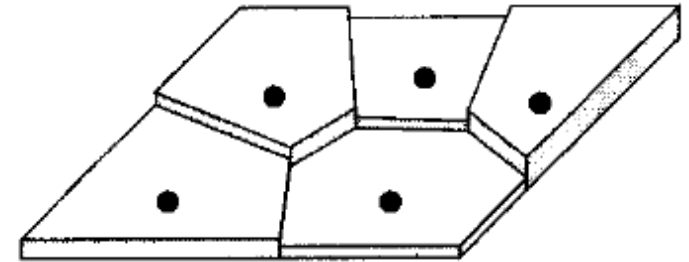


Figure 9.2
Obtaining a polyhedral terrain from a set of sample points

Subjective Edges... terrain, connecting dots

Star Constellations

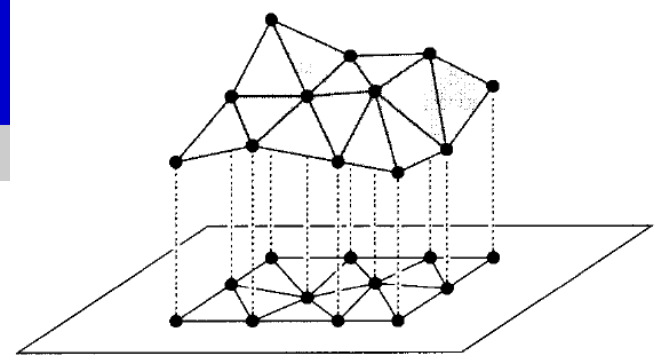
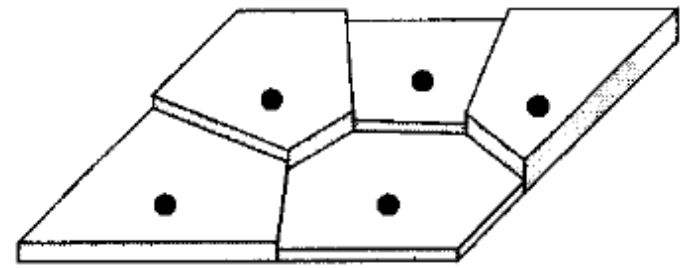
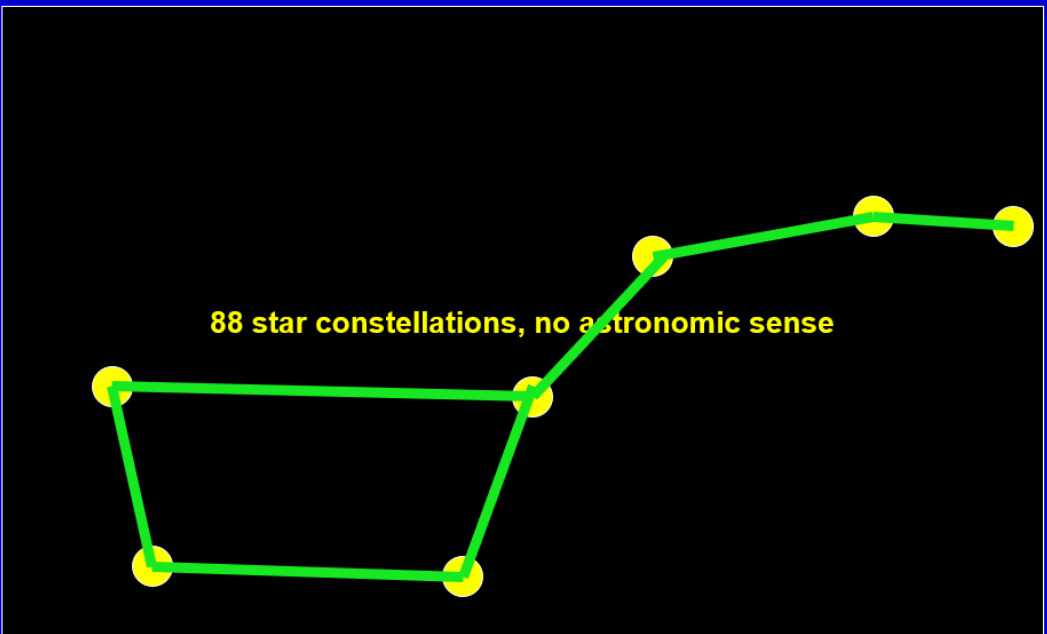
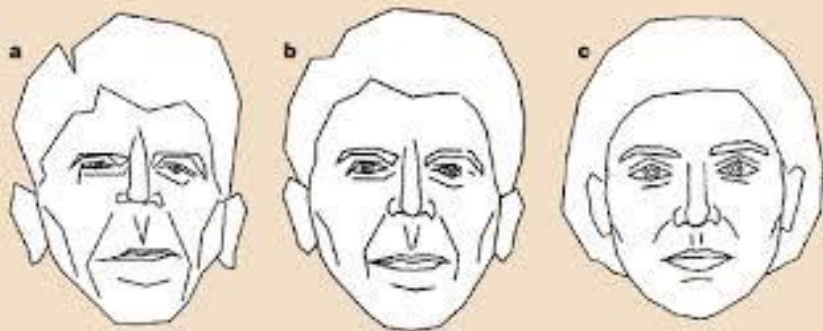
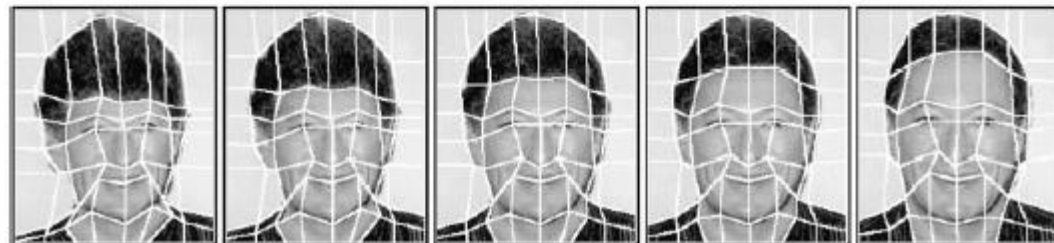
