

I12: Metody segmentace obrazů – prahování a shlukování, kombinatorické metody (GraphCut). Detekce významných bodů, Registrace obrazů. (DZO)

SEGMENTACE OBRAZŮ

Segmentace obrazu je proces, který umožňuje rozřadit pixely obrazu do několika segmentů (jinak řečeno několika množin nebo několika tříd). Pixelu je tedy přiřazen label, který odpovídá určitému segmentu. Pixely se stejným labelem sdílejí určité vlastnosti, na základě kterých jsou sdruženy do segmentu.

Typickou úlohou je rozpoznat na obraze objekt zájmu. Objekt jsou tedy všechny pixely obrazu, které jsou pro nás zajímavé. Na obraze může být i více objektů, odpovídajících více segmentům. Zbytek pixelů je pozadí.



Segmentace může být:

- Kompletní - Každému pixelu je přiřazen label. Obraz je tedy rozdělen do konečného počtu regionů, které se nepřekrývají.
- Částečná – Label se přiřadí jen určitým částem obrazu, které jsou důležité pro další analýzu a interpretaci. Typicky se vyberou jen regiony se zajímavou vlastností, skupiny významných hran nebo skupiny rohů apod. Zbytek pixelů se ignoruje.

Objekty (a segmenty, které jim odpovídají) na obraze rozpoznáváme na základě určitých vlastností (features). Využíváme tedy apriorní znalosti, které o objektu zájmu máme. Můžou to být:

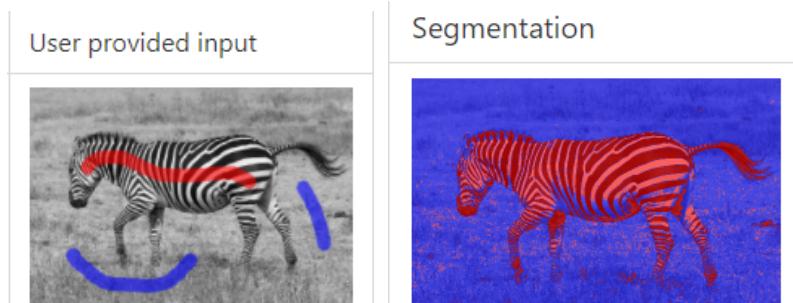
- Vlastnosti získané přímo ze senzoru – intenzita, barva
- Složitější vlastnosti – textura, tvar, předpokládaná pozice objektu na obraze, informace o vztahu objektu k jiným regionům obrazu (typicky prostorové vztahy – nad, pod, vlevo, vpravo, uvnitř, paralelně, kříží se atd.), v případě videa i způsob pohybu objektu

Metody segmentace obrazů

Důležité je, že volba metody segmentace vždy závisí na konkrétní aplikaci a na apriorní znalosti podoby obrazů a objektů. Nelze určit jednu univerzálně nejlepší metodu.

Metody jsou:

- Automatické – Algoritmus pracuje jen s obrazem a nedostává žádnou dodatečnou informaci od uživatele.
- Poloautomatické – Uživatel dodá informaci navíc (typicky určí vzorek pixelů objektu a pozadí – jako v úkolu z DZO, nebo hrubě ohraňčí objekt myší a zúží tím oblast hledání)



Prahování

Pixelům přiřadíme labely na základě globální vlastnosti obrazu. Typicky je to intenzita u obrazů ve stupních šedi. Globální vlastnost je v tomto případě reprezentována histogramem intenzity. Podle zvoleného prahu intenzity klasifikujeme pixely do dvou skupin, které reprezentují objekt a pozadí. Výsledkem je tedy binární obraz (maska) jedniček a nul.

Input image $f(i, j)$, output image $g(i, j)$.

For each pixel (i, j)

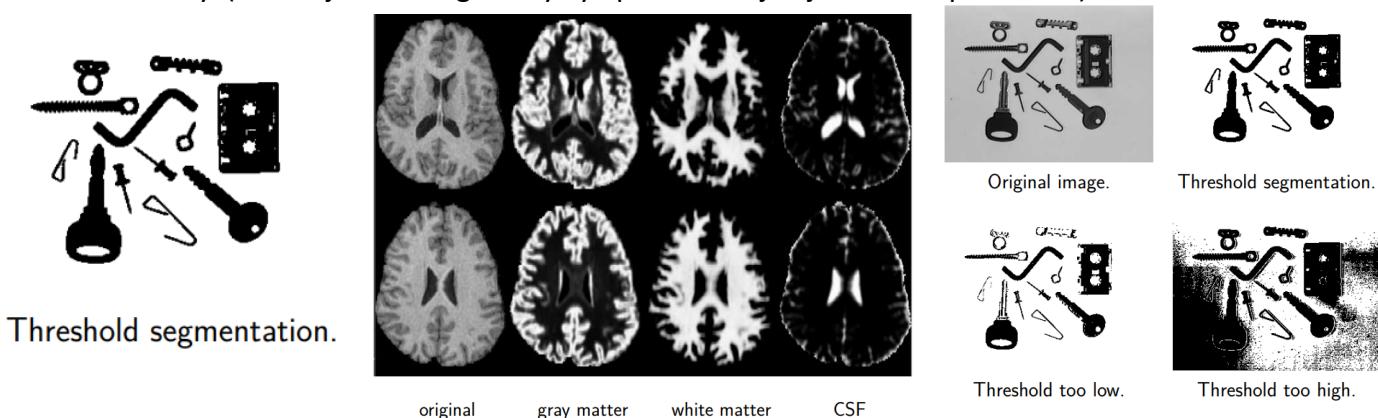
$$g(i, j) = \begin{cases} 1 & \text{for } f(i, j) \geq \text{Threshold}, \\ 0 & \text{for } f(i, j) < \text{Threshold}. \end{cases}$$

Prahování ale lze modifikovat i pro barevné RGB obrazy, potom jde o vícepásmové prahování (multiband thresholding). Práh se určuje v každém barevném kanálu zvlášť a pixel musí splnit 3 kritéria zároveň, aby byl klasifikován jako objekt. Dají se použít i jiné barevné modely než RGB.

V praxi je možné prahování využít pouze pro segmentaci obrazů, kde mají objekty jasně rozdílnou intenzitu než pozadí. V případě segmentace více objektů je třeba, aby měl každý druh objektu jinou intenzitu. Většinou je třeba i zařídit vhodné podmínky, aby intenzita pixelů nebyla ovlivněná např. nerovnoměrným osvětlením, vrženými stíny atd..

Využití:

- v průmyslu při kontrole součástek – při vhodném nasvícení (typicky back-light illumination, které z objektu udělá siluetu) lze velmi přesně kontrolovat defekty tvaru a hran součástky (např. s dalším zpracováním pomocí interpolace).
- Rozpoznání znaků od pozadí na naskenovaném obrazu textu
- v medicínských obrazech – např. základní algoritmus pro segmentaci šedé hmoty, bílé hmoty, mozkomíšní tekutiny a pozadí ve snímcích magnetické rezonance mozku. Využívá se podobně i pro CT snímky. (reálně jsou ale algoritmy vylepšené a nejde jen o čisté prahování)



+ Snadná implementace, rychlý výpočet, pro vhodné úkoly i dobrá přesnost

- Dobře funguje jen pro specifické případy úloh segmentace, problém s určováním prahu (natvrdo zadáný se často nedá použít -> je třeba vhodně zvolená automatická detekce prahu)

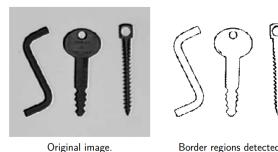
Určování prahu:

- Napevno předem – pokud máme zaručené stálé podmínky, intenzity objektů a pozadí jsou známé. (součástky při stálém nasvícení).

