

OpenCV

Andrej Lúčný

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

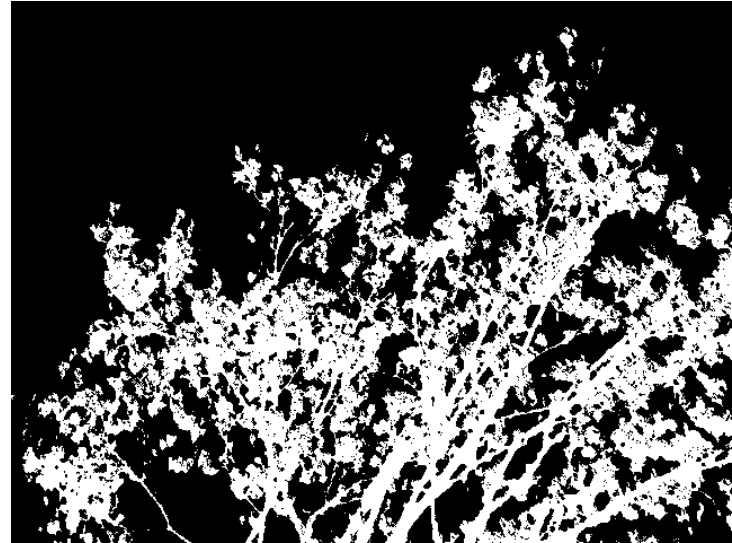
http://dai.fmph.uniba.sk/w/Andrej_Lucny

www.agentspace.org/opencv

BGRA model



α -blending (zmiešavanie)



Seamless Cloning

- čo keď nemáme alpha zložku?



- neprenášame obraz, ale deriváciu obrazu
- v 2D, rekonštrukcia obrazu z derivácii a intenzít na hranici nie je jednoduchý problém, rieši sa iteratívnym algoritmom, tzv. solverom Poissonovej rovnice

Normal Cloning

- berieme derivácie len z prenášaného obrazu



Mixed Cloning

- berieme maximum derivácii z oboch obrazov



Afina transformácia

$$x' = a_x x + b_x y + t_x$$

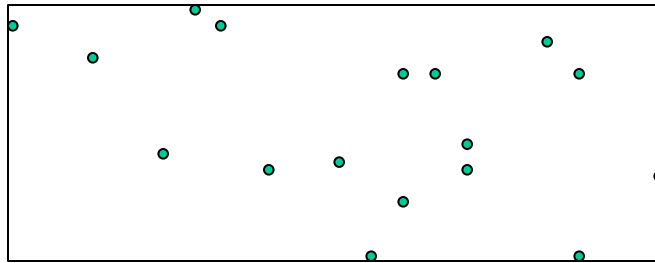
$$y' = a_y x + b_y y + t_y$$

- z troch bodov (neležiacich na priamke) je možné určiť afínnu transformáciu, ktorá projektuje trojuholník na trojuholník

`cv2.getAffineTransform`

`cv2.warpAffine`

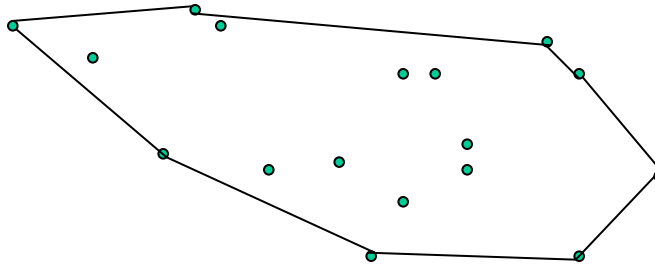
Ohraničujúci obdĺžnik



- Pre množinu bodov je možné ľahko vypočítať ohraničujúci obdĺžnik

`cv2.boundingBox`

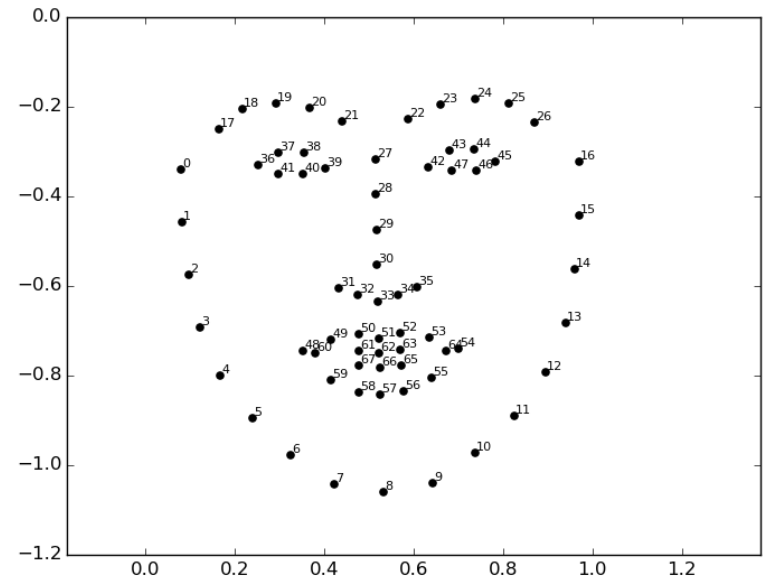
Konvexný obal



- Pre množinu bodov je možné vypočítať konvexný obal

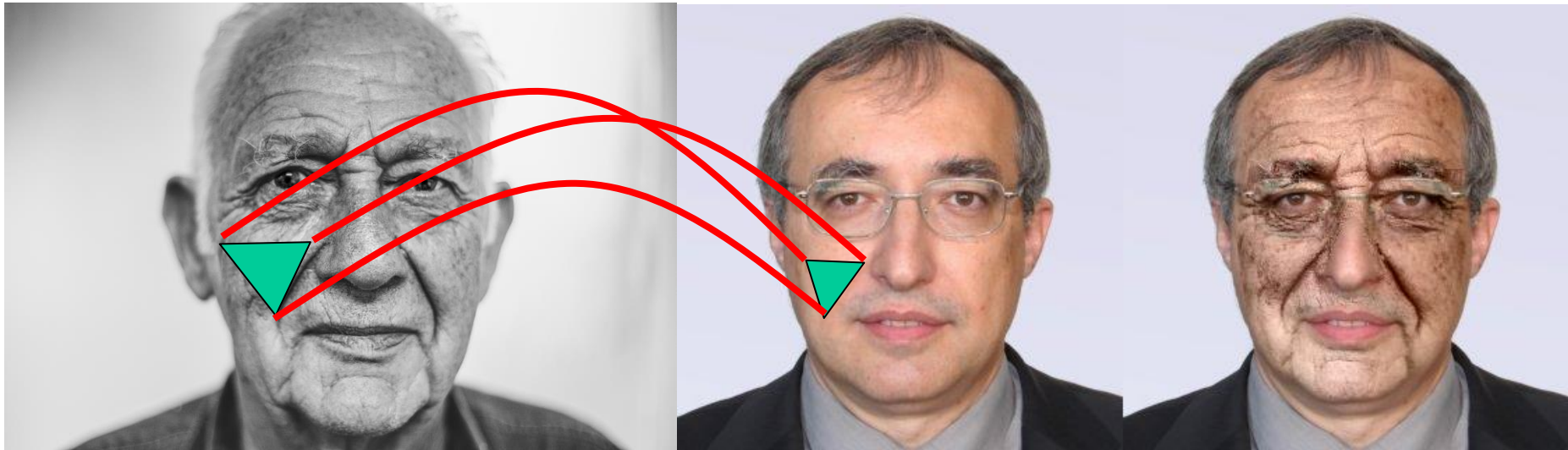
`cv2.convexHull`

Tvárové črty (face landmarks)



