

Felületi normálvektorok becslése affin transzformációból

Baráth Dániel

QJV3ZD

danini@freemail.hu

2013.

Konzulensek:

Hajder Levente

Molnár József

1.1.1. Tartalomjegyzék

2.	Bevezetés	3
2.1.	Motiváció	3
2.2.	Normálvektorok becslésének alaplépései	3
2.3.	Célkitűzéseink	5
3.	Elméleti áttekintés	6
3.1.	Fotometrikus sztereo.....	6
3.2.	SIFT Feature detektor	6
3.3.	SURF Feature detektor	7
3.4.	KLT feature tracker	7
3.5.	Iterative Multi-View Plane Fitting.....	7
3.6.	A surface-growing approach to multi-view stereo reconstruction	7
4.	A normálvektor becselő	9
4.1.	Definíciók és jelölések	9
4.2.	Képrészletek megfeleltetése	10
4.3.	Gyorsbecslő	14
4.3.1.	Specializáció perspektív kamerára.....	16
4.4.	Pontosítás	19
4.5.	Ismert mélység (si) esetén.....	19
4.5.1.	A feladat megoldása	20
4.5.2.	A két kvadratikus görbe metszéspontja (Optimális becselő)	24
4.5.3.	Közelítő megoldás lineáris módszerrel.....	26
4.6.	Ismeretlen mélység (si) esetén.....	27
4.6.1.	Közelítő megoldás lineáris módszerrel.....	28
4.6.2.	Konvergálás a (lokálisan) optimális megoldás felé (Alternálás)	29
5.	Affin transzformációt becselő eljárás	41
5.1.	Affin transzformáció (elméleti áttekintés)	41
5.2.	Becslés globális kereséssel	42
5.3.	Elforgatás meghatározása	43
5.4.	Skálázás meghatározása	45
5.5.	A nyírás meghatározása.....	46

5.6.	Az eltolás meghatározása	48
5.7.	Lokális paraméterfinomítás	50
5.8.	Az affin transzformációt becslő algoritmus menete	51
5.9.	Hibás megoldások kezelése	52
6.	Régió-alapú osztályozás	54
6.1.	Régiók meghatározása.....	54
6.2.	Régiókhoz tartozó normálvektor meghatározása	58
6.3.	Kiértékelés	60
7.	Kiértékelés.....	31
7.1.	Szintetikus tesztek	31
7.1.1.	Gyorsbecslés – Optimális becslő	31
7.1.2.	Gyorsbecslés – Alternálás.....	32
7.1.3.	Optimális becslő – Alternálás	34
7.1.4.	Optimális becslő – Lineáris közelítés ismert mélység esetén.....	36
7.1.5.	Alternálás – Lineáris közelítés ismeretlen mélység esetén	37
7.2.	Valós tesztek.....	39
8.	Konklúzió.....	61
9.	Hivatkozások.....	62

2. Bevezetés

2.1. Motiváció

Dolgozatomban a számítógépes látás egy eddig megoldatlan problémájával foglalkozom: megmutatom, hogyan lehet kalibrált projektív kamerák segítségével egy felületi ponthoz tartozó normálvektort megbecsülni, ha a két képen a régió középpontját és a vetületük közötti affin transzformációt ismerjük. Erre a problémára eddig ugyanis szinte kizárólag az epipoláris geometrián alapuló megoldások nyújtottak valamiféle megoldást, pedig a különböző rekonstrukciós, illetve klaszterező eljárások a normálvektorok pontosabb ismeretében hatékonyabban és robosztusabban működtethetők.

Fontos szempont még a becslő eljárás optimalitása is, ugyanis valós adatokon a lehető legnagyobb mértékben ki kell küszöbölni, minimalizálni kell a hibát – amely az adatok pontos ismeretének hiányában minden esetben lesz; a különböző becslési folyamatok során is (pl. feature pontok megfeleltetése, affin transzformáció becslése) hozzáadódik a rendszerhez.

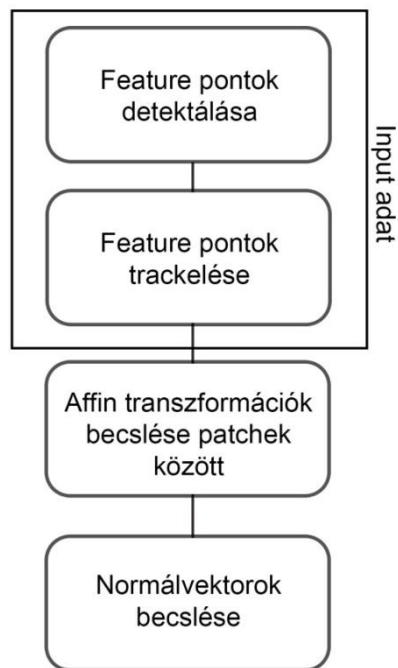
2.2. Normálvektorok becslésének alaplépései

A valós adatokon alapuló tesztek esetén a detektált feature pontokat [1], és azok sztereo képpáron való megfeleltetését [2][3] input adatnak tekintettük, ugyanis ezek meghatározására már jó néhány jól működő algoritmus elérhető.

Ezzel szemben jól működő és robosztus affin transzformációt becslő algoritmusra volt szükség – ezt nem vehettük input adatnak, hiszen az eljárás működése nagyban függ az affin transzformáció minőségétől. Ez az eljárás a dolgozat egy későbbi fejezetében kifejtésre kerül.

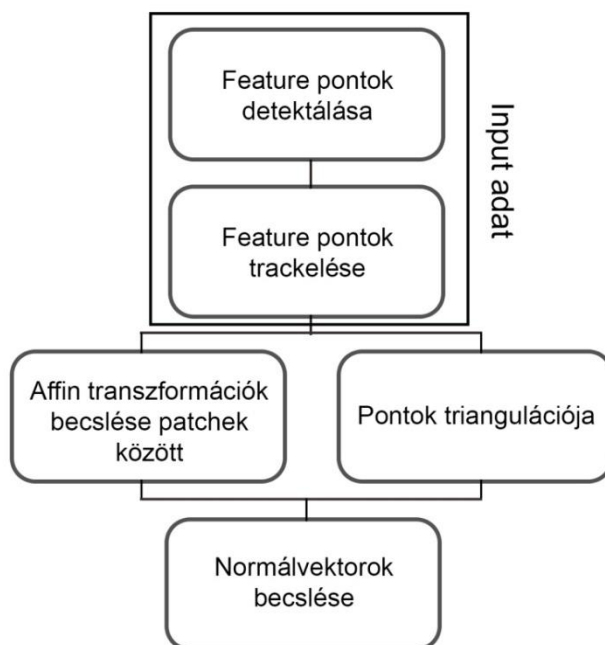
Az optimális becslőhöz szükségesek a detektált és megfeleltetett pontok háromdimenziós koordinátái is, ezeket a Hartley-Sturm páros által bevezetett optimális (legkisebb négyzetes értelemben) triangulációs eljárás [4] segítségével számoljuk ki.

Tehát az alapegyenleteket felhasználó eljárás lépései a következők:



I.1 Az alapegyenleteket használó eljárás menete

Az optimális becslő eljárás lépései:



I.2 Az optimális becslő eljárás menete

2.3. Célkitűzéseink

Célkitűzésünk egy olyan optimális normálvektor-becslő eljárás elméleti hátterének megalkotása, amely a szintetikus teszteken túl robosztusan alkalmazható valós környezetben mért adatokon is. Továbbá célkitűzések a későbbiekben a már letesztelt algoritmusok GPU-ra való megírása – a real time feldolgozás érdekében -, illetve a kinyert normálvektorok rekonstrukcióban való alkalmazása.

3. Elméleti áttekintés

3.1. Fotometrikus sztereo

Mióta Woodham kitalálta fotometrikus sztereo egy közkedvelt témakör lett a számítógépes látásban. Az elmélet sok további finomítása, illetve változata jelent meg az évtizedek folyamán, de itt csak a Woodham által kidolgozott elméletet tekintjük.

A fotometrikus sztereo eljárás fő koncepciója, hogy több olyan képből határozzuk meg a felület irányultságát (a felületi normálist), ahol a kamera pozíciója fix, de a megvilágítást változtatjuk. Általános esetben ilyenkor nincs információnk se a kameráról, se a fényforrásról, se a visszaverő felületről. Természetesen ebben az esetben a felületi normálist nem lehet pontosan meghatározni, tehát megkötéseket kell tennünk a környezetre vagy a felületre a precíz eredményekért cserébe.

A legáltalánosabb megkötés, hogy a kamera és a fények kalibráltak, illetve a felület Lambert-féle felület. Ekkor a felületi normálisok könnyen számolhatók egy lineáris egyenletrendszer megoldásával. Az ennél komplexebb BRDF-el rendelkező felületek a problémát sokkal bonyolultabbá tehetik.

3.2. SIFT Feature detektor

Scale-invariant feature transform (vagy SIFT) egy algoritmus a képi jellegzetes pontok detektálására és leírására. Az algoritmus David Lowe által lett publikálva 1999-ben. Sok eljárás felhasználja működése folyamán, például: objektum felismerés; robotikában a térképezés, illetve navigáció; 3D modellezés; stb.

Az alapkoncepció, hogy minden objektum, illetve képi jellemző pont leírható úgynevezett „feature descriptor”-ok segítségével. Fontos szempont ekkor, hogy ezek a meghatározott deskriptorok egymással könnyen összehasonlíthatóak legyenek, illetve robosztus legyen skálázásra, zajra és illumináció-változásra. Sok pont nagy kontrasztú régióknál jelenik meg, például objektum élei mentén.

Fontos tulajdonság még ezeket felül, hogy az párok közti relatív távolság ne, vagy csak minimális mértékben változzon az első képen detektálthoz képest. Általános esetben a SIFT detektor nagy mennyiségű feature pontot talál, amiket aztán különböző szempontok szerint (illesztési hiba, stb.) kiszűr. [8] [9]

3.3. SURF Feature detektor

A SURF (Speeded Up Robust Features) egy Herbert Bay által kidolgozott robusztus feature detektor, amit a tudomány sok területén alkalmaznak: gépi látás, objektum felismerés, 3D rekonstrukció. Részben SIFT deskriptorokra épülő deskriptorokat használ. A SURF egy általános változata néhányszor gyorsabb, mint a SIFT algoritmus, és robusztusabb a különböző képi transzformációkra. [10]

3.4. KLT feature tracker

A Kanade–Lucas–Tomasi (KLT) feature tracker egy olyan megoldás a feature regisztrációs problémára, ami az tradicionális megoldásoknál sokkal kisebb számítási költséggel, illetve robusztusabban működik. A KLT a térbeli intenzitás információk segítségével keresi a legjobban illeszkedő párt. [5] [6] [7]

3.5. Iterative Multi-View Plane Fitting

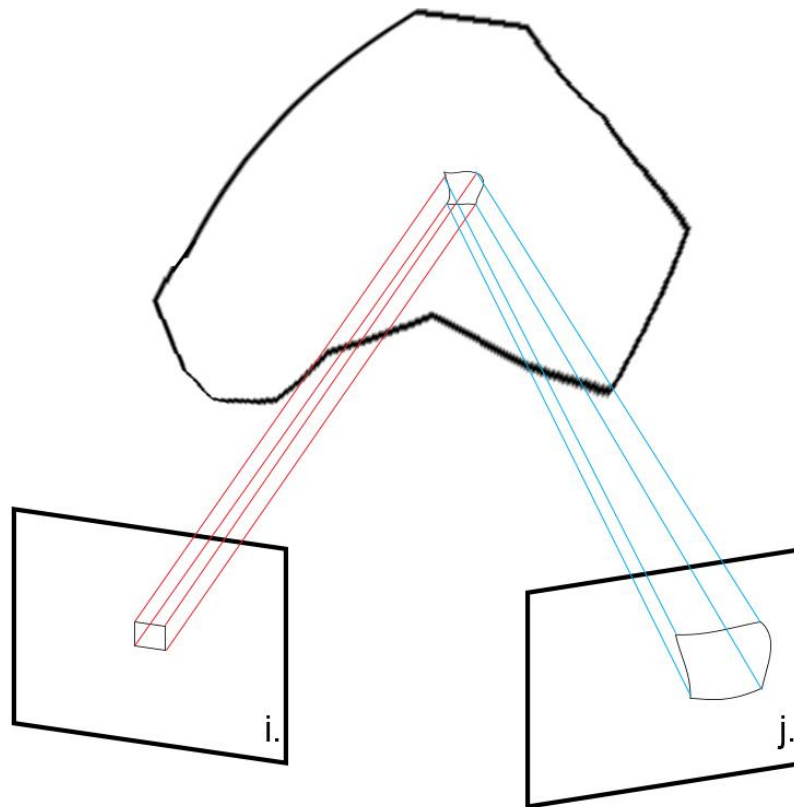
Ezen algoritmus alapkonceptiója, hogy sztereo képpáron próbálja meghatározni az egyes detektált feature pontokhoz a hozzájuk legjobban illeszkedő síkot. Ezt kétdimenziós projektív leképezések (homográfia) meghatározásának segítségével. [11]

3.6. A surface-growing approach to multi-view stereo reconstruction

Ez az algoritmus alapvetően egy régió-növesztéses technikával megkapott síklemezekből történő rekonstrukciós eljárást vázol fel. Az eljárás addig tart, amíg a kép teljes egésze le nincs fedve lemezekkel. Ekkor tovább finomíthatóak a régiókra vonatkozó információk (pl. normálvektor, 3D-s pozíció). [\[12\]](#)

4. A normálvektor becslő

A Molnár József által bevezetett alapegyenletek két kalibrált kamera esetén (az i . index jelen esetben nem az i . deriváltra, hanem az i . képindekre vonatkozik) a következők.



IV.1 Vetítés sztereo képpárra.

4.1. Definíciók és jelölések

A színtér a háromdimenziós euklideszi tér: $\mathbb{R}^3$, aminek i, j, k legyen az egymásra kölcsönösen merőleges standard bázisa, és X, Y, Z -vel jelöljük a

térkoordinátákat. Az objektumok legyenek felületükkel reprezentálva. Ezekről jelen esetben paraméteres formában beszélünk:

$$X, Y, Z: \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$u, v \in \mathbb{R}$$

IV.2

$$S(u, v) = X(u, v)i + Y(u, v)j + Z(u, v)k$$

Itt az X, Y, Z függvények a megfelelő térkoordinátákra képzik le az u, v paramétereket. A vetítési függvény (S) ekkor a felület minden (u, v) pontjához egy $[X, Y, Z]^T$ térkoordinátát rendel. Ekkor az i . vetítési függvény általános alakja:

$$x'_i, y'_i: \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$x_i, y_i: \mathbb{R}^2 \rightarrow \mathbb{R}$$

IV.3

$$x_i = x'_i(X(u, v), Y(u, v), Z(u, v)) = x_i(u, v)$$

$$y_i = y'_i(X(u, v), Y(u, v), Z(u, v)) = y_i(u, v)$$

Ekkor feltesszük, hogy a felület megfigyelt pontja körüli kicsiny nyílt koordinátakörnyezet képei is léteznek a vetületeken. Ez a függvények differenciálhatóságának a feltétele, illetve feltesszük a vetítési függvényekről, hogy minden (u, v) pontban létezik inverzük.

4.2. Képrészletek megfeleltetése

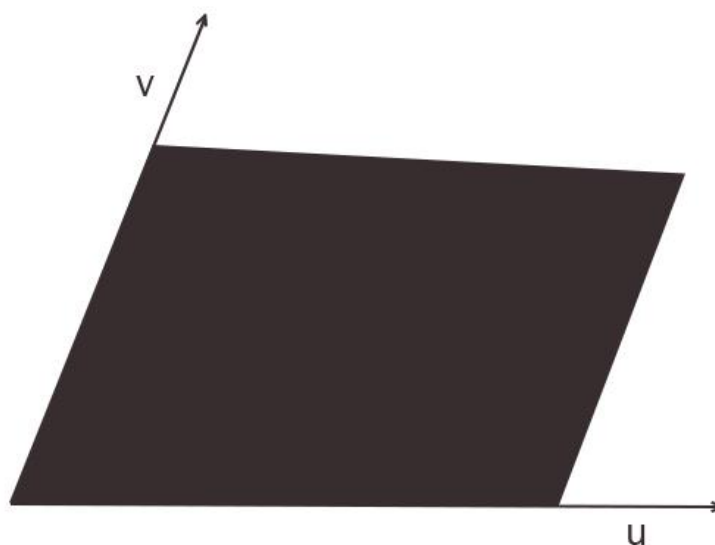
Ekkor az egyenletekkel adott függvényeket (u, v) -ben sorba fejthetjük. Innentől kezdve az i, j index a különböző képek indexei. A sorból csak az elsőrendű tagokat tartjuk meg. A megfigyelt pont körüli differenciálok ekkor:

$$dx_i = x_i(u + du, v + dv) - x_i(u, v)$$

$$dy_i = y_i(u + du, v + dv) - y_i(u, v)$$

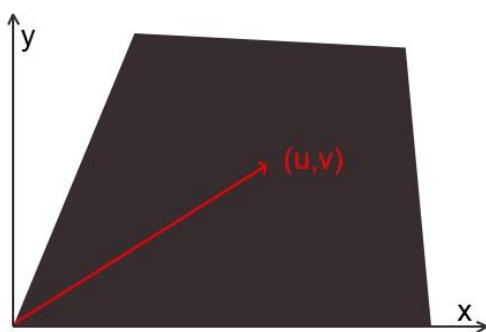
IV.4

Ahol adott egy háromdimenziós objektum felülete,

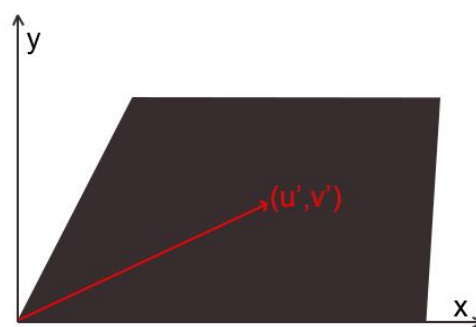


IV.5: (u,v)-vel paraméterezett felület

illetve annak két különböző vetülete.