Modifikace pevného prahu:

- o Lokálně adaptivní práh – obraz se rozdělí na podobrazy a práh se určí pro každý podobraz zvlášť
- o Pásmovej práh – pixel je klasifikován jako objekt, pokud má intenzitu v daném pásmu intenzit

$$g(i, j) = \begin{cases} 1 & \text{for } f(i, j) \in D, \\ 0 & \text{otherwise.} \end{cases}$$

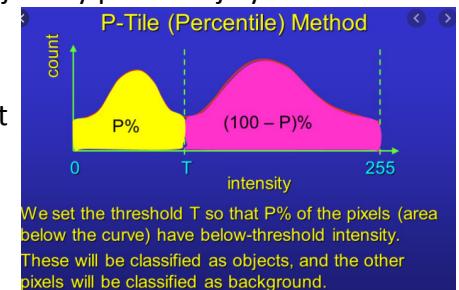


- o Více prahů pro více druhů objektů – multiple thresholds
- o Semi thresholding – vhodné jako předzpracování pro lidského pozorovatele. Práh určí pouze kde je pozadí, ale v popředí jsou ponechány původní hodnoty intenzity.

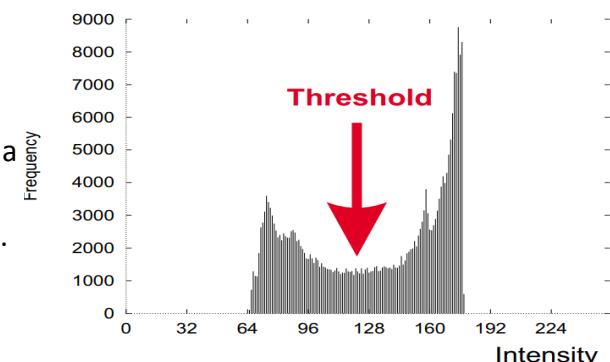
$$g(i, j) = \begin{cases} f(i, j) & \text{for } f(i, j) \geq \text{Threshold}, \\ 0 & \text{for } f(i, j) < \text{Threshold}. \end{cases}$$

- Automatická detekce prahu – až z konkrétního obrazu se určí práh, je tedy pokaždé jiný

- o P – tile thresholding – Použitelné, pokud předem víme, jaký podíl plochy obrazu zaujímá objekt a víme, jestli je objekt tmavší, nebo světlejší než pozadí. Např. víme, že tmavý text bude zaujímat $p=20\%$ stránky a podle toho zvolíme z histogramu práh (textu přisoudíme intenzity, které v histogramu odpovídají 20% nejtmavších pixelů).



- o Na základě tvaru histogramu – Histogram se typicky nejdříve vyhladí, aby nebyl „zubatý“ a dobře se určovaly jeho lokální extrémy. Poté se analyzuje jeho tvar. Určující jsou lokální maxima a minima. Typicky jde o úlohu, kde rozlišujeme objekt a pozadí. Histogram je tedy bimodální tzn. má 2 jasné vrcholy a mezi nimi „údolí“. Vrcholy korespondují s nejčastější intenzitou objektu a pozadí, údolí tedy určuje vhodný práh mezi nimi.



- o Na základě proložení pravděpodobnostními rozloženými – Nejpoužívanější je proložení N gaussovskými rozloženými (mixture of gaussians). Jde vlastně o optimalizační problém – snažíme se N gausiánů co nejlépe proložit histogramem pomocí minimalizace rozdílového kritéria. Hledáme tedy parametry gausiánů, pro které bude rozdíl mezi histogramem a rozloženými minimální. Na základě modelu se pak automaticky určí prahy pro přiřazení labelů pixelům. Pixel dostane label, který odpovídá gausiánu, který je pro intenzitu pixelu nejpravděpodobnější. Tato metoda už je blízko klastrování (EM algoritmus), ale není to totéž, EM pracuje s výpočty likelihood, tato metoda ne.

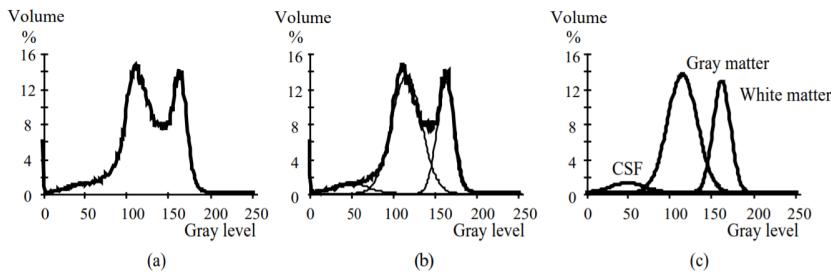
- ◆ h_{region} – local histogram.
- ◆ h_{model} – approximation of a histogram by n Gaussian distributions,

$$h_{\text{model}}(g) = \sum_{i=1}^n a_i e^{-\frac{(g-\mu_i)^2}{2\sigma_i^2}}.$$

- ◆ The minimized optimization criterion F ,

$$F = \sum_{g \in G} \left(h_{\text{model}}(g) - h_{\text{region}}(g) \right)^2.$$

- ◆ Desired classes: white matter, grey matter, celebro-spinal fluid (CSF)



Klastrování

Pixelům se přiřadí labely na základě příslušnosti ke klastrům. Pixely se do klastrů rozřazují podle nějaké vlastnosti. Typicky je to intenzita u šedotónových, nebo barva u barevných obrazů. Pro barevné RGB obrazy jde tedy o klastrování ve 3D prostoru, pro šedotónové v 1D prostoru.

Vlastnosti, podle které klastrování může být například i 1. parciální derivace, která vystihuje změny intenzity – slouží k klastrování na základě textury. Používají se i další složitější vlastnosti, které vystihují textury obrazu nebo gradienty v obrazu.



Image, courtesy Ondřej Drbohlav

- ◆ Feature used for clustering – the absolute value of 1st partial derivatives,

$$\left(\left| \frac{\partial I(x, y)}{\partial x} \right|, \left| \frac{\partial I(x, y)}{\partial y} \right| \right)$$

Klastrování lze provést různými metodami ([K-means](#), [EM](#), [mean shift](#)). Společnou vlastností K-means a EM je to, že musíme předem určit počet klastrů. Například pokud chceme v obraze rozlišit červenou a žlutou kytku od zeleného pozadí, chceme rozdělení do 3 klastrů podle barvy. V některých případech je problematické najít vhodný počet klastrů, protože neznáme počet objektů, které chceme segmentovat. Například bychom nevěděli, jestli se na obraze neobjeví ještě modrá a bílá kytka – 3 klastry by v tomto případě nejspíš dávali nesmyslný výsledek.

Obecný princip K-Means a EM je podrobně popsán v otázce I13. Tady odkaz na Hlaváčovu prezentaci o nich <http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/31PattRecog/52UnsupervLearning.pdf>

K-Means

Na principu minimalizace vzdálenosti bodů (které reprezentují pixely) od centroidů. Zhruba se dá říct, že v případě s kytkama by každý centroid reprezentoval typickou červenou a žlutou barvu kytek a zelenou barvu pozadí. Ostatní pixely by se přiřadily centroidům podle toho, jak je jejich barva blízká barvě centroidů.

Lloyd's Algorithm

- ◆ Initialize the pixels to belong to a random region.
- ◆ Compute the mean feature vector in each region.
- ◆ Move a pixel to another region if this decreases the error function (= total distance) J .

$$J = \sum_{n=1}^N \sum_{k=1}^K \|x_n - \mu_k\|^2,$$

where n points to individual pixels, N is number of pixels, K is an a priori given number of clusters, $K < N$, x_n is a pixel value, μ_k is the cluster representative (mean point of a cluster).