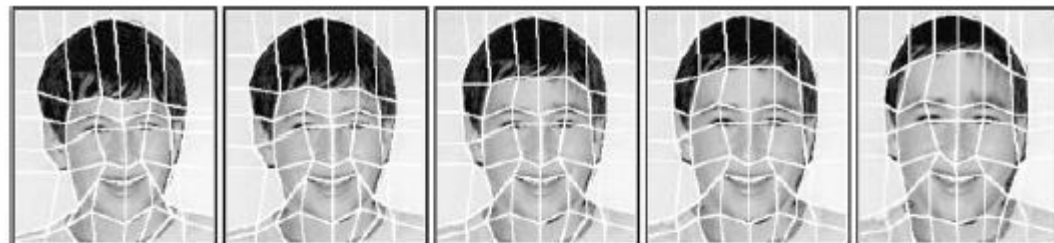
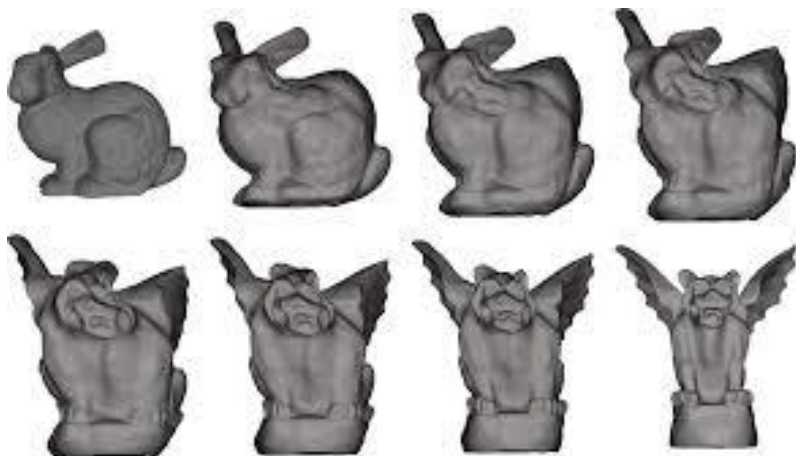


Figure 9.2
Obtaining a polyhedral terrain from a set of sample points

Morphing, warping



Autostitch

- **[BL03]**



25 of 57 images aligned



All 57 images aligned



Final Result

- <http://matthewalunbrown.com/autostitch/autostitch.html>

From digital image (Ruzicky) to IBR (OpenCV pipeline)

