Multi-H: Érintősíkok hatékony kinyerése képpárokból

Daniel Barath[1], Jiri Matas[2], Levente Hajder[1]

[1] MTA SZTAKI, Machine Perception Research Laboratory, Budapest
{barath.daniel,hajder.levente}@sztaki.mta.hu
[2] The Center for Machine Perception Department of Cybernetics Faculty of Electrical Engineering Czech Technical University, Prague
matas@cmp.felk.cvut.cz

Kivonat A Multi-H algoritmus[1] hatékony módszer érintősíkok kinyerésére pont-megfeleltetésekből, melyek kielégítik az epipoláris feltételeket. A problémát energia-minimalizációs feladatként fogalmazzunk meg, ahol a minimalizálandó energia egy adat és egy térbeli összetartozási kifejezésekből áll. A síkok számát a Mean-Shift és Alpha-Expansion algoritmusok segítségével kontrolláljuk.
A fountain-P11 háromdimenziós adathalmazon bemutatjuk, hogy a Multi-H módszer pontos becslést ad az érintősíkokra. Ezen felül az AdelaideRMF adathalmazon demonstráljuk, hogy a módszer felülmúlja a legkorszerűbb multi-homográfia illesztő technikákat. Mivel a Multi-H közel hiba nélküli eredményt ért el ezeken, készítettünk új képpárokat, melyeken a multi-homográfia illesztés lényegesen nehezebb feladat.

1.. Bevezetés

A külső és belső színterek megértése alapvető fontosságú a Számítógépes Látás számos területén. Az ember-készítette objektumok sokszor főleg síkszerű régiókból állnak – jellemzően városi környezetben vagy belső színtéren. A felmerülő feladatok megoldására sok algoritmus síkokat vagy síkszerű objektumokat használ. Ilyen feladatok például a kamera kalibráció [1,2,3], robot navigáció [4,5], kiterjesztett valóság [6] vagy éppen a háromdimenziós rekonstrukció [7,8].

Ez a cikk elsősorban a színtéren megjelenő érintősíkok pontos becslését tűzi ki célul, olyan módon, hogy két kép közötti pont-megfeleltetéseket – melyek kielégítik az epipoláris kényszert – az érintősíkuk hasonlósága alapján particionálja. Egy sík-sík megfeleltetés két kép között az úgynevezett homográfia mátrixszal írható le [9]. Ennek becslésére számos lehetőség adódik különböző típusú bemeneti adatok felhasználásával, például elérhetőek pont- [9], egyenes- [9], kúp- [10,11], lokális affin transzformáció- [12] vagy régió-alapú [13] módszerek is.

Több algoritmus létezik multi-homográfia becslésre – egyidőben több sík megtalálására képpárok között.[2] A közkedvelt RANSAC paradigmának elérhető

[1] A cikk angol nyelvű változata a BMVC2016 konferencián jelent meg.
[2] A könnyebb érthetőség kedvéért nem általános modellekről, hanem homográfiákról beszélünk, annak ellenére, hogy a tárgyalt módszerek jórészt általános eljárások, melyek alkalmasak tetszőleges típusú modell illesztésére.

több erre a feladatra kiterjesztett változata: a szekvenciális RANSAC [14] vagy a multiRANSAC [15]. A RANSAC mohó stratégiája annak ellenére, hogy egy modell illesztésére kiváltképp alkalmas, nehézségekbe ütközik, amennyiben egyidőben többet kell megtalálnia. A J-Linkage [16] vagy a nemrégiben publikált T-Linkage [17] a preferencia térben – ami lényegében a pontokpárok síkokhoz való rendelésének költségét írja le – véletlenszerűen generált homográfiákat klaszterez. A J-Linkage ezeket a klasztereket az úgynevezett Jaccard-távolság[3] alapján vonja össze iteratív módon. A T-Linkage algoritmus ezt az elgondolást terjeszti ki folytonos preferencia térre, és a preferencia-vektorok közötti távolság-függvényt a Tanimoto-távolságra cseréli. Mindkét algoritmus az alapján dönti el, hogy egy kimenetként kapott modell fontos-e vagy sem, hogy elegendő pont tartozik-e hozzá.

Az általunk bemutatott munkához legközelebbi eljárás a Boykov és mtsai. által publikált PEARL algoritmus [18]. A PEARL a multi-homográfia illesztési feladatot tisztán egy globális energia-függvény optimalizációjaként írja le. Ebben az energiában az adattag egy pont adott homográfiához való rendelésének költségét írja le. A második, térbeli összetartozást leíró kifejezés arra a feltételezésre épül, hogy azok a pontok, amelyek közel vannak egymáshoz, nagy valószínűséggel ugyanahhoz a síkhoz is tartoznak. Az energia harmadik tagja bünteti a felhasznált homográfiák számát, ezzel elérve azt, hogy a "gyenge" homográfiákat eldobja a rendszer.

A PEARL-höz hasonlóan mi is egy az energiát minimalizáló címkézés kereséseként írjuk le a problémát. Az általunk javasolt energia hasonló: ugyanazon adat és térbeli összetartozási tagokból áll. Azonban a javasolt Multi-H algoritmusban a harmadik tagot elhagytuk, és a homográfiák számát Mean-Shift [19] és α-expansion [20] segítségével kontrolláljuk.

Kiegészítve a PEARL stratégiát egy determinisztikus inicializálási technikával és a Mean-Shift algoritmussal a Multi-H felülmúlja a PEARL algoritmust mind pontosságban, mind sebességben. Felhasználjuk a Baráth és mtsai. [12] által javasolt módszert arra, hogy minden egyes pont-megfeleltetéshez és a hozzá tartozó lokális affin transzformációhoz egy egyedi homográfiát becsüljünk. Másik fontos tulajdonsága az algoritmusnak, hogy nem a pontok száma alapján döntjük el egy síkról, hogy erős-e, hiszen ez a felhasználási területtől függ. Például a kis kiterjedésű síkok hasznosak lehetnek rekonstrukcióban, míg a domináns síkok meghatározásában nem játszanak nagy szerepet.