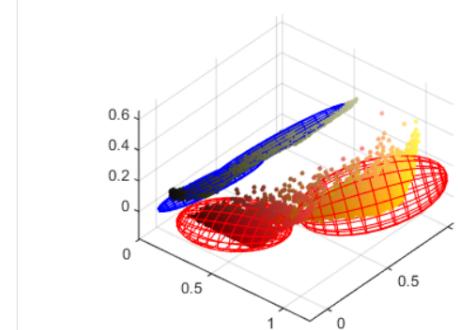
- ◆ Iterate until pixels do not move any longer.

+ snadná implementace dostupná jako funkce ve většině běžných softwarů (matlab, python...)

- konvergence k lokálnímu minimu, nutnost určení počtu klastrů, závislé na počáteční poloze klastrů, preferuje sférické klastry, protože pracuje na principu vzdálenosti (nevystihne „protáhlá“ rozdělení), citlivé na outliery (pracuje s těžištěm, což je průměr a ten je náchylný na outliery)

EM algorithm

EM pracuje na principu iterativního výpočtu likelihood proložení dat pravděpodobnostními rozloženími (v segmentaci většinou gaussiány). Snaží se co nejlépe approximovat MLE proložení. V příkladu s kytkami bychom došli k něčemu jako na obrázku níž (v úkolu ale nebyl použitý EM, ale jiný postup). Výsledkem EM je mixture od gaussians (případně jiného typu rozdělení). Pixelu se tedy přiřadí N čísel, které reprezentují pravděpodobnosti, že pixel náleží daným rozdělením. Výsledný label pixelu pak typicky odpovídá rozdělení s nejvyšší pravděpodobností. Přiřazení labelů se ale může modifikovat a zahrnout více kritérií.



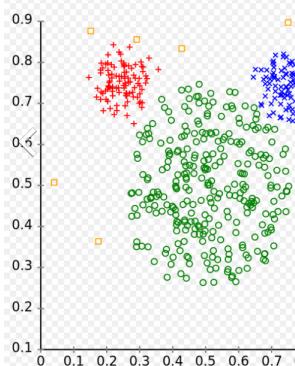
The foreground (red) and background (blue) sample pixels are used to estimate PDFs of foreground and background pixels modeled by GMMs. Individual components of GMM are formed by multivariate normal distributions. Both foreground and background GMMs are formed by two normal distributions in this case. The normal distributions are plotted as ellipsoids. The ellipsoid center corresponds to the mean vector and its shape to the covariance matrix of the respective normal distribution.

-opět nutné určení počtu klastrů, obecně konverguje pouze k lokálnímu maximu likelihood

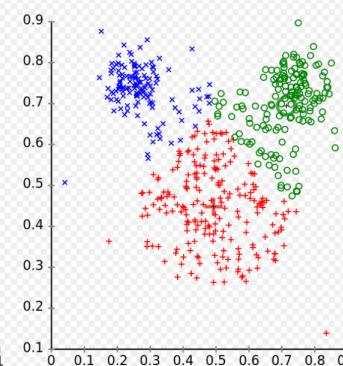
+ odstraňuje některé nedostatky k-means – nepreferuje sférické klastry (tvar klastru určený elipsoidem kovarianční maticí); pixelu není hned natvrdo přiřazen label segmentu, ale pouze pravděpodobnost, že náleží daným rozdělením -> možnost modifikovat přiřazování labelů.

Different cluster analysis results on "mouse" data set:

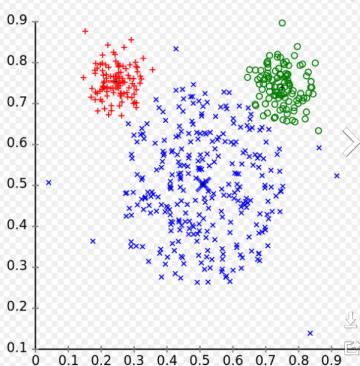
Original Data



k-Means Clustering



EM Clustering



Mean shift

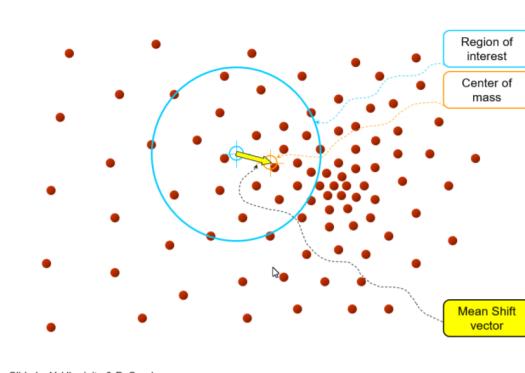
Spíš navíc pro zajímavost, výhoda je, že není třeba předem přesně znát počet klastrů, když tak tady
<http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/15ImageAnalysis/32-04SegmentationSpatialCoherence.pdf>

Mean shift algorithm (1)



29/4

- ◆ Input: points in the Euclidean (feature) space.
- ◆ Determine a search window size (usually small).
- ◆ Choose the initial location of the search window.
- ◆ Compute the mean location (centroid of the data) in the search window.
- ◆ Center the search window at the mean location computed in the previous step.
- ◆ Repeat until convergence.

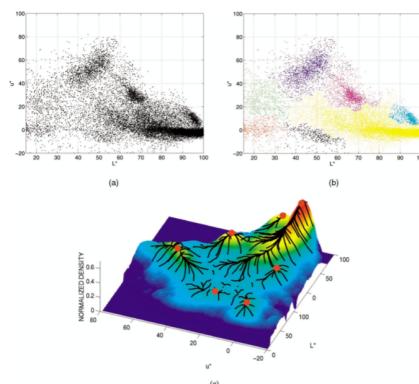


Mean shift image segmentation algorithm



31

1. Convert the image into tokens (via color, gradients, texture measures etc.).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same 'peak' or mode.
5. The data these merged windows traversed are clustered together.



Využití klastrování: Vhodné pro segmentaci objektů, které mají rozdílné vlastnosti než pozadí nebo jiné objekty. Také pro rozdělení obrazu na podobné regiony, které se dále interpretují





+ Široké využití na segmentaci regionů s odlišnými vlastnostmi (podle barvy, intenzity, gradientů..)

- Klastrování nezohledňuje prostorové souvislosti mezi pixely -> výsledné segmenty nejsou souvislé, bývají „rozdrobené“ a obsahují osamocené pixely

Graph cut

Hlaváč tomu moc času nedal, nevím jak do hloubky je to třeba vědět. Asi je zbytečný sem dávat, jak se formálně matematicky grafy popisují, na to by se snad u státnic ptát neměli. Popíšu teda s obrázkama postup Graph cut, který jsme dělali v úloze na DZO, snad to bude stačit. Existuje spousta modifikací a obecně ten termín Graph cut není úplně určitě vymezený.

Existuje mnoho algoritmů, které využívají převedení obrazu do grafové struktury k segmentaci obrazu, hledání hran v obrazech a obrysů objektů, nebo ořezávání obrazu. Graph cut je název pro jednu ze skupin algoritmů, které pro segmentaci používají grafy. Jinak se Graph cut algoritmem také říká Maximal flow, nebo Minimum cut algorithms, je to tentýž princip. Graph cut je vhodný na segmentaci objektu od pozadí.

Jiné typy grafových algoritmů na segmentaci jsou např. Minimal spanning tree/Kruskal alg (segmentace obrazů), Dijkstras alg/shortest path alg (inteligentní nůžky-detekují obrysy objektů).