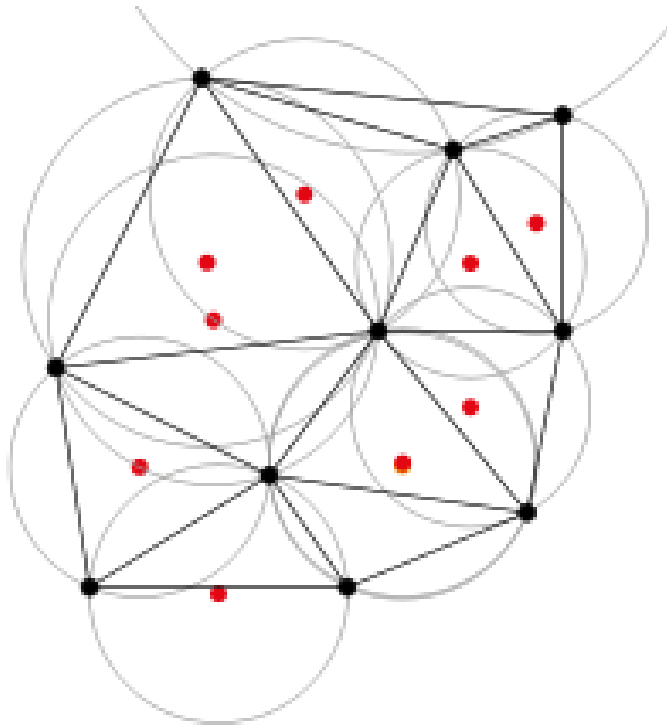
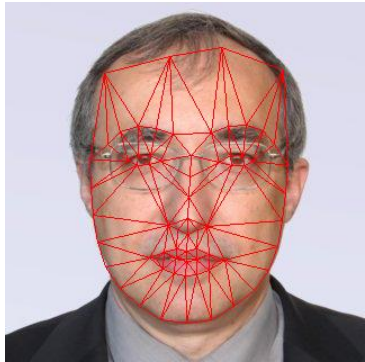
Morphing

- Ľubovoľný trojuholník vieme projektovať na ľubovoľný trojuholník
- Takže ak chceme projektovať jeden tvar na druhý, stačí ich rovnako triangulovať



Delanay triangulation

- Jednoznačnú a vhodnú trianguláciu poskytuje Delanayov algoritmus

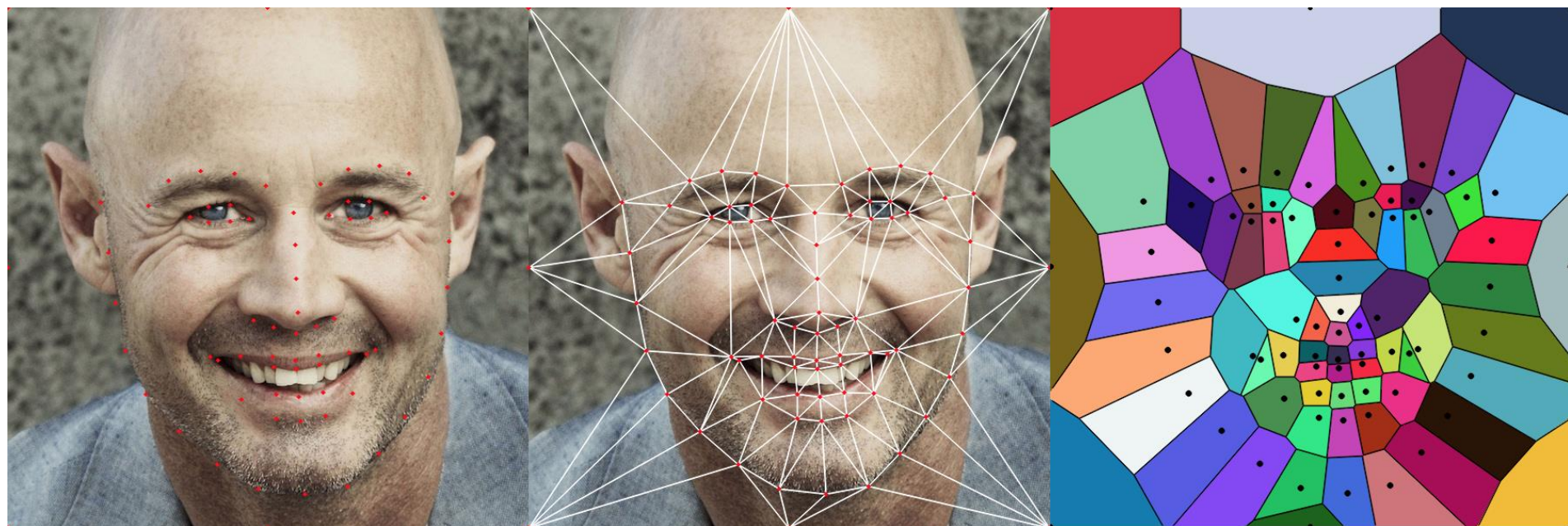


- Žiadny bod nesmie ležať v kružnici opísanej ľubovoľnému trojuholníku
- = Duálny problém k Voronoi diagramu
- Algoritmus pre výpočet oboch je efektívny

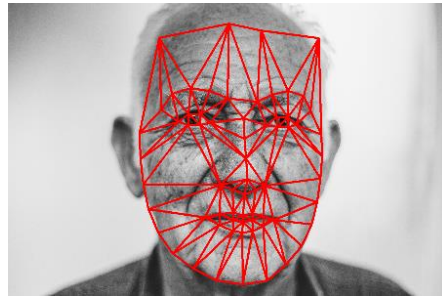
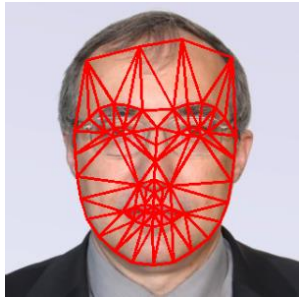
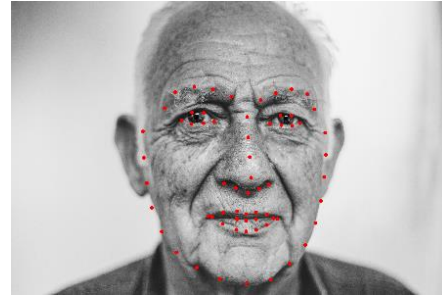
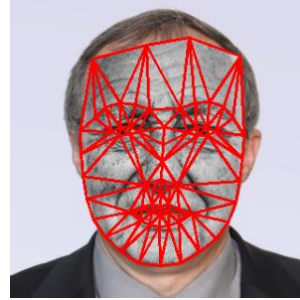
Face
landmarks

