# 6. Hledání korespondencí mezi obrazy. Houghova transformace (HT). Použití HT a metody RANSAC (Random Sample and Consensus) pro robustní nalezení transformace mezi obrazy.

# Korespondence

Hledání korespondencí je jedním ze základních problémů počítačového vidění. Cílem je najít odpovídající si části scény ve dvou nebo více obrazcích. Hledání korespondence pro každý pixel je výpočetně náročné.

Metoda založená na korespondencích lokálních příznaků proto v každém obrazu nejdřív nalezne tzv. detektorem množina 'záchytných' bodů resp. regionů, ze kterých budou vybrány korespondující (lícovací) páry. Je třeba najít v různých obrazech takové odpovídající si body, které lze opakovaně identifikovat nehledě na geometrické a fotometrické transformace obrazu. K pozici každého záchytného bodu se přiřadí zakódovaný popis jeho okolních pixelů. Může to být např. vektor jasů okolních pixelů, histogram jasů, histogram gradientů nebo jiný invariantní popis. Způsob zakódování ovlivňuje dvě věci: invarianci vůči transformacím a diskriminativitu popisu. Výslední popis je takzvaný lokální příznak, vektor zachycující charakteristiky obrazové funkce malého, dobře lokalizovaného okolí záchytného bodu. Při vlastním hledání korespondencí metodou lokálních příznaků se každý záchytný bod z prvního obrazu nalezneme bod z druhého obrazu s nejbližším popisem. Vlivem šumu, zákrytů a nepřesností modelů geometrické a fotometrické transformace jsou některé nalezené korespondence špatné, t.j. nespojují tytéž body scény. Potřebujeme tedy robustní algoritmus, jenž na základě známého modelu transformace scény vybere největší podmnožinu korespondencí, v níž se všechny korespondence transformují konzistentně s modelem. Jedním z takových algoritmů je **RANSAC** (RANdom SAmpling Concensus)

Po **detekci a generování popisu**, jenž probíhaly na každém obrázku scény **nezávisle**, následuje vlastní hledání **korespondencí**. První fází je nalezení tzv. **tentativních** **korespondencí** (někdy nazývaných i “*matchů*”). Tentativní korespondence je **pár** – **dvojice popisů**, jež jsou si v nějaké **vzdálenosti** **podobné**. Nejjednodušší metodou pro nalezení korespondencí je najít **vzdálenosti** mezi všemi **páry** popisů. Tuto tabulku nazveme **maticí vzdáleností**. Jako kritérium podobnosti popisů použijeme euklidovskou vzdálenost. Označme si obrázky, v nichž hledáme korespondence pravý a levý. Zkusíme si několik metod výběru tentativních korespondencí:

1. Tentativní korespondence **vzájemně nejbližších** popisů vznikne, pokud zvolený popis A z pravého obrázku má ze všech popisů z levého obrázku nejbližší popis B a zároveň, popis B má nejbližší ze všech popisů v pravém obrázku popis A.
2. Při (slabém) **stabilním párovaní** popisů, postupujeme jinak. V každém kroku algoritmu najdeme globálně nejbližší popisy A a B. Vytvoříme tentativní korespondenci a popisy A a B vyloučíme z dalšího hledání (např. nastavením vzdáleností příslušného řádku sloupce na nekonečno). Postup opakujeme, pokud zůstali v matici vzdáleností nějaké aktivní prvky (hodnoty menší než nekonečno).
3. Jiný postup je najít ke každému popisu z levého obrazu nejbližší dva v pravém obrazu. Když je poměr vzdáleností **prvního/druhého nejbližšího** popisu menší než např. 0.8, popisy korespondují, a zároveň je zaručeno, že v jistém “okolí” v prostoru popisů není žádný bod s kterým by jsme si ho mohli splést. Tuto metodu je možné použít pro diskriminativní popisy, např. **SIFT**.