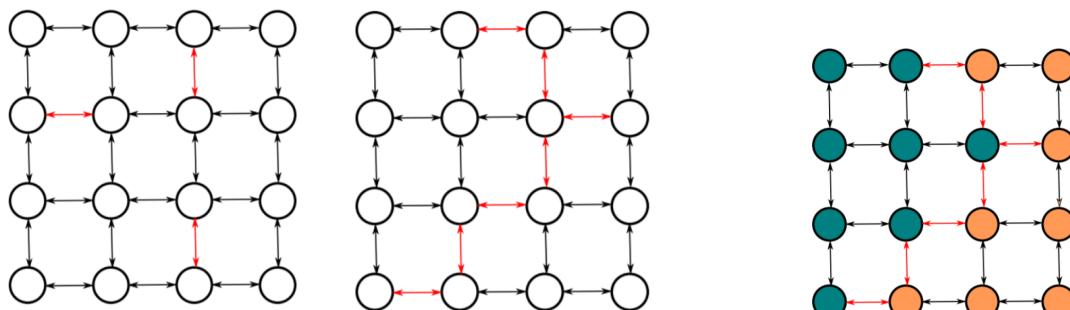
<http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/15ImageAnalysis/32-08SegmentationGraphBased.pdf>

Graf je struktura, která se skládá z uzlů a hran (vertices and edges). Základním principem je převedení informací z obrazu do grafové struktury. Na začátku tedy máme jeden propojený graf, který reprezentuje celý obraz. Uzly v grafu reprezentují jednotlivé pixely a hrany reprezentují vztahy mezi pixely. V některých algoritmech se do grafu přidají další uzly a hrany, které představují další kritéria pro segmentaci.

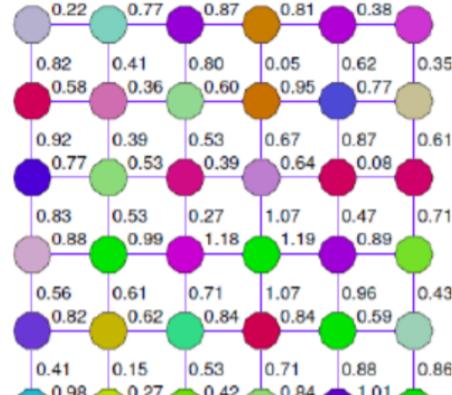
Cílem graph cut algoritmu je podle nějakého pravidla rozdělit původní graf na 2 podgrafy, které reprezentují segmenty objektu a pozadí obrazu (dá se představit jako rozseknutí některých hran). Formálně je tedy graph cut množina hran, které když se z grafu odstraní, rozdělí graf na 2 nezávislé části. Jiná formulace stejného postupu je, že uzlům přiřadíme labely segmentů a tím graf rozdělíme na podgrafy.

- Do not form a cut
- Form a cut

Relation between labelings and cuts



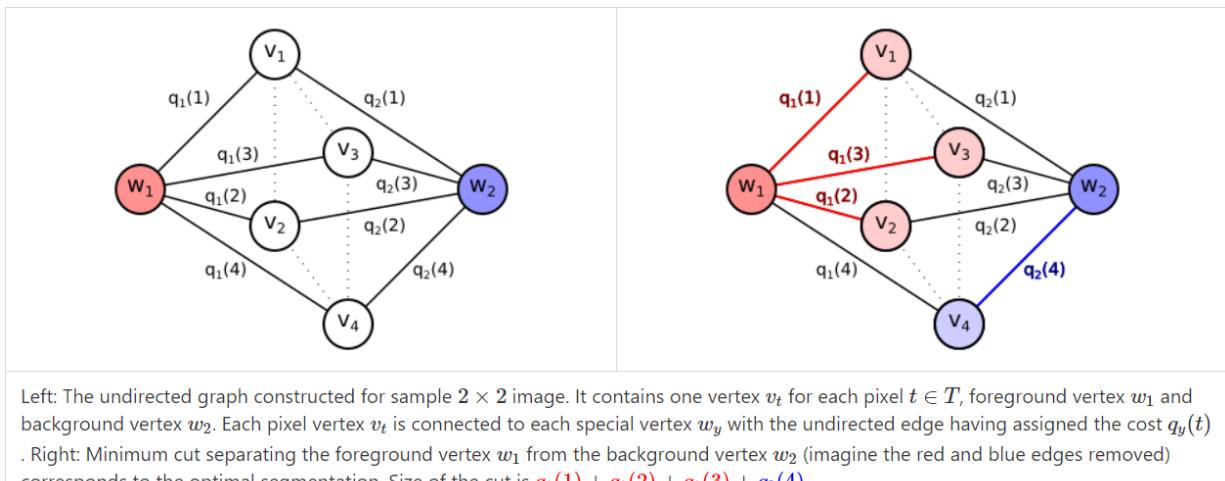
- ◆ Nodes correspond to pixels.
- ◆ Edges connect neighboring pixels. 4-neighbors are considered in the example.
- ◆ Edges weights express the similarity between the neighboring pixels (binary relation).



graph with weighted edges

Postup Graph cut z DZO úkolu:

- Uživatel označí myší vzorek objektu a pozadí
- Nejdříve se provede proložení označených pixelů mixture of gaussians a tím získáme pravděpodobnosti, že dané pixely náleží buď popředí, nebo pozadí. (viz obrázek s kytkama)
- Potom se uzlům, které reprezentují pixely přiřadí ceny podle těchto pravděpodobností. To se provede tak, že se zavedou dodatečné 2 uzly, které reprezentují objekt a pozadí. Každý z pixelových uzelů je s nimi pak 2 hranami spojen. Hraně mezi pixelem a objektem/pozadím se přiřadí cena úměrná pravděpodobnosti, že pixel náleží do objektu/pozadí. V úkolu cena odpovídá $-\log(p)$, maximální pravděpodobnosti tedy odpovídá minimální cena q.



- Následně se přiřadí ceny i hranám mezi jednotlivými pixely. Uvažuje se, že hrany spojují pixely z okolí (4 nebo 8-okolí). Cena hran mezi pixely v sobě zahrnuje 2 kritéria – prostorová vzdálenost pixelů v obraze (ty diagonálně v 8 okolí jsou dál), eukleidovská vzdálenost pixelů v RGB barevném prostoru. Výsledek je, že mezi pixely, které pravděpodobně náleží do stejného segmentu bude vysoká cena hrany. Naopak mezi pixely, které spolu nemají společné vlastnosti bude cena nízká a bude vyšší pravděpodobnost, že tudy povede graphcut.

Formally, for each $\{t, s\} \in P$ we compute the following cost:

$$r(t, s) = \lambda_1 + \frac{\lambda_2}{d(t, s)} \exp\left(-\frac{\|x_t - x_s\|^2}{2\beta}\right)$$

The parameter λ_1 scales the constant part of the weight $r(t, s)$. The constant part is usually called Ising prior. The parameter λ_2 scales the part dependent on t and s . The recommended values are $\lambda_1 = 5$ and $\lambda_2 = 45$. Using higher values of λ_1 and λ_2 encourages smoother segmentation. Lower values prefer pixel-wise segmentation based on the estimated foreground and background color model.

The term $d(t, s)$ denotes Euclidean distance of pixels in the image grid. It is equal to 1 for horizontally and vertically neighboring pixels. It is equal to $\sqrt{2}$ for diagonal neighbors.