IV.6.: Első vetület



IV.7.: Második vetület

Ebben az esetben ha első rendben közelítjük a differenciálok, akkor megkapjuk a

$$\begin{bmatrix} dx_i \\ dy_i \end{bmatrix} \approx \begin{bmatrix} \frac{\partial x_i}{\partial u} & \frac{\partial x_i}{\partial v} \\ \frac{\partial y_i}{\partial u} & \frac{\partial y_i}{\partial v} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = J_i \begin{bmatrix} du \\ dv \end{bmatrix} \quad \text{IV.8}$$

egyenletet ($J \in \mathbb{R}^{2 \times 2}$ a Jakobi mátrix). Ezentúl hagyjuk el a $\approx$ jelet, legyen helyette egyenlőség. Kifejezve innen $\begin{bmatrix} du \\ dv \end{bmatrix}$ -t adódik, hogy

$$\begin{bmatrix} du \\ dv \end{bmatrix} = J_i^{-1} \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} \quad \text{IV.9}$$

Ezt behelyettesítve a második vetülethez tartozó egyenletbe

$$\begin{bmatrix} dx_j \\ dy_j \end{bmatrix} = J_j J_i^{-1} \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} \quad \text{IV.10}$$

A megjelenő $A_{ij}(J_j J_i^{-1}) \in \mathbb{R}^{2 \times 2}$ mátrixot nevezzük ezentúl affin transzformációnak. Tehát a

$$\begin{bmatrix} dx_j \\ dy_j \end{bmatrix} = A_{ij} \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} \quad \text{IV.11}$$

egyenlet adódik. Ennek az A_{ij} transzformációs mátrixnak a komponensei jelenleg paraméteres formában adóttak, célunk, hogy invariáns formában adjuk meg őket. Ehhez figyelembe kell venni, hogy a **(IV.3)** vetítési függvények a felület pontjainak térkoordinátái **(IV.2)** szerinti összetett függvények, így a differenciáloperátorok között egy tetszőleges f függvényre alkalmazva a következő összefüggés áll fenn:

$$\begin{aligned} \frac{\delta f}{\delta u} &= \frac{\delta X}{\delta u} \frac{\delta f}{\delta X} + \frac{\delta Y}{\delta u} \frac{\delta f}{\delta Y} + \frac{\delta Z}{\delta u} \frac{\delta f}{\delta Z} = S_u * \Delta f \\ \frac{\delta f}{\delta v} &= \frac{\delta X}{\delta v} \frac{\delta f}{\delta X} + \frac{\delta Y}{\delta v} \frac{\delta f}{\delta Y} + \frac{\delta Z}{\delta v} \frac{\delta f}{\delta Z} = S_v * \Delta f \end{aligned} \quad \text{IV.12}$$

$$f \in \{x_i, x_j, y_i, y_j\}$$

Az így adódó J_i, J_j komponensek:

$$J_i = \begin{bmatrix} S_u * \Delta x_i & S_v * \Delta x_i \\ S_u * \Delta y_i & S_v * \Delta y_i \end{bmatrix} \quad \text{IV.13}$$

$$J_j = \begin{bmatrix} S_u * \Delta x_j & S_v * \Delta x_j \\ S_u * \Delta y_j & S_v * \Delta y_j \end{bmatrix} \quad \text{IV.14}$$

Ezeket az összefüggéseket behelyettesítve az **IV.10**-be az $A_{ij} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ komponensei így adódnak:

$$\det(J_i) = (S_u * \Delta x_i)(S_v * \Delta y_i) - (S_v * \Delta x_i)(S_u * \Delta y_i)$$

$$\begin{bmatrix} dx_j \\ dy_j \end{bmatrix} = J_j J_i^{-1} \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} = \frac{1}{\det(J_i)} * \begin{bmatrix} S_u * \Delta x_j & S_v * \Delta x_j \\ S_u * \Delta y_j & S_v * \Delta y_j \end{bmatrix} * \begin{bmatrix} S_v * \Delta y_i & -S_v * \Delta x_i \\ -S_u * \Delta y_i & S_u * \Delta x_i \end{bmatrix} * \begin{bmatrix} dx_i \\ dy_i \end{bmatrix}$$

$\Updownarrow$

$$a_{11} = \frac{1}{\det(J_i)} * [(S_u * \Delta x_j)(S_v * \Delta y_i) - (S_v * \Delta x_j)(S_u * \Delta y_i)]$$

$$a_{12} = \frac{1}{\det(J_i)} * [-(S_u * \Delta x_j)(S_v * \Delta x_i) + (S_v * \Delta x_j)(S_u * \Delta x_i)]$$

$$a_{21} = \frac{1}{\det(J_i)} * [(S_u * \Delta y_j)(S_v * \Delta y_i) - (S_v * \Delta y_j)(S_u * \Delta y_i)]$$

IV.15

$$a_{22} = \frac{1}{\det(J_i)} * [-(S_u * \Delta y_j)(S_v * \Delta x_i) + (S_v * \Delta y_j)(S_u * \Delta x_i)]$$

Ez és az összes komponens felírható a felület normál egységvektorával, így a determináns átalakítása:

$$\begin{aligned} & (S_u * \Delta x_i)(S_v * \Delta y_i) - (S_v * \Delta x_i)(S_u * \Delta y_i) = \\ & \Delta x_i * (S_u S_v) * \Delta y_i - \Delta x_i * (S_v S_u) * \Delta y_i = \\ & -\Delta x_i * (S_v S_u - S_u S_v) * \Delta y_i = \\ & -\Delta x_i * N_{\times} * \Delta y_i = -|N| * |\Delta x_i \mathbf{n}^T \Delta y_i| \end{aligned}$$

IV.16

$$\mathbf{n} \in \mathbb{R}^3$$

a_{11} átalakítása:

$$\begin{aligned} & (S_u * \Delta x_j)(S_v * \Delta y_i) - (S_v * \Delta x_j)(S_u * \Delta y_i) = \\ & \Delta x_j * (S_u S_v) * \Delta y_i - \Delta x_j * (S_v S_u) * \Delta y_i = \\ & -\Delta x_j * (S_v S_u - S_u S_v) * \Delta y_i = \\ & -\Delta x_j * N_{\times} * \Delta y_i = -|N| * |\Delta x_j \mathbf{n}^T \Delta y_i| \end{aligned}$$

IV.17

$$\mathbf{n} \in \mathbb{R}^3$$

a_{12} átalakítása hasonlóan

$$-\Delta x_i * N_x * \Delta x_j = -|N| * |\Delta x_i \mathbf{n}^T \Delta x_j|$$

IV.18

$$\mathbf{n} \in \mathbb{R}^3$$

a_{21} átalakítása hasonlóan

$$-\Delta y_j * N_x * \Delta y_i = -|N| * |\Delta y_j \mathbf{n}^T \Delta y_i|$$

IV.19

$$\mathbf{n} \in \mathbb{R}^3$$

a_{22} átalakítása hasonlóan

$$-\Delta x_i * N_x * \Delta y_j = -|N| * |\Delta x_i \mathbf{n}^T \Delta y_j|$$

IV.20

$$\mathbf{n} \in \mathbb{R}^3$$

Tehát az átalakítások után adódik, hogy

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \frac{|\Delta x_j \mathbf{n}^T \Delta y_i|}{|\Delta x_i \mathbf{n}^T \Delta y_i|} & \frac{|\Delta x_i \mathbf{n}^T \Delta x_j|}{|\Delta x_i \mathbf{n}^T \Delta y_i|} \\ \frac{|\Delta y_j \mathbf{n}^T \Delta y_i|}{|\Delta x_i \mathbf{n}^T \Delta y_i|} & \frac{|\Delta x_i \mathbf{n}^T \Delta y_j|}{|\Delta x_i \mathbf{n}^T \Delta y_i|} \end{bmatrix}$$

IV.21

$$\Updownarrow$$

$$\begin{bmatrix} |\Delta x_j \mathbf{n} \Delta y_i| & |\Delta x_i \mathbf{n} \Delta x_j| \\ |\Delta y_j \mathbf{n} \Delta y_i| & |\Delta x_i \mathbf{n} \Delta y_j| \end{bmatrix} = |\Delta x_i \mathbf{n} \Delta y_i| \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

4.3. Gyorsbecslő

Feltételezések:

1. A_{ij} komponensei ismertek. Ezek az affin transzformációt becslő eljárásból adódnak.
2. A_{ij} mátrix rangja 2.
3. A gradiensek ismertek a vizsgált pontok környezetében.

Kiegészítve ezt azzal a kritériummal, hogy normálvektor hossza egységnyi ($|\mathbf{n}| \equiv 1 (= \sqrt{n_x^2 + n_y^2 + n_z^2})$) ez egy túlhatározott egyenletrendszer lesz. Átrendezve az egyenletet adódik

$$\begin{bmatrix} \mathbf{n}^T(\Delta y_i \times \Delta x_j) & \mathbf{n}^T(\Delta x_j \times \Delta x_i) \\ \mathbf{n}^T(\Delta y_i \times \Delta y_j) & \mathbf{n}^T(\Delta y_j \times \Delta x_i) \end{bmatrix} = \mathbf{n}^T(\Delta y_i \times \Delta x_i) \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{IV.22}$$

A 2. feltételezésből következik, hogy a sorok, illetve oszlopok lineárisan függetlenek egymástól, ezért a sorok, vagy oszlopok komponenseinek egymással való osztásával kiejthetjük a közös szorzót. Ezt a műveletet háromféleképpen tehetjük meg

1.

$$\frac{\mathbf{n}^T(\Delta y_i \times \Delta x_j)}{\mathbf{n}^T(\Delta y_j \times \Delta x_i)} = \frac{a_{11}}{a_{22}} \quad \text{IV.23}$$

$$\frac{\mathbf{n}^T(\Delta y_i \times \Delta x_j)}{\mathbf{n}^T(\Delta y_j \times \Delta x_i)} = \frac{a_{12}}{a_{21}}$$

2.

$$\frac{\mathbf{n}^T(\Delta y_i \times \Delta x_j)}{\mathbf{n}^T(\Delta y_i \times \Delta y_j)} = \frac{a_{11}}{a_{21}} \quad \text{IV.24}$$

$$\frac{\mathbf{n}^T(\Delta x_j \times \Delta x_i)}{\mathbf{n}^T(\Delta y_j \times \Delta x_i)} = \frac{a_{12}}{a_{22}}$$

3.

$$\frac{\mathbf{n}^T(\Delta y_i \times \Delta x_j)}{\mathbf{n}^T(\Delta x_j \times \Delta x_i)} = \frac{a_{11}}{a_{12}} \quad \text{IV.25}$$

$$\frac{\mathbf{n}^T(\Delta y_i \times \Delta y_j)}{\mathbf{n}^T(\Delta y_j \times \Delta x_i)} = \frac{a_{21}}{a_{22}}$$

A következő levezetés mind a három esetben ugyanazon lépésekből áll, ezért csak az 1. esetre írom le:

$$a_{22}\mathbf{n}^T(\Delta y_i \times \Delta x_j) - a_{11}\mathbf{n}^T(\Delta y_j \times \Delta x_i) = 0$$

$$a_{21}\mathbf{n}^T(\Delta y_i \times \Delta x_j) - a_{12}\mathbf{n}^T(\Delta y_j \times \Delta x_i) = 0$$

$\Updownarrow$

$$\mathbf{n}^T[a_{22}(\Delta y_i \times \Delta x_j) - a_{11}(\Delta y_j \times \Delta x_i)] = 0$$

IV.26

$$\mathbf{n}^T [a_{21}(\Delta y_i \times \Delta x_j) - a_{12}(\Delta y_j \times \Delta x_i)] = 0$$

Tehát van két egyenletünk (IV.26), hogy meghatározzuk $\mathbf{n}$ -et, ami merőleges kell legyen erre a két megadott vektorra:

$$\mathbf{p} = a_{22}(\Delta y_i \times \Delta x_j) - a_{11}(\Delta y_j \times \Delta x_i)$$

$$\mathbf{q} = a_{21}(\Delta y_i \times \Delta x_j) - a_{12}(\Delta y_j \times \Delta x_i)$$

↓

$$\mathbf{n} = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p} \times \mathbf{q}|}$$

IV.27

4.3.1. Specializáció perspektív kamerára

Kalibrált perspektív kamera [13] esetén adott a kamera projekciós mátrixa, ami tartalmazza a belső paramétereket, illetve a rotációt, translációt:

$$P^{(i)} = K^{(i)} * (R^{(i)} | T^{(i)}) = \begin{bmatrix} \boldsymbol{\pi}_1^{(i)T} \\ \boldsymbol{\pi}_2^{(i)T} \\ \boldsymbol{\pi}_3^{(i)T} \end{bmatrix} = \begin{bmatrix} p_{11}^{(i)} & p_{12}^{(i)} & p_{13}^{(i)} & p_{14}^{(i)} \\ p_{21}^{(i)} & p_{22}^{(i)} & p_{23}^{(i)} & p_{24}^{(i)} \\ p_{31}^{(i)} & p_{32}^{(i)} & p_{33}^{(i)} & p_{34}^{(i)} \end{bmatrix}$$

IV.28

$$i \in \{1,2\}$$

Egy adott $\mathbf{V}$ vektor homogénkoordinátás alakja $\mathbf{V}' = [X \ Y \ Z \ 1]^T$. A projekció speciális formája ekkor racionális függvény:

$$x_i = \frac{\boldsymbol{\pi}_1^{(i)T} \mathbf{V}'}{\boldsymbol{\pi}_3^{(i)T} \mathbf{V}'} = \frac{p_{11}^{(i)}X + p_{12}^{(i)}Y + p_{13}^{(i)}Z + p_{14}^{(i)}}{p_{31}^{(i)}X + p_{32}^{(i)}Y + p_{33}^{(i)}Z + p_{34}^{(i)}}$$

$$y_i = \frac{\boldsymbol{\pi}_2^{(i)T} \mathbf{V}'}{\boldsymbol{\pi}_3^{(i)T} \mathbf{V}'} = \frac{p_{21}^{(i)}X + p_{22}^{(i)}Y + p_{23}^{(i)}Z + p_{24}^{(i)}}{p_{31}^{(i)}X + p_{32}^{(i)}Y + p_{33}^{(i)}Z + p_{34}^{(i)}}$$

IV.29

$$i \in \{1,2\}$$

Vezessük be a $s_i = \boldsymbol{\pi}_3^{(i)T} \mathbf{V}'$ kifejezést. Ekkor az x képkoordinátára:

$$\frac{\delta x_i}{\delta X} = \frac{1}{s_i} p_{11}^{(i)} - \frac{1}{s_i^2} p_{31}^{(i)} \boldsymbol{\pi}_1^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{11}^{(i)} - p_{31}^{(i)} x_i)$$

IV.30

$$\frac{\delta x_i}{\delta Y} = \frac{1}{s_i} p_{12}^{(i)} - \frac{1}{s_i^2} p_{32}^{(i)} \boldsymbol{\pi}_1^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{12}^{(i)} - p_{32}^{(i)} x_i)$$

$$\frac{\delta x_i}{\delta Z} = \frac{1}{s_i} p_{13}^{(i)} - \frac{1}{s_i^2} p_{33}^{(i)} \boldsymbol{\pi}_1^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{13}^{(i)} - p_{33}^{(i)} x_i)$$

$$i \in \{1, 2\}$$

Az y képkoordinátára:

$$\frac{\delta y_i}{\delta X} = \frac{1}{s_i} p_{21}^{(i)} - \frac{1}{s_i^2} p_{31}^{(i)} \boldsymbol{\pi}_2^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{21}^{(i)} - p_{31}^{(i)} y_i)$$

$$\frac{\delta y_i}{\delta Y} = \frac{1}{s_i} p_{22}^{(i)} - \frac{1}{s_i^2} p_{32}^{(i)} \boldsymbol{\pi}_2^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{22}^{(i)} - p_{32}^{(i)} y_i)$$

$$\frac{\delta y_i}{\delta Z} = \frac{1}{s_i} p_{23}^{(i)} - \frac{1}{s_i^2} p_{33}^{(i)} \boldsymbol{\pi}_2^{(i)T} \mathbf{V}' = \frac{1}{s_i} (p_{23}^{(i)} - p_{33}^{(i)} y_i)$$

$$i \in \{1, 2\}$$

IV.31

Ezek az egyenlőségek (IV.30, IV.31) a lineáris vetítés formulái. A gradiensek csak a projekciós mátrix baloldali 3×3 -as részmátrixát használják. Ezeket az eredményeket felhasználva látszik, hogy a skála faktorok (s_i) kiesnek az egyenletekből. Ezzel a közös skálafaktorral felszorozva használhatjuk a skálázott vektorokat is ($\mathbf{P} = s_1 s_2 \mathbf{p}$ $\mathbf{Q} = s_1 s_2 \mathbf{q}$) a normálvektor meghatározásához. Ekkor az alábbi formában adódik a normálvektor:

$$\mathbf{P} = [a_{22} s_1 s_2 (\Delta \mathbf{y}_1 \times \Delta \mathbf{x}_2) - a_{11} s_1 s_2 (\Delta \mathbf{y}_2 \times \Delta \mathbf{x}_1)]$$

$$\mathbf{Q} = [a_{21} s_1 s_2 (\Delta \mathbf{x}_2 \times \Delta \mathbf{x}_1) - a_{12} s_1 s_2 (\Delta \mathbf{y}_1 \times \Delta \mathbf{y}_2)]$$

$\Downarrow$

IV.32

$$\mathbf{n} = \frac{\mathbf{P} \times \mathbf{Q}}{|\mathbf{P} \times \mathbf{Q}|}$$

A teljesség kedvéért:

$$\mathbf{P} = a_{22} \begin{bmatrix} p_{21}^1 - p_{31}^1 y_1 \\ p_{22}^1 - p_{32}^1 y_1 \\ p_{23}^1 - p_{33}^1 y_1 \end{bmatrix} \times \begin{bmatrix} p_{11}^2 - p_{31}^2 x_2 \\ p_{12}^2 - p_{32}^2 x_2 \\ p_{13}^2 - p_{33}^2 x_2 \end{bmatrix} - a_{11} \begin{bmatrix} p_{21}^2 - p_{31}^2 y_2 \\ p_{22}^2 - p_{32}^2 y_2 \\ p_{23}^2 - p_{33}^2 y_2 \end{bmatrix} \times \begin{bmatrix} p_{11}^1 - p_{31}^1 x_1 \\ p_{12}^1 - p_{32}^1 x_1 \\ p_{13}^1 - p_{33}^1 x_1 \end{bmatrix}$$

IV.33

$$\mathbf{Q} = a_{21} \begin{bmatrix} p_{11}^2 - p_{31}^2 x_2 \\ p_{12}^2 - p_{32}^2 x_2 \\ p_{13}^2 - p_{33}^2 x_2 \end{bmatrix} \times \begin{bmatrix} p_{11}^1 - p_{31}^1 x_1 \\ p_{12}^1 - p_{32}^1 x_1 \\ p_{13}^1 - p_{33}^1 x_1 \end{bmatrix} - a_{12} \begin{bmatrix} p_{21}^1 - p_{31}^1 y_1 \\ p_{22}^1 - p_{32}^1 y_1 \\ p_{23}^1 - p_{33}^1 y_1 \end{bmatrix} \times \begin{bmatrix} p_{11}^2 - p_{31}^2 y_2 \\ p_{12}^2 - p_{32}^2 y_2 \\ p_{13}^2 - p_{33}^2 y_2 \end{bmatrix}$$

Itt $i \in \{1,2\}$ esetben az (x_i, y_i) képpontok az affin régiók centroidjait jelölik.

4.4. Pontosítás

Ettől a résztől kezdődik az én munkám. Továbbhaladva az előző fejezetben ismertetett eljárással a normálvektorok becslésének két módja válik nyilvánvalóvá:

1. Vektor becslése ismert mélység (s_i) segítségével (3.5-ös fejezet).
2. Vektor becslése ismeretlen mélység (s_i) segítségével (3.6-os fejezet).

4.5. Ismert mélység (s_i) esetén

Az $s_i = \pi_3^{(i)} \mathbf{V}' = p_{31}^{(i)} X + p_{32}^{(i)} Y + p_{33}^{(i)} Z + p_{34}^{(i)}$ projekciós mélységi értéket abban az esetben tudjuk kiszámítani, ha ismert a projekciós mátrix, illetve a pontok háromdimenziós koordinátái is. Két kép - sztereo probléma – esetén a megfeleltetett pontok háromdimenziós koordinátáinak kiszámítására megfelelő a képfeldolgozásban ismert Hartley-Sturm triangulációs eljárás [4], ami legkisebb négyzetes értelemben optimális megoldást ad erre a problémára.