Výstupem hledání tentativních korespondencí jsou **dvojice indexů**, párů z **levého** a **pravého** obrázku s nejpodobnějšími popisy.

# Houghova transformace (HT)

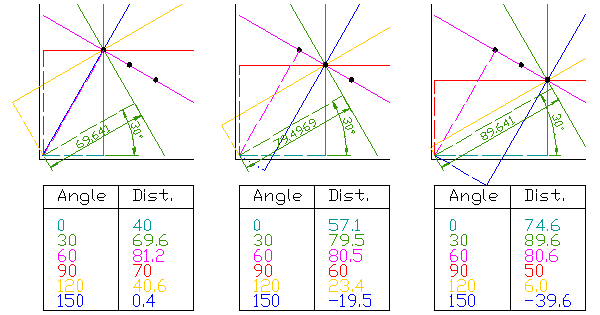
Houghova transformace je metoda pro nalezení parametrického popisu objektů v obraze. Při implementaci je třeba znát analytický **popis tvaru hledaného objektu**. Proto je tato metoda používána pro **detekci jednoduchých objektů** v obraze jakou jsou **přímky**, **kružnice**, **elipsy**. Hlavní výhodou této metody je **robustnost** vůči **nepravidelnostem** a **porušení** hledané křivky.

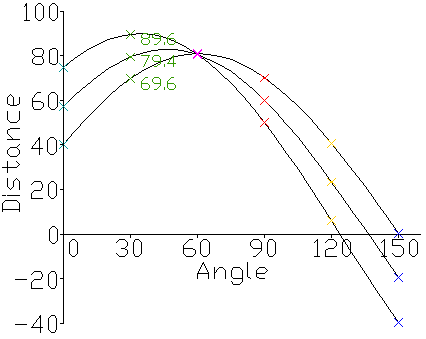
**Workflow:**

1. definuj **parametry** hledaného objektu ( přímka: fi, r kružnice: x, y, r )
2. rozděl Houghův prostor: pro každý parametr urči meze a prostor rozděl do určeného počtu buněk
3. vytvoř akumulátor A(fi, r) a nastav vše na 0
4. pro **všechny vstupní body**: pro každý možný model objektu (určený parametry) zahlasuj do akumulátoru.
5. vyber z akumulátoru takovou sadu parametrů, která má největší počet hlasů

**Paměťová náročnost** = jedinou paměťově náročnou proměnnou je Akumulátor, ten rapidně stoupá s počtem parametrů

**Příklad**:





Máme body v obrázku (ty tři černé tečky) a hledáme optimální přímku, přímka je definována 2 parametry 1) vzdálenost od počátku soustavy a 2) úhel natočení, tak třeba ta zelená přímka je definována tak, že je vzdálená 69,641 a úhel je 30°.

V každém černém bodě zkonstruujme spoustu (zde právě 6) různých přímek, takže volíme různé parametry úhlu a vzdálenosti tak, aby přímka procházela tím černým bodem. Všechny možné přímky, které prochází černým bodem odpovídají té křivce v grafu pod tím.

Pro každý černý bod zkonstruuju takovouhle křivku, takže tady jsou 3 křivky. Každá křivka "hlasuje", tam kde je nejvíc hlasů - tam kde se křivky protínají je nejoptimálnější přímka.

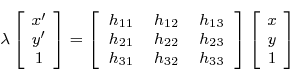
Pak vyberu ty parametry - tedy **úhel 60**, **vzdálenost 80** a mám **optimální přímku** na které všechny ty tři body leží.