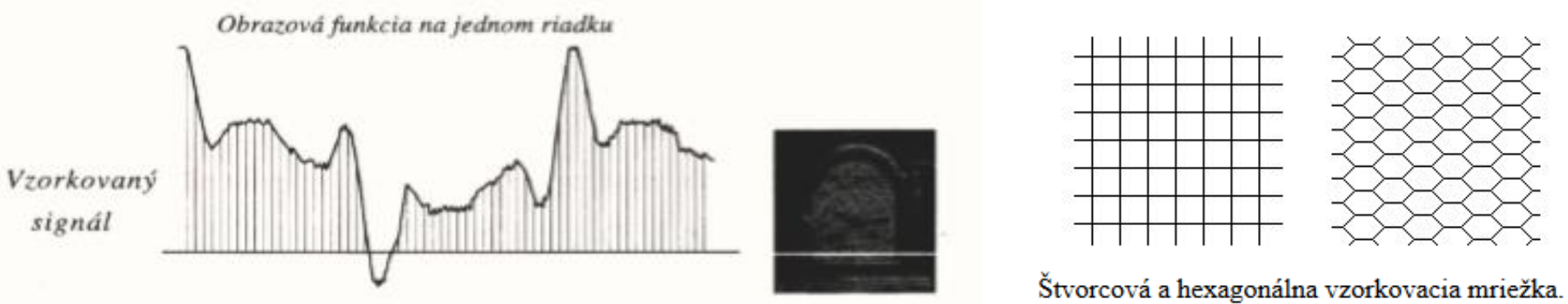


Obr. 6.1 Obrazová funkcia získaná vzorkovaním

Matematické základy spracovania obrazu	75
6.1. Úvod do Fourierovej transformácie	75
6.2. Diskrétna Fourierova transformácia	78
6.3. Niektoré vlastnosti Fourierovej transformácie	82
Jasová korekcia a filtrácia obrazu	87
7.1. Úvod	87
7.2. Jasová korekcia	87
7.3. Zlepšenie obrazu pomocou histogramu	89
7.4. Filtrácia	95
7.5. Vyhľadanie obrazu	98
7.6. Ostrenie obrazu	102
Segmentácia a hranica obrazu	107
8.1. Úvod	107
8.2. Segmentácia prahovaním	107
8.3. Hranica a obrys binárneho obrazu	112
8.4. Segmentácia obrysom	120
8.5. Segmentácia narastaním oblastí	122
Morfologické transformácie a skelet	125
9.1. Úvod do morfológie	125
9.2. Konštrukcia elementárnych transformácií	126
9.3. Kostra (skelet) množiny	131
9.4. Základné pojmy	131
9.5. Algoritmy skeletovania	135



Digital Image



1.2 Vlastnosti digitálneho obrazu

V tejto učebnici slovo **obraz**, alebo šedotónový obraz bude vyjadrovať dvojrozmernú jasovú funkciu $f(x,y)$. Definičným oborom obrazovej funkcie bude rovinná oblasť R :

$$R = \{(x, y), 0 \leq x \leq x_n, 0 \leq y \leq y_m\} \quad (1)$$

kde x, y sú celé čísla, x_n, y_m sú maximálne súradnice. Obor hodnôt je celočíselná množina jasových hodnôt.

V digitálnom obraze môžeme zaviesť **vzdialenosť** medzi dvoma bodmi. Nech (i,j) (k,l) sú dva obrazové elementy, potom vzdialenosť môžeme definovať nasledujúcimi spôsobmi:

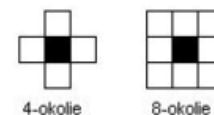
$$D_E = \sqrt{(i-k)^2 + (j-l)^2} \quad (2)$$

$$D_4 = |i-k| + |j-l| \quad (3)$$

$$D_8 = \max\{|i-k|, |j-l|\} \quad (4)$$

Kde D_E je Euklidovská vzdialenosť, ktorá ale nie je vhodná pre diskretný obraz, lebo nemusí vrátiť celé číslo. Vzdialenosti D_4, D_8 určujú najmenší počet jednotkových krokov mriežke. V prípade D_4 posun je povolený len vo zvislom alebo vo vodorovnom smere. V prípade D_8 sú povolené aj diagonálne pohyby.

Ďalším dôležitým pojmom je **susednosť**. Rozlišujeme **4-susednosť** a **8-susednosť**. 4-susedia daného obrazového elementu sú body s jednotkovou vzdialenosťou v D_4 . Tiež sa hovorí **4-okolie**, alebo **8-okolie**.