Azt feltételezzük, hogy az affin paraméterek zajosak, de a projekciós mátrixokat pontosan ismerjük, akkor az alapegyenletekre az optimális becslőt olyan módon állítjuk elő, hogy a normálvektorokra nézve minimalizáljuk az affin paraméterekre kiszámított hibát.

$$\begin{bmatrix} \mathbf{n}^T * (\Delta y_1 \times \Delta x_2) & \mathbf{n}^T * (\Delta x_2 \times \Delta x_1) \\ \mathbf{n}^T * (\Delta y_1 \times \Delta y_2) & \mathbf{n}^T * (\Delta y_2 \times \Delta x_1) \end{bmatrix} = |\Delta x_1 \mathbf{n}^T \Delta y_1| \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$\Leftrightarrow$

$$\begin{bmatrix} \mathbf{n}^T * (\Delta y_1 \times \Delta x_2) & \mathbf{n}^T * (\Delta x_2 \times \Delta x_1) \\ \mathbf{n}^T * (\Delta y_1 \times \Delta y_2) & \mathbf{n}^T * (\Delta y_2 \times \Delta x_1) \end{bmatrix} = \mathbf{n}^T * (\Delta y_1 \times \Delta x_1) \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

IV.34

$\Leftrightarrow$

$$\frac{1}{\mathbf{n} * (\Delta y_1 \times \Delta x_1)} \begin{bmatrix} \mathbf{n}^T * (\Delta y_1 \times \Delta x_2) & \mathbf{n}^T * (\Delta x_2 \times \Delta x_1) \\ \mathbf{n}^T * (\Delta y_1 \times \Delta y_2) & \mathbf{n}^T * (\Delta y_2 \times \Delta x_1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Ez (III.1) a mátrixokat tartalmazó egyenlet megfelel ekkor négy egyenletnek, amiket az alábbi alakban tudunk felírni:

$$\frac{\mathbf{n}^T * p_k}{\mathbf{n}^T * p_5} = a_k$$

IV.35

$$k \in \{1,2,3,4\}$$

ahol

$$a_1 = a_{11}, a_2 = a_{12}, a_3 = a_{21}, a_4 = a_{22}$$

$$p_1 = \Delta y_1 \times \Delta x_2, p_2 = \Delta x_2 \times \Delta x_1, p_3 = \Delta y_1 \times \Delta y_2, p_4 = \Delta y_2 \times \Delta x_1$$

Tehát az optimális becslő feladata minimalizálni a következő kifejezést,

$$\arg \min_{\mathbf{n}} \sum_{k=1}^4 \left(\frac{\mathbf{n}^T * p_k}{\mathbf{n}^T * p_5} - a_k \right)^2$$

IV.36

ahol csupán egyetlen ismeretlen mennyiségünk van: az $\mathbf{n}$ vektor. Fontos megjegyezni, hogy ez a kifejezés $\mathbf{n}$ hosszára érzéketlen, ezért erre megkötést kell tenni.

A következő 4.5.1-es pontnál számolt megoldásnál ebből az esetből kiindulva $\alpha = 1$ áll fent. (Későbbi problémánál is visszahivatkozunk 4.5.1-es pontra, ezért kerül felírásra a levezetés általánosabb, α -t használó formában).

4.5.1. A feladat megoldása

Tehát adott az alábbi feladat:

$$\arg \min_{\mathbf{n}} \sum_{k=1}^4 \left(\frac{\mathbf{n}^T * p_k}{\alpha \mathbf{n}^T * p_5} - a_k \right)^2$$

IV.37

A kifejezést $\mathbf{n}$ szerint kellene deriválnunk, azonban figyelniük kellene arra, hogy $\mathbf{n}^T \mathbf{n}$ hossza egységnyi legyen, ezt általában Lagrange-multiplikátor segítségével szokták megtenni, de ebben az esetben nagyon elbonyolódna az egyenleteink. Ezzel szemben inkább tételezzük fel, hogy $\mathbf{n}$ egyik komponense 1 ($n_i = 1, i \in \{1,2,3\}$). Ez értelemszerűen három esetet vet fel, de az alábbi eljárás

mindegyikre ugyanúgy adódik, szóval azt csak az $n_3 = 1$ -re írjuk le. A három különböző eset lefedése azért fontos, mert adódhat olyan eset, amikor a kiválasztott koordináta értéke nem lehet 1 (pl. tegyük fel, hogy $n_3 = 1$ és adódik, hogy $\mathbf{n} = [1, 0, 0]^T$), ezt azonban az összes eset vizsgálatával könnyen lehet detektálni és kiszűrni.

Tehát legyen $n_3 = 1$, és a maradék két koordinátát pedig nevezzük el $\mathbf{m}$ vektornak. Tehát $\mathbf{n} = [\mathbf{m}^T, 1]^T$. A $\mathbf{p}_k$ vektorokat is ketté kéne bontani, úgy hogy $\mathbf{p}_k = [\mathbf{q}_k, p_{k,3}]$, ahol $\mathbf{q}_k$ kétdimenziós vektor, illetve a $p_{k,3}$ pedig a $\mathbf{p}_k$ vektor harmadik koordinátája. Így adódik behelyettesítéssel (IV.35-osba) az alábbi egyenlet:

$$\arg \min_{\mathbf{m}} \sum_{k=1}^4 \left(\frac{\mathbf{m}^T * \mathbf{q}_k + p_{k,3}}{\alpha \mathbf{m}^T * \mathbf{q}_5 + \alpha p_{5,3}} - a_k \right)^2 \quad \text{IV.38}$$

Ebben az esetben kinyerhetőek a szélsőértékek az egyenletek $\mathbf{m}$ szerinti deriválásával, ha megkeressük a derivált zérushelyeit:

$$f(\mathbf{m}) = \sum_{k=1}^4 \left(\frac{\mathbf{m}^T \mathbf{q}_k + p_{k,3}}{\alpha \mathbf{m}^T \mathbf{q}_5 + \alpha p_{5,3}} - a_k \right)^2 \quad \text{IV.39}$$

$$\frac{\delta f}{\delta \mathbf{m}} = 2 \sum_{k=1}^4 \left(\frac{\mathbf{m}^T \mathbf{q}_k + p_{k,3}}{\alpha \mathbf{m}^T \mathbf{q}_5 + \alpha p_{5,3}} - a_k \right) \left(\alpha \frac{(\mathbf{m}^T \mathbf{q}_5 + p_{5,3}) \mathbf{p}_k - (\mathbf{m}^T \mathbf{q}_k + p_{k,3}) \mathbf{p}_5}{(\alpha \mathbf{m}^T \mathbf{q}_5 + \alpha p_{5,3})^2} \right) = 0$$

A bal oldali tagot hozzuk közös nevezőre, így csak a számlálónak kell majd nullának lennie.

$$\sum_{k=1}^4 \left((\mathbf{m}^T \mathbf{q}_k + p_{k,3}) - (\alpha \mathbf{m}^T \mathbf{q}_5 + \alpha p_{5,3}) a_k \right) \left((\mathbf{m}^T \mathbf{q}_5 + p_{5,3}) \mathbf{p}_k - (\mathbf{m}^T \mathbf{q}_k + p_{k,3}) \mathbf{p}_5 \right) = 0$$

$\Leftrightarrow$

$$\sum_{k=1}^4 (\mathbf{m}^T \mathbf{q}_k + p_{k,3} - \alpha \mathbf{m}^T \mathbf{q}_5 a_k - \alpha p_{5,3} a_k) \left((\mathbf{m}^T \mathbf{q}_5 \mathbf{p}_k + p_{5,3} \mathbf{p}_k) - (\mathbf{m}^T \mathbf{q}_k + p_{k,3}) \mathbf{p}_5 \right) = 0$$

$\Leftrightarrow$

$$\sum_{k=1}^4 (\mathbf{m}^T (q_k - \alpha q_5 a_k) + p_{k,3} - \alpha p_{5,3} a_k) (\mathbf{m}^T (q_5 p_k - q_k p_5) + p_{5,3} p_k - p_{k,3} p_5) = 0$$

Két egyenletre bontva ezt a vektorokat tartalmazó egyenletet:

$$\sum_{k=1}^4 (\mathbf{m}^T (q_k - \alpha q_5 a_k) + p_{k,3} - \alpha p_{5,3} a_k) \begin{bmatrix} \mathbf{m}^T (q_5 p_{k,1} - q_k p_{5,1}) + p_{5,3} p_{k,1} - p_{k,3} p_{5,1} \\ \mathbf{m}^T (q_5 p_{k,2} - q_k p_{5,2}) + p_{5,3} p_{k,2} - p_{k,3} p_{5,2} \end{bmatrix} = 0$$

$\Updownarrow$

$$\sum_{k=1}^4 (\mathbf{m}^T \begin{bmatrix} p_{k,1} - \alpha p_{5,1} a_k \\ p_{k,2} - \alpha p_{5,2} a_k \end{bmatrix} + p_{k,3} - \alpha p_{5,3} a_k) \begin{bmatrix} \mathbf{m}^T \begin{bmatrix} p_{5,1} p_{k,1} - p_{k,1} p_{5,1} \\ p_{5,2} p_{k,1} - p_{k,2} p_{5,1} \end{bmatrix} + p_{5,3} p_{k,1} - p_{k,3} p_{5,1} \\ \mathbf{m}^T \begin{bmatrix} p_{5,1} p_{k,2} - p_{k,1} p_{5,2} \\ p_{5,2} p_{k,2} - p_{k,2} p_{5,2} \end{bmatrix} + p_{5,3} p_{k,2} - p_{k,3} p_{5,2} \end{bmatrix} = 0$$

$\Updownarrow$

$$\sum_{k=1}^4 (\mathbf{m}^T \begin{bmatrix} p_{k,1} - \alpha p_{5,1} a_k \\ p_{k,2} - \alpha p_{5,2} a_k \end{bmatrix} + p_{k,3} - \alpha p_{5,3} a_k) \begin{bmatrix} \mathbf{m}^T \begin{bmatrix} 0 \\ p_{5,2} p_{k,1} - p_{k,2} p_{5,1} \end{bmatrix} + p_{5,3} p_{k,1} - p_{k,3} p_{5,1} \\ \mathbf{m}^T \begin{bmatrix} p_{5,1} p_{k,2} - p_{k,1} p_{5,2} \\ 0 \end{bmatrix} + p_{5,3} p_{k,2} - p_{k,3} p_{5,2} \end{bmatrix} = 0 \quad \text{IV.40}$$

Ekkor amennyiben bevezetjük az $\mathbf{m} = [x, y]^T$ jelölést a két egyenlet (IV.40) felírható az alábbi formában:

$$\sum_{k=1}^4 (\varphi_1^{(1)} x + \varphi_2^{(1)} y + \varphi_3^{(1)}) \begin{bmatrix} \varphi_1^{(2)} x + \varphi_2^{(2)} y + \varphi_3^{(2)} \\ \varphi_1^{(3)} x + \varphi_2^{(3)} y + \varphi_3^{(3)} \end{bmatrix} = 0 \quad \text{IV.41}$$

ahol

$$\varphi_{1,k}^{(1)} = p_{k,1} - \alpha p_{5,1} a_k$$

$$\varphi_{2,k}^{(1)} = p_{k,2} - \alpha p_{5,2} a_k$$

$$\varphi_{3,k}^{(1)} = p_{k,3} - \alpha p_{5,3} a_k$$

$$\varphi_{1,k}^{(2)} = 0$$

$$\varphi_{2,k}^{(2)} = p_{5,2} p_{k,1} - p_{k,2} p_{5,1}$$

$$\varphi_{3,k}^{(2)} = p_{5,3} p_{k,1} - p_{k,3} p_{5,1}$$

$$\varphi_{1,k}^{(3)} = p_{5,1}p_{k,2} - p_{k,1}p_{5,2}$$

$$\varphi_{2,k}^{(3)} = 0$$

$$\varphi_{3,k}^{(3)} = p_{5,3}p_{k,2} - p_{k,3}p_{5,2}$$

Ebből (IV.41) pedig az egyenletek felbontásával adódik két kvadratikus görbe

$$\sum_{k=1}^4 (\varphi_{1,k}^{(1)}x + \varphi_{2,k}^{(1)}y + \varphi_{3,k}^{(1)}) \begin{bmatrix} \varphi_{1,k}^{(2)}x + \varphi_{2,k}^{(2)}y + \varphi_{3,k}^{(2)} \\ \varphi_{1,k}^{(3)}x + \varphi_{2,k}^{(3)}y + \varphi_{3,k}^{(3)} \end{bmatrix} = 0$$

$$\Updownarrow$$

$$\sum_{k=1}^4 \begin{bmatrix} (\varphi_{1,k}^{(1)}x + \varphi_{2,k}^{(1)}y + \varphi_{3,k}^{(1)})(\varphi_{1,k}^{(2)}x + \varphi_{2,k}^{(2)}y + \varphi_{3,k}^{(2)}) \\ (\varphi_{1,k}^{(1)}x + \varphi_{2,k}^{(1)}y + \varphi_{3,k}^{(1)})(\varphi_{1,k}^{(3)}x + \varphi_{2,k}^{(3)}y + \varphi_{3,k}^{(3)}) \end{bmatrix} = 0$$

$$\Updownarrow$$

$$\sum_{k=1}^4 (A_k^l x^2 + B_k^l y^2 + C_k^l xy + D_k^l x + E_k^l y + F_k^l) = 0$$

IV.42

ahol

$$A_k^l = \varphi_{1,k}^{(1)} \varphi_{1,k}^{(l+1)}$$

$$B_k^l = \varphi_{2,k}^{(1)} \varphi_{2,k}^{(l+1)}$$

$$C_k^l = \varphi_{1,k}^{(l+1)} \varphi_{2,k}^{(1)} + \varphi_{2,k}^{(l+1)} \varphi_{1,k}^{(1)}$$

$$D_k^l = \varphi_{1,k}^{(l+1)} \varphi_{3,k}^{(1)} + \varphi_{3,k}^{(l+1)} \varphi_{1,k}^{(1)}$$

$$E_k^l = \varphi_{2,k}^{(l+1)} \varphi_{3,k}^{(1)} + \varphi_{3,k}^{(l+1)} \varphi_{2,k}^{(1)}$$

$$F_k^l = \varphi_{3,k}^{(l+1)} \varphi_{3,k}^{(1)}$$

$l \in \{1,2\}$ – a görbe sorszáma

4.5.2. A két kvadratikus görbe metszéspontja (Optimális becslő)

$$\sum_{k=1}^4 (A_k^l x^2 + B_k^l y^2 + C_k^l xy + D_k^l x + E_k^l y + F_k^l) = 0, (l \in \{1,2\})$$

$\Leftrightarrow$

$$\sum_{k=1}^4 A_k^l x^2 + \sum_{k=1}^4 B_k^l y^2 + \sum_{k=1}^4 C_k^l xy + \sum_{k=1}^4 D_k^l x + \sum_{k=1}^4 E_k^l y + \sum_{k=1}^4 F_k^l = 0, (l \in \{1,2\}) \quad \text{IV.43}$$

Így ez a speciális feladat két egyenletből áll:

$$0 \cdot x^2 + B^{(1)} y^2 + C^{(1)} xy + D^{(1)} x + E^{(1)} y + F^{(1)} = 0 \quad \text{IV.44}$$

$$A^{(2)} x^2 + 0 \cdot y^2 + C^{(2)} xy + D^{(2)} x + E^{(2)} y + F^{(2)} = 0 \quad \text{IV.45}$$

Fejezzük ki a második egyenletből (IV.45) y -t:

$$y = -\frac{A^{(2)} x^2 + D^{(2)} x + F^{(2)}}{E^{(2)} + C^{(2)} x}$$

Ezt visszahelyettesítve az első egyenletbe (IV.44)

$$\begin{aligned} B^{(1)} \left(\frac{A^{(2)} x^2 + D^{(2)} x + F^{(2)}}{E^{(2)} + C^{(2)} x} \right)^2 - C^{(1)} x \frac{A^{(2)} x^2 + D^{(2)} x + F^{(2)}}{E^{(2)} + C^{(2)} x} + D^{(1)} x \\ - E^{(1)} \frac{A^{(2)} x^2 + D^{(2)} x + F^{(2)}}{E^{(2)} + C^{(2)} x} + F^{(1)} = 0 \end{aligned} \quad \text{IV.46}$$

Szorozzuk fel az egyenletet (IV.46) $(E^{(2)} + C^{(2)} x)^2$ -el

$$\begin{aligned} B^{(1)} (A^{(2)} x^2 + D^{(2)} x + F^{(2)})^2 - C^{(1)} x (A^{(2)} x^2 + D^{(2)} x + F^{(2)}) (E^{(2)} + C^{(2)} x) + \\ + D^{(1)} x (E^{(2)} + C^{(2)} x)^2 - E^{(1)} (A^{(2)} x^2 + D^{(2)} x + F^{(2)}) (E^{(2)} + C^{(2)} x) + \\ + F^{(1)} (E^{(2)} + C^{(2)} x)^2 = 0 \end{aligned} \quad \text{IV.47}$$

Ekkor x -re nézve ez egy negyedfokú egyenlet (III.47). Az együtthatók pedig:

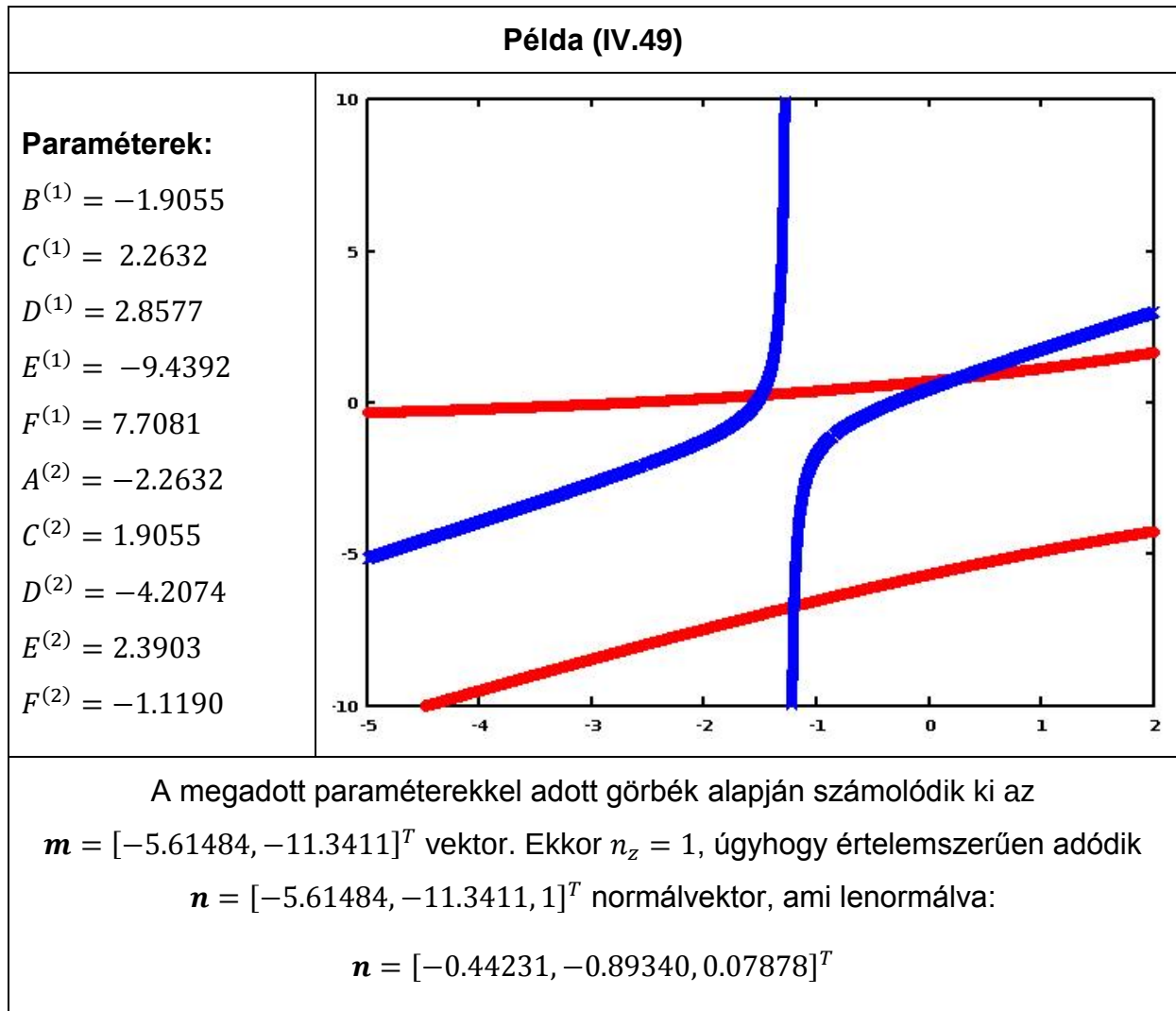
$$\begin{aligned} x^4: & B^{(1)} A^{(2)} A^{(2)} + C^{(1)} C^{(2)} A^{(2)} \\ x^3: & 2B^{(1)} A^{(2)} D^{(2)} - C^{(1)} A^{(2)} E^{(2)} - C^{(1)} D^{(2)} C^{(2)} + D^{(1)} C^{(2)} C^{(2)} - 2E^{(1)} A^{(2)} C^{(2)} \\ x^2: & B^{(1)} D^{(2)} D^{(2)} - C^{(1)} D^{(2)} E^{(2)} - C^{(1)} F^{(2)} C^{(2)} + 2D^{(1)} C^{(2)} E^{(2)} - E^{(1)} A^{(2)} E^{(2)} - \end{aligned} \quad \text{IV.48}$$

$$-E^{(1)}D^{(2)}C^{(2)} + F^{(1)}C^{(2)}C^{(2)}$$

$$x^1: 2B^{(1)}D^{(2)}F^{(2)} - C^{(1)}F^{(2)}E^{(2)} + D^{(1)}E^{(2)}E^{(2)} - E^{(1)}D^{(2)}E^{(2)} - \\ -E^{(1)}F^{(2)}C^{(2)} + 2F^{(1)}E^{(2)}C^{(2)}$$

$$x^0: B^{(1)}F^{(2)}F^{(2)} - E^{(1)}F^{(2)}E^{(2)} + F^{(1)}E^{(2)}E^{(2)}$$

Lentebb egy példa bemenetre meghatározott metszéspontok és a kiértékelt normálvektor látható (IV.49)



Megjegyzés:

1. Fontos megjegyezni, hogy a $E^{(2)} + C^{(2)}x = 0 \Leftrightarrow x = -\frac{E^{(2)}}{C^{(2)}}$ esetet le kell kezelni. Ebben az esetben az első egyenlet teljesen független y -től, abból x meghatározható.

2. A gyökök bármely lineáris algebrai programcsomag segítségével könnyen számolhatóak (pl. Matlab, Octave, stb.).

4.5.3. Közelítő megoldás lineáris módszerrel

A közelítő eljárás azon az alapon nyugszik, hogy ha az adott egyenletünket beszorozzuk a nevezővel (np_5), akkor lineárisrá válik a probléma:

$$\begin{aligned} \arg \min_n \sum_{k=1}^4 \left(\frac{np_k}{np_5} - a_k \right)^2 \\ \Downarrow \\ \arg \min_n \sum_{k=1}^4 (np_k - a_k np_5)^2 \\ \Downarrow \\ \arg \min_n \sum_{k=1}^4 (n(p_k - a_k p_5))^2 \end{aligned}$$

Ez a probléma felírható $An = 0$ alakban, ahol

$$A = \begin{bmatrix} p_1 - a_1 p_5 \\ p_2 - a_2 p_5 \\ p_3 - a_3 p_5 \\ p_4 - a_4 p_5 \end{bmatrix}$$

Ekkor

$$\begin{aligned} \arg \min_n \sum_{k=1}^4 (n(p_k - a_k p_5))^2 \\ \Downarrow \\ \arg \min_n \sum_{k=1}^4 (An)^2 \\ \Downarrow \end{aligned}$$

$$\arg \min_{\mathbf{n}} \sum_{k=1}^4 \mathbf{n}^T A^T A \mathbf{n}$$

Ekkor a lineáris algebrából ismert összefüggés alapján az egyenlet optimális megoldása a $A^T A$ mátrix legkisebb sajátértékéhez tartozó sajátvektora [14], amennyiben azt a megkötést is figyelembe vesszük, hogy $\mathbf{n}$ vektor hossza egységnyi.

4.6. Ismeretlen mélység (s_i) esetén

Ebben az esetben az előző fejezetekben felvázolt alapegyenletek továbbra is érvényes, ellenben a gradiensek pontos értéke ismeretlen – nem tudjuk meghatározni a skálázás mértékét. Értékük egyetlen skálázás erejéig ismert, a gradiensek elemeit csak közelítjük:

$$\frac{\delta x_i}{\delta X} \leftarrow (p_{11}^{(i)} - p_{31}^{(i)} x_i)$$

$$\frac{\delta x_i}{\delta Y} \leftarrow (p_{12}^{(i)} - p_{32}^{(i)} x_i)$$

$$\frac{\delta x_i}{\delta Z} \leftarrow (p_{13}^{(i)} - p_{33}^{(i)} x_i)$$

$$\frac{\delta y_i}{\delta X} \leftarrow (p_{21}^{(i)} - p_{31}^{(i)} y_i)$$

$$\frac{\delta y_i}{\delta Y} \leftarrow (p_{22}^{(i)} - p_{32}^{(i)} y_i)$$

$$\frac{\delta y_i}{\delta Z} \leftarrow (p_{23}^{(i)} - p_{33}^{(i)} y_i)$$

$$i = 1, 2$$

A pontosan meghatározott értékek, és a közelítés között a különbség a skálával (s_i) való osztás hiánya. Az alapegyenletek a közelített értékeket felhasználva a következőképpen változnak meg:

$$\begin{bmatrix} \frac{1}{s_1 s_2} * \mathbf{n} * (\Delta y_1 \times \Delta x_2) & \frac{1}{s_1 s_2} * \mathbf{n} * (\Delta x_2 \times \Delta x_1) \\ \frac{1}{s_1 s_2} * \mathbf{n} * (\Delta y_1 \times \Delta y_2) & \frac{1}{s_1 s_2} * \mathbf{n} * (\Delta y_2 \times \Delta x_1) \end{bmatrix} = \frac{1}{s_1^2} * \mathbf{n} * (\Delta y_1 \times \Delta x_1) \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Ekkor felszorozva a kifejezést $s_1 s_2$ -vel, illetve bevezetve az $\alpha = \frac{s_2}{s_1}$ kifejezést az egyenlet erre az alakra hozható:

$$\begin{bmatrix} \mathbf{n} * (\Delta y_1 \times \Delta x_2) & \mathbf{n} * (\Delta x_2 \times \Delta x_1) \\ \mathbf{n} * (\Delta y_1 \times \Delta y_2) & \mathbf{n} * (\Delta y_2 \times \Delta x_1) \end{bmatrix} = \alpha * \mathbf{n} * (\Delta y_1 \times \Delta x_1) \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

4.6.1. Közelítő megoldás lineáris módszerrel

A fenti mátrix négy elemére felírhatjuk a következő egyenleteket:

$$\mathbf{n}(\Delta y_1 \times \Delta x_2 - \alpha a_{11}(\Delta y_1 \times \Delta x_1)) = 0$$

$$\mathbf{n}(\Delta x_2 \times \Delta x_1 - \alpha a_{12}(\Delta y_1 \times \Delta x_1)) = 0$$

$$\mathbf{n}(\Delta y_1 \times \Delta y_2 - \alpha a_{21}(\Delta y_1 \times \Delta x_1)) = 0$$

$$\mathbf{n}(\Delta y_2 \times \Delta x_1 - \alpha a_{22}(\Delta y_1 \times \Delta x_1)) = 0$$

Ez a négy egyenlet lényegében négy ismeretlent tartalmaz: $\mathbf{n}$ három komponensét, és α -t. Ekkor az ismeretlenek x vektorral kifejezve egy homogén lineáris egyenletrendszer kapunk, ami azzal a feltétellel, hogy $\mathbf{n}$ hossza egységnyi egy a lineáris algebrából jól ismert összefüggéshez vezet vissza – így megoldható. $Ax = 0$ homogén lineáris egyenletrendszer optimális megoldása $x^T x = 1$ [14] esetben az A mátrix legkisebb sajátértékéhez tartozó sajátvektora. Esetünkben az A együttthatómátrix meghatározható, mint

$$A = \begin{bmatrix} (\Delta y_1 \times \Delta x_2)^T & -a_{11} \\ (\Delta x_2 \times \Delta x_1)^T & -a_{12} \\ (\Delta y_1 \times \Delta y_2)^T & -a_{21} \\ (\Delta y_2 \times \Delta x_1)^T & -a_{22} \end{bmatrix}$$

Az ismeretleneket tartalmazó x vektor pedig mint

$$\mathbf{x} = \begin{bmatrix} \alpha \mathbf{n}^T (\Delta \mathbf{y}_1 \times \Delta \mathbf{x}_1) \end{bmatrix}$$

Ez esetünkben túlhatározott probléma lesz, hiszen az $\mathbf{n}$ felületi normális meghatározásához elegendő lenne A -nak 3×4 -nek lennie, ezzel szemben a konkrét méret 4×4 . Így a 4 dimenziós $\mathbf{x}$ vektor első három koordinátája megadja a normálvektor komponenseit, míg a normálvektor ismeretében az utolsó egyenletből kiszámolható az $\alpha = \frac{s_2}{s_1}$ kifejezés.

4.6.2. Konvergálás a (lokálisan) optimális megoldás felé (Alternálás)

Az ismert s_i -ket felhasználó módszerhez hasonlóan itt is felírható egy legkisebb négyzetes értelemben optimális megoldást adó feladat:

$$\arg \min_{\mathbf{n}} \sum_{k=1}^4 \left(\frac{\mathbf{n} * p_k}{\alpha \mathbf{n} * p_5} - a_k \right)^2$$

A megoldásra nem találtunk egyelőre zárt alakot. Ezért a legközelebbi megoldást úgynevezett alternáló megoldással keressük meg. Az alternálás azt jelenti, hogy konvergenciáig két lépést ismételünk:

1. Az első lépésben finomítjuk α értéket, míg $\mathbf{n}$ értékét nem változtatjuk.
2. Pontosítjuk $\mathbf{n}$ értékét, míg α értékét nem változtatjuk.
3. Folytatjuk az 1. ponttól, ha még nem értük el a konvergenciát.

Mindkét esetben megadható az optimális érték.

α optimális becslése: A minimalizálandó kifejezés ekkor $\frac{1}{\alpha}$ szerint egy egyismeretlenes inhomogén egyenletrendszer, tehát felírható $A \frac{1}{\alpha} = b$ alakban, ahol

$$A = \begin{bmatrix} \mathbf{n}^T p_1 \\ \mathbf{n}^T p_5 \\ \mathbf{n}^T p_2 \\ \mathbf{n}^T p_5 \\ \mathbf{n}^T p_3 \\ \mathbf{n}^T p_5 \\ \mathbf{n}^T p_4 \\ \mathbf{n}^T p_5 \end{bmatrix}, b = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

Mivel túlhatározott inhomogén egyenletrendszerrel beszélünk, az optimális megoldást az úgynevezett Moore-Penrose pszeudoinverzrel ($A^+ = (A^T A)^{-1} A^T$) számolhatjuk [14]:

$$\frac{1}{\alpha} = (A^T A)^{-1} A^T b = \frac{1}{\sum_k (\frac{\mathbf{n}^T p_k}{\mathbf{n}^T p_5})^2} \sum_k \frac{\mathbf{n}^T p_k}{\mathbf{n}^T p_5} a_k = \frac{\mathbf{n}^T p_5}{\sum_k (\mathbf{n}^T p_k)^2} \sum_k \mathbf{n}^T p_k a_k$$

$\Downarrow$

$$\alpha = \frac{\sum_k (\mathbf{n}^T p_k)^2}{\mathbf{n}^T p_5 \sum_k \mathbf{n}^T p_k a_k}$$

n értékének optimális becslése: A megoldás menete 4.2. pontban ismertetett módon zajlik.

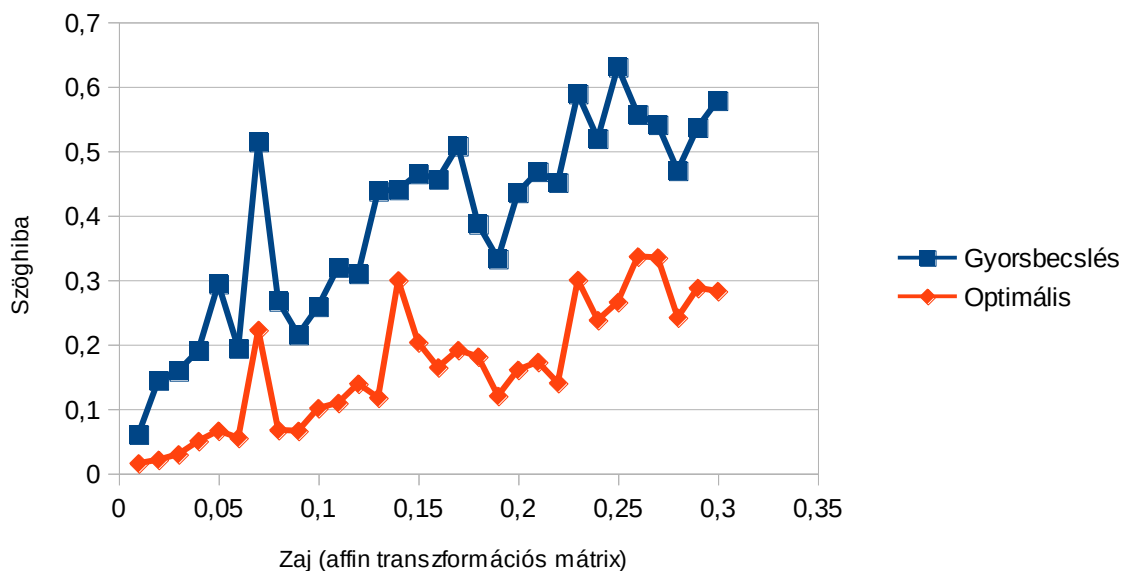
5. Kiértékelés

5.1. Szintetikus tesztek

A megoldott feladatok szintetikus tesztekre kiértékelésre kerültek. Sajnos egy összefoglaló diagram – ami megjeleníti az összes módszert egyszerre – áttekinthetetlen lenne, így csak a lényeges különbségeket hordozó párok lesznek az alábbiakban összehasonlítva.

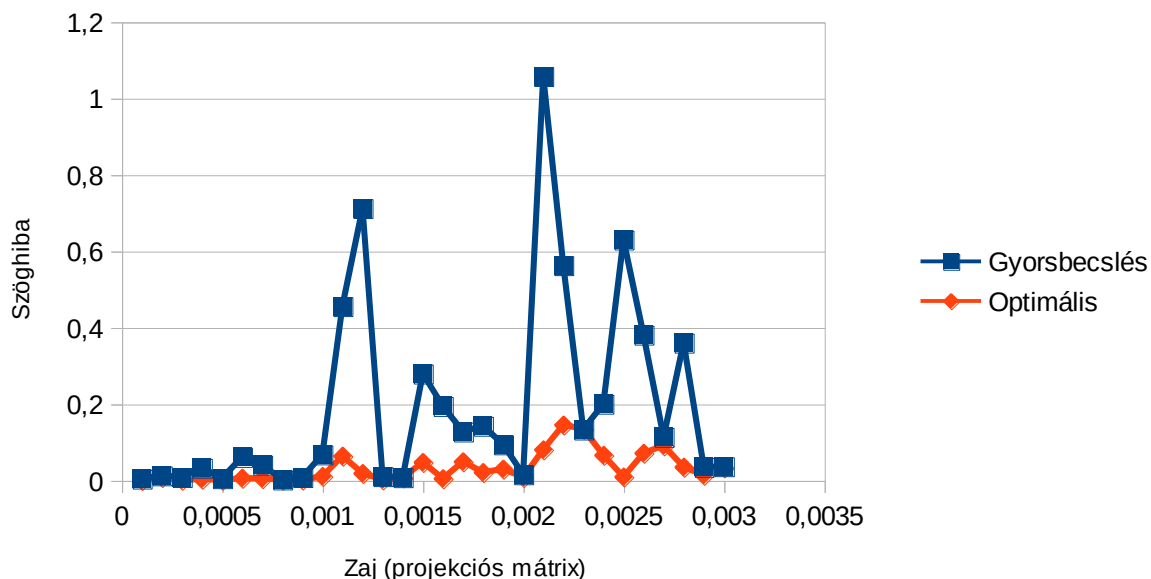
5.1.1. Gyorsbecslés – Optimális becslő

A kiértékelések során az az eredmény látszott egyértelműen, hogy az Optimális, illetve a Gyorsbecslő közül az Optimális minden esetben legalább másfélszer jobb eredményt ad egyre zajosabb bemeneti paraméterekre vizsgálva (V.1, V.2) az eredményt. A hibát mint a becsült normálvektor az eredetihez mért szöghibáját tekintjük. Tehát az Optimális becslő sokkal robusztusabb – kevésbé érzékeny a hibára.



V.1 Gyorsbecslés – Optimális becslés viselkedése affin hiba növelésére.

A **V.1** diagram jól szemlélteti, hogy egyre zajosabb affin transzformációval számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan az Optimális megoldás minden esetben kisebb hibát ad.

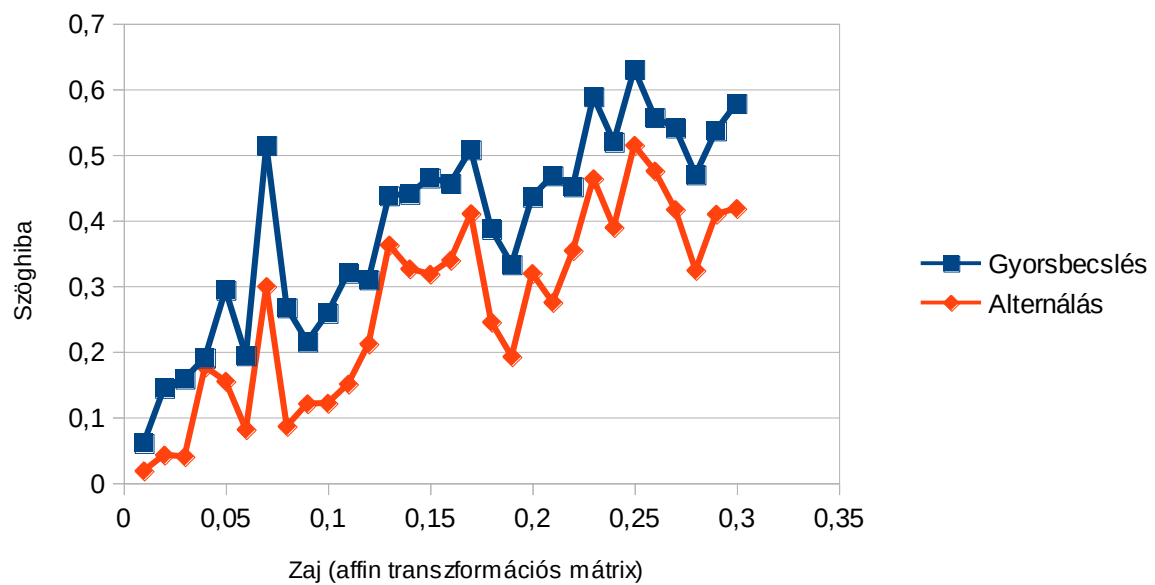


V.2 Gyorsbecslés – Optimális becslés viselkedése projekciós hiba növelésére.

