

Gráf-vágás RANSAC

Baráth Dániel^{1,2}, Matas Jiri²

¹ MTA SZTAKI, Gépi érzékelés kutatólaboratórium, Budapest, Magyarország

² Centre for Machine Perception, Czech Technical University, Prága, Csehország

Kivonat Ezen értekezésben egy új robusztus becslő módszert javasolunk, a Graph-Cut RANSAC-et. Ez a módszer az inlierek és outlierok elkülönítésére a gráf-vágás algoritmust használja lokális optimalizációs (LO) lépésként, a becslött modellek paramétereinek javítására. A javasolt LO lépés koncencionálisan egyszerű, könnyen implementálható, globálisan optimális és hatékony. A tesztek azt mutatják, hogy a GC-RANSAC geometriailag pontosabb modellekhez vezet, mint a state-of-the-art algoritmusok számos probléma esetén: egyenes illesztésre, homográfia, affin transzformáció, fundamentális és esszenciális mátrix becslésre. A módszer számos esetben valós idejű, hasonló sebességgel (ezredmásodpercek standard CPU-n), mint a kevésbé pontos technikák.¹

1. Bevezetés

A RANSAC (RANdom SAMple Consensus) algoritmust 1981-ben javasolta Fischler és Bolles [7] és azóta a számítógépes látás egyik legerterjettebb robusztus modell becslő algoritmusává vált. A RANSAC és hasonló *hipotetizáló-és-verifikáló* megközelítéseket a számítógépes látás számos problémájára alkalmazták sikerrel. Például, rövid bázis távolságú sztereó [25,27] problémákra, széles bázis távú megfeleltetésre [21,16,17], mozgás szegmentációra [25], kép mozaikolásra [9], geometriai primitívek detekciójára [23], multi-modell illesztésre [29], vagy akár multi-modell illesztő algoritmusok inicializálására [12,20]. Röviden, a RANSAC algoritmus ismételten véletlenszerű mintákat választ ki a bemeneti adatok közül, a kiválasztott mintára illeszt egy modellt, pl. egyenest két pontra, vagy fundamentális mátrixot hét pont-megfeleltetésre. Második lépésként megszámolja, hogy az adott modellnek hány inliere van (olyan pont, mely egy megadott küszöbnél közelebb esnek a modellhez). Végezetül, a legtöbb inlierrel rendelkező modell paramétereit finomítja, például, legkisebb-négyszetes illesztéssel.

Az utóbbi három évtizedben, számos változata jelent meg a RANSAC algoritmusnak. Például, a NAPSAC [19], PROSAC [4] vagy EVSAC [8] módszerek a mintavételezési stratégia módosításával próbálják annak az esélyét növelni, hogy minél előbb egy jó, csak inlierekből álló mintát találjunk. A NAPSAC azt feltételezi, hogy az inlierek térben összefüggőek struktúrát alkotnak; a PROSAC azt használja fel, hogy a legtöbb esetben van valamilyen kezdeti feltételezésünk arról, hogy mekkora valószínűséggel inliere egy adott pont; az EVSAC algoritmus egy valószínűségi modellt épít fel arra, hogy egyes pontok mekkora valószínűséggel lehetnek inlierek, és ezt a modellt frissíti adaptívan az algoritmus futása közben. A modell minőségének mérésére is számos javaslat

¹ A cikk megjelent a CVPR 2018 konferencián. Kuba Attila dőmjra pŰlyŰqŰş cikk.

született. Az MLESAC [26] és MSAC [10] technikákban a minőséget egy maximum likelihood eljárással mérik. A gyakorlat azt mutatja, hogy az MLESAC megközelítés gyakran jobb, mint a RANSAC inlier számolása és kevésbé érzékeny a felhasználó által megadott küszöb paraméterre. Azt, hogy a RANSAC mikor terminál egy, a felhasználó által megadott η konfidenciával kontrolláljuk, és akkor történik meg, amikor annak a valószínűsége, hogy az eddigi legjobbnál jobb modell találjunk η alá csökken.²