Delaunay
triangulation

Voronoi
diagram



Oldify

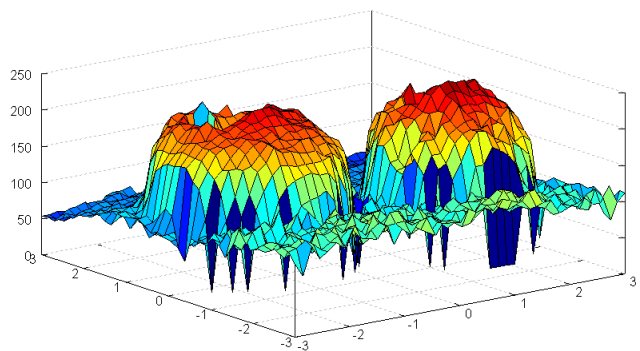


Inpainting

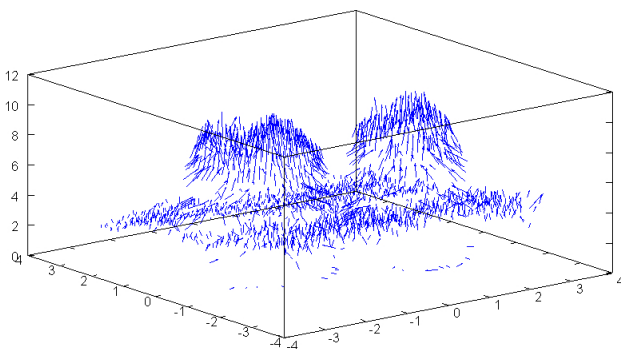


- Odstránenie rýh zo starých fotografií, odstránenie nápisov cez obraz, odstránenie vandalizácie obrazu či nežiadúcich prvkov na obraze

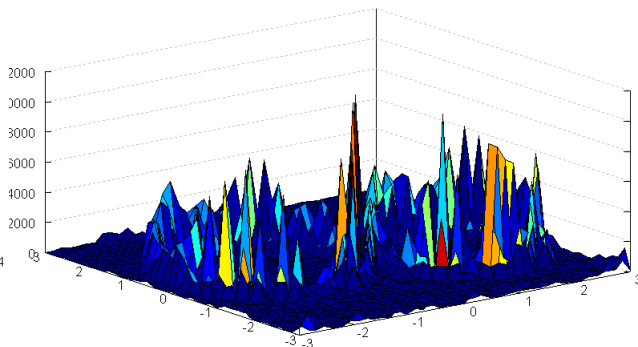
Obraz ako funkcia



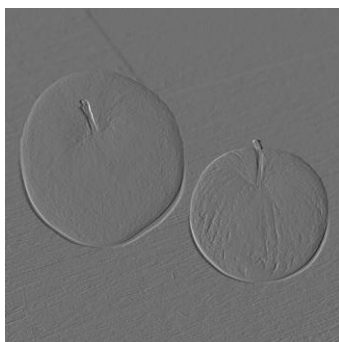
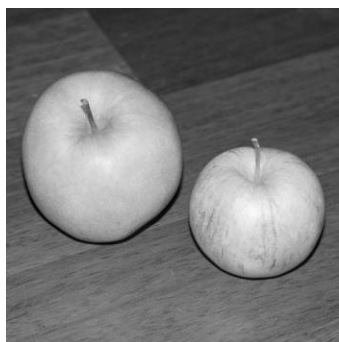
I



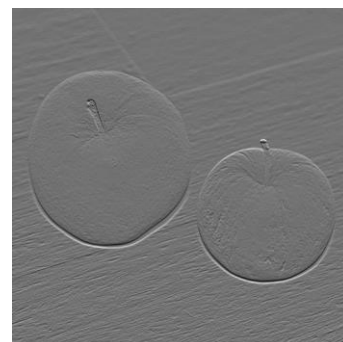
$$\nabla I = \left(\frac{dI}{dx}, \frac{dI}{dy} \right)$$