A cikk eredményei: (i) egy módszer, amely a pont-megfeleltetéseket közös síkokhoz rendeli az érintősíkjaik hasonlósága alapján – így becsül pontos felületi normálisokat. Azáltal, hogy nem döntünk egy sík fontosságáról, mind a gyenge, mind az erős síkokból hasznos információ származik. (ii) Megmutattuk, hogy az általánosan használt véletlenszerű kezdeti homográfia-generálás javítható és felcserélhető egy determinisztikus algoritmussal. A Multi-H eredményei ezáltal jelentősen felülmúlják a korszerű multi-homográfia becslő módszereket.

[3] A Jaccard-távolság két halmaz átfedését méri.

(iii) Bevezetünk és publikusan elérhetővé teszünk új, nagyobb kihívást jelentő képpárokat.[4].

2.. Multi-homográfia Becslés – Multi-H

A Multi-H algoritmus minden pontmegfeleltetéshez egy érintősíkot rendel, olyan módon, hogy közös síkok szerint csoportosítja őket. Az egyetlen elvárt bemenet egy képpár. A kimenet homográfiák halmaza – melyek leírják a közös érintősíkokat – és egy cimke minden egyes megfeleltetéshez, amely hozzárendeli azt egy síkhoz.

2.1.. Pontmegfeleltetések és Lokális Affin Transzformációk

Számos módszer létezik lokális affin transzformációk becslésére. Mi az affin-kovariáns jellemzőpont detektorokat részesítjük előnyben [21], hiszen ezek mind a pontokat, mind a lokális transzformációkat egyazon időben szolgáltatják. A MODS[5] [22] algoritmust preferáljuk, mivel az lényegesen gyorsabb, mint az ASIFT [23]. A MODS jó minőségű affinokat szolgáltat és az epipoláris geometriát is megbecsüli, amivel a kimenetként adott pont-megfeleltetések konzisztensek. Természetesen más affin-kovariáns detektor is használható, azonban a megfeleltetéseknek konzisztensnek kell lenniük az F fundamentális mátrixszal, hiszen ezt a tulajdonságot a Multi-H felhasználja.

Jelöljük az i. homogén pontot a k. képen az alábbi módon: $\mathbf{p}_k^i = [p_k^{i,x} \quad p_k^{i,y} \quad 1]^T$, $i \in [1, N]$, $k \in \{1, 2\}$, és a hozzá tartozó affin transzformációt pedig $\mathbf{A}_k^i$-val. Pontpárok végtelen kicsi környezetei közötti transzformáció az, amely az első affint a másikban transzformálja: $\mathbf{A}^i \mathbf{A}_1^i = \mathbf{A}_2^i$. Ezért $\mathbf{A}^i = \mathbf{A}_2^i (\mathbf{A}_1^i)^{-1}$. Az $\mathbf{A}^i$ elemei sorfolytonos módon írva: a_{11}^i, a_{12}^i, a_{21}^i és a_{22}^i. Az 1. ábra néhány lokális affin

1. ábra. Egymásnak megfelelő affin transzformációk ellipszisekkel ábrázolva.

[4] `http://web.eee.sztaki.hu/~dbarath/`

[5] `http://cmp.felk.cvut.cz/wbs/`

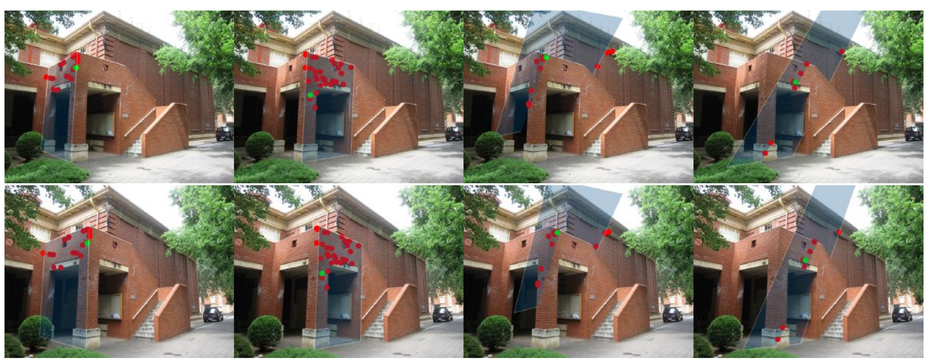
transzformációt ábrázol ellipszisekkel. Annak érdekében, hogy ezeket az affinokat pontosítsuk, alkalmazzuk rájuk az EG-L_2-Opt korrekciót [24].