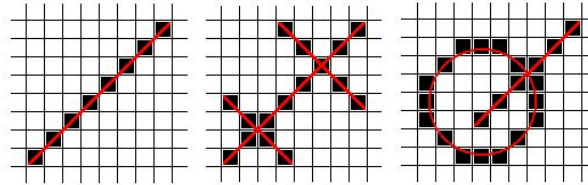


Digital Image Properties

Oblasť je súvislá množina obrazových elementov, pre ktorú platí, že medzi každými dvoma bodmi existuje cesta patriaca celá do tejto množiny. Predpokladajme, že R_i sú oblasti obrazu. Nech R je oblasť ktorá vznikne zjednotením všetkých oblastí R_i . Potom R^c je množinovým doplnkom oblasti R , nazývame ho pozadím.

Objekty sú oblasti, ktoré obvykle odpovedajú entitám zobrazovaného sveta. V jednoduchom praktickom prípade, keď má bod jas väčší ako určitý prah, priradíme ho k objektu.

Súvislosť a susednosť definovaná na diskretnej štvorcovej mriežke nás privedie k určitým paradoxom. Predstavme si úsečku s 45 stupňovým sklonom v digitálnom obraze. Ak uvažujeme 4-susednosť, potom táto úsečka je v každom svojom bode nesúvislá. Ďalším paradoxom je, že dve pretínajúce sa úsečky v digitálnom obraze sa len dotýkajú.



a) v prípade 4-susednosti, úsečka je v každom svojom bode nesúvislá.

b) v pravo hore sa úsečky pretínajú, kým ľavo dole sa len dotýkajú, t.j. nemajú spoločný bod.

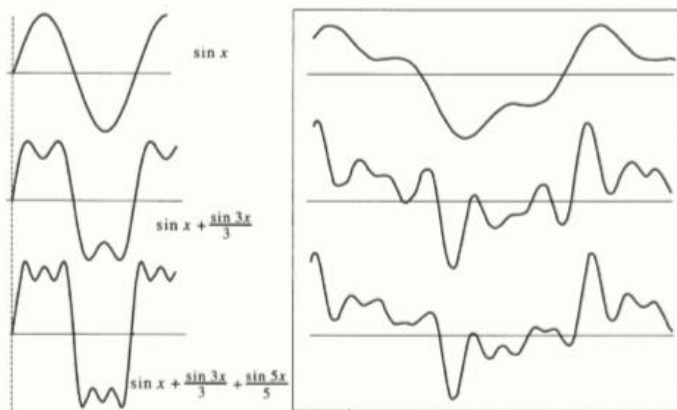
c) z euklidovskej geometrie platí, že uzavretá krivka delí priestor na dve časti. V digitálnom obraze to ale nemusí byť pravda. Na obrázku vidíme kruh, t.j. uzavretú krivku, a úsečku ktorá ju nepretína, ale spája body z vnútra s bodmi z vonkajška.

Jedným riešením pre tieto paradoxy je použiť 8-susednosť pre objekty a 4-susednosť pre pozadie. Je to ale nepraktické riešenie. Ďalšou možnosťou je použiť hexagonálnu mriežku, v ktorej paradoxy nevznikajú. Narazíme ale na realizačný problém, pretože väčšina grafických zariadení podporuje štvorcový raster.

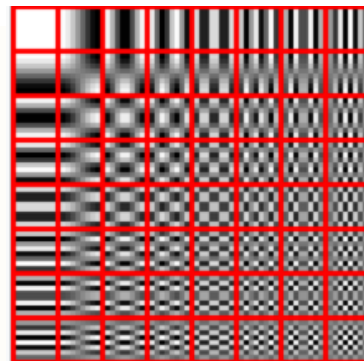
Pod pojmom **hranica oblasti** rozumieme množinu všetkých bodov, ktoré majú aspoň jedného suseda, ktorý nepatrí do oblasti. V digitálnom obraze rozlišujeme **vonkajšiu** a **vnútornú** hranicu. Pre vnútornú hranicu potom platí predošlá definícia. Vonkajšia hranica je hranicou pozadia.