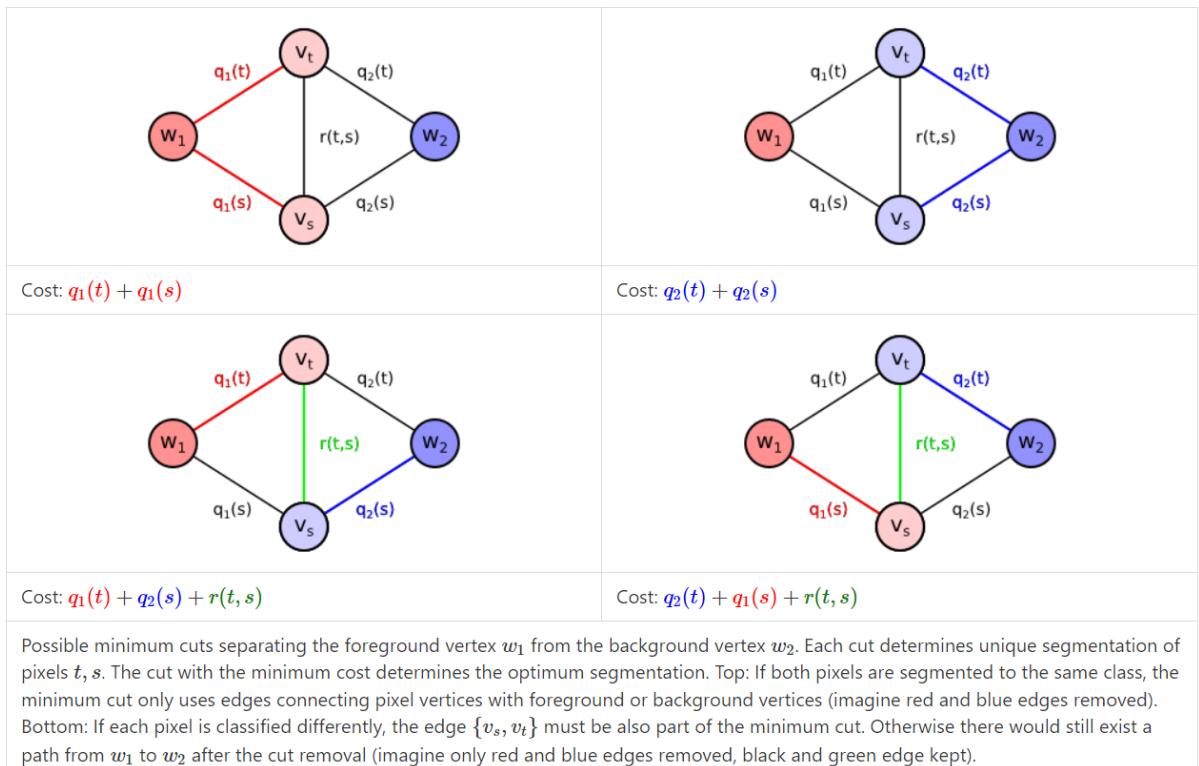
The term $\|x_t - x_s\|^2$ denotes squared Euclidean distance of pixel colors. The pixel color $x_t \in [0, 1]^3$ is 3D vector of RGB components.

$$\|x_t - x_s\|^2 = (x_t^R - x_s^R)^2 + (x_t^G - x_s^G)^2 + (x_t^B - x_s^B)^2$$

The term β normalizes Euclidean distance of pixel colors. It is equal to the expected squared Euclidean distance of pixel colors over all image, i.e. over each pair of neighbors $\{t', s'\} \in P$:

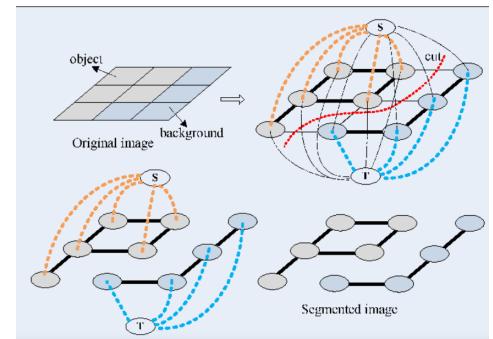
$$\beta = \frac{1}{|P|} \sum_{\{t', s'\} \in P} \|x_{t'} - x_{s'}\|^2$$

- Poté se hledá taková cesta grafem, která oddělí uzel objektu od uzlu pozadí a zároveň bude mít co nejnižší cenu. (představte si že černé hrany na obrázku zmizí a sečtou se ceny těch barevných). *Přímo to hledání cesty jsme měli naimplementované a nevím, jak to fungovalo na pozadí, neřešili jsme to.*



+Oproti klastrování a prahování zohledňuje i prostorové vztahy mezi pixely, to vede k hladké segmentaci bez „děr“ a osamělých pixelů. Umožňuje využít mnoha různých variant cost function na přřazování cen uzlům a hranám -> flexibilní.

– Pouze pro odlišení objektu a pozadí -> v této podobě nelze použít pro segmentaci na více objektů.



DETEKCE VÝZNAMNÝCH BODŮ

Tady trochu tápu, vzhledem k tomu, že to Hlaváč zapomněl odpřednášet :D

Slouží většinou jako mezikrok před dalším zpracováním. Při jejich hledání se nezohledňuje sémantika obrazu. Významné body mohou být použity pro následnou registraci obrazů, rekonstrukci 3D scény z 2D obrazů, sledování pohybu objektu na videu, rozpoznání objektů na obrazech, navigaci robotů a další.

Významné body jsou rohy, nebo body na významných oblinách/kulatých rozích (blobs) objektů.

Významné body by měly splňovat následující vlastnosti:

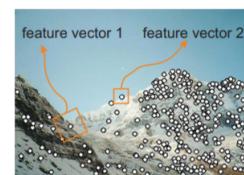
- Jejich počet je významě menší než počet pixelů obrazu
- Významné body a jejich deskriptory jsou invariantní vůči malým affinním transformacím obrazu
- Malým radiometrickým změnám typu změna osvětlení scény, rozmažání, šum
- Mírná změna vzorkování a kvantizace obrazu

(např. 4 rohy stolu by se měly detektovat i pokud trochu změníme úhel, pod kterým stůl vyfotíme, trochu u toho pohneme foťákem a rozmaže se to, nad stolem rozsvítíme lustr, takže se změní intenzity na obraze, použijeme trochu nižší rozlišení fotky)

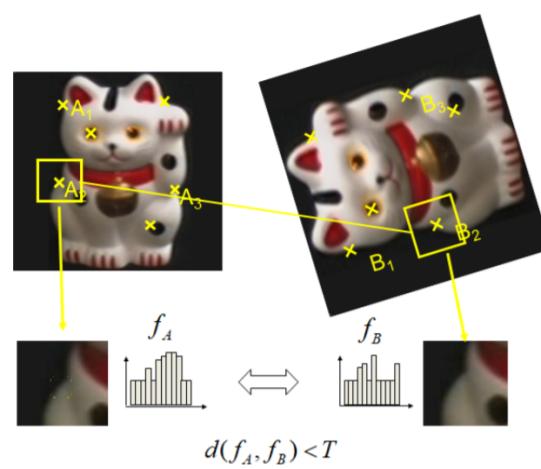
Obecně má detekce významných bodů 3 hlavní kroky:

- Samotné vyhledání významného bodu v obrazu některým z detektorů (viz níž)
- Popis jeho okolí tak, aby se dalo následně porovnávat s jinými místy obrazu a vyhodnocovat podobnost – tzv. vytváření lokálního deskriptoru (jinak řečeno feature vectoru) významného bodu.
- Vyhodnocování podobnosti a hledání korespondujících bodů v jiném obrazu

1. **Detection** identifies the interest points (also keypoints) or regions.
2. **Description** calculates descriptors (feature vectors) from the local neighborhood of each interest point or region.
3. **Matching** compares feature description pairwise across views, ranks them, and selects the most prominent one (or a few most prominent ones to increase robustness).



1. Find a set of distinctive interest points.
2. Define a region around each interest point.
3. Extract and normalize the region content.
4. Compute a local descriptor from the normalized region.
5. Match local descriptors.



Detektory významných bodů

Tady fakt moc nevím, snad bude stačit znát Harris corners.