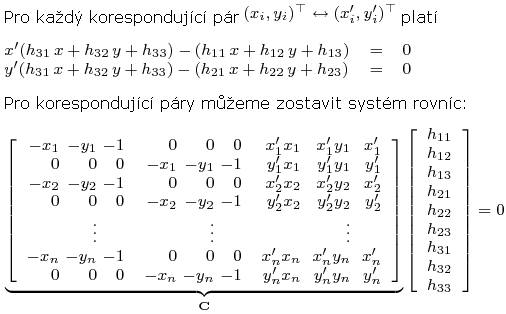
# RANSAC(Random Sample Consensus)

**Dvojice bodů** (tentativní korespondence) získané v předchozím kroku na základě podobnosti zpravidla obsahují mnoho nekorespondujících bodů (nazývají se **outliers**), tedy tentativních korespondencí, jenž mají podobné popisy ale **neodpovídají jednomu fyzickému prvku scény**. Cílem poslední fáze hledání korespondencí je **zbavit** se těchto bodů za předpokladu možné transformace scény mezi dvěma pohledy. Výsledkem je nalezení modelu – **transformace scény** a množiny tzv. **inlierů**, tentativních korepsondencí, jenž jsou **konzistentní s nalezeným modelem**.

### Nalezení transformace rovinné scény

Transformace roviny mezi dvěma pohledy perspektivní kamery je lineární transformace (nazývaná taky **homografie**) 3-dimenzionálních homogenních vektorů (korespondujících bodů v homogenních souřadnicích) reprezentovaná regulární Graph maticí Graph:





Tento homogenní systém rovnic má 9 neznámých. Jedno netriviální řešení dostaneme jako pravý nulový prostor matice Graph. Pro jednoznačné řešení potřebujeme, aby matice Graph měla alespoň 8 lineárně nezávislých řádků, tj. 8 omezení ze 4 korespondujících bodů v obecné poloze (žádné tři nesmí ležet na přímce).

Pro nalezení modelu scény – správné projektivní transformace dvou rovin, potřebujeme alespoň čtyři bezchybné korespondence, které neleží na přímce. Abychom nalezli nejlepší projektivní transformaci mezi získanými dvojicemi bodů z tentativních korespondencí, použijeme algoritmus M.Fischlera a R.C.Bollese, **RANSAC**.

Iterativní metoda pro odhad parametrů matematického modelu z pozorovaných dat zatížených šumem (obsahujicích **outliery**). Je to randomizovaný algoritmus, funguje s ohledem na míru spolehlivosti.

1. Náhodně vybereme 4 korespondující páry, tzv. vzorek (sample).
2. Z těchto bodů vypočteme homografii Graph.
3. Zjistíme tzv. **podporu** nebo-li **support** nalezeného modelu. Je-li Graph spočtená ze správných korespondencí, měly by transformované body Graph ležet na místech korespondujících bodů Graph v druhém obraze. Podporu modelu kvantifikujeme počtem bodů konzistentních s modelem. Tj. bodů se vzdáleností transformovaného bodu z **levého** obrázku a odpovídajícího bodu v **pravém** obrázku menší než zadaný **práh** (např. 1-3 pixely).
4. Pokud je hodnota podpory nalezené homografie doteď největší, uložíme si její hodnotu, doteď-nejlepší-model Graph a množinu **inlierů** nebo-li bodů, které podpořili tento model.
5. Body 1-4 opakujeme **iterativně**, a hledáme takovou Graph, která správně transformuje největší počet korespondencí, tj. maximalizuje kritérium definované v bodě 3.

**Vstupem** budou dvojice souřadnic u (matice 6xN, dvojic odpovídajících si bodů v homogénních souřadnicích - souřadnice šesti tentativních korespondencí).

**Výstupem** bude nejlepší nalezený model Graph a pole inl 1xN, kde 0 bude znamenat outlier a 1 inlier, kde N je počet bodů, sloupců matice u.