Minden egyes pontpárhoz és $\mathbf{A}_i$-hez megbecslünk egy $\mathbf{H}_i$ homográfiát a HAF módszer [12] felhasználásával. Ez a módszer alkalmas arra, hogy már egyetlen pontpárból, a hozzá tartozó affin transzformációból és a fundamentális mátrixból kiszámolja a homográfiát. A becsléshez felír egy lineáris, inhomogén $\mathbf{Cx} = \mathbf{b}$ alakú egyenletrendszert, ahol az együttható mátrix:

$$\mathbf{C} = \begin{bmatrix} a^i_{11}p^{i,x}_1 + p^{i,x}_2 - e^x & a^i_{11}p^{i,y}_1 & a^i_{11} \\ a^i_{12}p^{i,y}_1 + p^{i,x}_2 - e^x & a^i_{12}p^{i,x}_1 & a^i_{12} \\ a^i_{21}p^{i,x}_1 + p^{i,y}_2 - e^y & a^i_{21}p^{i,y}_1 & a^i_{21} \\ a^i_{22}p^{i,y}_1 + p^{i,y}_2 - e^y & a^i_{22}p^{i,x}_1 & a^i_{22} \end{bmatrix}, \tag{1}$$

az $\mathbf{e} = [e^x \quad e^y]^T$ pedig az epipólus a második képen. A $\mathbf{b} = [f_{21} \quad f_{22} \quad -f_{11} \quad -f_{12}]$ vektor tartalmazza a inhomogén részeit a négy egyenletnek, míg az $\mathbf{x} = [h_{31} \quad h_{32} \quad h_{33}]^T$ az ismeretlen paramétereket. A legkisebb négyzetes értelemben optimális megoldás az $\mathbf{x} = \mathbf{C}^\dagger\mathbf{b}$ egyenlettel kapható, ahol $\mathbf{C}^\dagger$ a $\mathbf{C}$ mátrix Moore-Penrose pszeudo-inverze. A homográfia mátrix végül az utolsó sorának felhasználásával kinyerhető: $h_{1j} = e^x h_{3j} + f_{2j}$, $h_{2j} = e^y h_{3j} + f_{1j}$, ahol $j \in \{1, 2\}$ és $f_{lm}, l, m \in \{1, 2, 3\}$ az $\mathbf{F}$ elemei.

2.2.. Alternáló Minimalizáció



2. ábra. A felső és alsó sorban levő képek a `johnsona` teszt két képe. A kék színű poligonok homográfiákat vizualizálnak, melyek egybeesnek (első két oszlop) vagy nem esnek egybe (második két oszlop) érintősíkokkal. A megfeleltetés, amelyből az adott homográfiát becsültük, zölddel van jelölve. A ϵ-küszöböléssel kapott inliereket pirossal jelöltük, ahol $\epsilon = 3.0$ pixel.

A 2.1. fejezetben ismertetett inicializáció után a meghatározott homográfiákat egy három lépésből álló alternáció segítségével javítjuk (lásd az 1. algoritmust).

Algorithm 1 A Multi-H Algoritmus.

Bemenet: I_1, I_2 – képek; $P, A, F :=$ MODS(I_1, I_2) [22]
P - pont-megfeleltetések; A – affin transzformációk; F – fundamentális mátrix
Kimenet: $\mathcal{H}$ – a kapott homográfiák; L – a címkézés

```
ℋ⁰ := HAF(P, A, F) [12]                          ▷ Homográfiák becslése pontonként
i := 0;
repeat                                           ▷ Alternáló minimalizáció
    i := i + 1;
    ℋⁱ := MeanShift(ℋⁱ⁻¹)                         ▷ Alapértelmezett ϵ = 2.7
    Lⁱ := α-expansion(P, ℋⁱ)                      ▷ Alapértelmezett λ = 0.5, γ = 0.005
    ℋⁱ := LSQHomographyRefinement(P, A, Lⁱ, F)
until Convergence                                ▷ if ℋⁱ = ℋⁱ⁻¹ ∧ Lⁱ = Lⁱ⁻¹
ℋ := ℋⁱ; L := Lⁱ
```

(1) Mean-Shift. A 2. ábra azt mutatja, hogy az inicializáció után számos homográfia egybeesik (első két oszlop) és számos nem esik egybe (második két oszlop) valódi érintősíkokkal. Az egyes oszlopokban a pont, amelyből a homográfiát számoltuk, zölddel van jelölve, míg a ϵ-inlierei pirossal, ahol $\epsilon = 3.0$ pixel küszöböt használtunk. Az érintősíkokat kék poligonokkal jelöltük.

Azt feltételezzük, hogy azok a síkhomográfiák, amelyek több ponthoz is tartoznak, csomópontokként jelennek meg a homográfiák terében. Mivel a színtéren levő homográfiák számát nem ismerjük, a Mean-Shift [19] algoritmust használtuk ezen csomópontok kinyerésére. Az i. homográfia leírására a következő 6D teret konstruáltuk:

$$\mathbf{v}^i = \begin{bmatrix} w_1^{i,x} & w_1^{i,y} & w_2^{i,x} & w_2^{i,y} & w_3^{i,x} & w_3^{i,y} \end{bmatrix}, \tag{2}$$

ahol

$$\mathbf{w}_1^i = \frac{\mathbf{H}^i[0 \quad 0 \quad 1]^T}{H_{33}^i}, \quad \mathbf{w}_2^i = \frac{\mathbf{H}^i[1 \quad 0 \quad 1]^T}{H_{13}^i + H_{33}^i}, \quad \mathbf{w}_3^i = \frac{\mathbf{H}^i[0 \quad 1 \quad 1]^T}{H_{23}^i + H_{33}^i}.$$

Az egyes $\mathbf{w}^i$-k nevezője a transzformált ponthoz tartozó projektív mélység. A $\mathbf{v}^i$ vektorok meghatározzák a leírt homográfiát, hiszen az a $[0 \quad 0 \quad 1]^T$, $[1 \quad 0 \quad 1]^T$, $[0 \quad 1 \quad 1]^T$ pontokból és a transzformáltjukból kiszámolható, amennyiben ismert a fundamentális mátrix [12,9]. Annak ellenére, hogy számos lehetséges reprezentációja van egy homográfiának (pl. az elemei felhasználásával, négy pont vetítésével, stb.), mi a lehető legkevesebb dimenziót preferáljuk – a Mean-Shift algoritmus futási ideje erősen összefügg a dimenziószámmal.