DIP - Digital Image Processing, Interaktívna učebnica spracovania obrazu
Copyright©2003-06 Gábor Blásovits, Katedra aplikovanej informatiky FMFI UK Bratislava

[Ru], W_i , [Kalra]

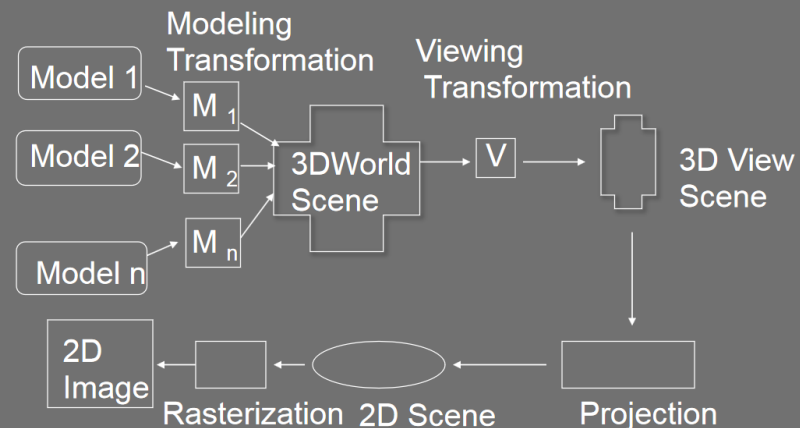


a) b)
Obr. 6.2 Signál vyjadrený frekvenciou harmonických funkcií



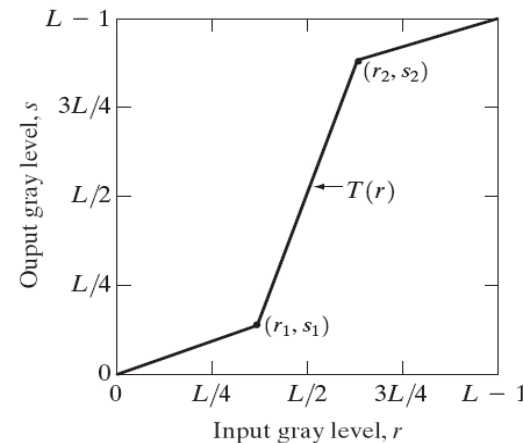
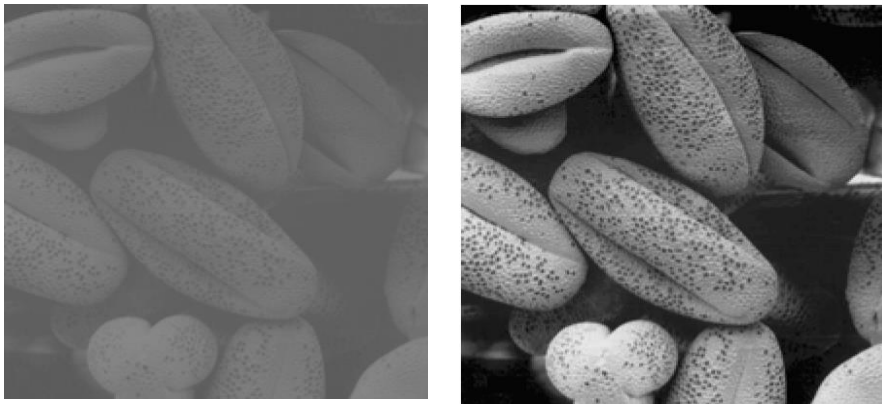
The DCT transforms an 8x8 block of input values to a linear combination of these 64 patterns. The patterns are referred to as the two-dimensional DCT *basis functions*, and the output values are referred to as *transform coefficients*. The horizontal index is u and the vertical index is v .

Graphics Rendering Pipeline



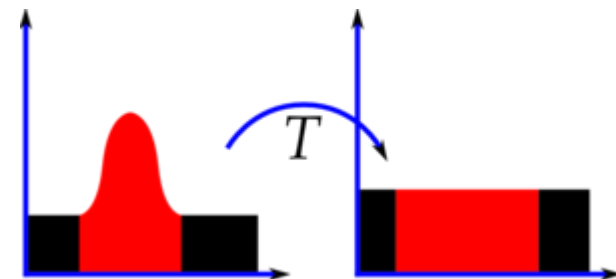
Digital Image Processing (Low Level)

- The negative of an image with gray levels in the range $[0, 255]$, $s = 255 - r$
- Contrast stretching by increasing the dynamic range [Benesova]



- The histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the k -th gray level and n_k is the number of pixels in the image having gray level r_k [Benesova] >> EQUALIZATION

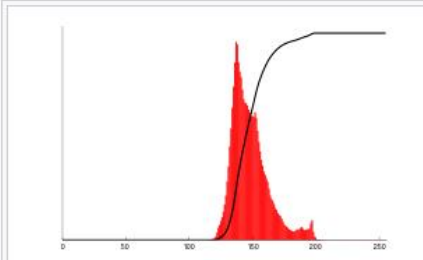
- Image averaging
- Image subtraction
- Smoothing/sharpening