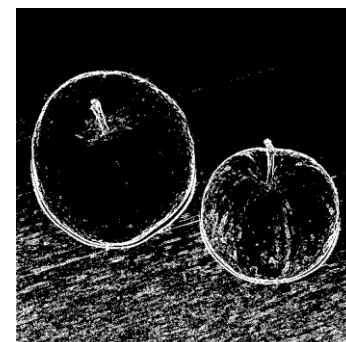
$$\nabla I \cdot \nabla I = \Delta I$$



dx



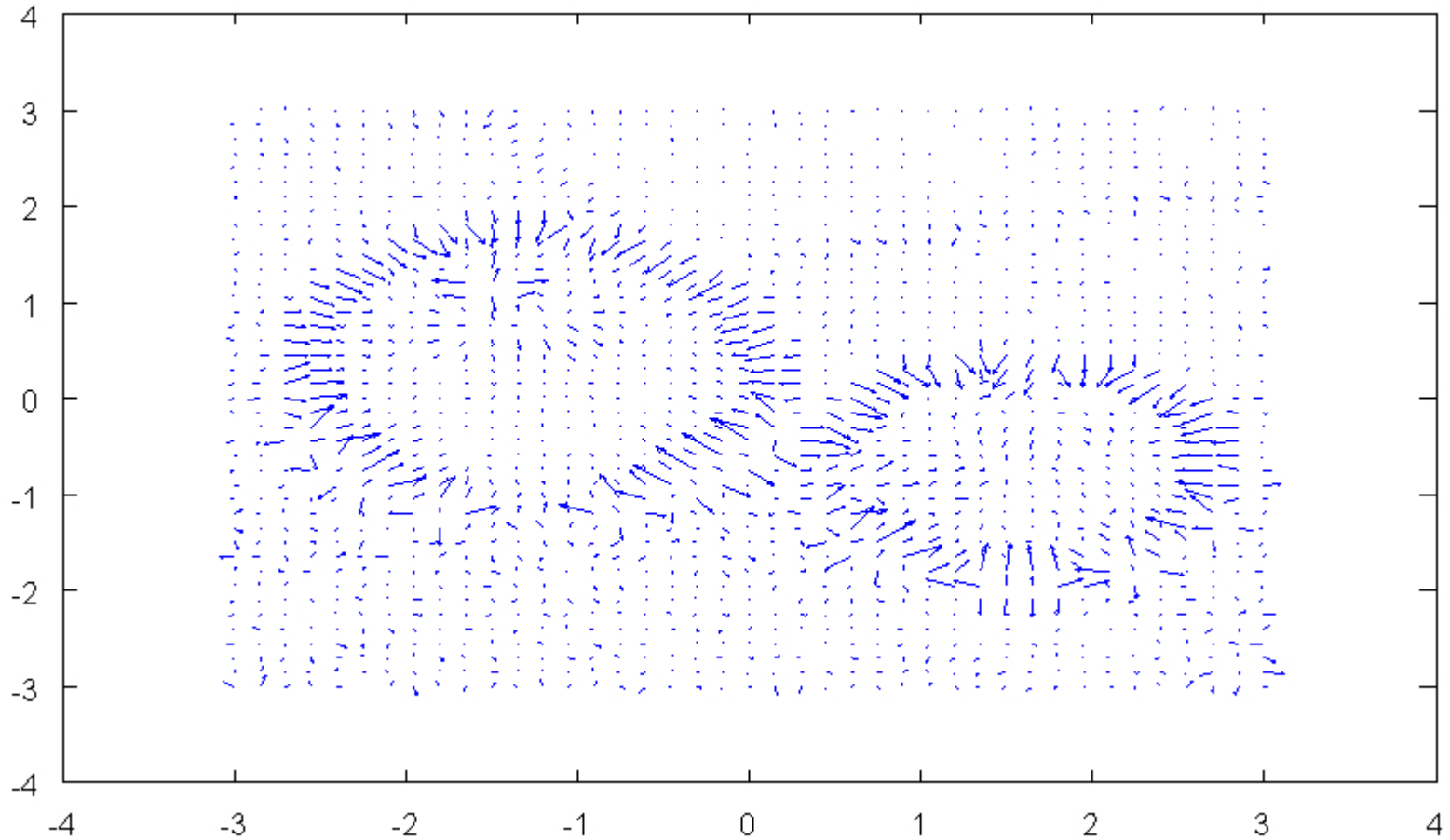
dy



Obraz

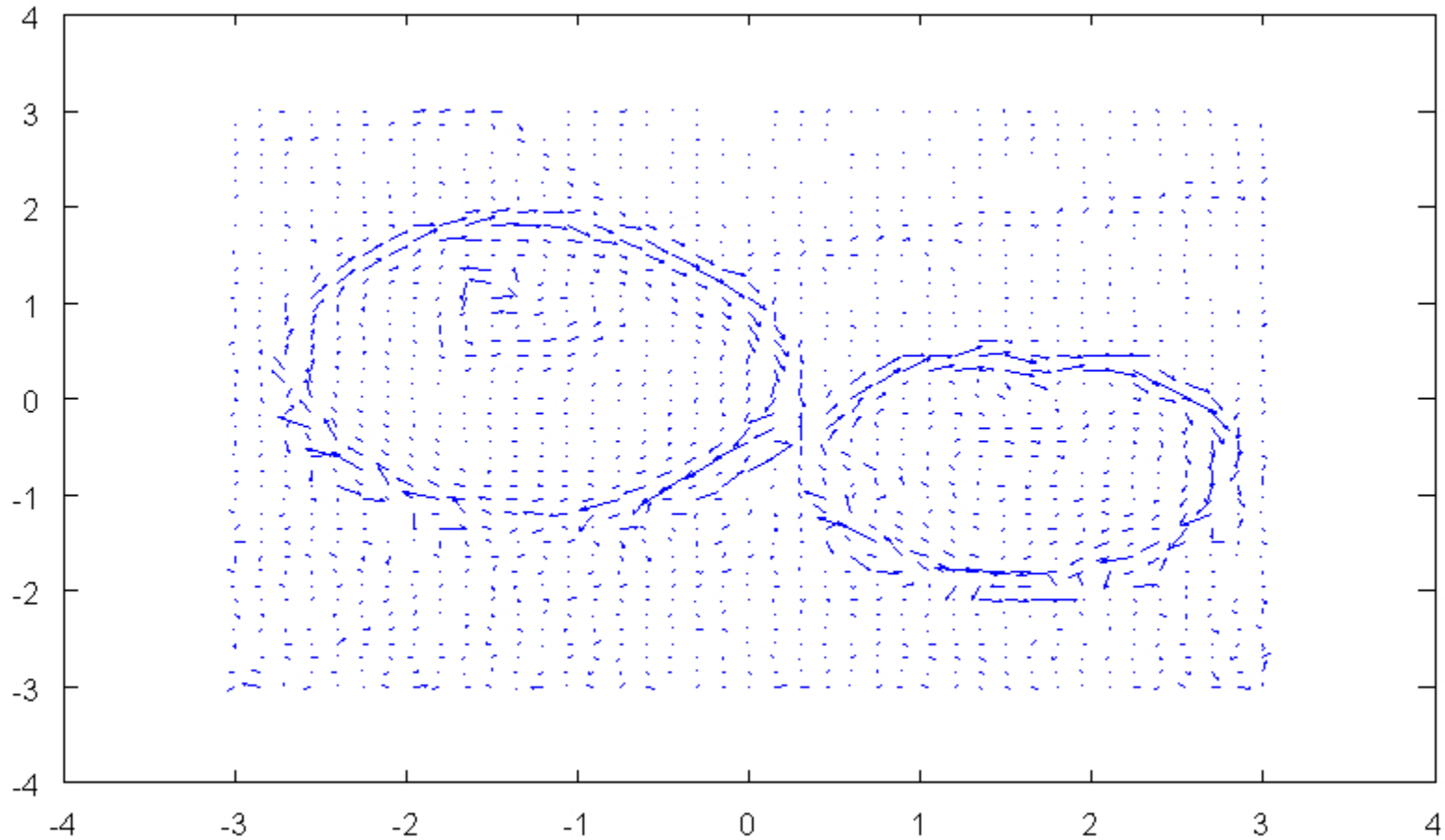


Gradient (strmost')



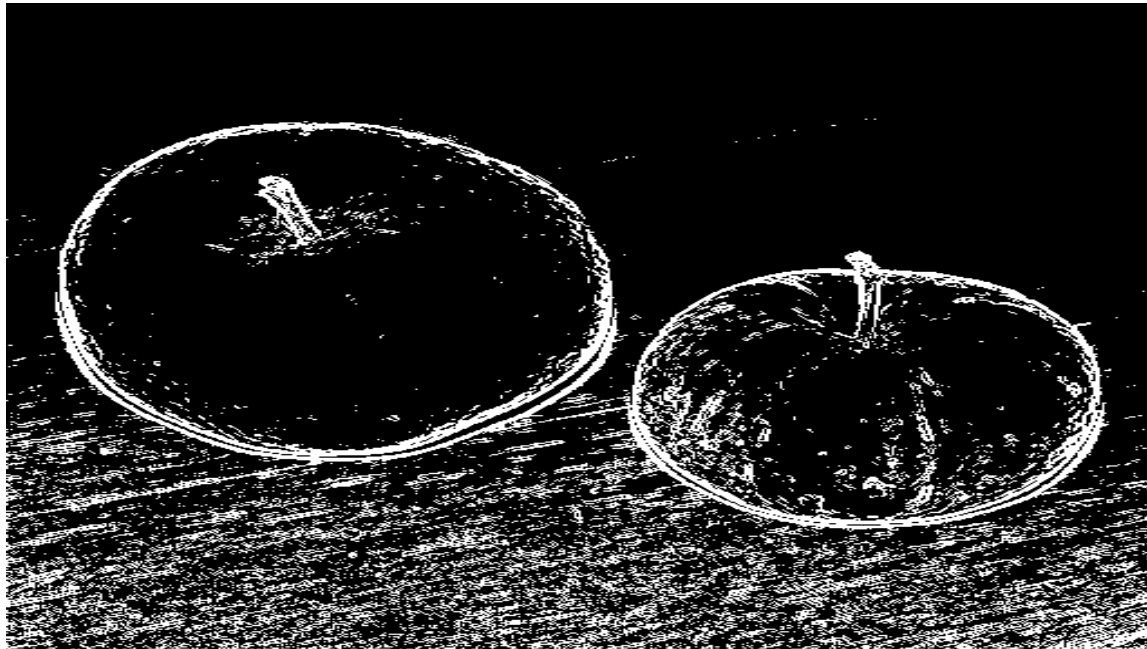
$$\nabla I = \left(\frac{dI}{dx}, \frac{dI}{dy} \right)$$