Mivel az egyes $[v_k^i \quad v_{k+1}^i]$, $k \in \{1, 3, 5\}$ koordináta-párok pontjait a $\mathbf{H}^i$-vel transzformáltjuk, a d távolságnak megfelelő választás az átlagos Euklideszi-távolsága a három koordináta párnak. Tehát a távolság az i. és a j. vektor között az alább látható:

$$d(\mathbf{v}^i, \mathbf{v}^j) = \frac{1}{3} \sum_{k=1}^{3} ||[v_{2(k-1)+1}^i \quad v_{2(k-1)+2}^i]^T - [v_{2(k-1)+1}^j \quad v_{2(k-1)+2}^j]^T||_2.$$

(2) Az α-expansion-t [20] használó lépés a következő energiát minimalizálja:

$$E(L) = \frac{1}{\lambda} E_d(L) + \lambda E_s(L), \tag{3}$$

ahol L az aktuális címkézés, $E_d(L)$ és $E_s(L)$ pedig az adattag és a térbeli összetartozást leíró kifejezés. A λ paraméter az egyes kifejezések egymáshoz viszonyított fontosságát konfigurálja. Az adattag az alábbi módon definiált:

$$E_d(L) = \sum_{i=1}^{N} \| \mathbf{p}_2^i - \frac{\mathcal{H}^{l_i} \mathbf{p}_1^i}{\mathcal{H}_{31}^{l_i} p_1^{i,x} + \mathcal{H}_{32}^{l_i} p_1^{i,y} + \mathcal{H}_{33}^{l_i}} \|_2, \tag{4}$$

ahol $\mathcal{H}^{l_i}$ az i. megfeleltetés $l_i \in L$ cimkéjére vonatkozó homográfia.

Az E_s tag arra feltételezésre reflektál, miszerint a közeli pontok nagyobb valószínűséggel tartoznak ugyanahhoz a homográfiához. Értéke egészen pontosan a különböző cimkével ellátott szomszédos pontok száma:

$$E_s(L) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{A}_{ij} [\![l_i \neq l_j]\!], \tag{5}$$

ahol N a megfeleltetések száma, a $[\![.]\!]$ Iverson zárójel egyenlő eggyel, amennyiben a tartalmazott kifejezés igaz, ellenkező esetben pedig nulla. Az $\mathcal{A}_{ij}$ a szomszédossági mátrix egy eleme és egyenlő eggyel, ha az i. és j. megfeleltetések szomszédok, egyébként pedig nulla az értéke. Két pont-megfeleltetést akkor tekintünk szomszédoknak, ha a távolságuk a 4D összefűzött koordináta rendszerben γ-nál kisebb – az egyes megfeleltetéshez tartozó vektor $[p_1^x \quad p_1^y \quad p_2^x \quad p_2^y]^T$. Az $\mathcal{A}$ mátrix hatékonyan számolható a FLANN algoritmus [25] segítségével.

Az energia ebben a lépésben nem növekedhet az α-expansion algoritmus természete miatt. Azokat a pontokat nem rendeljük semelyik síkhoz, amelyek távolsága a legközelebbi síktól nagyobb, mint 3ϵ, ahol az ϵ egy empirikus úton beállított küszöb.

(3) A Legkisebb Négyzetes Homográfia Illesztés a HAF [12] módszert használja az adott homográfiák paramétereinek finomhangolására a címkézés alapján. Minden megfeleltetést felhasználva, amit a címkézés az adott homográfiához rendel, legkisebb négyzetes illesztés hajtható végre, ezzel minimalizálva az L_2-es hibát. Ezen lépés során nem változhat a homográfiák száma. Az energia csökken, vagy nem változik, mivel E_d a visszavetítési hibák összege és ezt minimalizáljuk itt. Az E_s értéke változatlan marad, hiszen a címkézés nem változik a lépés során.

A konvergenciát eléri az algoritmus, amikor se a homográfiák száma, se az energia nem változik két iteráció alatt. Mivel az első lépés nem növeli a számot, a többi csökkenti az energiát és a lehetséges címkézések száma véges, a konvergencia biztosított. A 3.. fejezetben bemutatott kísérleteken az 1. algoritmus legfeljebb nyolc iterációs lépés alatt konvergált.

3.. Kísérleti Eredmények

3.1.. Összehasonlítás Multi-homográfiát Becslő Eljárásokkal

Ezen fejezetben a Multi-H algoritmust teszteljük multi-homográfia illesztési feladatra és megmutatjuk, hogy felülmúlja a korszerű módszereket.

Lényeges síkok meghatározása. Annak érdekében, hogy meghatározzuk, mikor lényeges egy sík és mikor nem, anélkül, hogy erős megkötéseket tennénk a minimum inlier számra, az alábbi algoritmust használjuk: (1) Először minden olyan síkot eltávolítunk, amelyhez kevesebb, mint négy pont tartozik. (2) Újraillesztjük a homográfiákat a normalizált négy pontos algoritmussal [9], majd a kapott eredményt a Levenberg-Marquardt optimalizációval finomítjuk, ezzel minimalizálva a visszavetítési hibát. (3) Az ismert kompatibilitási megkötés [9] homográfiára és fundamentális mátrixra vonatkozóan az alábbi: $\mathbf{H}^T\mathbf{F} + \mathbf{F}^T\mathbf{H} = 0$. Ezt felhasználva törlünk minden olyan $\mathbf{H}_i$ homográfiát, melyre $||\mathbf{H}_i^T\mathbf{F} + \mathbf{F}^T\mathbf{H}_i||_F > \theta$. A paraméter hangolása után a $\theta = 1.0$ értéket megfelelőnek találtuk a célra.

1. táblázat. Félreosztályozási hiba (%) kétképes sík szegmentációra. A kiválasztott képpárok az AdelaideRMF adathalmazból ugyanazok, mint [17]-ben. A ground truth síkok számát R-el jelöltük.