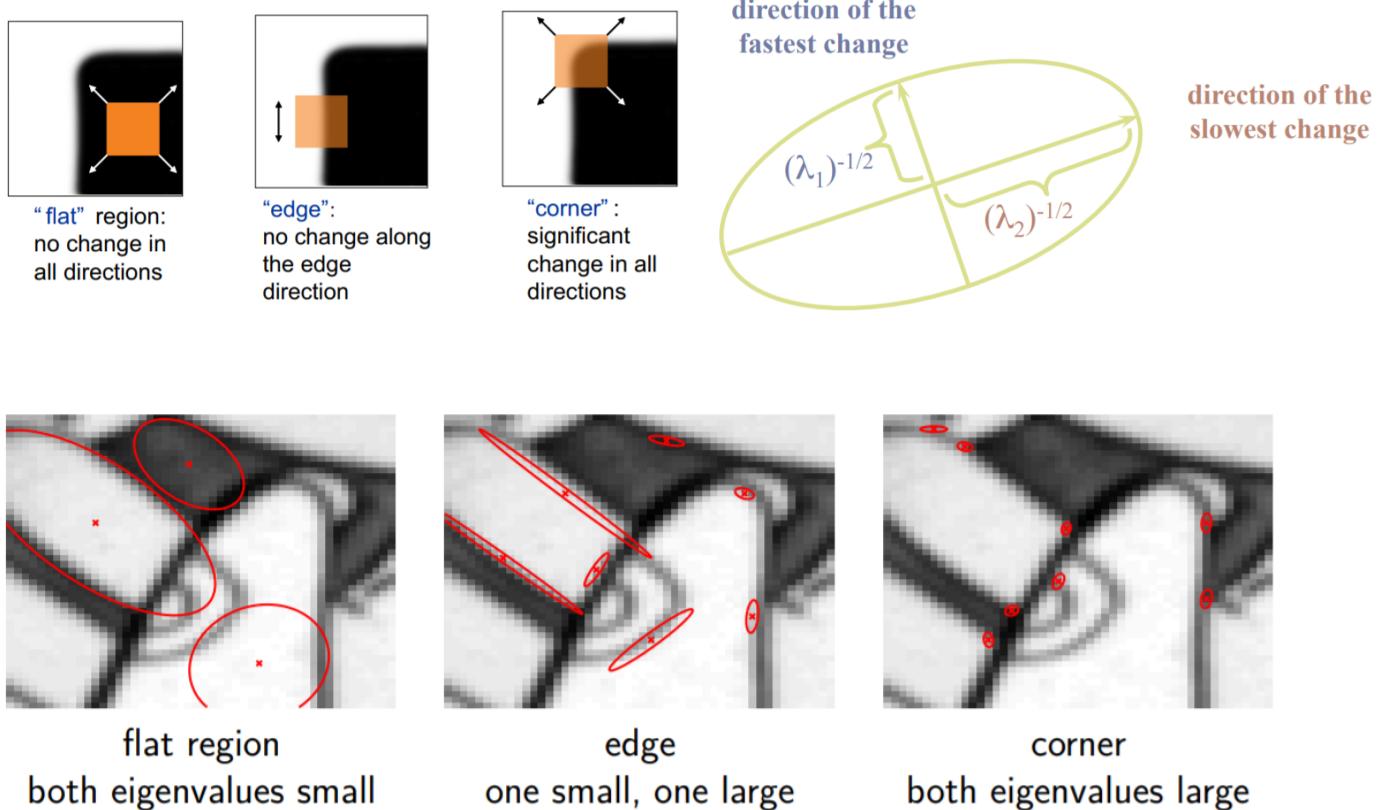
Hessian & Harris	[Beaudet 1978], [Harris 1988]
Laplacian, DoG	[Lindeberg '98], [Lowe 1999]
Harris-/Hessian-Laplace	[Mikolajczyk & Schmid 2001]
Harris-/Hessian-Affine	[Mikolajczyk & Schmid 2004]
EBR and IBR	[Tuytelaars & Van Gool 2004]
MSER	[Matas et al. 2002]
Salient Regions	[Kadir & Brady 2001]
Others ...	

Abbreviations:

DoG – Difference of Gaussians; EBR – edge-based region; IBR – intensity-based region; MSER – maximally stable extremal regions.

Harris corners

Vychází z aproximace autokorelační funkce obrazové funkce pomocí kvadratické formy. Ta tedy přibližně vyjadřuje, jak podobný je obraz své posunuté kopii pro každý směr posunu. Intuitivně je jasné, že v stejnorodých regionech si bude obraz a posun podobný, kdežto v rozích naopak ne. To se dá vyjádřit pomocí vlastních čísel kvadratické formy autokorelační funkce. Malá vlastní čísla v obou směrech → pomalá změna → plochý region. Jedno velké a jedno malé vlastní číslo → ve směru kolmo na hranu velká změna, po hraně malá → hraná, obě malá vlastní čísla → rychlá změna v obou směrech → roh.

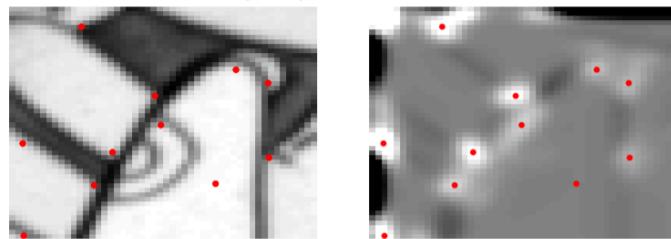


Aby se nemusela vlastní čísla explicitně počítat, vymyslel Harris kritérium na posouzení „rohovatosti“ obrazu v daném bodě (Harris's cornerness criterion), který závisí na determinantu (což je součin vlastních čísel) a stopě (což je součet vlastních čísel). Tam, kde je na obrazu lokální maximum cornerness, je určen roh.

Characterize ‘cornerness’ $H(x, y)$ by eigenvalues of $Q(x, y)$:

Lxx/Lyy

- $Q(x, y)$ is symmetric and positive definite $\Rightarrow \lambda_1, \lambda_2 > 0$
- $\lambda_1\lambda_2 = \det Q(x, y) = AC - B^2$, $\lambda_1 + \lambda_2 = \text{trace } Q(x, y) = A + C$
- Harris suggested: Cornerness $H = \lambda_1\lambda_2 - 0.04(\lambda_1 + \lambda_2)^2$
- Image $I(x, y)$ and its cornerness $H(x, y)$:



Find corner points as **local maxima** of cornerness $H(x, y)$:

- Local maximum in the image is defined as a point greater than its neighbors (in 3×3 or even 5×5 neighborhood)

Souhrn algoritmu Harris corners:

- ◆ Compute partial derivatives $I_x(x, y)$, $I_y(x, y)$ by finite differences:

$$I_x(x, y) \approx I(x+1, y) - I(x, y), \quad I_y(x, y) \approx I(x, y+1) - I(x, y)$$

Before this, it is good (but not necessary) to smooth image with Gaussian with $\sigma \sim 1$, to eliminate noise.

- ◆ Compute images

$$A(x, y) = \sum_W I_x(x, y)^2, \quad B(x, y) = \sum_W I_x(x, y) I_y(x, y), \quad C(x, y) = \sum_W I_y(x, y)^2$$

E.g., image $A(x, y)$ is just the convolution of image $I_x(x, y)^2$ with the Gaussian. Use MATLAB function conv2.

- ◆ Compute cornerness $H(x, y)$
- ◆ Find local maxima in $H(x, y)$. This can be parallelized in MATLAB by shifting the whole image $H(x, y)$ by one pixel left/right/up/down.

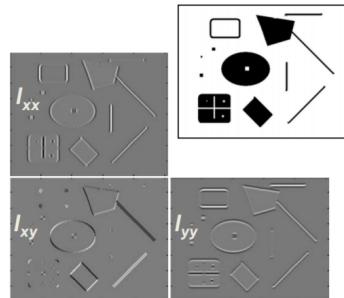
Hessiánové detektory

Hessian Detector [Beaudet78]

- Hessian determinant

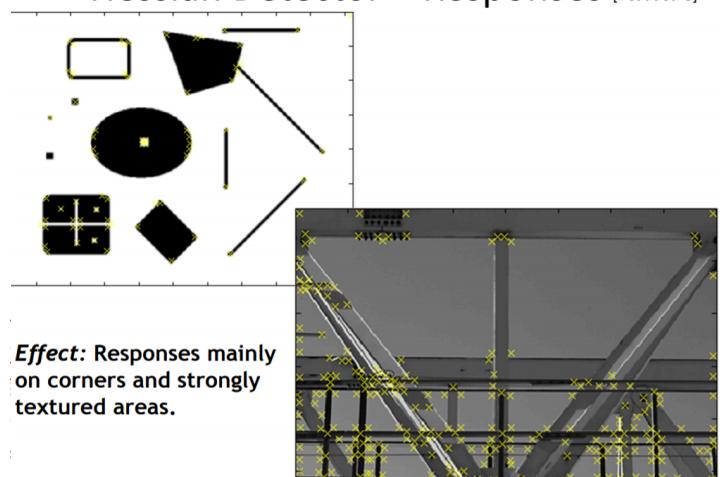
$$\text{Hessian}(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

Note: these are 2nd derivatives!