Isofoty („úbočia“)



$$\nabla^{\perp} I = \left(-\frac{dI}{dy}, \frac{dI}{dx} \right)$$

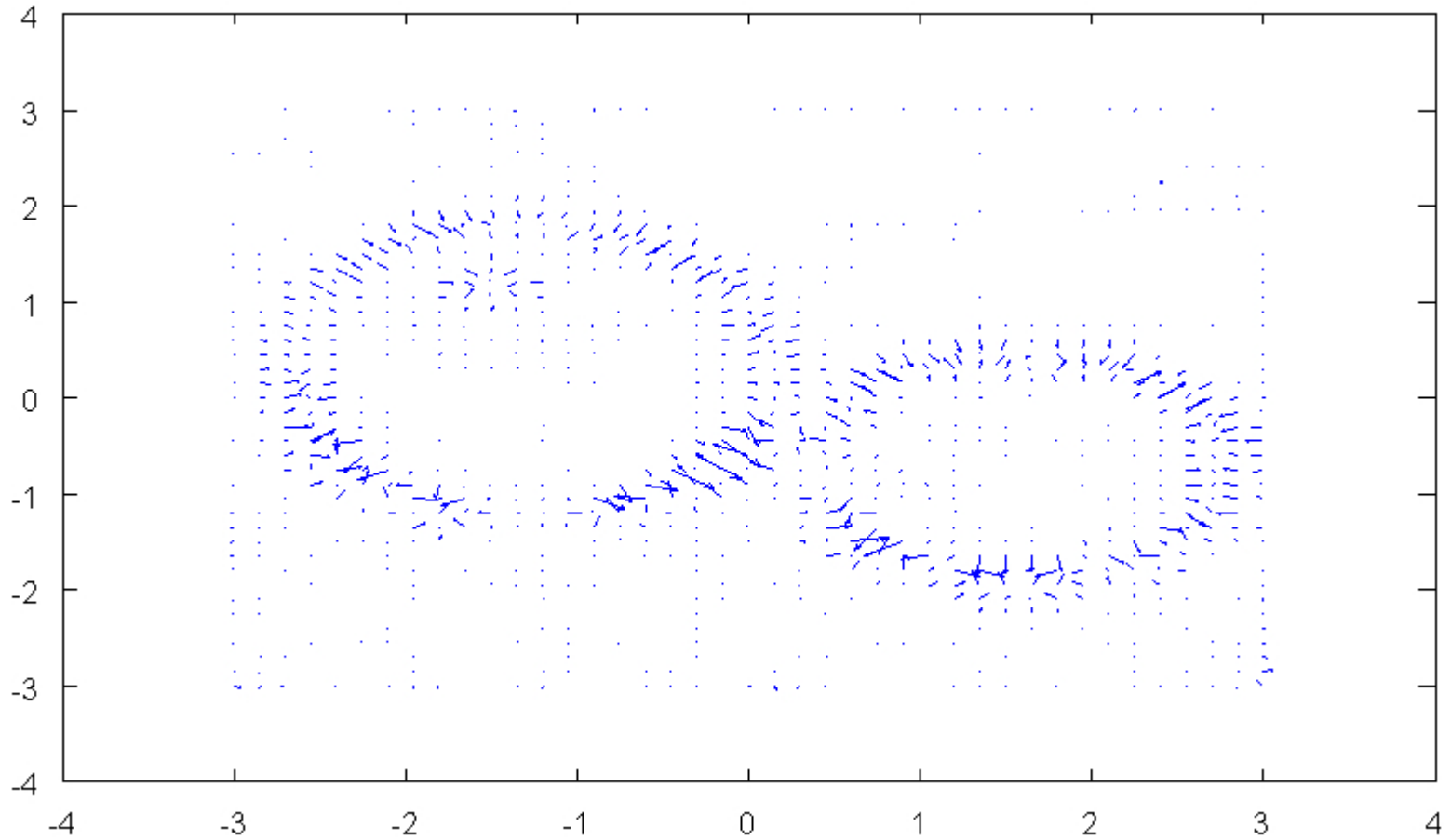
Laplacian (spojitý)