	R	PEARL	QP-MF	FLOSS	ARJMC	SA-RCM	J-Lnkg	T-Lnkg	**Multi-H**
johnsonna	4	4.02	18.50	4.16	6.48	5.90	5.07	4.02	**2.41**
johnsonnb	7	18.18	24.65	18.18	21.49	17.95	18.33	18.17	**4.46**
ladysymon	2	5.49	18.14	5.91	5.91	7.17	9.25	5.06	**0.00**
neem	3	5.39	31.95	5.39	8.81	5.81	3.73	3.73	**0.00**
oldclassicswing	2	1.58	13.72	1.85	1.85	2.11	0.27	0.26	**0.00**
sene	2	0.80	14.00	0.80	0.80	0.80	0.84	0.40	**0.00**
mean		5.91	20.16	6.05	7.56	6.62	6.25	5.30	**1.19**
median		4.71	18.32	4.78	6.20	5.86	4.40	3.87	**0.00**

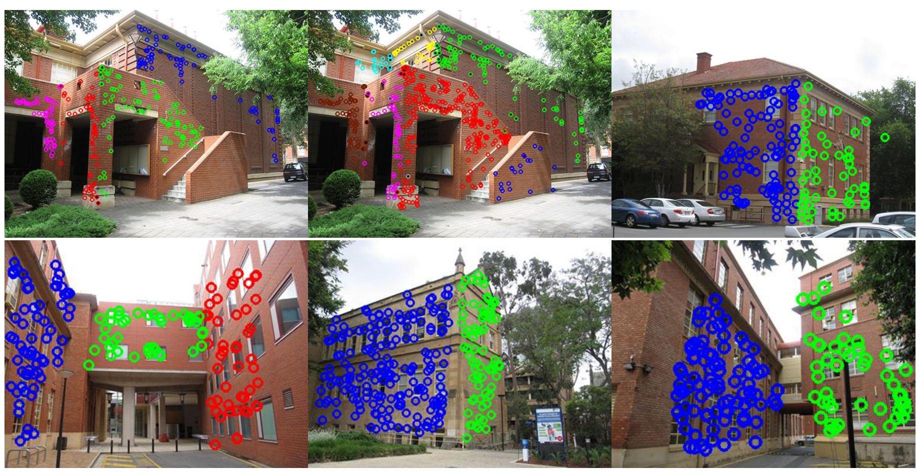
	J-Lnkg	T-Lnkg	RPA	SA-RCM	Grdy-RansaCov	ILP-RansaCov	**Multi-H**
mean	25.50	24.66	17.20	28.30	26.85	12.91	**4.40**
median	24.48	24.53	17.78	29.40	28.77	12.34	**2.41**

A Multi-H-t a [17]-ben javasolt módon teszteltük az AdelaideRMF adathalmazon – amely minden egyes képpárhoz tartalmaz pontmegfeleltetéseket domináns síkokhoz rendelve. Ellenben az adathalmaz lokális affin transzformációkat nem tartalmaz, ezért a MODS [22] detektorral annyi megfeleltetést generáltunk, amennyit lehetséges – ez egy paramétere a MODS-nak. Végül megkerestük a legközelebbi detektált megfeleltetést minden egyes párhoz az adathalmazból. Ezeket a megfeleltetéseket és a hozzájuk tartozó affinitásokat használtuk bemenetként a Multi-H-hoz.

A félreosztályozási hibát (FH) a következőképp számoltuk: először megkerestük a leképezést az L_{GT} ground truth és a Multi-H által javasolt L címkézések között. Erre egy iteratív módszer használtunk, amely minden egyes Multi-H homográfiához azt a ground truth homográfiát rendeli, amellyel a legnagyobb az átfedése. Megjegyezzük, hogy amennyiben ez a hozzárendelés nem optimális, akkor a félreosztályozási hibákat felülről becsültük. Az FH egyenlő $\sum_{i=1}^{N} [\![l_{GT}^i \neq l^i]\!]$-vel, amely a rosszul osztályozott és az összes pont aránya.

Az első kísérletben a Multi-H-t a T-Linkage [17], ARJMC [26], PEaRL [18], QP-MF [27], FLoSS [28], J-Linkage [16] és SA-RCM [29] algoritmusokkal hasonlítottuk össze (lásd 1. táblázatot). Minden módszer paraméterei, beleértve a javasoltat, minden egyes képpárra külön lettek behangolva. Habár, mi a fix paraméterekkel kapott eredményeket preferáljuk – ezeket a fejezet végén be is mutatjuk –, a képpáronként hangolt eredményeket bemutatása teszi lehetővé, hogy az irodalomban elérhető eredményekkel összehasonlítsuk a javasolt módszert.

Az 1. táblázat azt mutatja, hogy a Multi-H szolgáltatja a legkisebb átlagos és medián hibákat azon a hat képpáron amelyet a [17] cikk is használ. A 3. ábra a Multi-H pontokat mutatja, színekkel jelölve a síkot, amelyhez tartoznak.

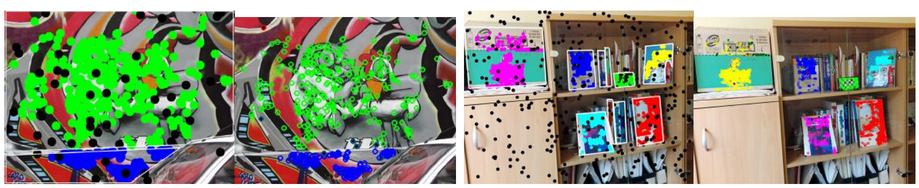


3. ábra. A Multi-h által adott particionális az AdelaideRMF adathalmazon. A síkokat színekkel jelöltük. Látható néhány félreosztályozott pont a bal-felső és bal-középső képeken az élek közelében. Ezeket apró, kitöltött, fekete körökkel jelöltük.