Azt megfigyelve, hogy a RANSAC a gyakorlatban több iterációt igényel, mint amit az elmélet indokolna, Chum és mtsai. [5] felismerte a problémát, hogy nem minden csak inlierekből álló minta „jó”, vagyis vezet olyan modellhez, ami eléggé pontos az összes inlier megkülönböztetéséhez. Ilyen eset fennállhat például a kiválasztott pontok rosszul kondicionáltsága miatt is. Ezen probléma megoldására javasolják az ún. lokálisan optimalizált RANSAC-et (LO-RANSAC), mely kiegészíti az eredeti módszert egy, csak a legjobb modellekre alkalmazott lokális optimalizációs lépéssel. Az eredeti cikkben [5] ez a lokális optimalizáció egy belső RANSAC-be ágyazott iterált legkisebb-négyzetes illesztés, ahol a használt inlier-outlier küszöb folyamatosan csökken. Az LO lépést csak az aktuális modell inliereire alkalmazzuk. A bemutatott teszteken az LO-RANSAC felülmúlta a standard RANSAC-et mind pontosság, mind a legjobb modell megtalálásához szükséges iterációk számának tekintetében. Az LO futtatások száma közel áll a verifikációk számának logaritmusához, ezért nem jelent szignifikáns futási idő növekedést a legtöbb esetben. Ennek ellenére, ahogy azt Lebeda és mtsai. [15] megmutatták, a sok inlierral rendelkező modellek esetén ez a lokális optimalizációs lépés futási ideje kritikussá válhat az iterált legkisebb négyzetes illesztés számítási igénye miatt. Ezen probléma enyhítésére azt javasolják, hogy csak $7m$ méretű véletlenszerű inlier mintákra alkalmazzuk az LO lépést, ahol az m a minimális minta mérete (pl. $m = 2$ egyenesek esetén); és a 7 faktort számos adatbázison végzett kimerítő kereséssel állították meg. A lokális optimalizációt a state-of-the-art RANSAC módszerekbe, pl. USAC [22], is beépítették. Ellenben az eredeti LO ad hoc, komplex és számos paraméterre van szüksége.

Ebben az értekezésben a számítógépes látás két különböző szálát kombináljuk össze annak érdekében, hogy egy state-of-the-art RANSAC-et kapjunk. A RANSAC-hez kapcsolódó kutatásokban az inlier-outlier döntés mindig egyes adatpont a modelltől vett távolságának függvénye és minden egy ponthoz külön-külön, a többitől függetlenül döntjük el. Geometriai modell-illesztésben azonban sok esetben feltételezhetjük, hogy az inlierek és outlierek térben összefüggő struktúrákat alkotnak. Tehát egy pont, ami közel van egy inlierhez nagy valószínűséggel maga is inlier. A térbeli összefüggőséget, amit a Potts modell [3] megfogalmaz, a számítógépes látás számos problémájánál vették már figyelembe. Például szegmentációnál [28], multi-modell illesztésnél [12,20] vagy mintavételezésnél [19]. RANSAC esetén eddig csak és kizárólag a mintavételezés javítására használták a NAPSAC [19] algoritmusban. Annak érdekében, hogy bevonjuk ezt a feltételezést a RANSAC algoritmusba, egy kézenfekvő megoldás a gráf-vágás alkalmazása, mely segítségével tetszőleges energia-tagok bevonhatóak a procedúrába. Az azonban megengedhetetlenül magas számítási igényhez vezetne, amennyiben az egyes RANSAC verifikációkat gráf-vágási problémaként fogalmazzuk meg. Ellenben, egy lokális optimalizációs lépésként, mint [5]-ben, amit csak a legjobb modellekre alkalmazunk, a gráf-vágások száma közel logaritmus a végrehajtott verifikációk számának.

² Ezen interpretációja az η -nak kizárólag a standard minőség-függvényre érvényes.

A javasolt módszer, a Graph-Cut RANSAC (GC-RANSAC), egy lokálisan optimalizált RANSAC, amely a gráf-vágást és a modell újraillesztést iterálja LO lépésként. A GC-RANSAC felülmúlja az LO-RANSAC-et több szempontból. Ez a megközelítés alkalmas az inlierek és outlierok térbeli kapcsolatainak figyelembevételére. Az LO lépés koncepcionálisan egyszerű, könnyen implementálható kevés intuitív paraméterrel az eredeti LO-RANSAC ad hoc, iteratív és komplex lépéseivel szemben. Ezen felül tesztekkel támasztjuk alá, hogy a GC-RANSAC mind geometriai pontosságát, mind a szükséges iterációk számát tekintve felülmúlja az LO-RANSAC-et és annak több variánsát számos valós, publikusan elérhető adatbázison. Sok problémára futási idejét tekintve gyorsabb, mint a többi módszer. Végezetül, meglepődve lettünk figyelmesek rá, hogy a GC-RANSAC sokszor az elméletileg szükséges iterációs szám *előtt* terminál. Ennek az oka, hogy a lokális optimalizáció a térbeli összefüggőség figyelembevételével sokszor „jó” modellhez konvergál már egy nem csak inlierekből álló kezdeti minta esetén is.