druhá mocnina veľkosti magnitúdy

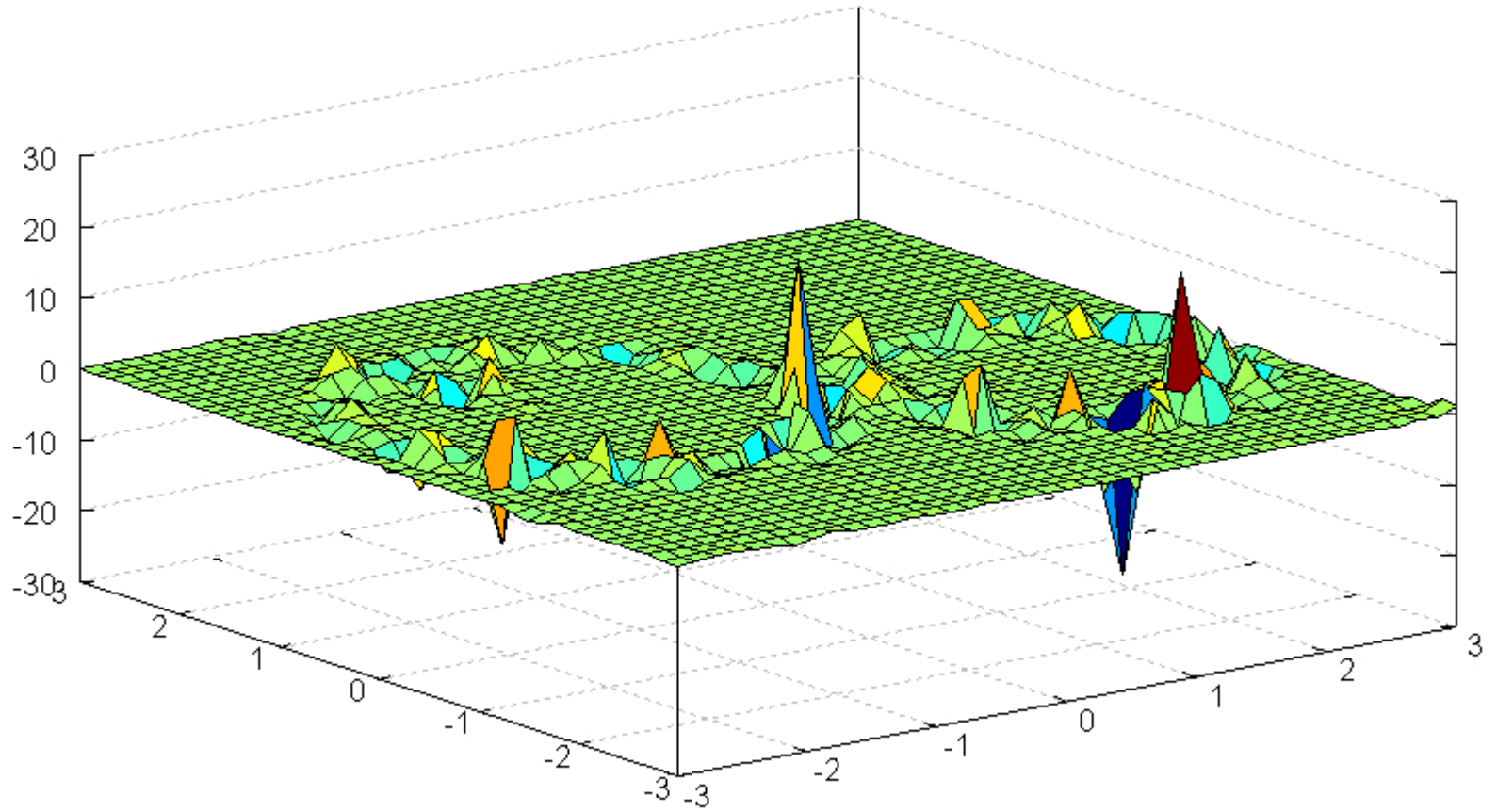
$$\Delta = \nabla \nabla$$

Gradient Laplacianu (GoL)



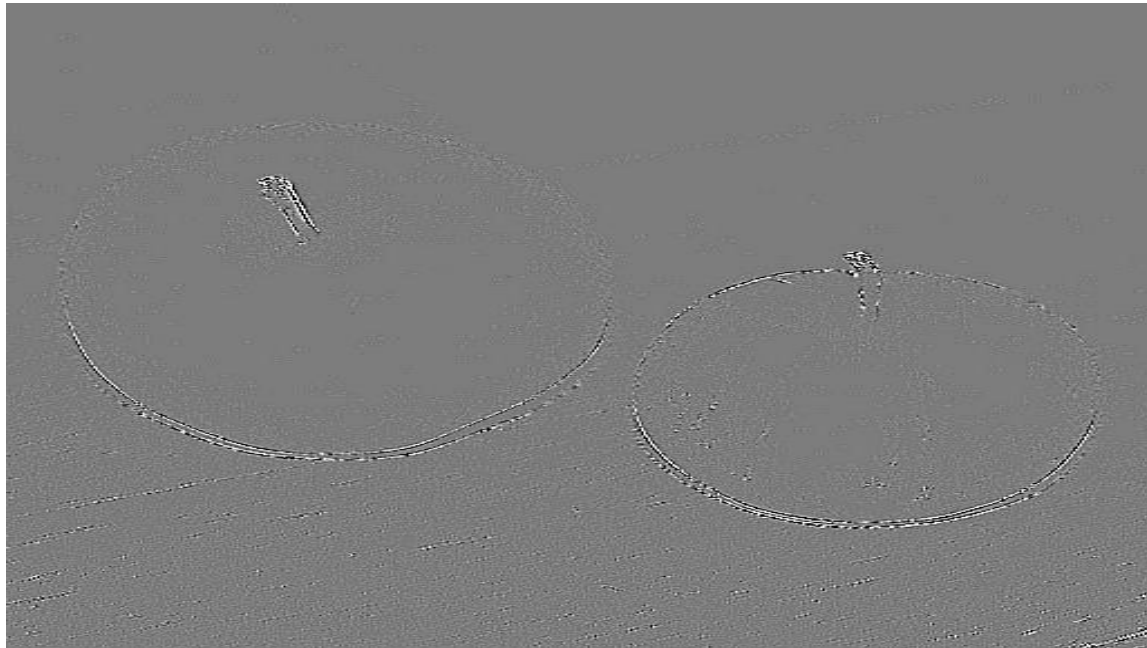
$$\nabla (\Delta I)$$

Hrebene



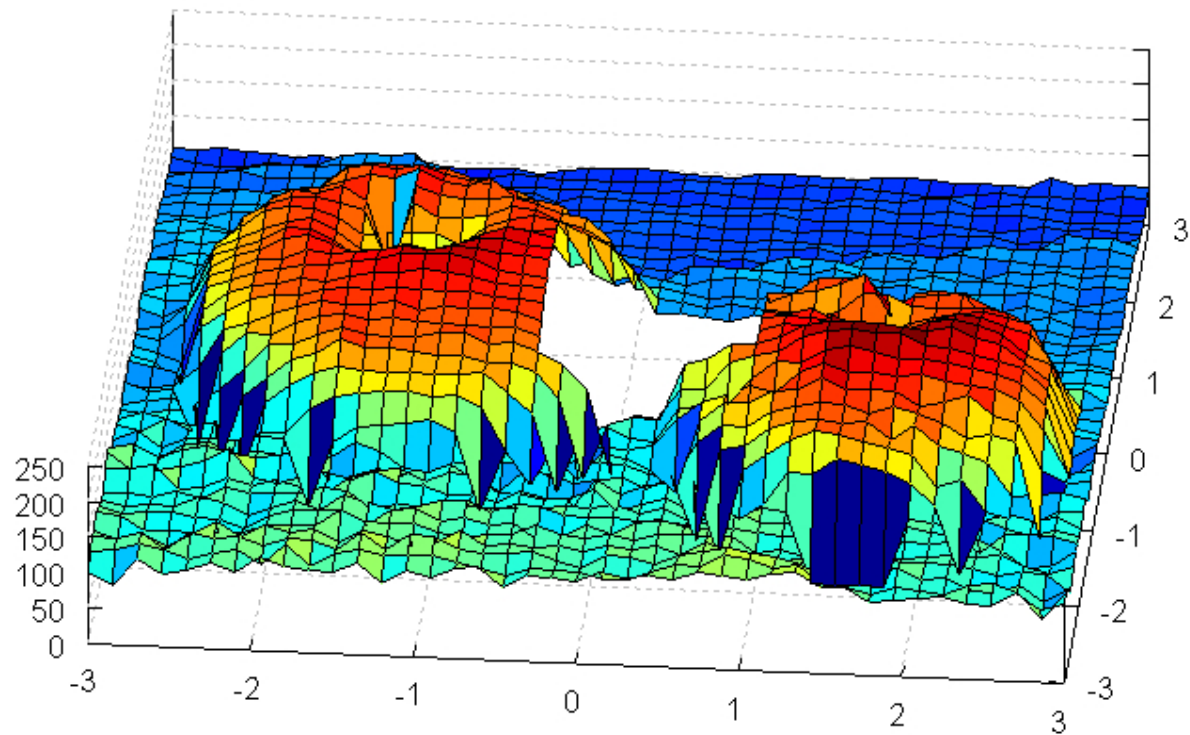
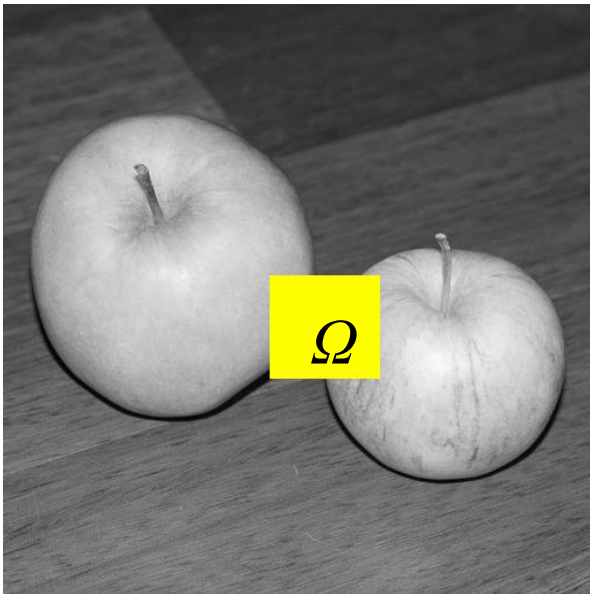
$$\nabla^{\perp} I \cdot \nabla (\Delta I)$$