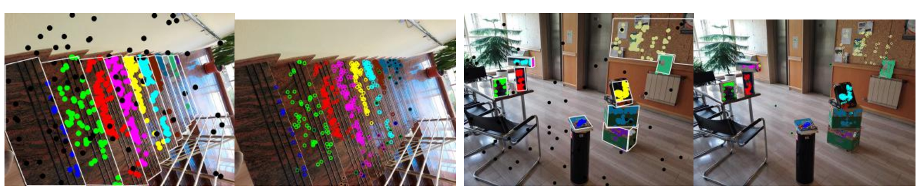
A 3.1.. táblázat az átlag és medián félreosztályozási hibákat mutatja az AdelaideRMF mind a 19 képén. A többi módszer a T-Linkage [17], J-Linkage [16], RPA [30], SA-RCM [29], Greedy-RansaCov [31] és az ILP-RansaCov [31]. Multi-

H jelentősen felülmúlja az összes megvizsgált módszert. Megjegyezzük, hogy jelentős különbség látszik azon a hat képpáron – amit általában használnak az irodalomban (1. táblázat) – és az összesen elért eredmények között.

Annak ellenére, hogy ez az adathalmaz a leggyakrabban használt multi-homográfia illesztésre, általánosságban könnyű teszteseteket tartalmaz, amelyeknél a megfigyelt síkok merőlegesek vagy messze vannak egymástól. Annak érdekében, hogy teszteljük a Multi-H pontosságát, nagyobb kihívást jelentő képpárokat készítettünk – ezek láthatóak a 4. ábrán. Ezeken a képeken a MODS [22] algoritmussal detektáltunk pontmegfeleltetéseket, majd manuálisan síkokhoz rendeltük őket. Legvégül outliereket, hibás pontpárokat adtunk a detektált halmazhoz. A 4. ábra egyes képpárainál az első kép a ground truth szegmentáció, a második a kinyert síkpartícionális. Az outliereket fekete pontokkal abrázoltuk a képeken. A 4(a). pár ismert a közkedvelt `graffiti` tesztsorozatból.[6] Itt két alig különböző sík látható – az alsó sík alig láthatóan közelebb van a kamerához, mint a felső. Ennek ellenére a Multi-H pontosan különbözteti a két síkot és alacsony, 1.19%-os hibát ér el. A 4(b). képpár egy könyvespolcot, míg a 4(c) egy lépcsőt ábrázol felülről nézve. Az utolsó két képen (4(d).) egy szoba látható dobozokkal és síkszerű objektumokkal. Ezek a tesztek nehezebbek egy multi-homográfia detektáló algoritmusnak, hiszen az épületekkel ellentétben a megfigyelt síkok közel vannak egymáshoz, az orientációjuk egészen hasonló is lehet, és kis kiterjedésűek (lásd például a könyvgerinceket a `glasscasea` példán).



(a) `graffiti` (FH = 1.19%) (b) `glasscasea` (FH = 3.69%)



(c) `stairs` (FH = 8.74%) (d) `boxesandbooks` (FH = 3.14%)

4. ábra. Négy képpár az új adathalmazból. A pontok, érintősíkjuktól függően, különböző színűek. Kézi annotáció (bal) és Multi-H eredmény (jobb). Az FH a félreosztályozási hiba.

[6] `http://www.robots.ox.ac.uk/~vgg/research/affine/`

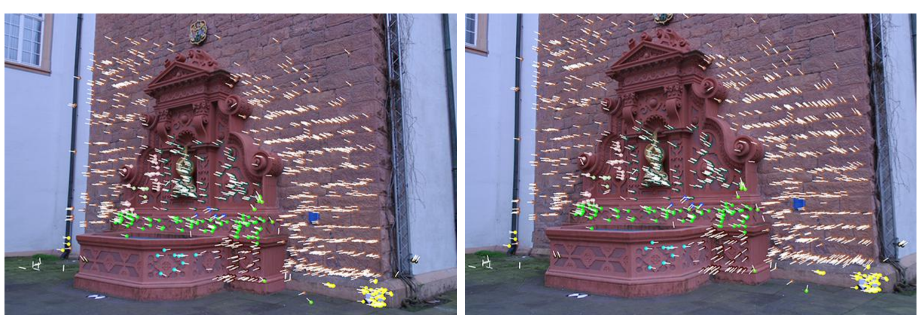
2. táblázat. Öt futtatás átlagos félreosztályozási hibája (%) fix paramétereket használva. A következő rövidítéseket használtuk: `johnsonna` (`johnsa`), `johnsonnb` (`johnsb`), `oldclassicswing` (`old`).

	`johnsa`	`johnsb`	`ladysymon`	`neem`	`old`	`sene`	`mean`	`median`
Multi-H	**9.33**	**10.14**	**4.49**	**2.00**	**1.79**	**0.00**	**4.79**	**3.74**
T-Lnkg	34.28	24.04	24.67	25.65	20.66	7.63	22.82	24.36
SA-RCM	36.73	16.46	39.50	41.45	21.30	20.20	29.27	29.02
RPA	10.76	26.76	24.67	19.86	25.25	0.42	17.95	22.27

A javasolt konfiguráció. Praktikus okok miatt előnyös, hogy ha egyetlen fix paraméter-beállítás lefedi az esetek többségét. Kimerítő kereséssel meghatároztuk, hogy a $\lambda = 0.5$, $\epsilon = 2.7$ és a $\gamma = 0.005$ paraméterek jó eredményeket adnak. A 2. táblázat a félreosztályozási hibát mutatja az AdalaideRMF adathalmazon (5 futtatás átlaga). Az eredmények rosszabbak, mint a képenként hangoltak (lásd 1. táblázat), de a Multi-H így is lényegesen jobban teljesít, mint a többi módszer fix paraméterekkel[7].