Histogram Equalization



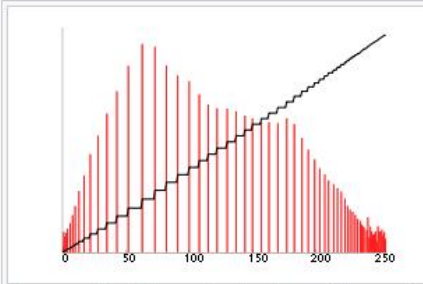
Before Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)



After Histogram Equalization

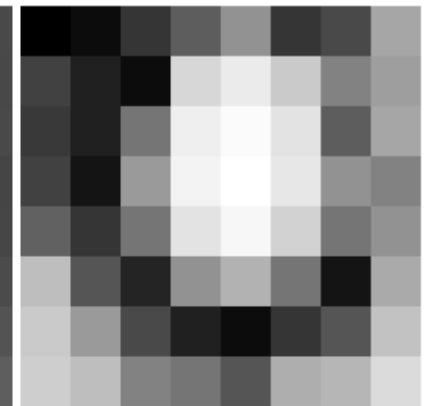


Corresponding histogram (red) and cumulative histogram (black)

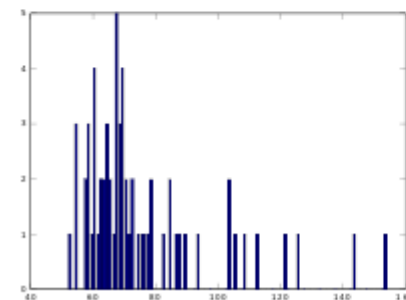
https://en.wikipedia.org/wiki/Histogram_equalization



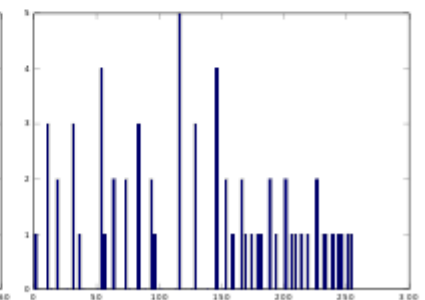
Original



Equalized



Histogram of Original image



Histogram of Equalized image

Digital Image Transformations

- Translation, rotation, scale, symmetry, skew >> dip.sccg.sk
- Pixel approximation: nearest neighbour (1 pixel), bilinear interpolation (4 pixels), bicubic interpolation (9 pixels)

V praxi sa táto rovnica nahradzuje bilineárnou transformáciou, alebo afinnou transformáciou. Bilineárna transformácia má tvar:

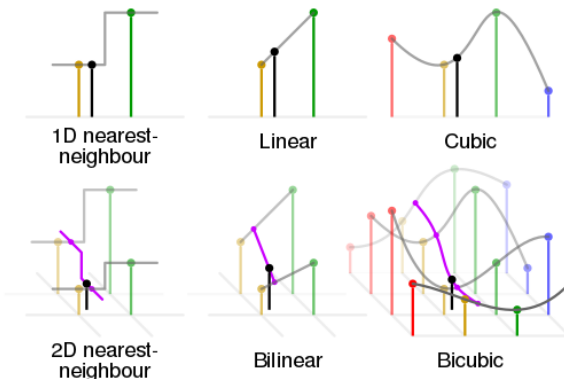
$$\begin{aligned} x_1 &= a_0 + a_1x + a_2y + a_3xy \\ y_1 &= b_0 + b_1x + b_2y + b_3xy \end{aligned} \quad (3)$$

Na jeho určenie potrebujeme štyri dvojice vstupných a výstupných bodov.
Na určenie afinnej transformácie stačia tri dvojice bodov, a má tvar:

$$\begin{aligned} x_1 &= a_0 + a_1x + a_2y \\ y_1 &= b_0 + b_1x + b_2y \end{aligned} \quad (4)$$

Pomocou homogénnych súradníc môžeme afinné transformácie vyjadriť v maticovom tvare

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$



https://en.wikipedia.org/wiki/Bicubic_interpolation#/media/File:Comparison_of_1D_and_2D_interpolation.svg

- Fourier transform >> dip.sccg.sk, Ruzicky, Sikudova



Spracovanie obrazu apl.

Andrej FERKO

Comenius University Bratislava

PG1, 7. 12. 2020, FMFI UK