A **V.2** diagram jól szemlélteti, hogy egyre zajosabb projekciós mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan az Optimális megoldás minden esetben kisebb hibát ad.

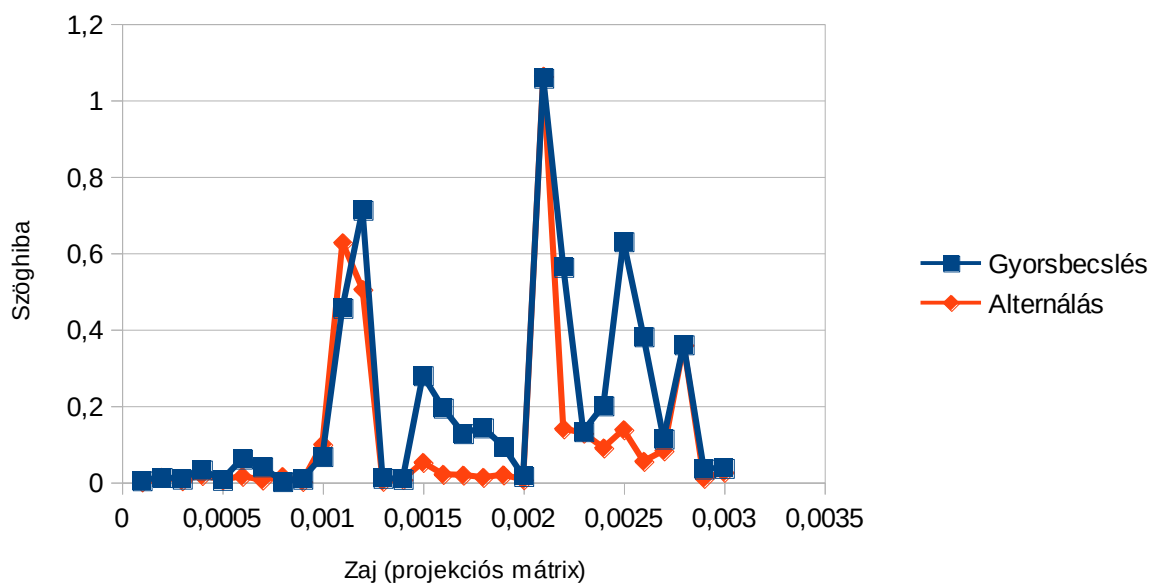
5.1.2. Gyorsbecslés – Alternálás

A szintetikus tesztek kiértékelése során egyértelműen látszott, hogy az Alternálás az esetek többségében jobb eredményt ad, mint a Gyorsbecslő. Az Alternálás az affin transzformációs mátrix hibájára közel fele olyan érzékeny (**V.3**), azonban a projekciós mátrixban fellépő zajra közel ugyanúgy reagál a két módszer (**V.4**).



V.3. Gyorsbecslés – Alternálás viselkedése affin hiba növelésére.

A **V.3** diagram azt szemlélteti, hogy egyre zajosabb affin transzformációs mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Az alternálás minden esetben jobb eredményt ad, mint az eredeti egyenletekből származó Gyorsbecslés.

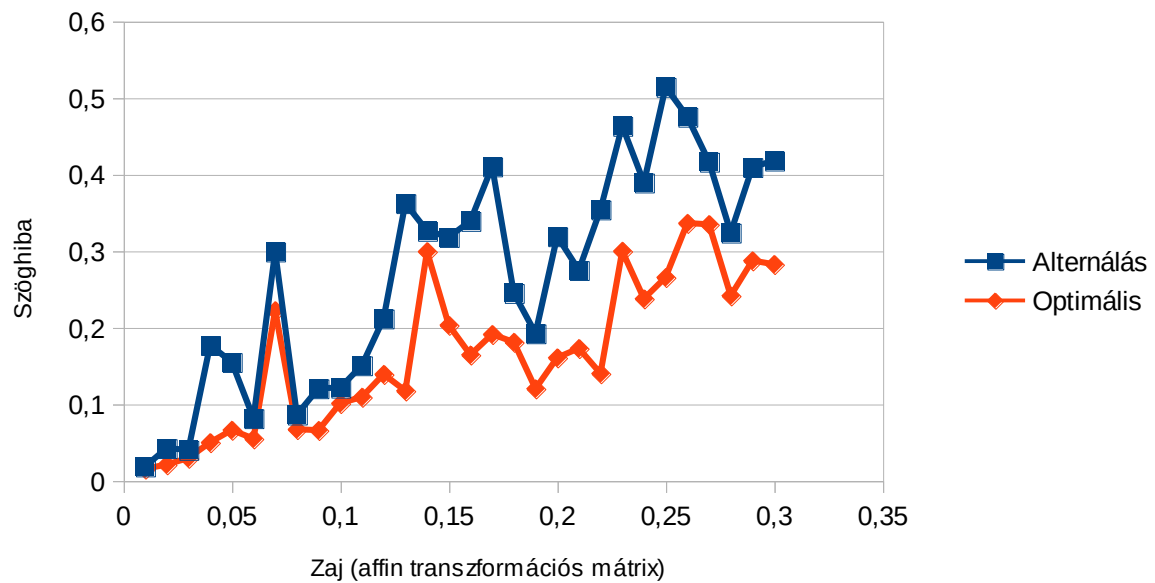


V.4 Gyorsbecslés – Alternálás viselkedése projekciós hiba növelésére.

A **V.4** diagram azt szemlélteti, hogy egyre zajosabb projekciós mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Az Alternálás érzékenysége a projekciós mátrix hibájára láthatóan kisebb, mint a Gyorsbecslőé.

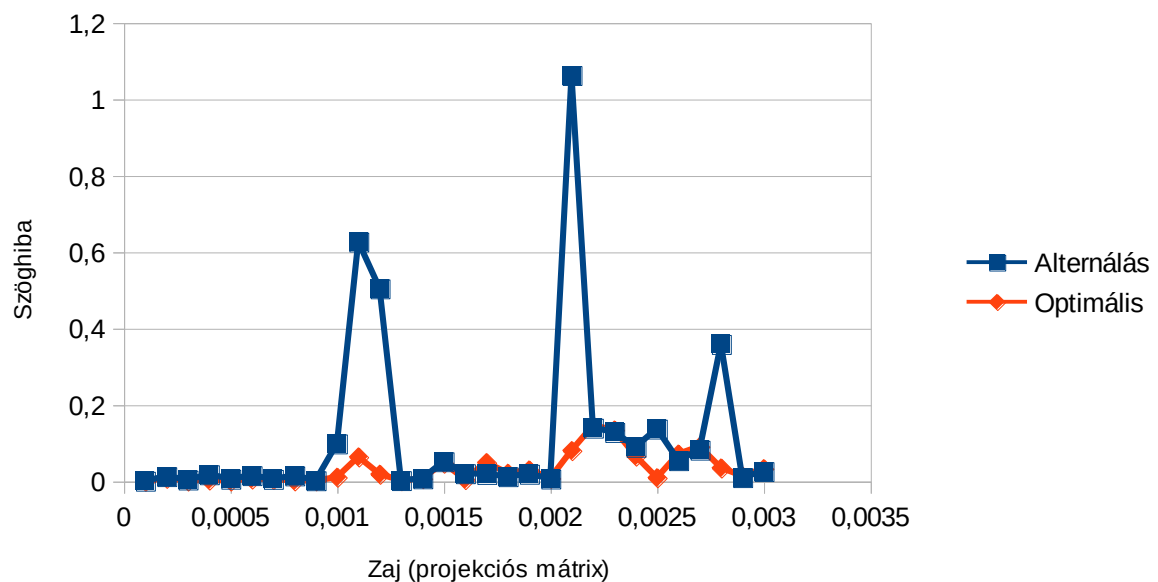
5.1.3. Optimális becslő – Alternálás

A szintetikus tesztek során egyértelműen megjelent, hogy bár az Alternálás jó eredményeket ad affin transzformációs hiba esetén az Optimális megoldás minden esetben jobb nála (**V.5**). A projekciós mátrixban megjelenő hibára az Optimális becslő sokkal kevésbé érzékeny (**V.6**). Összességében az Optimális becslő robosztusabb eljárás, mint az Alternálás.



V.5 Optimális becslő – Alternálás viselkedése affin hiba növelésére.

A V.5 diagram azt szemlélteti, hogy egyre zajosabb affin transzformációs mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan az Optimális megoldás jobb eredményt ad minden esetben.

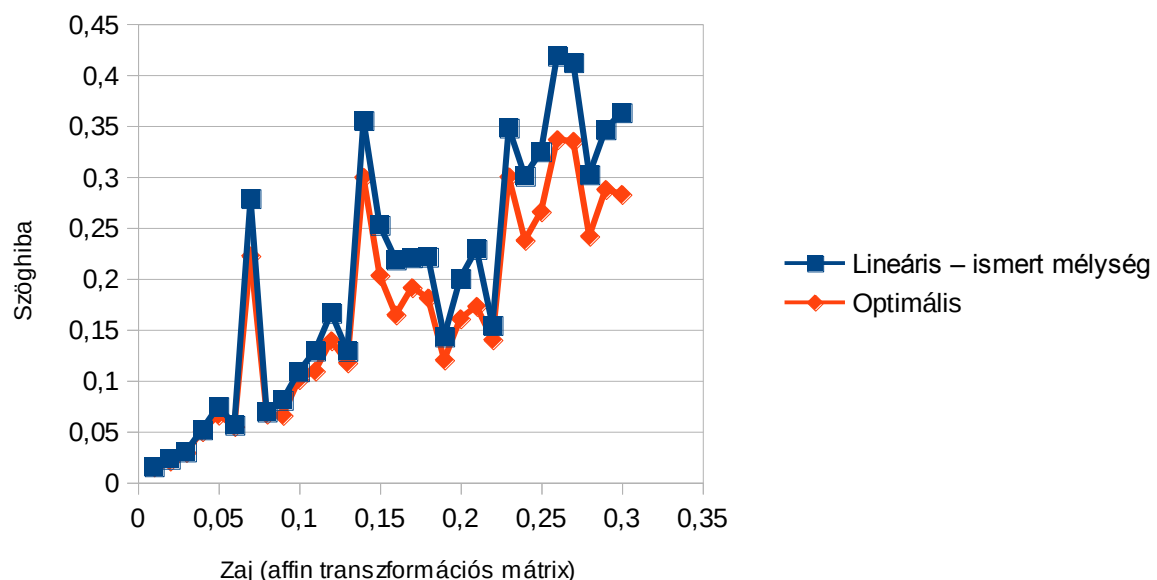


V.6 Optimális becslő – Alternálás viselkedése projekciós hiba növelésére.

A **V.6** diagram azt szemlélteti, hogy egyre zajosabb projekciós mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Egyértelműen megjelenik, hogy az Optimális megoldás sokkal kevésbé érzékeny a projekciós mátrixban megjelenő hibára, mint az Alternálás.

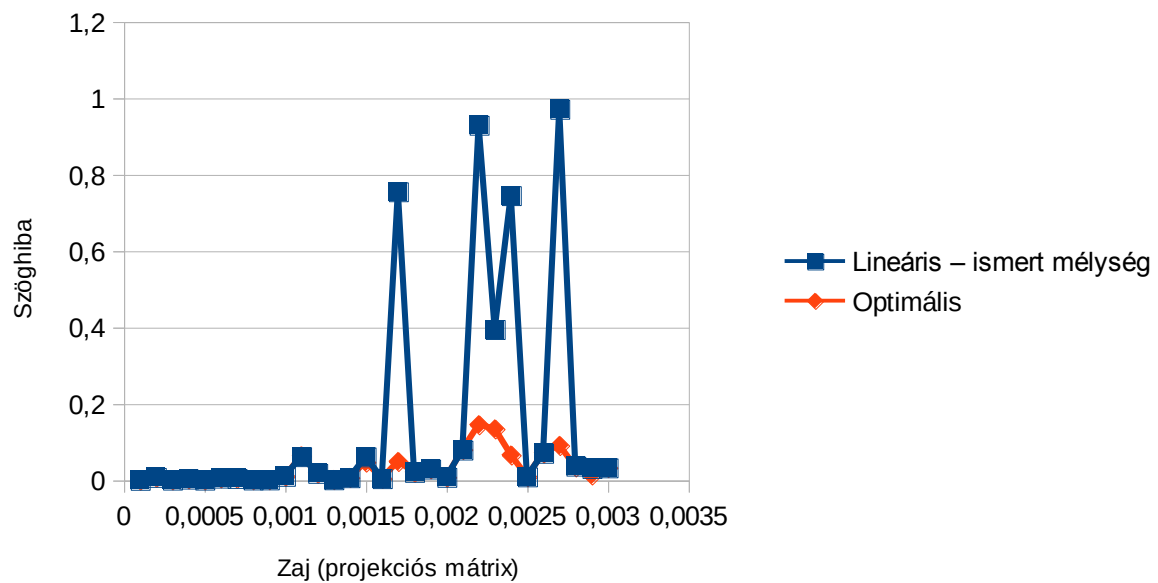
5.1.4. Optimális becslő – Lineáris közelítés ismert mélység esetén

Ezen két becslési eljárás közül az Optimális megoldás jobb eredményt ad, mint a lineáris közelítése, de közel azonos mértékben érzékeny az affin transzformációs mátrixban megjelenő zajra (**V.7**). Ezzel szemben az Optimális megoldás sokkal robosztusabb a projekciós mátrixban zajára (**V.8**).



V.7 Optimális becslő – Lineáris közelítés viselkedése affin hiba növelésére.

A **V.7** diagram azt szemlélteti, hogy egyre zajosabb affin transzformációval számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan a két megoldás – az Optimális, illetve az Optimális lineáris közelítése ismert mélységre – közel azonos, azonban az Optimális megoldás némileg minden esetben jobb eredményt ad.

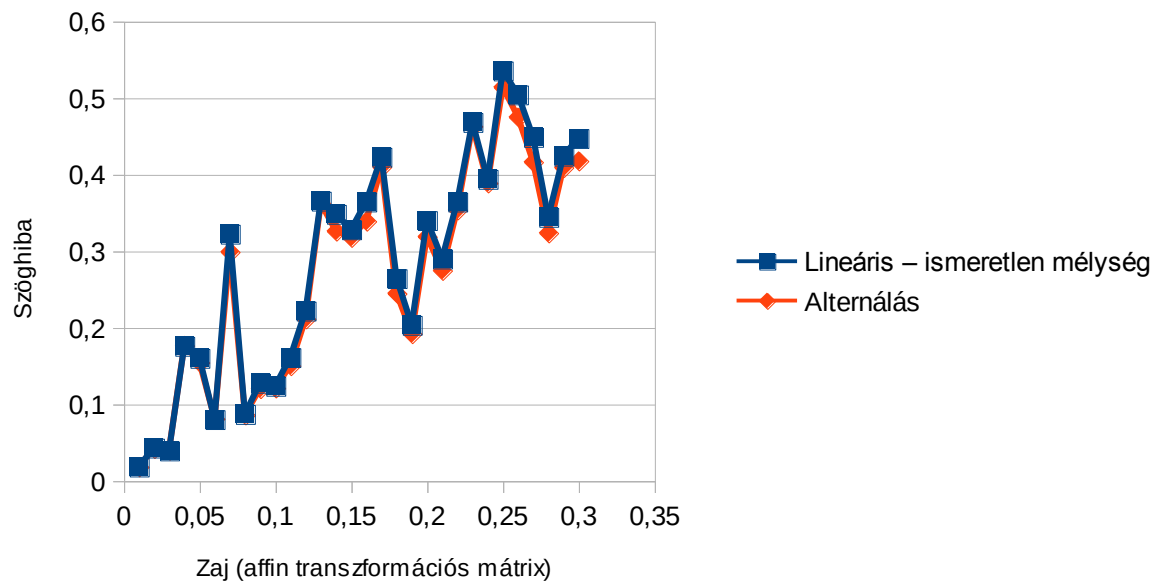


V.8 Optimális becslő – Lineáris közelítés viselkedése projekciós hiba növelésére.

A **V.8** diagram azt szemlélteti, hogy egyre zajosabb projekciós mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan a két megoldás – az Optimális, illetve az Optimális lineáris közelítése ismert mélységre – közel azonos, azonban az Optimális megoldás némileg minden esetben jobb eredményt ad.

5.1.5. Alternálás – Lineáris közelítés ismeretlen mélység esetén

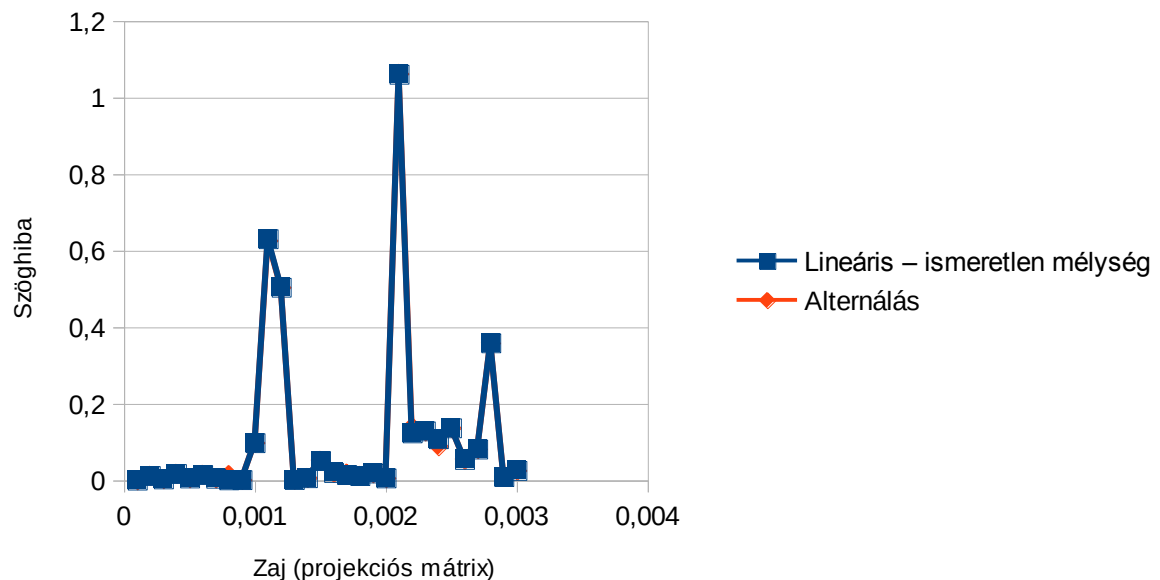
A két eljárás közel azonos mértékben érzékeny mind az affin transzformációs mátrixban (**V.9**), mind a projekciós mátrixban fellépő zajra (**V.10**). Ezzel szemben a lineáris becslő kisebb műveletigénnyel bír, mint az Alternálás.