**Algoritmus obecný**

1. inicializuj hodnotu maximalního počtu iterací nutných k nalezení modelu s určenou spolehlivostí (niter)
2. i = 0
3. Dokud i < niter:
4. Náhodný výběr vzorku m bodů (m je počet bodů charakteristický pro daný model - např. 2 pro přímku, 3 pro kružnici, atd.)
5. Výpočet parametrů modelu, který „sedí“ na vybraný vzorek.
6. Výpočet odchylky všech bodů od modelu
7. Výpočet podpory (inlierů, tzn. bodů, které podporují tento model), pokud je nový model lepší než doposud nalezený, ulož jeho parametry a přepočítej (sniž) maximální počet iterací (niter)
8. Zvyš číslo iterace i a opakuj výběr vzorku

**Workflow**

|  |  |  |
| --- | --- | --- |
| *Select sample of m points at random* | *Calculate model parameters that fit the data in the sample* | *Calculate error function for each data point* |
| *Select data that support current hypothesis* | *Repeat sampling ...* |  |
|  |  |  |

**Vlastnosti**

* Výhodou RANSACu je schopnost robustního odhadu parametrů modelu (i přes relativně vysoký počet outlierů dosahuje vysoké spolehlivosti)
* Nevýhodou je absence horní meze časové složitosti. Dále korektnost výsledků (degenerace) a přesnost (model je odhadován z minimálního vzorku)

**Parametry**

* N - Počet bodů
* I - počet inlierů
* m - velikost vzorku pro odhad modelu
* p - požadovaná spolehlivost (např. 0.95)

# Porovnání HT a RANSAC

**HT**

**výhody**

* velice efektivní pro rozpoznávání libovolných obrazců a objektů
* vypořádá se s velkým procentem out-lierů (95%)
* vytáhne seskupení ze změti v lineárnim čase

**nevýhody**

* problém kvantizace
* použitelné pouze pro modely s parametry do dimenze 4

**RANSAC**

**výhody**

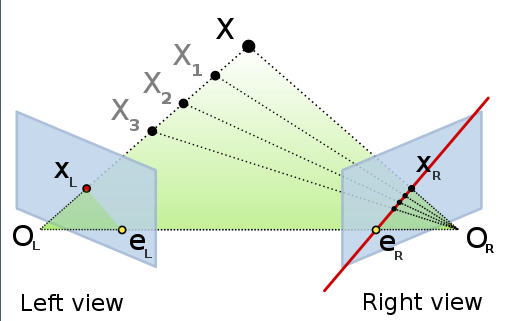
* obecný algoritmus použitelný pro širokou množinu problémů
* jednoduchý na implementaci
* nezávislý na dimenzi parametrů modelu.

**nevýhody**

* schopný pracovat pouze s průměrným počtem out-lierů (50%)

## Vzájemná poloha dvou kamer, epipolární geometrie

Aby jste mohli ověřit korespondence i pro jiné než rovinné scény, budeme potřebovat obecnější vztah svazující dva body ve dvou kamerách. Tento vztah se pro dvojici perspektivních kamer jmenuje epipolární geometrie. Epipolární geometrie je geometrický vztah korespondujících bodů Grapha Graph, obrazů bodu ve scéně viděného dvojicí perspektivních kamer:



Tenhle vztah se dá matematicky vyjádřit: Graph

Epipolární geometrie je reprezentována maticí Graph, zvanou **fundamentální** matice. Pro její nalezení je potřebných alespoň 7 korespondujících bodů.

Aby náš **RANSAC** fungoval pro hledání epipolární geometrie, budeme potřebovat dvě věci, funkci pro výpočet modelu – fundamentální matice Graph a funkci pro nalezení vzdálenosti bodů od modelu, tj. vzdálenosti od epipolární přímky (obrazu paprsku v druhé kameře). Funkce budou mít stejné parametry jen matice Graph se zamění za matici Graph.

**Vstupem** budou dvojice souřadnic odpovídajících si bodů (dle tentativních korespondencí) v homogenních souřadnicích.

**Výstupem** bude nejlepší nalezený model Graph a pole inl 1xN, kde 0 bude znamenat outlier a 1 inlier, kde N je počet bodů, sloupců matice u.