Intuition: Search for strong derivatives in two orthogonal directions

Hessian Detector – Responses [Beaudet78]



Vytvoření lokálního deskriptoru

Hlaváč tam o tom moc nemá, popisuje jen samotnou detekci významných bodů. Je o tom část té prezentace ze Stanfordu nebo i tady:

http://gertkootstra.com/ecse/day_2_part_1_interest_points.pdf?PHPSESSID=a184c46825164d8df1657872de228201

Deskriptor se vytvoří na základě okolí významného bodu. Je založený na vystihnutí textury v okolí bodu. Je důležité, aby deskriptor okolí dobře vystihoval a byl co nejunikátnější, aby se významné body správně přiřadily protějškům v jiném obrazu. Také je výhodné, pokud je deskriptor invariantní vůči škálování tzn. bod se přiřadí ke správnému protějšku i když mají obrazy jiné měřítko. Existují na to různé metody.

Matching

Zase je toho víc v prezentaci ze Stanfordu.

Viz registrace – na základě nějakého kritéria se posuzuje podobnost významných bodů s body jiného obrazu. Najdou se parametry, kdy je podobnost nejvyšší, a obrazy se zarovnají.

REGISTRACE OBRAZŮ

Postup, který vede k zarovnání obrazu a jiným referenčním obrazem. Cílem je často možnost automatického porovnání mezi obrazem a referencí, nebo mezi více obrazy, které se zarovnají na stejnou referenci. Jiné využití je nalezení podobrazu v obrazu, nebo obecně nalezení nějakého templatu/ vzoru v obrazu.

Matching – “Two regions in the image (two instances of the object) resembling or corresponding one to the another.”

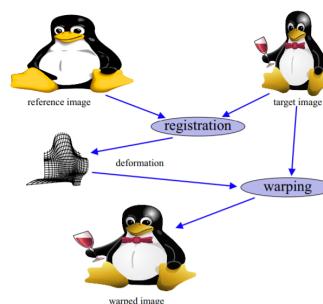
- ◆ Matching is a core of all three following concepts.
- ◆ Corresponding regions have to be found in the source and target data.

Registration – “To adjust so as to be properly aligned.”

- ◆ Geometric (and possibly photometric) alignment of one image with another, i.e. process of transforming different sets of data into one coordinate system.
- ◆ Images may be of same or different modalities (e.g. MR, CT).
- ◆ It is needed to be able to compare or integrate the data obtained from different measurements

Fusion – “Something new is created by a mixture of qualities, ideas, or things.”

Warping – “Become or cause to become bent or twisted out of shape, typically as a result of the effects of heat or dampness.”



Využití:

- ◆ **Photography** – Stitching photos together, panoramic images
- ◆ **Astronomy** – Stitching photos together and fusion of different wavelengths, modalities, e.g. optical, radar.
- ◆ **Chemistry** – Finding similar images of molecules



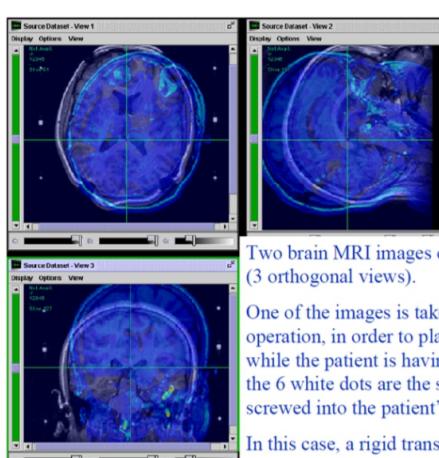
- ◆ **Signature comparison**



- ◆ **Human motion analysis** – temporal alignment of prototypical postures, e.g. gesture understanding
- ◆ **Stereo and shape from motion** – correspondence between key point
- ◆ **Video stabilization** – correspondence between key point

Registrace je velmi důležitá v oblasti medicínských obrazů. Využití:

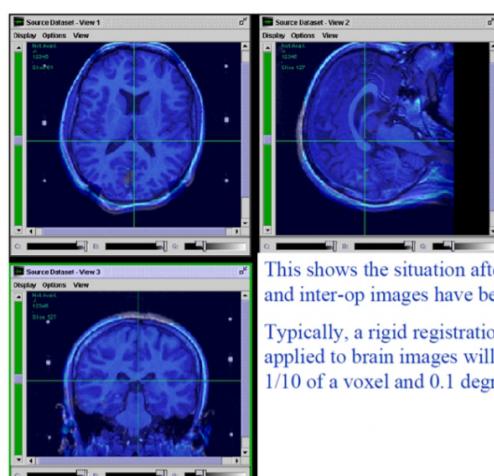
- Sestavení výsledného obrazu, který posuzuje lékař, z více nasnímaných obrazů. Například kolonoskopie, kde je původní obraz 1D „řetěz“ a registruje se tak, aby se sestavil 2D obraz případně 3D model střev.
- Registrace různých vyšetření jednoho pacienta. Pacient má pokaždé tělo v jiné poloze a obrazy tak na sebe nesedí a musí se koregistrovat. Umožní to automatické zpracování obrazů a velmi se tím usnadní práce lékaři. Může jít o stejnou modalitu (před a pooperační MRI mozku) nebo i různé modality (CT koregistrované na MRI mozku a jiné).



Two brain MRI images of the same patient (3 orthogonal views).

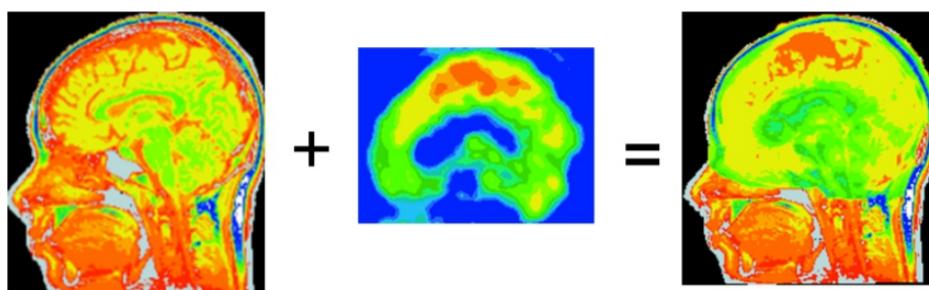
One of the images is taken prior to the operation, in order to plan it; the second while the patient is having the operation: the 6 white dots are the stereotactic frame screwed into the patient's skull.

In this case, a rigid transform suffices



This shows the situation after the pre-op and inter-op images have been aligned.