VII.9 Alternálás – Lineáris közelítés viselkedése affin hiba növelésére.

A **VII.9** diagram azt szemlélteti, hogy egyre zajosabb affin transzformációs mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz.

Láthatóan a két megoldás közel azonos minőségű eredményt ad.

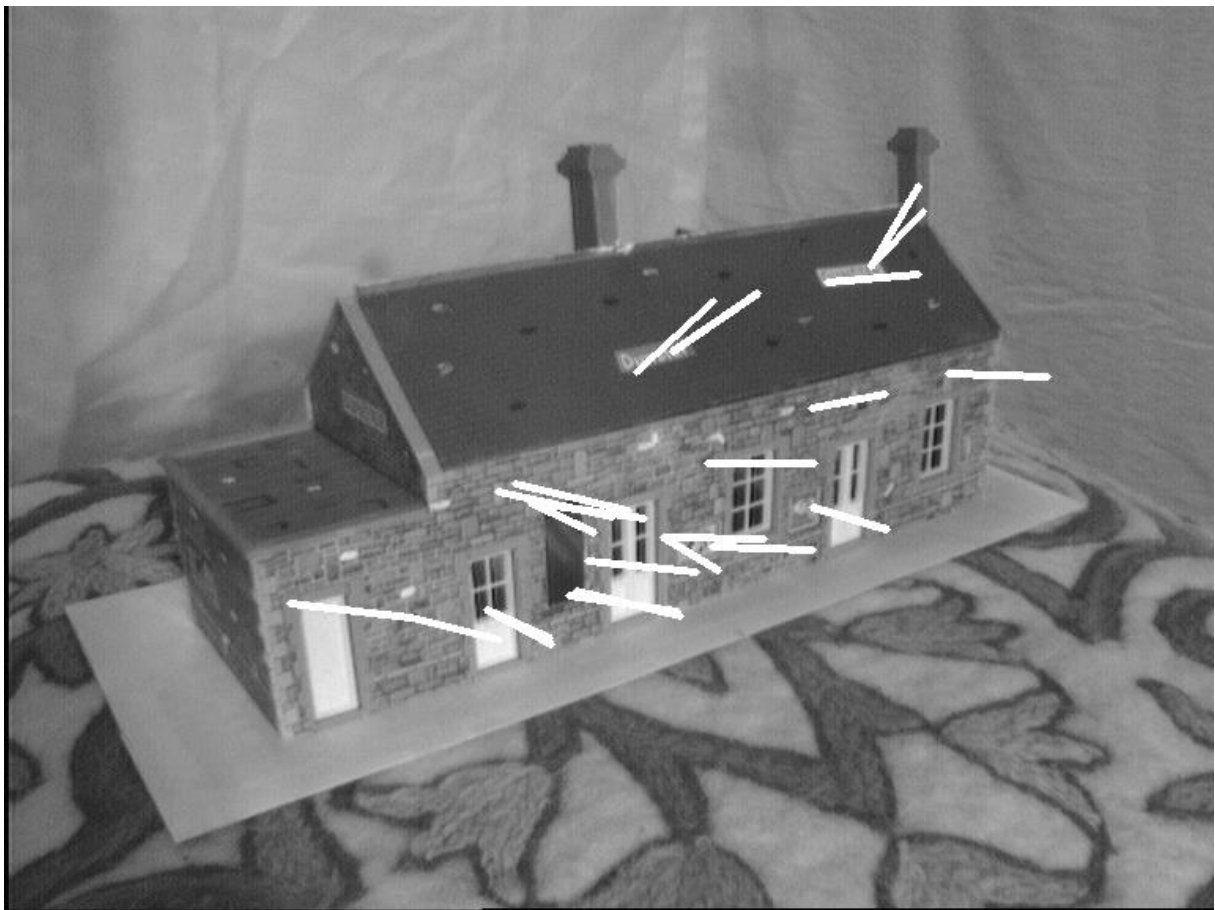


VII.10 Alternálás – Lineáris közelítés viselkedése projekciós hiba növelésére.

A **VII.10** diagram azt szemlélteti, hogy egyre zajosabb projekciós mátrixszal számolt normálvektorok esetén a két eljárás hibája hogyan viszonyul egymáshoz. Láthatóan a két megoldás közel azonos minőségű eredményt ad.

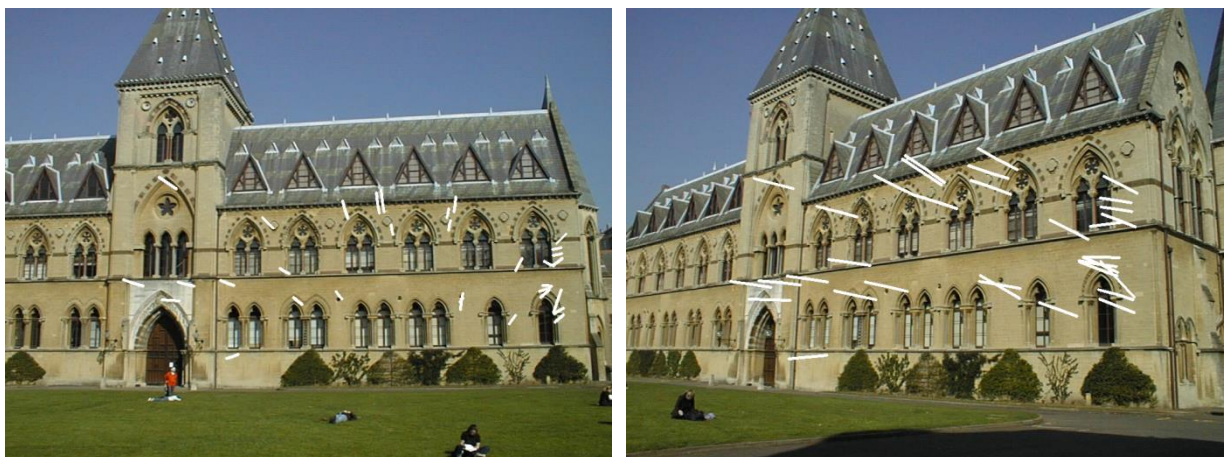
5.2. Valós tesztek

A valós tesztek (**V.11**, **V.12**) kiértékelése során egyértelműen megjelent az a tény, hogy fent megadott eljárások egyedüli érzékeny pontja az affin transzformációban fellépő hiba, azonban jó minőségű transzformációval az eredmények kiválóak.



V.11 Egy modell ház a normálvektoraival.

A **V.11** képen egy kocka látható a becsült normálvektoraival. Látszik, hogy a becsült értékek a valódi síkok felületi normálisát jól közelítik.



V.12 Sztereo képpár a detektált normálvektorokkal.

A **V.12** képen egy épület látható a becsült normálvektoraival. Látszik, hogy a becsült értékek a valódi síkok felületi normálisát jól közelítik. Jelen esetben ez talán első ránézésre nehezen meghatározható, azonban a becsült régiók sok helyen az ablakok, illetve a különböző objektumokon, azok élén helyezkednek el és nem a fal konkrét síkján.

6. Affin transzformációt becselő eljárás

6.1. Affin transzformáció (elméleti áttekintés)

Az affin transzformáció egy aránytartó leképezése pontok halmazának. Ezt kétdimenziós esetben egy $A \in \mathbb{R}^{2 \times 3}$ mátrixszal ábrázoljuk, ami a mátrixszorzás asszociativitásából fakadóan felírható a különböző transzformációk szorzataként [16]:

$$\begin{aligned} R * S_1 * S_2 * T * x &= \\ (R * S_1 * S_2 * T) * x &= \\ A * x \end{aligned}$$

$$R, S_1, S_2, T \in \mathbb{R}^{3 \times 3}$$

$$x \in \mathbb{R}^3$$

Ekkor a pontok homogén koordinátás alakját vesszük. Gyakorlatban az affin transzformáció felső 2×3 -as szubmátrixát vesszük.

Transzformáció fajták (két dimenzióban):

1. Elforgatás (R)

$$x' = R * x = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\alpha \in \mathbb{R}$ - az elforgatás szöge

2. Skálázás (S_1)

$$x' = S_1 * x = \begin{bmatrix} s_{1,1} & 0 & 0 \\ 0 & s_{1,2} & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$s_{1,1}, s_{1,2} \in \mathbb{R}$ - az különböző tengelyek menti skálázás mértéke

3. Nyírás (S_2)

$$\mathbf{x}' = S_1 * \mathbf{x} = \begin{bmatrix} 1 & s_{2,2} & 0 \\ s_{2,1} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$s_{2,1}, s_{2,2} \in \mathbb{R}$ - az különböző tengelyek menti nyírás mértéke.

Valójában egy nyíró paraméter van (megoldható a külön tengelyek menti nyírás 1 paraméterrel plusz egy forgatással), de a jó eredmény gyorsabb elérése érdekében vesszük a külön tengelyek menti nyírást.

4. Eltolás (T)

$$\mathbf{x}' = T * \mathbf{x} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$t_x, t_y \in \mathbb{R}$ - az különböző tengelyek menti eltolás mértéke

6.2. Becslés kereséssel

Az előző fejezetekből kiderül, hogy fontos szerepet játszik a normálvektorok megbecslése közben a helyesen meghatározott affin transzformáció

A felületi normális a fentiekben vázolt becslésének egyik legérzékenyebb pontja a megfelelő affin transzformáció megtalálása.

Az affin transzformáció kinyerése általában feature pontok detektálásával és megfeleltetésével történik. Ekkor lényegében a megkapott kétdimenziós pontfelhő egymásba transzformálásáról van szó, ami könnyen számítható, hiszen már kialakult matematikája van. A mi esetünkben ez nem kivitelezhető az affin régiók kis méretéből adódóan – nem detektálható és megfeleltethető rajtuk a számításokhoz elegendő mennyiségű pont.

Ezért a mi esetünkben a képrészletek közötti affin transzformációt becslő eljárás egy numerikus hibaminimalizációs módszer lesz, amely az első patch-hez keresi azt az affin transzformációt, amellyel az a lehető legjobban korrelál a második képrészlettel.

$$A^{opt} = f(I_1, I_2) = \arg \min_{A \in \mathbb{R}^{2 \times 3}} g(I_1, I_2 \circ A)$$

$I_l \in \mathbb{R}^{n \times m}$ ($l \in \{1, 2\}$) – A képrészleteket tartalmazó mátrixok

$A \in \mathbb{R}^{2 \times 3}$ – affin transzformációs mátrix

$A \circ (: \mathbb{R}^{n \times m} \times \mathbb{R}^{2 \times 3} \rightarrow \mathbb{R}^{n \times m})$ operátor legyen egy olyan transzformációs operátor, amely a megadott képrészlet pixeleire, mint kétdimenziós pontokra alkalmazza az adott transzformációs mátrixot a megfelelő interpolációs lépésekkel együtt.

$A \ g(: \mathbb{R}^{n \times m} \times \mathbb{R}^{n \times m} \rightarrow \mathbb{R})$ költségfüggvény úgynevezett fotokonzisztencia-vizsgálatra épül - a két kép pixelei közötti normalizált kereszt-korrelációt adja meg [15]:

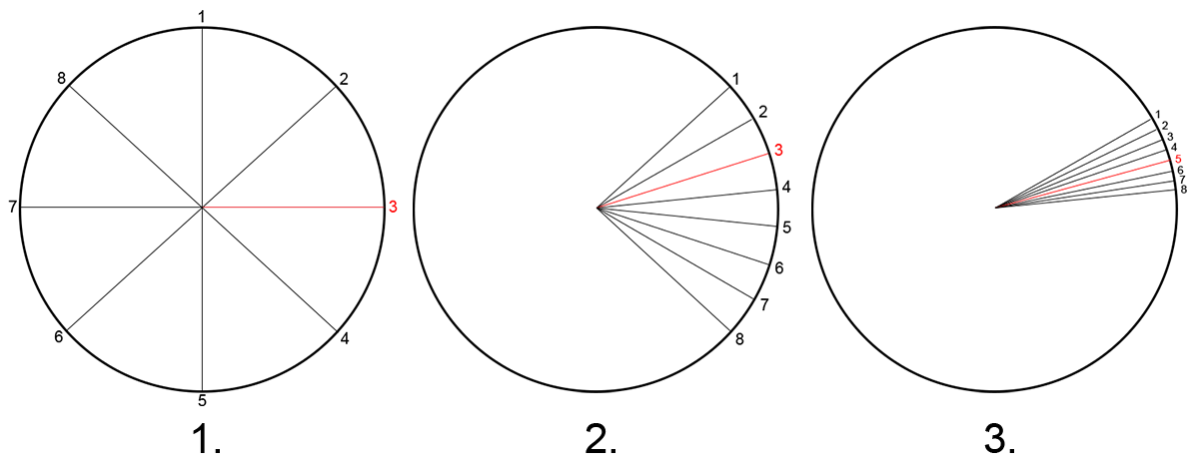
$$g(I_1, I_2) = 1 - \frac{\sum_{(i,j) \in W} I_{1,i,j} * I_{2,i,j}}{\sqrt{\sum_{(i,j) \in W} I_{1,i,j}^2 * \sum_{(i,j) \in W} I_{2,i,j}^2}}$$

Esetünkben tehát egy globális keresésről beszélünk, amely eredménye az a transzformáció, amelyet alkalmazva a második affin régió a számított hiba minimális.

Ez a gyakorlatban túlságosan nagy számítási igényű feladat. Ezért az algoritmus a különböző transzformáció-típusok mértékének megbecslésével, majd az eredmény finomításával foglalkozik.

6.3. Elforgatás meghatározása

Az elforgatás megbecslése egy olyan numerikus eljárás esetünkben, ahol fokozatos finomítással keressük a legkisebb hibát adó szöveget.



VII.1 Keresési tér finomítása - Forgatás

IV.1 kép a keresés finomításának szemléltetése. Az első képen a keresési tér a teljes kör, itt a legjobbnak vélt elforgatás a 3-as indexű szög. A második képen a 3-as indexű szög környezetében folytatódik a keresés. A 3. kép az előző keresés további finomítását szemlélteti. A becselő algoritmus menete:

1. Inicializálja az algoritmust, beállítja a keresési teret $([0^\circ, 360^\circ])$, a kezdeti lépésközt (60), illetve a finomítási paramétert $(\frac{1}{10})$.
2. Veszi a keresési teret, majd az intervallum alsó határától indulva a megadott lépésközzel változtatva a rotációt megvizsgálja az illesztési hibát.
3. A legkisebb hibával rendelkező szög egy környezetét veszi az új keresési térnek, majd újratekdi 2. ponttól az algoritmust, illetve a paramétereket finomítja. Ez az algoritmus addig fut, amíg el nem éri az előre meghatározott mélységet.

Formálisan:

$A^{(1)}$ - A kiindulási transzformációs mátrix.

$$\forall i \in \mathbb{N}: \delta_i^{(1)}, \alpha_i^{(1)}, \beta_i^{(1)} \in [0, 2\pi]$$

$$\alpha_i^{(1)} < \beta_i^{(1)}, \delta_i^{(1)} < \delta_{i-1}^{(1)}, \alpha_i^{(1)} \geq \alpha_{i-1}^{(1)}, \beta_i^{(1)} \leq \beta_{i-1}^{(1)}$$

$$\alpha_0^{(1)} (= 0) - \text{A kezdeti alsó határ}$$

$$\beta_0^{(1)} (= 2\pi) - \text{A kezdeti felső határ}$$

$$\delta_0^{(1)} - \text{A kezdeti lépésköz}$$

$$\mu^{(1)} (= \frac{1}{10}) - \text{A finomítási paraméter}$$

$\varphi^{(1)}: \mathbb{R}^{2 \times 3} \times [0, 2\pi] \rightarrow \mathbb{R}^{n \times m}$ - Hozzáveszi a paraméterként megadott affin transzformációhoz az adott elforgatást.

$$\alpha_i^{(1)} = \left\lfloor \frac{f_i^{(1)}}{\delta_{i-1}^{(1)}} \right\rfloor \delta_{i-1}^{(1)}$$

$$\beta_i^{(1)} = (\left\lfloor \frac{f_i^{(1)}}{\delta_{i-1}^{(1)}} \right\rfloor + 1) \delta_{i-1}^{(1)}$$

$$\delta_i^{(1)} = \delta_{i-1}^{(1)} * \mu^{(1)}$$

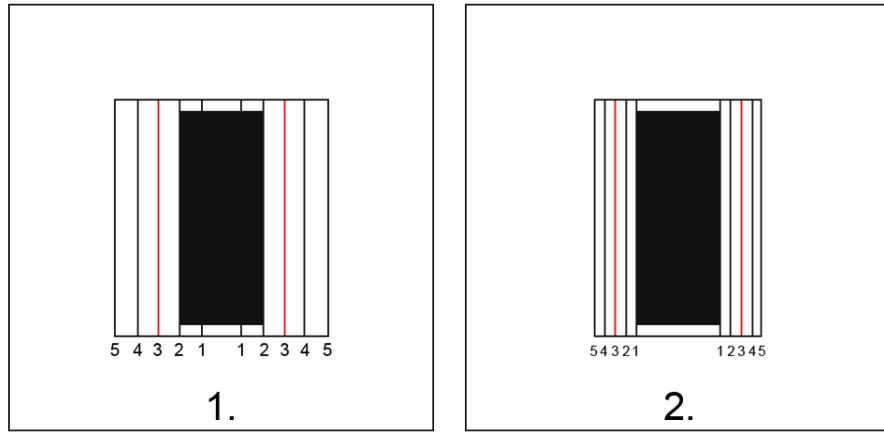
$$f_0^{(1)} = 0$$

$$f_i^{(1)} = \arg \min_{\gamma \in \Gamma_i^{(1)}} g(I_1, I_2 \circ \varphi^{(1)}(A^{(1)}, \gamma))$$

$$\Gamma_i^{(1)} = \{\gamma \mid \gamma \in [\alpha_{i-1}, \beta_{i-1}], (\gamma - \alpha_{i-1}) \bmod \delta_{i-1}^{(1)} = 0\}$$

6.4. Skálázás meghatározása

A skálázás becslése egy olyan numerikus eljárás esetünkben, ahol fokozatos finomítással keressük a legkisebb hibát adó skála faktort x és y tengelyek mentén szeparáltan. A skálázás centroidja az affin régió középpontja.



IV.2 A keresési tér finomítása - Skálázás.

IV.2-nál az első képen a teljes keresési tér látható, itt a legjobbnak vélt skálafaktor a 3-as indexű. A második képen a 3-as indexű faktor környezetében folytatódik a keresés. A becslő algoritmus menete (külön vizsgáljuk a két tengely menti skálázást):

1. Inicializálja az algoritmust, beállítja a keresési teret $([0.1, 2])$, a kezdeti lépésközt (0.1), illetve a finomítási paramétert $(\frac{1}{10})$.
2. Veszi a teljes keresési teret, majd az intervallum alsó határától indulva diszkrét lépésekkel haladva változtatja a skálafaktort és megvizsgálja a skálázott kép illesztési hibáját.

3. A legkisebb hibával rendelkező skála egy környezetét veszi új keresési térnek, majd újratekdi 2. ponttól az algoritmust, illetve a paramétereket finomítja. Ez az algoritmus addig fut, amíg el nem éri az előre meghatározott mélységet.