Hrebene



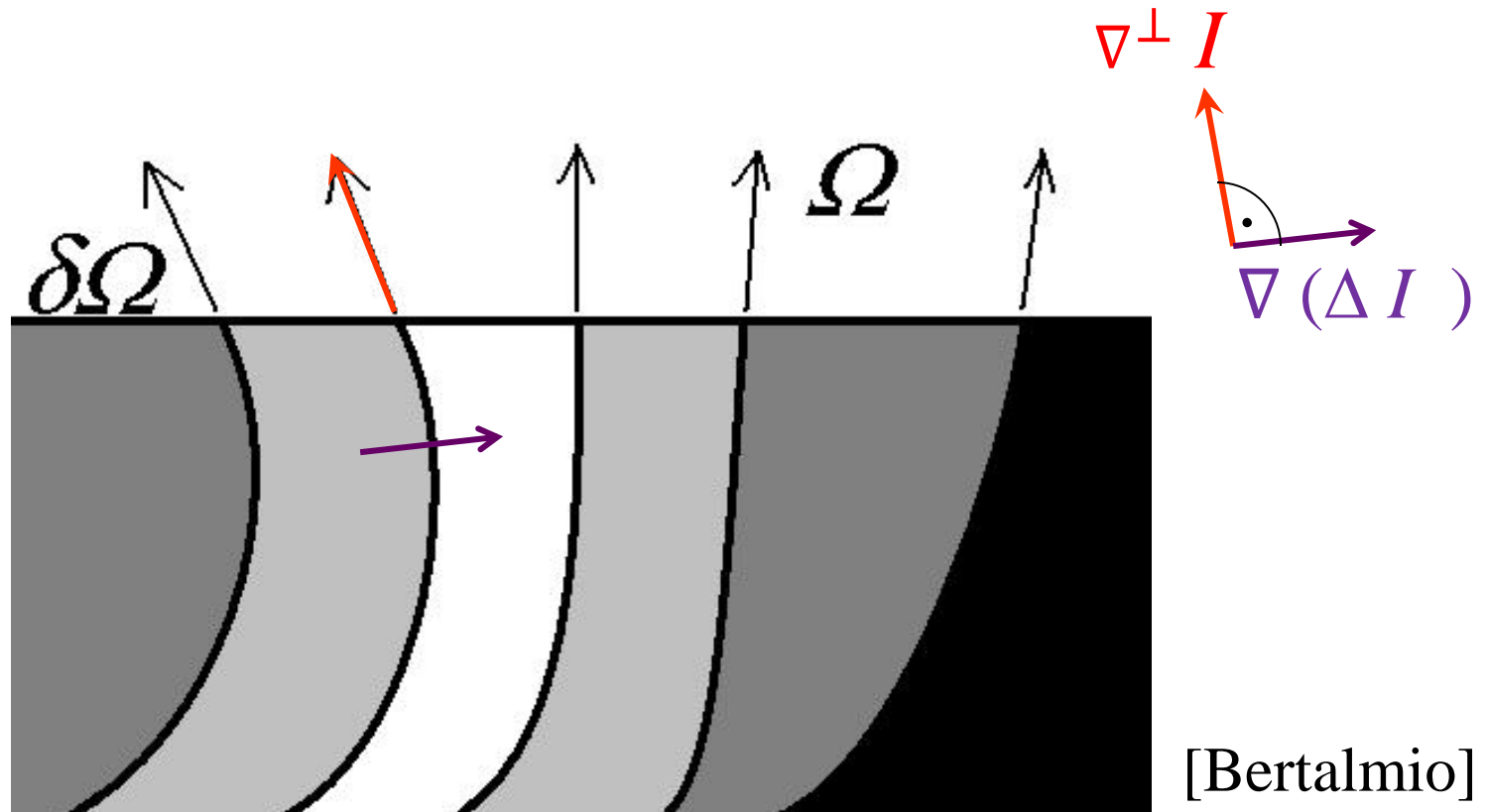
Úloha inpaintingu

- Nahradit' chýbajúce hodnoty funkcie na časti definičného oboru Ω

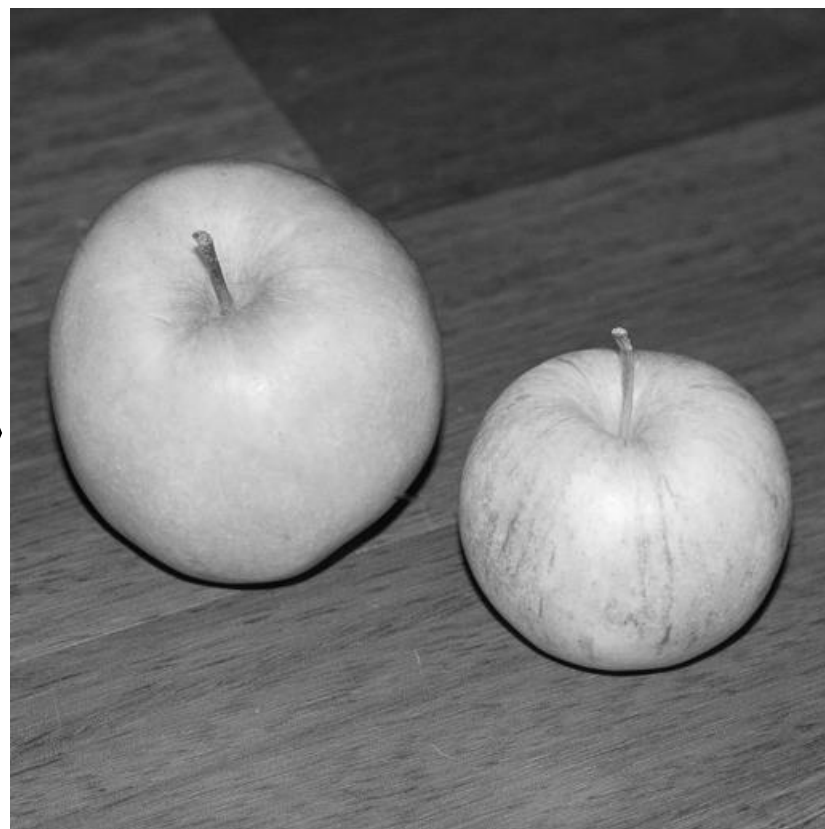
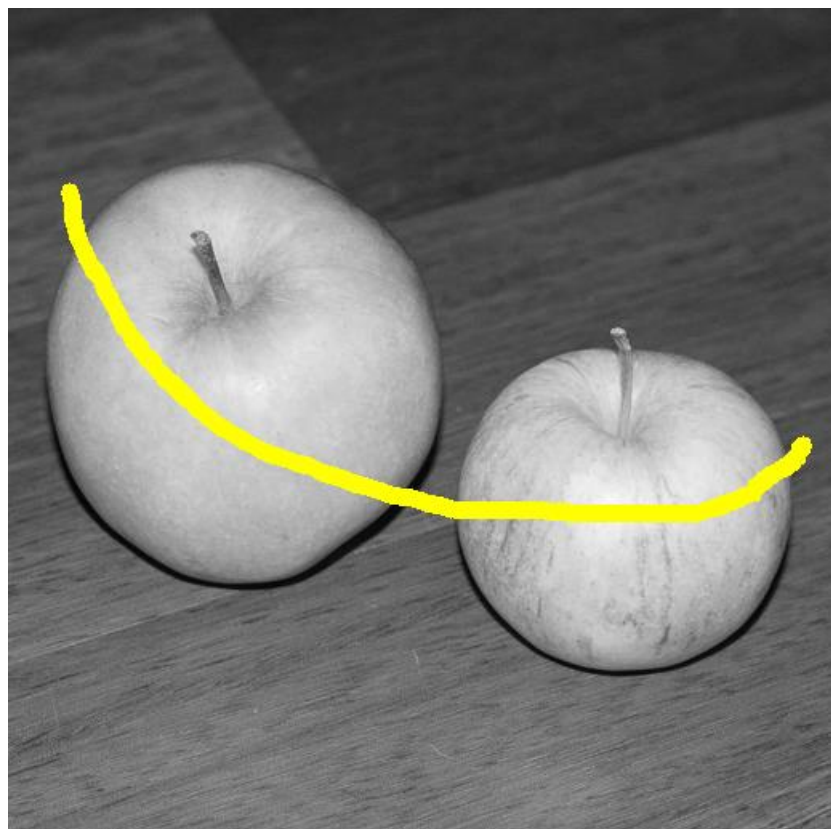