3.2.. Felületi Normálisok Becslése

Ebben a fejezetben a Multi-H által becsült felületi normálisokat hasonlítjuk össze az affin kovariáns detektorok pontpáronként kapott eredményeivel. A pontosságot a fountain-P11 adathalmazon [32] értékeltük ki, amely 11 darab 3072×2048-as képet, projekciós és kalibrációs mátrixokat, illetve 3D-s irányított pontfelhőket tartalmaz. Pontmegfeleltetéseket MODS [22]-al detektáltunk a képpárok között – átlagosan 920-at.



5. ábra. Érintősíkukhoz rendelt megfeleltetések a fountain-P11 első két frame-én. A síkokat színek jelölik, a felületi normálisokat pedig fehér szakaszok.

[7] Az eredményeket a [30]-ből másoltuk.

A Multi-H az érintősíkjaik alapján particionálja a megfeleltetéseket (lásd 6. ábra), majd egyetlen homográfiát illeszt a klaszter összes pontjára. Ekkor a felületi normális könnyen számolható a homográfia paramétereiből, hiszen a kamerák paramétereit ismerjük. A 3. táblázat (3. sor a Multi-h eredménye) mutatja az átlagos és medián szöghibákat a becsült és a ground truth normálisok között. A Multi-H általt becsült normálisok lényegesen pontosabbak, mint a detektorok által becsült lokális transzformációkból számoltak (3. táblázat, első sor). Az EG-L_2-Opt procedúra [8] által becsült normálisok is lényegesen kevésbé pontosak (3. táblázat, második sor).

3. táblázat. A becsült normálisok átlagos és medián szöghibái (fokban) a kiválasztott képpárok között.

Frames	1 − 2	3 − 5	1 − 5	6 − 8	5 − 9
Affin Detektor	35.7 | 32.7	24.9 | 20.3	19.0 | 15.8	22.5 | 18.6	20.0 | 15.4
EG-L_2-Optimális	35.5 | 32.5	23.1 | 19.8	16.7 | 13.9	19.9 | 16.6	17.8 | 14.4
Multi-H	**14.4 | 9.4**	**9.0 | 7.5**	**7.0 | 5.8**	**8.8 | 7.3**	**7.1 | 5.7**

3.3.. Feldolgozási Idő és az Implementáció Részletei

A Multi-H procedúra sebességét egy 100 és egy 500 megfeleltetést tartalmazó ponthalmazon teszteltük. Az 1. algoritmus átlagos iteráció száma megközelítőleg 6 volt mindkét esetben. Az átlagos feldolgozási idő pedig 0.04 és 0.80 másodperc a két ponthalmazra egy asztali PC-n Intel Core i5-4690 CPU, 3.50 GHz processzorral 4 magot használva.

A 6(a). ábra egyes oszlopai az egyes képpárokra vonatkozó feldolgozási időt jelölik tizedmásodpercben. Az egyes sávok az algoritmus különböző lépéseit mutatják. Az adatok azt mutatják, hogy a Multi-H időigénye elhanyagolható a pontdetekciós lépéshez képest (MODS). A szomszédossági mátrix és a pontonkéni homográfiák kiszámolására vonatkozó sávok nem láthatóak, hiszen azok időigénye megközelítőleg $4-6$ tizedmásodperc.

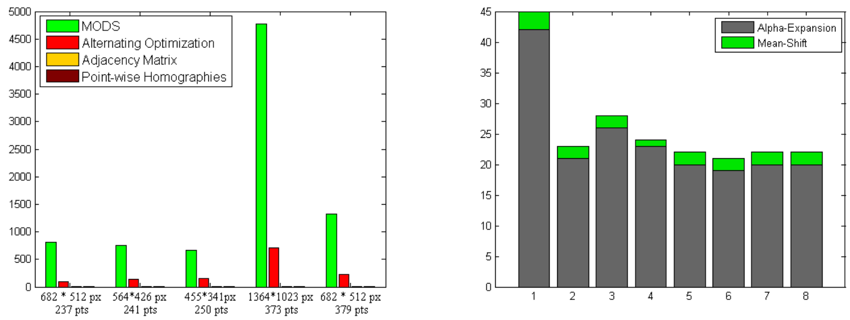
A 6(b). ábra az alternáló minimalizáció feldolgozási idejét mutatja. Láthatólag lényegesen lecsökken az idő az első lépés után, majd konstansnak tűnik. A csökkenés a Mean-Shift algoritmusnak köszönhető, amely lényegesen redukálja a homográfiák számát, ezzel gyorsulást előidézve a α-expansion algoritmusban.

A Multi-H-t C++ nyelven implementáltuk és a GCOptimization[9] könyvtárban implementált α-expansion-t használtuk. Gyors Mean-Shift implementációt a webről töltöttünk le[10].

[8] Az EG-L_2 módszer a lokális affin transzformációkat javítja az epipoláris geometria felhasználásával

[9] `http://vision.csd.uwo.ca/code/`

[10] `http://scikit-learn.org/stable/modules/clustering.html#mean-shift`



(a) (b)

6. ábra. **(a)** A Multi-H feldolgozási ideje (tizedmásodpercben) különböző képpárokon. A vízszintes tengely az egyes oszlopoknál a képfelbontást és megfeleltetés számot jelöli. **(b)** Az 1-8. iterációs lépések futási ideje a `hartley` teszten.