Formálisan:

$A^{(2)}$ - A kiindulási transzformációs mátrix

$$\forall k \in \{1,2\}, i \in \mathbb{N}: \delta_{k,i}^{(2)}, \alpha_{k,i}^{(2)}, \beta_{k,i}^{(2)} \in \mathbb{R}^+$$

$$0 < \alpha_{k,i}^{(2)} < \beta_{k,i}^{(2)}, 0 < \delta_{k,i} < \delta_{k,i-1}, \alpha_{k,i}^{(2)} \geq \alpha_{k,i-1}, \beta_{k,i}^{(2)} \leq \beta_{k,i-1}^{(2)}$$

$$\delta_i^{(2)} = \begin{bmatrix} \delta_{1,i}^{(2)} \\ \delta_{2,i}^{(2)} \end{bmatrix} - \text{A lépésköz } x \text{ és } y \text{ tengelyek mentén.}$$

$\varphi_k^{(2)}: \mathbb{R}^{2 \times 3} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{n \times m}$ – Megadja egy affin transzformációs mátrix skálafaktorral módosított változatát. $k = 1$ esetben x tengely menti skálázásról, $k = 2$ esetben y tengely menti skálázásról beszélünk.

$\alpha_{k,0}^{(2)}$ – A kezdeti alsó határ

$\beta_{k,0}^{(2)}$ – A kezdeti felső határ

$\delta_{k,0}^{(2)}$ – A kezdeti lépésköz

$\mu_k^{(2)} (= \frac{1}{10})$ – A finomítási paraméter

$$\alpha_{k,i}^{(2)} = \left\lfloor \frac{f_{k,i}^{(2)}}{\delta_{k,i-1}^{(2)}} \right\rfloor \delta_{k,i-1}^{(2)}$$

$$\beta_{k,i}^{(2)} = \left(\left\lfloor \frac{f_{k,i}^{(2)}}{\delta_{k,i-1}^{(2)}} \right\rfloor + 1 \right) \delta_{k,i-1}^{(2)}$$

$$\delta_{k,i}^{(2)} = \delta_{k,i-1}^{(2)} * \mu_k^{(2)}$$

$$f_{k,0}^{(2)} = 1$$

$$f_{k,i}^{(2)} = \arg \min_{\gamma \in \Gamma_{k,i}^{(2)}} g(I_1, I_2 \circ \varphi_k^{(2)}(A^{(2)}, \gamma))$$

$$\Gamma_{k,i}^{(2)} = \{\gamma \mid \gamma \in [\alpha_{k,i-1}^{(2)}, \beta_{k,i-1}^{(2)}], (\gamma - \alpha_{k,i-1}^{(2)}) \bmod \delta_{k,i-1}^{(2)} = 0\}$$

6.5. A nyírás meghatározása

A nyírás meghatározása az előzőekhez hasonlóan egy olyan numerikus eljárás, ahol fokozatos finomítással keressük a legkisebb hibát adó nyírási mértéket x és y tengelyek mentén szeparáltan. A nyírás centroidja az affin régió középpontja lesz a számolásoknál.

Tegyük fel, hogy $|I_1|_x = |I_2|_x$, $|I_1|_y = |I_2|_y$, de az algoritmus könnyen kiterjeszthető különböző méretű patch-ek esetére is.

A becselő algoritmus menete (külön vizsgáljuk a két tengely menti nyírást):

1. Inicializálja az algoritmust, beállítja a keresési teret $[-1.5, 1.5]$, a kezdeti lépésközt $(\frac{1}{10})$, illetve a finomítási paramétert $(\frac{1}{10})$.
2. Veszi a teljes keresési teret, majd az intervallum alsó határától indulva a diszkrét lépésekkel haladva a megadott nyírási faktorról nyírja a képet és megvizsgálja a transzformált kép illesztési hibáját.
3. A legkisebb hibával rendelkező nyírási faktor egy környezetét veszi új keresési térnek, majd újakezdi 2. ponttól az algoritmust, illetve a paramétereket finomítja. Ez az algoritmus addig fut, amíg el nem éri az előre meghatározott mélységet.

Formálisan:

$A^{(3)}$ - A kiindulási transzformációs mátrix

$$\forall k \in \{1, 2\}, i \in \mathbb{N}: \delta_{k,i}^{(3)}, \alpha_{k,i}^{(3)}, \beta_{k,i}^{(3)} \in \mathbb{R}$$

$$\alpha_{k,i}^{(3)} < \beta_{k,i}^{(3)}, \delta_{k,i}^{(3)} < \delta_{k,i-1}^{(3)}, \alpha_{k,i}^{(3)} \geq \alpha_{k,i-1}^{(3)}, \beta_{k,i}^{(3)} \leq \beta_{k,i-1}^{(3)}$$

$$\delta_i^{(3)} = \begin{bmatrix} \delta_{1,i}^{(3)} \\ \delta_{2,i}^{(3)} \end{bmatrix} - \text{A lépésköz } x \text{ és } y \text{ tengelyek mentén.}$$

$\varphi_k^{(3)}: \mathbb{R}^{n \times m} \times \mathbb{R} \rightarrow \mathbb{R}^{n \times m}$ – Megadja a megadott affin transzformációs mátrix nyírási paraméterrel felülírt változatát. $k = 1$ esetben x tengely menti nyírásról, $k = 2$ esetben y tengely menti nyírásról beszélünk.

$\alpha_{k,0}^{(3)}$ – A kezdeti alsó határ

$\beta_{k,0}^{(3)}$ – A kezdeti felső határ

$\delta_{k,0}^{(3)}$ – A kezdeti lépésköz

$\mu_k^{(3)} (= \frac{1}{10})$ – A finomítási paraméter

$$\alpha_{k,i}^{(3)} = \left\lfloor \frac{f_{k,i}^{(3)}}{\delta_{k,i-1}^{(3)}} \right\rfloor \delta_{k,i-1}^{(3)}$$

$$\beta_{k,i}^{(3)} = \left(\left\lfloor \frac{f_{k,i}^{(3)}}{\delta_{k,i-1}^{(3)}} \right\rfloor + 1 \right) \delta_{k,i-1}^{(3)}$$

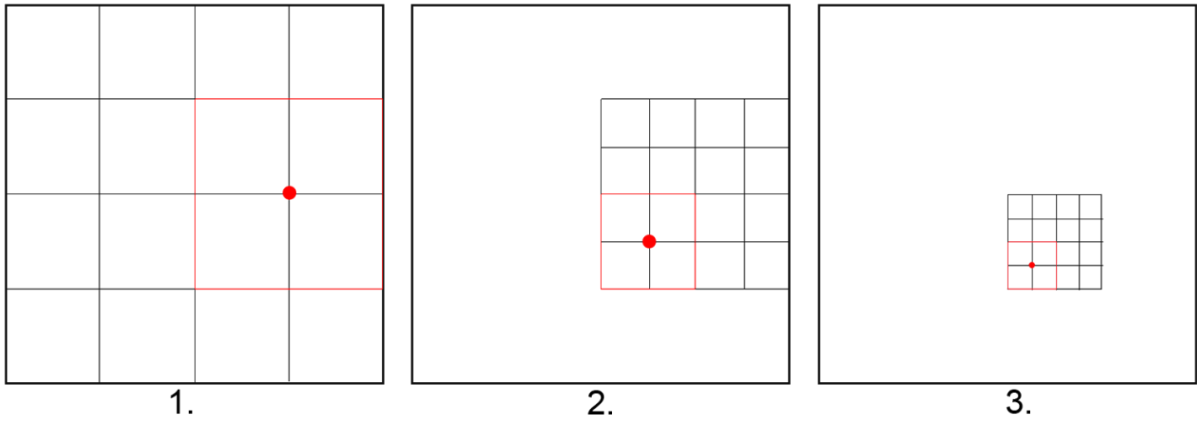
$$\delta_{k,i}^{(3)} = \delta_{k,i-1}^{(3)} * \mu_k^{(3)}$$

$$f_{k,i}^{(3)} = \arg \min_{\gamma \in \Gamma_{k,i}^{(3)}} g(I_1, I_2 \circ \varphi_k^{(3)}(A^{(3)}, \gamma))$$

$$\Gamma_{k,i}^{(3)} = \{\gamma \mid \gamma \in [\alpha_{k,i-1}^{(3)}, \beta_{k,i-1}^{(3)}], (\gamma - \alpha_{k,i-1}^{(3)}) \bmod \delta_{k,i-1}^{(3)} = 0\}$$

6.6. Az eltolás meghatározása

A transláció meghatározása a többihez hasonlóan egy olyan numerikus eljárás, ahol fokozatos finomítással keressük a legkisebb hibát adó pozíciót. Az koncepció, hogy a mintánkat egy rács pontjain mozgatjuk, majd a legkisebb hibát adó pontot kiválasztva folytatjuk az algoritmust annak a környezetében.



VI.3 Keresési tér finomítása - Eltolás

A IV.3 képen a keresési tér fokozatos finomítása látható. Az első képen a keresési tér a teljes képet befoglaló négyzet, itt a legjobbnak vélt pozíció a megjelölt. A második képen az előző képen kijelölt környezetben folytatódik a keresés. A 3. kép az 2. további finomítását szemlélteti. A becslő algoritmus menete:

1. Inicializálja az algoritmust, beállítja a keresési teret (az egész képet tartalmazó négyzet), a felosztás számát (10).
2. Vesz a teljes keresési rácsot, majd egyenként a rádspontokra illeszti képet.
3. A legkisebb hibával rendelkező rádspont egy környezetét vesz új keresési térnek, majd újratekdi 2. ponttól az algoritmust, illetve a paramétereket finomítja. Ez az algoritmus addig fut, amíg el nem éri az előre meghatározott mélységet.

Formálisan:

$A^{(4)}$ - A kiindulási transzformációs mátrix

$$\forall j \in \{1,2\}, i \in \mathbb{N}: \delta_i^{(4)}, \alpha_i^{(4)}, \beta_i^{(4)} \in \mathbb{R}^2$$

$$\alpha_{i,j}^{(4)} < \beta_{i,j}^{(4)}, \delta_{i,j}^{(4)} < \delta_{i-1,j}^{(4)}, \alpha_{i,j}^{(4)} \geq \alpha_{i-1,j}^{(4)}, \beta_{i,j}^{(4)} \leq \beta_{i-1,j}^{(4)}$$

$\varrho^{(4)} \in \mathbb{N} > 1$ – A vizsgált képrészlet felosztásainak száma

$\alpha_0^{(4)} (= \begin{bmatrix} 0 \\ 0 \end{bmatrix})$ – Kezdetben a bal felső sarokpontja a hálónak

$\beta_0^{(4)} (= \begin{bmatrix} |I_1|_x \\ |I_1|_y \end{bmatrix})$ – Kezdetben a jobb alsó sarokpontja a hálónak

$$\delta_0^{(4)} (= \begin{bmatrix} \frac{|I_1|_x}{\varrho} \\ \frac{|I_1|_y}{\varrho} \end{bmatrix})$$
 – Kezdeti lépésköz

$\varphi^{(4)}: \mathbb{R}^{2 \times 3} \times \mathbb{R}^2 \rightarrow \mathbb{R}^{n \times m}$ – Megadja egy adott affin transzformációs mátrix egy eltolási vektorral módosított változatát. Az eltolás centroidja legyen a mátrix közepe.

$$\alpha_i^{(4)} = \begin{bmatrix} \left\lceil \frac{f_{i,1}^{(4)}}{\delta_{i-1,1}^{(4)}} \right\rceil \delta_{i-1,1}^{(4)} \\ \left\lceil \frac{f_{i,2}^{(4)}}{\delta_{i-1,2}^{(4)}} \right\rceil \delta_{i-1,2}^{(4)} \end{bmatrix}$$

$$\beta_i^{(4)} = \begin{bmatrix} \left(\left\lfloor \frac{f_{i,1}^{(4)}}{\delta_{i-1,1}^{(4)}} \right\rfloor + 1 \right) \delta_{i-1,1}^{(4)} \\ \left(\left\lfloor \frac{f_{i,2}^{(4)}}{\delta_{i-1,2}^{(4)}} \right\rfloor + 1 \right) \delta_{i-1,2}^{(4)} \end{bmatrix}$$

$$\delta_i^{(4)} = \begin{bmatrix} \delta_{i-1,1}^{(4)} / \varrho \\ \delta_{i-1,2}^{(4)} / \varrho \end{bmatrix}$$

$$f_0^{(4)} = A^{(4)}$$

$$f_i^{(4)} = \arg \min_{\gamma \in \Gamma_i^{(4)}} g \left(I_1, I_2 \circ \varphi_t(A^{(4)}, \gamma) \right)$$

$$\Gamma_i^{(4)} = \{ \gamma \mid \gamma \in \mathbb{R}^2, \gamma_j \in [\alpha_{i-1,j}^{(4)}, \beta_{i-1,j}^{(4)}], (\gamma_{j-1} - \alpha_{i-1,j}^{(4)}) \bmod \delta_{i-1,j}^{(4)} = 0, \\ j \in \{1, 2\} \}$$

6.7. Lokális paraméterfinomítás

Ezen lépés az affin transzformációs mátrixot nem mint különböző típusú transzformációk szorzatát veszi, hanem az egyes paraméterek ε sugarú környezetében keresi az optimális megoldást fokozatos finomítással.

$A^{(5)}$ - A kiindulási transzformációs mátrix

$\varepsilon_0 \in \mathbb{R}^{2 \times 3}$ - A kezdeti környezeti sugarakat tartalmazó mátrix.

$\varrho^{(5)} \in \mathbb{R}^{2 \times 3}$ - A kezdeti felosztásszámokat tartalmazó mátrix.

$\mu^{(5)} (= \frac{1}{10})$ - A finomítási paraméter.

$\varphi^{(5)}: \mathbb{R}^{n \times m} \times \mathbb{R} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}^{n \times m}$ - Módosítja az adott mátrix megadott indexű elemét.

$\delta_{i,j,k}^{(5)} \left(= \frac{\varepsilon_{i,j,k}}{\varrho_{j,k}^{(5)}} \right)$ - A lépésközöket tartalmazó mátrix.

$$\varepsilon_{i,j,k} = \varepsilon_{i-1,j,k} * \mu^{(5)}$$

$$f_0^{(5)} = A^{(5)}$$

$$f_{i,j,k}^{(5)} = f_{i-1,j,k}^{(5)} + \arg \min_{\gamma \in \Gamma_{i,j,k}^{(5)}} g \left(I_1, I_2 \circ \varphi^{(5)}(f_{i-1}^{(5)}, \varepsilon_{i,j,k}) \right)$$

$$\Gamma_{i,j,k}^{(5)} = \{ \gamma \mid \gamma \in [f_{i-1,j,k}^{(5)} - \varepsilon_{i,j,k}, f_{i-1,j,k}^{(5)} + \varepsilon_{i,j,k}],$$

$$(\gamma - f_{i-1,j,k}^{(5)} + \varepsilon_{i,j,k}) \bmod \delta_{i,j,k}^{(5)} = 0 \}$$

$$f_i^{(5)} = \begin{bmatrix} f_{i,0,0}^{(5)} & f_{i,0,1}^{(5)} & f_{i,0,2}^{(5)} \\ f_{i,1,0}^{(5)} & f_{i,1,1}^{(5)} & f_{i,1,2}^{(5)} \end{bmatrix}$$

6.8. Az affin transzformációt becslő algoritmus menete

A kiindulási mátrix egy identitás affin transzformáció.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$\lambda^{(l)} \in \mathbb{N}^+, (l \in \{1,2,3,4,5\})$ - A különböző becslési eljárásokhoz tartozó finomítási mélység.

Erre az A mátrixra első lépésben alkalmazzuk a rotáció-becslő algoritmust úgy, hogy

$$f_0^{(1)} = 0$$

$$A^{(1)} = A$$

Adódik $f_{\lambda^{(1)}}^{(1)}$ értéke és így $A^{(1)}$ módosításával $A^{(2)}$ is. Az eredmény transzformációval futtatjuk az x tengely menti skálázást

$$f_{1,0}^{(2)} = 1$$

$$A^{(2)} = \varphi^{(1)}(A^{(1)}, f'_{\lambda^{(1)}}^{(1)})$$

majd az y tengely szerintit

$$f'_{2,0}^{(2)} = 1$$

$$A^{(2)'} = \varphi_1^{(2)}(A^{(1)}, f_{1,\lambda^{(2)}}^{(2)})$$

Adódik $f_{2,\lambda^{(2)}}^{(2)}$ értéke és így $A^{(2)'}$ módosításával $A^{(3)}$ is. Következő lépésben az így megkapott transzformációs mátrixra számítsuk ki a skálázáshoz hasonlóan a nyírás mértékét:

$$f_{1,0}^{(3)} = 1$$

$$A^{(3)} = \varphi_2^{(2)}(A^{(2)'}, f_{2,\lambda^{(2)}}^{(2)})$$

$$f_{2,0}^{(3)} = 1$$

$$A^{(3)'} = \varphi_1^{(3)}(A^{(3)}, f_{1,\lambda^{(3)}}^{(3)})$$

Adódik $f_{2,\lambda^{(3)}}^{(3)}$ értéke és így $A^{(3)'}$ módosításával $A^{(4)}$. Az így kapott mátrixhoz keressük meg a megfelelő translációt

$$f_0^{(4)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A^{(4)} = \varphi_2^{(3)}(A^{(3)'}, f_{2,\lambda^{(3)}}^{(3)})$$

Adódik $f_{\lambda^{(4)}}^{(4)}$ értéke és így $A^{(4)}$ módosításával $A^{(5)}$. Alkalmazzuk a lokális paraméterfinomítást:

$$A^{(5)} = \varphi^{(4)}(A^{(4)}, f_{\lambda^{(4)}}^{(4)})$$

$$f_0^{(5)} = f'_{\lambda^{(4)}}^{(4)}$$

Ekkor adódik $f_{\lambda^{(5)}}^{(5)}$ értéke is, amivel módosítva $A^{(5)}$ affin transzformációt megkapjuk az optimális transzformációt:

$$A^{opt} = \varphi^{(5)}(A^{(5)}, f_{\lambda^{(5)}}^{(5)})$$

6.9. Hibás megoldások kezelése

Hibásan becsült A^{opt} detektálására és a kizárására két szempont adódik:

1. A hibára be kell vezetni egy μ küszöböt, aminél nagyobb hiba esetén $(g(I_1, I_2 \circ A^{opt}) > \mu)$ a kapott transzformációs mátrixot dobjuk el.
2. Az algoritmus a skálázás becslésénél hajlamos lokális minimumba konvergálni. Ez könnyen detektálható úgy, ha a skálázási paraméter túl közel van a skálázás felső, vagy alsó határához.

Ezek a

$$A_{1,1}^{opt} - \alpha_{1,0}^{(2)} < \varepsilon$$

$$A_{2,2}^{opt} - \alpha_{2,0}^{(2)} < \varepsilon$$

$$\beta_{1,0}^{(2)} - A_{1,1}^{opt} < \varepsilon$$

$$\beta_{2,0}^{(2)} - A_{2,2}^{opt} < \varepsilon$$

$$\varepsilon \in \mathbb{R}$$

esetek.