Základná myšlienka inpaintingu

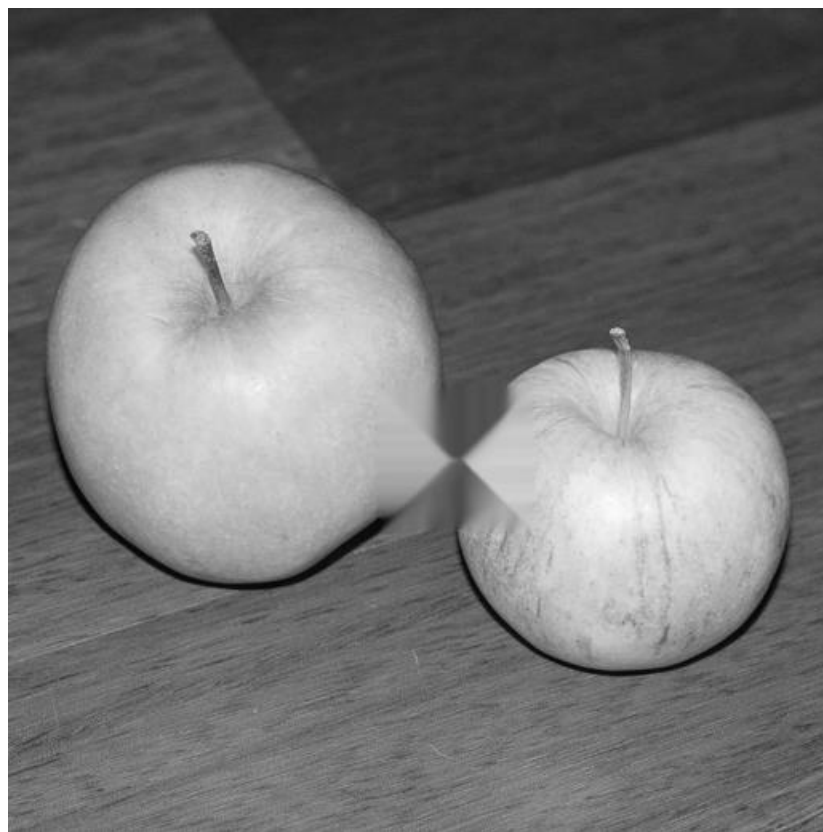
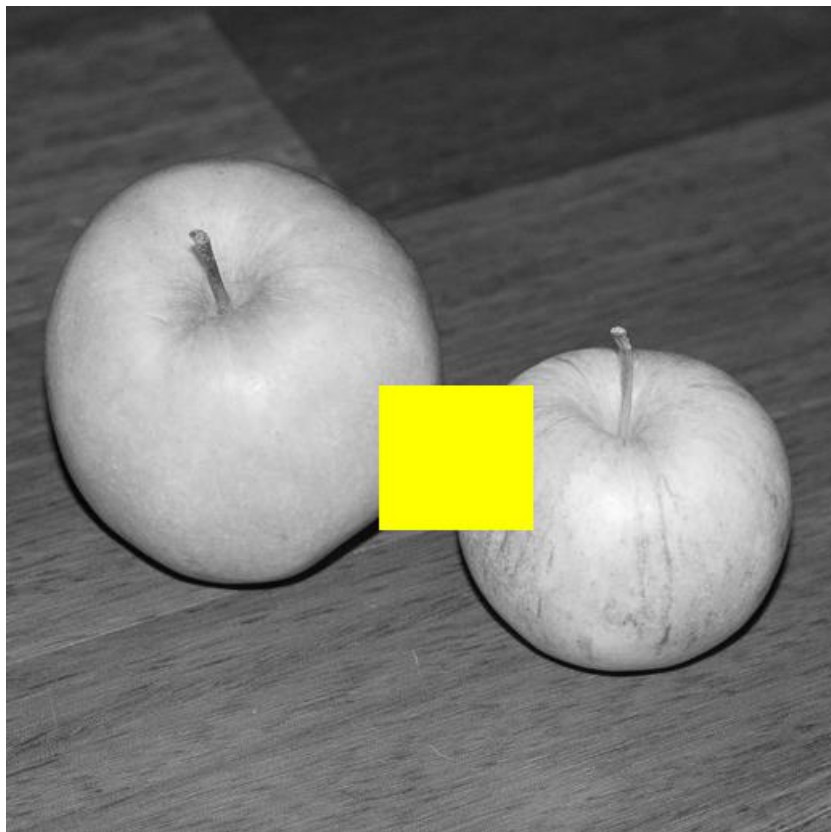
- Chýbajúci kus funkcie nahradíme tak, že isofoty budú kolmé na gradient Laplacianu



Výsledky inpaintingu



Výsledky inpaintingu



Iteratívne metódy

- Naivný algoritmus v mieste stretu postupu rekonštrukcie obrazu vytvára artefakty
- Lepšie výsledky v pomalšom čase poskytujú iteratívne metódy, ktoré artefakty čiastočne eliminujú difúziou (Navier-Stokes)
[Bertalmio - Sapiro]
- Alternatívna metóda: [Telea], výrazne pomalšia a len porovnateľne dobrá