Typically, a rigid registration algorithm applied to brain images will be accurate to 1/10 of a voxel and 0.1 degrees of rotation

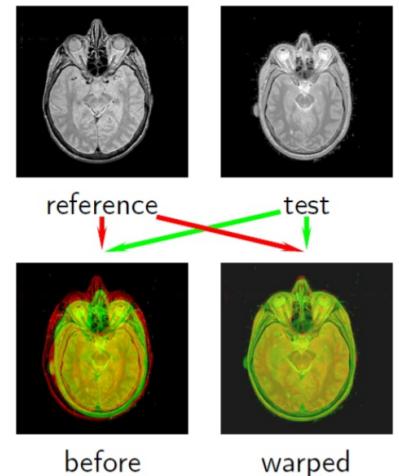


MR

SPECT

registered

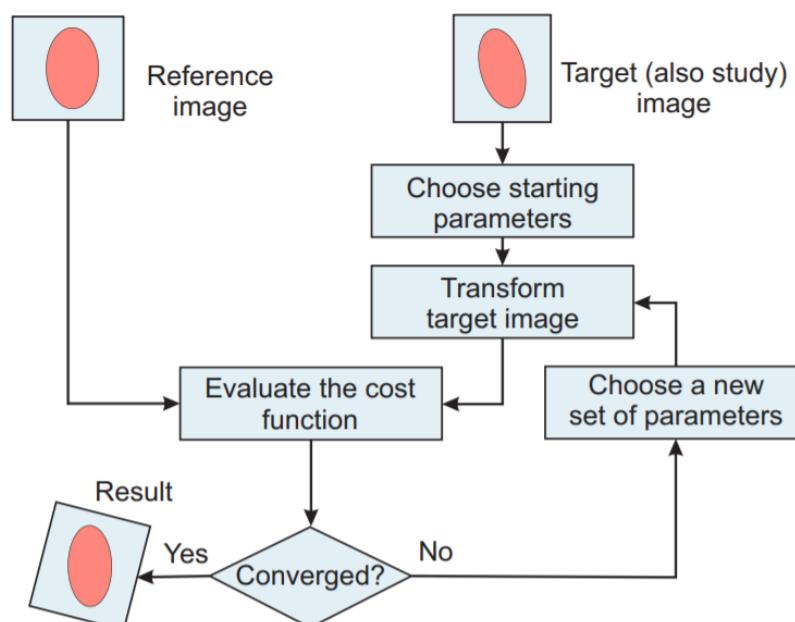
- Registrace na template. Umožní porovnání snímků stejné části těla mezi různými pacienty případně porovnání snímků konkrétního pacienta se zdravou referenční skupinou. Je velmi důležitá pro zpracování obrazů mozku, zejména MRI. Existuje tzv. atlasový mozek, který byl vytvořen jako průměrný MR obraz mozku stovek zdravých dobrovolníků. Ten se považuje za referenční template. U každého pacienta se najde registrační deformační transformace, která jeho snímky zarovná a deformuje tak, aby seděly na template. To umožní další automatickou analýzu pacientových snímků.



Registrace je obecně špatně řešitelný problém – dá se říct, že jde o nedourčenou soustavu. Máme totiž na začátku k dispozici méně dat než je parametrů, které potřebujeme určit. Musíme tedy řešení omezit jen na nějaký předem definovaný okruh registračních transformací, pak existuje jednoznačné řešení. Můžeme například povolit pouze rotaci, posun a škálování (rigidní registrace), nebo můžeme uvažovat i deformace (warping registration). Základní algoritmus pak funguje tak, že po krocích procházíme všechny parametry povolených transformací. V každém kroku vyhodnotíme nějakou cost function a hledáme parametry odpovídající „nejlevnějšímu“ řešení.

- ◆ Reminder:
An ill-posed problem has more degrees of freedom compared to available data.
- ◆ In registration:
If the reference image I_a has a total number of N pixels and the target image I_b has M pixels, then we need to find $2 * N$ coordinate values given $N + M$ values of input data.
- ◆ Consequently finding the best correspondence function φ (i.e. the transformation)

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \varphi \left(\begin{bmatrix} x_a \\ y_a \end{bmatrix} \right)$$
 is an Ill-posed problem!
- ◆ Solution: Restrict the set of admissible transformations φ .



Rigid registration

- ◆ Landmark based
- ◆ Edgels / surface based
- ◆ Pixel / voxel intensity based
- ◆ Information theory based

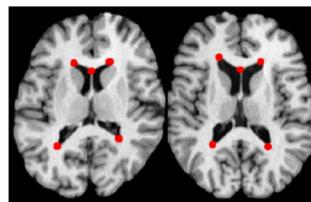
Non-rigid registration

(also deformable registration)

- ◆ Registration using basis functions
- ◆ Registration using curves surfaces (e.g. splines)
- ◆ Physics based, e.g. elastic, fluid, optical flow, etc.

Základní typy regisračních algoritmů:

- Založené na porovnávání intenzit všech odpovídajících pixelů v původním a referenčním obrazu. Pro každý krok algoritmu se pro dané parametry transformace spočítá podobnost obrazů. Při výpočtu podobnosti podle kritéria (MSE, korelační koeficient, vzájemná informace viz níž) se zahrnují všechny pixely – postupně se projede klouzavým oknem přes celý obraz. (v praxi se nedoporučuje)
- Založené na manuálně označených landmark points (významných bodech). Landmarks mohou být označeny dodatečně na obrazu, nebo už jsou pořízeny s obrazem (různé značky, co se lepí nebo kreslí na tělo pacienta). Transformace se pak hledá pouze na základě toho, jak na sebe sedí landmarks, nezohledňují se všechny pixely.
 - ◆ Manually defined landmarks serve to infer the deformation.
 - ◆ Landmarks
 - Extrinsic landmarks, artificial objects attached to, e.g., a patient.
 - Intrinsic landmarks, e.g. obvious anatomical structures.
 - ◆ Translation is estimated from landmark centroids shifts.
- Založené na automaticky určených významných bodech, hranách, obrysech apod. Opět se pak transformace hledá pouze na základě podobnosti těchto důležitých bodů, ne podle všech pixelů.



Kritéria pro posuzování podobnosti obrazů:

- MSE – vhodné pro stejné modality ze stejného přístroje (např. před a pooperační MRI)
 - ◆ Mean squared error $MSE = \sum_{x,y} (I_a(x,y) - I_b(x,y))^2$
 - ◆ Advantages:
 - Simple and intuitive.
 - Positive differences do not cancel negative differences.
 - Ensures exact intensity matching.
 - ◆ Disadvantages:
 - Sensitivity to outliers.
 - Linear mean squares will not be the best fit for data that is not linear.
 - Images must have almost the same brightness and contrast.
 - Does not work if modalities differ.

- Korelační koeficient – také na stejně modality, ale funguje i pro lineární změny v intenzitě (MRI pořízená na jiných přístrojích s jinými intenzitami)
- The (Pearson's) correlation coefficient ρ of the two random variables is the covariance of two variables divided (normalized) by their standard deviations σ .

In the case of images I_a (the reference image) and I_b (the target image),

$$\rho = \frac{E((I_a - \mu_a)(I_b - \mu_b))}{\sigma_a \sigma_b},$$

where E expresses the expectation of a probability distribution and μ is the population mean. If f is the discrete probability density of the distribution X then

$$E(X) = \sum_{x \in X} x f(x)$$

◆ Advantages:

- Corrects for linear intensity changes, which is common in practice.
- It is efficient to evaluate.
- Can be generalized to non-scalar signals using the “Canonical Correlation Analysis”, see [overview paper, tutorial](#).

◆ Disadvantages:

- There is the underlying assumption about Gaussian distributions due to the Central limit theorem.
- Does not work for (very) different modalities.

- Vzájemná informace – funguje i pro různé modality (např. CT + MRI).
 - ◆ The mutual information of two discrete random variables X and Y is based on the joint probability distribution $p(X, Y)$, which is approximated by the joint histogram in practice.
 - ◆ The mutual information is defined as

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right),$$

where $p(x)$, $p(y)$ are marginal probability density functions.

- ◆ The mutual information measures the information that X and Y share. It tells how much knowing one of these variables reduces uncertainty about the other.

◆ Advantages:

- Rather general approach.
- Works for different modalities.
- Works for non-scalar signals, e.g. RGB images.

◆ Disdvantages:

- Can be tricky to implement.
- Slower evaluation as compared to MSE and correlation.