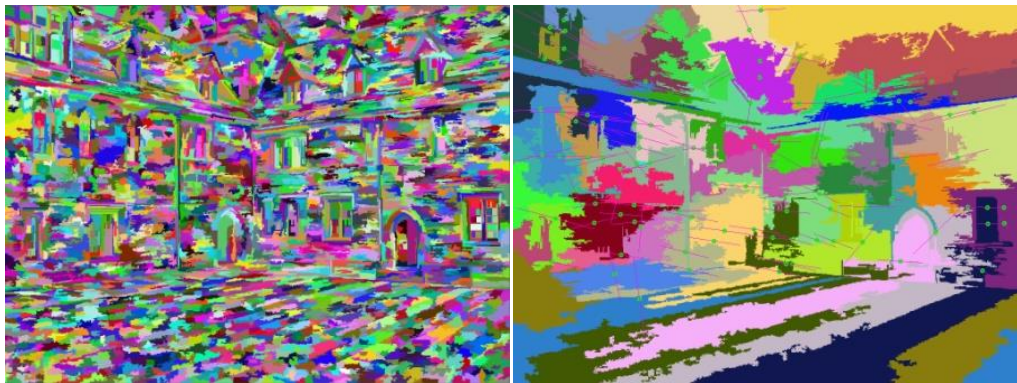
Hibás transzformációs mátrix esetén érdemes kiszámolni a $I'_1 = I_2$, $I'_2 = I_1$ esetre a becsült affin transzformációt. Amennyiben az így kapott transzformáció valószínűsíthetően jó (nem hibás a megadott szempontokra nézve), akkor $A^{opt} = (A^{opt'})^{-1}$ lesz.

7. Régió-alapú osztályozás

7.1. Régiók meghatározása

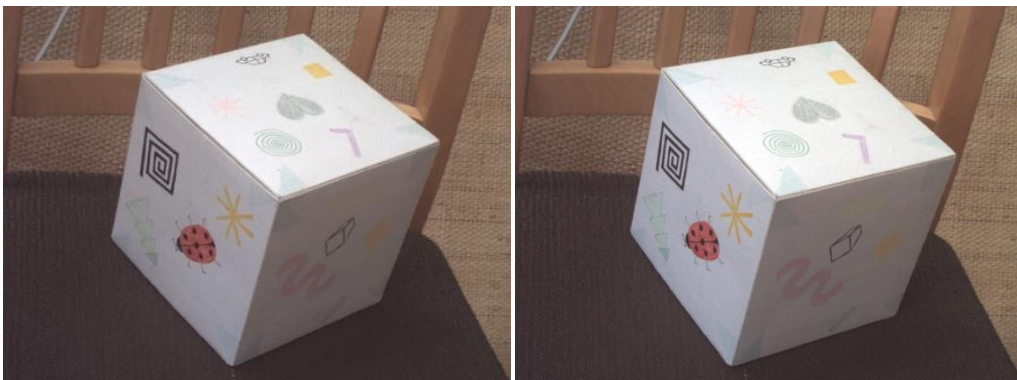


VII.1 Input képpár (1. teszt)

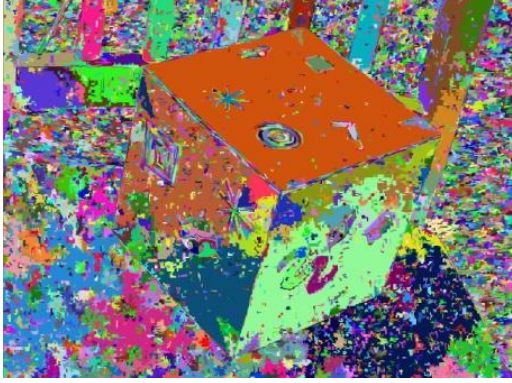


VII.2 Szegmentált kép

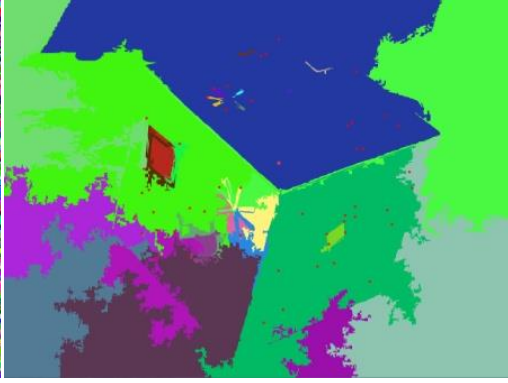
VII.3 Összevont régiók



VII.4 Input képpár (2. teszt)



VII.5 Ssegmentált kép



VII.6 Összevont régiók

A szegmentálás és régióegyesítés menete jól látszik a VII.2, VII.3, és a VII.5 VII.6-os képeken (VII.1, VII.4 az eljárásoknál használt input képpárok). Jól látszik, hogy a sok apró régióból összevont nagyobb régiók a megfelelő síkok mentén helyezkednek el. Az, hogy adott síkot több régió is fed a normálvektorok további pontosításával, illetve a küszöbök módosításával kezelhető.

Az előzőekben kiszámolt normálvektorok tartalmazó síkok szerinti klaszterezése fontos feladat, erre a problémára egy alternatív megoldást jelent egyfajta régió-alapú osztályozás bevezetése. Az algoritmus menete:

1. Válasszuk ki a sztereó képpár egyik képét, a hozzá tartozó affin régiók centroidjaival (feature pontok), illetve a normálvektorokkal együtt, ekkor

$5n \in \mathbb{N}$ legyen a régiók száma.

$n_i^{(1)} \in \mathbb{N}$ legyen az i . régió képpontjainak száma (lényegében a régió területe).

$n_i^{(2)} \in \mathbb{N}$ legyen az i . régió által tartalmazott centroidok száma.

$H_i \in \mathbb{R}^{n_i^{(1)} \times 2}$ legyen az i . régióhoz tartozó képpontokat tartalmazó mátrix.

$$H_i = \begin{bmatrix} x_1 & y_1 \\ \dots & \dots \\ x_{n_i^{(1)}} & y_{n_i^{(1)}} \end{bmatrix}$$

$P_i \in \mathbb{R}^{n_i^{(2)} \times 2}$ legyen az i . régióhoz tartozó centroidokat tartalmazó mátrix.

$$P_i = \begin{bmatrix} x_1 & y_1 \\ \dots & \dots \\ x_{n_i^{(2)}} & y_{n_i^{(2)}} \end{bmatrix}$$

$N_i \in \mathbb{R}^{n_i^{(2)} \times 3}$ legyen az i . régióhoz tartozó centroidok normálvektorait tartalmazó mátrix.

$$N_i = \begin{bmatrix} x_1 & y_1 & z_1 \\ \dots & \dots & \dots \\ x_{n_i^{(2)}} & y_{n_i^{(2)}} & z_{n_i^{(2)}} \end{bmatrix}$$

$T \in \{0,1\}^{n \times n}$ legyen a régiók szomszédossági mátrixa, ami minden olyan T_{ij} cellájában 1-et tartalmaz, ahol az i . régiónak van szomszédos képpontja a j . régióval, ellenkező esetben 0-t. Ezen felül $\forall i, j \in [1, n]: (T_{ij} = 1 \Rightarrow T_{ji} = 1)$

2. Szegmentáljuk a képet viszonylag kisméretű régiókra úgy, hogy a szomszédossági viszonyok is rendelkezésünkre álljanak. Tároljunk el a régiókhoz valami egyedi azonosítót úgy, hogy az azonosító párok között definiálható legyen valamiféle távolság (megfelelő azonosító például a RGB színvektor).

S legyen a lehetséges azonosítóknak egy olyan halmaza, ami a szorzásra, illetve az összeadásra testet alkot.

$D_i \in S$ legyen az i . régióhoz tartozó azonosító.

$\varphi: D_i \times D_i \rightarrow \mathbb{R}$ legyen a távolságfüggvény.

3. Töltsünk fel $\forall P_i, N_i$ mátrixot. Tehát $\forall (x, y) \in \mathbb{R}^2$ centroidot osszuk be ahhoz a régióhoz, amely tartalmazza azt. Jelen esetben a tartalmazás azt jelenti, hogy $(x, y) \in H_i$.
4. Ekkor indítsunk el egy rekurzív régió-növesztő algoritmust. Minden olyan régióhoz, amely tartalmaz legalább egy darab centroidot vizsgáljuk meg a szomszédos régiókat:
 - a. Ha nincs szomszédja az aktuális régiónak, akkor folytassuk az algoritmust a következő régiónál.
 - b. Ha az adott i . régiónak van olyan szomszédja, amely tartalmaz legalább egy centroidot ($n_i^{(2)} > 0 \Rightarrow (\exists j \in [1, n]: (T_{ij} = 1 \wedge n_j^{(2)} > 0))$), akkor vizsgáljuk meg a régióhoz tartozó átlag normálvektor és a szomszédhoz tartozó átlag normálvektor által bezárt szöget.

$N_{i,k}$ legyen az N_i mátrix k -dik sora.

$$\alpha_i = \frac{\sum_{k=1}^{n_i^{(2)}} N_{i,k}^T}{n_i^{(2)}} \text{ ekkor az } i. \text{ régió normálvektorainak átlaga.}$$

$$\gamma_{ij} = \cos^{-1} \frac{\alpha_i \times \alpha_j}{|\alpha_i| * |\alpha_j|} \text{ ekkor az } i. \text{ régió átlag normálvektora és a } j. \text{ régió}$$

átlag normálvektora által bezárt szög.

μ egy megfelelő küszöb

Amennyiben $\gamma_{ij} < \mu$ vonjuk össze a két régiót és módosítsuk a megfelelő mátrixokat. Ekkor folytassuk az algoritmust a következő régiónál.

$$N'_i = \begin{bmatrix} N_i \\ N_j \end{bmatrix}, P'_i = \begin{bmatrix} P_i \\ P_j \end{bmatrix}, H'_i = \begin{bmatrix} H_i \\ H_j \end{bmatrix},$$

$$n_i^{(1)} = n_i^{(1)} + n_j^{(1)}, n_i^{(2)} = n_i^{(2)} + n_j^{(2)}$$

$$\forall k \in [1, n]: (T_{i,k} = 1 \vee T_{j,k} = 1) \Rightarrow (T'_{i,k} = 1)$$

$$D'_i = \frac{n_i^{(1)}}{n_i^{(1)} + n_j^{(1)}} D_i + \frac{n_j^{(1)}}{n_i^{(1)} + n_j^{(1)}} D_j$$

- c. Amennyiben nem sikerült az összevonás, keressük meg az aktuális (i . indexű) régióhoz legjobban hasonlító j indexű szomszédját és vonjuk össze vele.

$$j = \arg \min_{j \in [1, n_i^{(2)}]} \varphi(D_i, D_j)$$

Amennyiben sikerült az összevonás módosítsuk a megfelelő mátrixokat. Folytassuk az algoritmust a következő régiónál.

- d. Ha $i = n$ és történt legalább egy összevonás, akkor indítsuk újra az algoritmus a 4. ponttól. Ha nem történt összevonás, akkor végeztünk.

5. Minden olyan i régiót, amely nem tartalmaz centroidot ($n_i^{(2)} = 0$) töröljünk.

7.2. Régiókhöz tartozó normálvektor meghatározása

Az 5.1. pontban leírt módszer után maradó régiókból meg lehet határozni a hozzájuk legjobban illeszkedő normálvektort. Ez több módszerrel is lehetséges:

1. A normálvektorok átlaga ($\frac{\sum_j^{n_i^{(2)}} N_{i,j}^T}{n_i^{(2)}}$). Ez sajnos nem eléggé robusztus, néhány outlier az optimális megoldástól elhúzza az eredményt.
2. Valamilyen síkillesztési algoritmus (például PCA) segítségével a normálvektorok közé egy síkot illesztve a régióhoz tartozó megfelelő normálvektor a sík egyik alkotó vektora lesz. Ez a normálvektorok átlagától vett távolság alapján könnyen kiválasztható. Ezzel a módszerrel is ugyanaz lesz a probléma, mint az átlagvektor számításával is: túlságosan érzékeny hibára.

A síkot a következőképpen kapjuk meg PCA segítségével amennyiben adott régióhoz legalább három darab normálvektor tartozik ($n_i^{(2)} \geq 3$):

A sík két alkotó vektora $A^T A$ mátrix két legnagyobb sajátértékéhez tartozó sajátvektorai lesznek, ahol

$$N_{i,j} = [N_{i,j,1} \quad N_{i,j,2} \quad N_{i,j,3}]$$

$$O_i = \frac{\sum_j^{n_i^{(2)}} N_{i,j}^T}{n_i^{(2)}}$$

$$A = \begin{bmatrix} N_{i,1} - O_i \\ \dots \\ N_{i,n_i^{(2)}} - O_i \end{bmatrix}$$

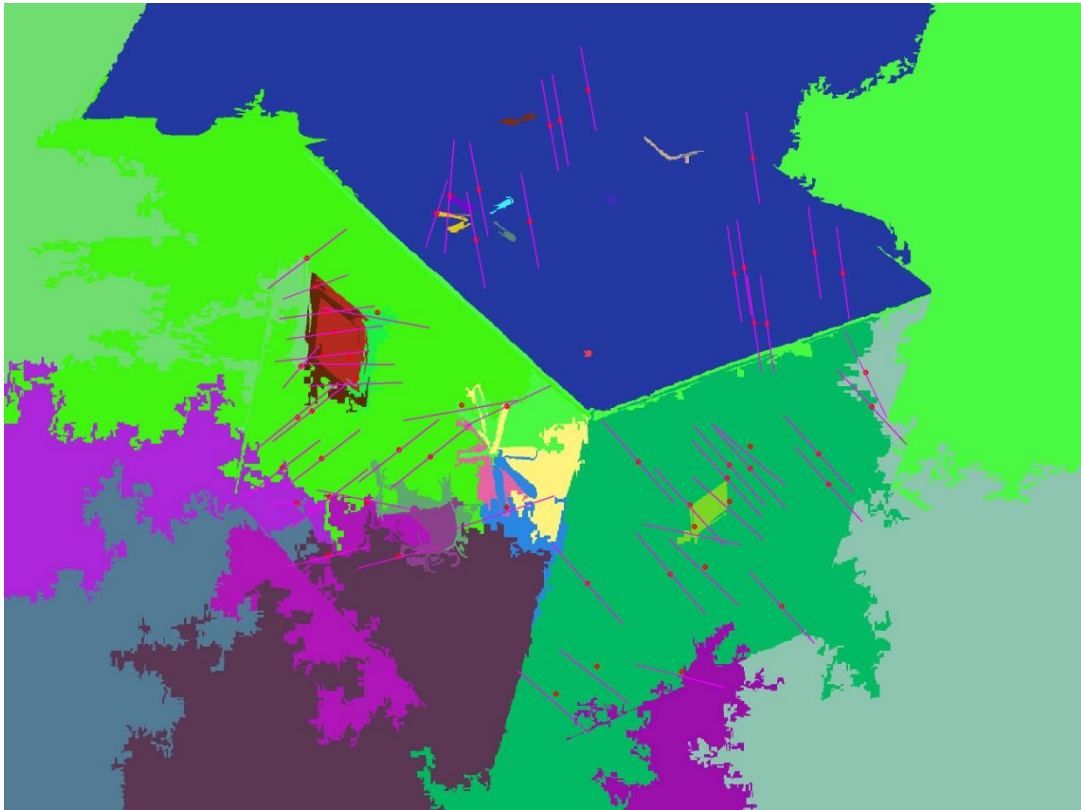
3. A normálvektorok mediánja lesz a régióhoz tartozó felületi normális. Az a módszer lesz a legrobusztusabb, amennyiben az outlierok száma kisebb, mint a minták számának a fele.

$$N_{i,j} = [N_{i,j,1} \quad N_{i,j,2} \quad N_{i,j,3}]$$

$$\gamma = \arg \min_{j \in [1, n_i^{(2)}]} \sum_{k=1}^{n_i^{(2)}} \left(\cos^{-1} \frac{N_{i,j}^T \times N_{i,k}^T}{\sqrt{N_{i,k} N_{i,j}^T} * \sqrt{N_{i,k} N_{i,k}^T}} \right)^2$$

$$n_i = N_{i,\gamma}$$

7.3. Kiértékelés



VII.6 Régiókhöz tartozó normálvektorok.

A VII.6 kép az egyes régiókhöz mediánszűrővel meghatározott normálvektort jeleníti meg tartozó összes tartalmazott affin centroidnál.

A normálvektorokat felhasználó régió-alapú síkmeghatározás komplex összetételű színtérre is robosztusan szeparálja a felületi normálisakat, amikből aztán további információk felhasználásával (pl. háromdimenziós pozíció, síkok metszése egymással) könnyen kialakítható a színtér háromdimenziós formájában.

8. Konklúzió

Tehát a normálvektorok meghatározására Molnár József által bevezetett alap összefüggések további finomításával, optimalizálásával sokkal robosztusabb, és a becslési hibát tekintve átlagosan másfélszer jobb eredményeket adó eljárásokat sikerült kidolgozni.

A kidolgozott módszerek megnevezései:

- Optimális megoldás ismert mélységre (Optimális becselő)
- Lineáris becslés ismert mélységre
- Optimális megoldás ismeretlen mélységre (Alternálás)
- Lineáris becslés ismeretlen mélységre

A Gyorsbecslőnél mindegyik megoldás lényegesebb jobb eredményt ad – kevésbé érzékeny az input adatok hibájára. Értelmszerűen az ismeretlen mélységgel számoló eljárások valamivel bizonytalanabban közelítik az optimális megoldást. Így a lehető legjobb eredmény – ahogy a fenti diagramokból is látszik - az Optimális eljárással adható – vagy annak lineáris közelítése is megfelelő az esetek többségében. Ezeken felül a képfeldolgozásban ritkaságnak számít egy problémára az optimális (legkisebb négyzetes értelemben) megoldás megléte és kiszámíthatósága.

A módszer további finomításának lehetőségei is könnyen felvethetők, akár a becslőkre vonatkozó valószínűségi apriori adatok a rendszerhez való hozzáadásával, illetve további optimalizációs eljárások felhasználásával (pl. Kálmán-szűrő).

Az affin becslőnk más hasonló elven működő becslőkkel összehasonlítva jóval több patch párra ad eredményt és a kapott transzformációs mátrixok sokkal jobban közelítik a valódi transzformációt.

A normálvektorok régió-alapú szegmentálása erősen függ a becsült normálisok minőségétől – elsősorban az outlierok mennyiségétől – azonban jó normálvektorokra megfelelően meghatározza tartalmazó síkok szerint a vektorcsoportokat.

9. Hivatkozások

- [1] Tomasi, C. and Shi, J. Good Features to Track. In IEEE Conf. Computer Vision and Pattern Recognition, pages 593–600, 1994.
- [2] C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Robotics Institute, Carnegie Mellon University, 1991.
- [3] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In International Joint Conference on Artificial Intelligence, pages "674–679", "1981".
- [4] Richard I. Hartley and Peter Sturm. Triangulation. Computer Vision and Image Understanding: CVIU, 68(2):146–157, 1997.
- [5] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [6] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991.
- [7] Jianbo Shi and Carlo Tomasi. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- [8] Lowe, David G. (1999). "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision* **2**. pp. 1150–1157.
- [9] "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004
- [10] Ryuji Funayama, Hiromichi Yanagihara, Luc Van Gool, Tinne Tuytelaars, Herbert Bay, "ROBUST INTEREST POINT DETECTOR AND DESCRIPTOR", published 2009-09-24

- [11] Martin Habbecke, Leif Kobbelt, „Iterative Multi-View Plane Fitting” (2006)
- [12] M Habbecke, Kobbelt: A surface-growing approach to multi-view stereo reconstruction
- [13] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
- [14] Rózsa P. Lineáris algebra és alkalmazásai. Nemzeti Tankönyvkiadó, Budapest, 1991.
- [15] Csetverikov Dmitrij – ELTE Elektronikus jegyzet
- [16] Valasek Gábor – ELTE Elektronikus jegyzet