4.. Összefoglalás

Egy újszerű megközelítést javasoltunk érintősíkok becslésére képpárok közötti pont-megfeleltetések particionálása által. A módszer pontos és felülmúlja a korszerű multi-homográfia-becslő technikákat mind fix, mind képről-képre hangolt paraméterekkel. A kísérletek azt mutatták, hogy a szabványos adathalmaz relatíve könnyű, ezért kiegészítettük az adatokat új, nagyobb kihívásnak bizonyuló képpárokkal.

A legtöbb alkalmazásban a Multi-H lényegesen gyorsabb lesz, mint amilyen gyorsan az affin kovariáns detektorok nyerik ki az eredményüket. Az algoritmus valós idejű, amennyiben a pontok száma legfeljebb 300. Azonban az α-expansion GPU implementációja [33] valós idejűvé teheti nagyobb problémákra is.

A forráskód és az annotált adathalmaz elérhető a `http://web.eee.sztaki.hu/home4/node/56` címről.

Hivatkozások

1. Z. Zhang, „A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence (PAMI)*, pp. 1330–1334, 2000.
2. T. Ueshiba and F. Tomita, „Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices," in *International Conference on Computer Vision (ICCV)*, 2003, pp. 966–973.
3. Z. Chuan, T. D. Long, Z. Feng, and D. Z. Li, „A planar homography estimation method for camera calibration," in *Computational Intelligence in Robotics and Automation*, 2003, pp. 424–429.
4. J. Zhou and B. Li, „Robust ground plane detection with normalized homography in monocular sequences from a robot platform," in *Image Processing*, 2006, pp. 3017–3020.

5. J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, „Homography-based visual servo tracking control of a wheeled mobile robot," *Robotics*, pp. 406–415, 2006.
6. G. Simon, A. W. Fitzgibbon, and A. Zisserman, „Markerless tracking using planar structures in the scene," in *International Symposium on Augmented Reality (ISAR)*, 2000, pp. 120–128.
7. Z. Zhang and A. R. Hanson, „3d reconstruction based on homography mapping," *Proc. ARPA96*, pp. 1007–1012, 1996.
8. T. Werner and A. Zisserman, „New techniques for automated architectural reconstruction from photographs," in *European Conference on Computer Vision (ECCV)*, 2002, pp. 541–555.
9. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.
10. P. K. Mudigonda, C. V. Jawahar, and P. J. Narayanan, „Geometric structure computation from conics." in *ICVGIP*, 2004, pp. 9–14.
11. A. Sugimoto, „A linear algorithm for computing the homography from conics in correspondence," *Journal of Mathematical Imaging and Vision (JMIV)*, pp. 115–130, 2000.
12. D. Barath and L. Hajder, „Novel ways to estimate homography from local affine transformations," in *Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2016, pp. 432–443.
13. J. Nemeth, C. Domokos, and Z. Kato, „Recovering planar homographies between 2d shapes," in *12th International Conference on Computer Vision (ICCV)*, 2009, pp. 2170–2176.
14. E. Vincent and R. Laganiére, „Detecting planar homographies in an image pair," in *Image and Signal Processing and Analysis (ISPA)*, 2001, pp. 182–187.
15. M. Zuliani, C. S. Kenney, and B. S. Manjunath, „The multiransac algorithm and its application to detect planar homographies," in *International Conference on Image Processing (ICIP)*, 2005, pp. III–153.
16. R. Toldo and A. Fusiello, „Robust multiple structures estimation with j-linkage," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 537–547.
17. L. Magri and A. Fusiello, „T-linkage: a continuous relaxation of j-linkage for multi-model fitting," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3954–3961.
18. H. Isack and Y. Boykov, „Energy-based geometric multi-model fitting," *International Journal of Computer Vision (IJCV)*, pp. 123–147, 2012.
19. D. Comaniciu and P. Meer, „Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence (PAMI)*, pp. 603–619, 2002.
20. Y. Boykov, O. Veksler, and R. Zabih, „Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence (PAMI)*, pp. 1222–1239, 2001.
21. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, „A comparison of affine region detectors," *International Journal of Computer Vision*, pp. 43–72, 2005.
22. D. Mishkin, J. Matas, and M. Perdoch, „MODS: Fast and robust method for two-view matching," *Computer Vision and Image Understanding*, pp. 81–93, 2015.
23. J.-M. Morel and G. Yu, „ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, pp. 438–469, 2009.
24. D. Barath, J. Matas, and L. Hajder, „Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry," in *27th British Machine Vision Conference (BMVC)*, 2016.

25. M. Marius and G. D. Lowe, „Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application (VISAPP)*, 2009, pp. 331–340.
26. T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, „Simultaneous sampling and multi-structure fitting with adaptive reversible jump mcmc," in *Advances in Neural Information Processing Systems*, 2011, pp. 540–548.
27. J. Yu, T.-J. Chin, and D. Suter, „A global optimization approach to robust multi-model fitting," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2041–2048.
28. N. Lazic, I. Givoni, B. Frey, and P. Aarabi, „Floss: Facility location for subspace segmentation," in *International Conference on Computer Vision (ICCV)*, 2009, pp. 825–832.
29. T.-T. Pham, T.-J. Chin, J. Yu, and D. Suter, „The random cluster model for robust geometric fitting," *Pattern Analysis and Machine Intelligence (PAMI)*, pp. 1658–1671, 2014.
30. L. Magri and A. Fusiello, „Robust multiple model fitting with preference analysis and low-rank approximation."
31. ——, „Multiple model fitting as a set coverage problem," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3318–3326.
32. C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, „On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.
33. V. Vineet and P. J. Narayanan, „Solving multilabel mrfs using incremental α-expansion on the gpus," in *Computer Vision (ACCV)*, 2009, pp. 633–643.