2. Lokális optimalizáció és térbeli összefüggőség

Ebben a szakaszban energia-minimalizációs feladatként fogalmazzuk meg a RANSAC inlier kiválasztását a pontok térbeli összefüggőségének figyelembevételével. A javasolt lokális optimalizáció lényegében egy iterált energia minimalizáció, mely bináris címkézést használ (outlier – 0 és inlier – 1). Az egyszerűség kedvéért az eredeti RANSAC sémából indulunk ki, majd a maximum likelihood becslést fogalmazzuk meg energy minimalizációval. A térbeli összetartozást egy energiatag lesz hivatott megfogalmazni.

2.1. Formalizáció energy minimalizációként

Tételezzük fel, hogy adott egy $\mathcal{P} \subseteq \mathbb{R}^n$ ($n > 0$) ponthalmazunk, egy $\theta \in \mathbb{R}^m$ ($m > 0$) paramétervektorával reprezentált modellünk és egy távolságfüggvényünk: $\phi : \mathcal{P} \times \mathbb{R}^m \rightarrow \mathbb{R}$, mely a pontok modellhez rendelésének költségét adja meg. Az eredeti RANSAC sémára, amely egy bináris illeszkedési függvényt használ (1 – közel, 0 – távol), az energia a következő:

$$E_{\{0,1\}}(L) = \sum_{p \in \mathcal{P}} \|L_p\|_{\{0,1\}},$$

ahol

$$\|L_p\|_{\{0,1\}} = \begin{cases} 0 & \text{if } (L_p = 1 \wedge \phi(p, \theta) < \epsilon) \vee \\ & (L_p = 0 \wedge \phi(p, \theta) \geq \epsilon) \\ 1 & \text{ellenkező esetben.} \end{cases}$$

Az $L \in \{0, 1\}^{|\mathcal{P}|}$ paraméter a címkézés (melyet a standard RANSAC-ben ignorálunk), az $L_p \in L$ a $p \in \mathcal{P}$ pont címkéje, a $|\mathcal{P}|$ a pontok száma és az ϵ az inlier-outlier küszöb. Az $E_{\{0,1\}}$ energiát használva ugyanazt az eredményt kapjuk mint az eredeti RANSAC esetén hiszen csak két esetet nem büntet: (i) amennyiben a p címkéje szerint inlier és közelebb van a modellhez, mint az ϵ küszöb, illetve, (ii) ha p outlier és messzebb van, mint a küszöb. Ez egészen pontosan, amit a RANSAC csinál.

A RANSAC publikációja óta számos cikk, pl. [15], tárgyalta, a $\{0, 1\}$ veszteség kicserélését valamilyen $K : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ kernel függvényvel, pl. Gauss-kernel. Ezen

elgondolás közel áll az MLESAC [26] maximum likelihood becsléséhez, illetve növeli a pontosságot és csökkenti a felhasználó által megadott ϵ -ra való érzékenységet. Az E_K energia, mely a K kernelt használja, az alábbi: $E_K(L) = \sum_{p \in \mathcal{P}} \|L_p\|_K$, ahol

$$\|L_p\|_K = \begin{cases} 1 - K(\phi(p, \theta), \epsilon) & \text{if } L_p = 1 \\ K(\phi(p, \theta), \epsilon) & \text{if } L_p = 0 \end{cases} \quad (1)$$

és

$$K(\delta, \epsilon) = e^{-\frac{\delta^2}{2\epsilon^2}}, \quad (2)$$

ami egyenlő egygel, amennyiben a távolság nulla. A GC-RANSAC-ban a gráf-vágás alapú verifikációban ezt az E_K energiát használjuk.

2.2.. Térbeli összefüggőség

Annak köszönhetően, hogy a problémát egy bináris címkézéssel való energia minimalizációként fogalmazzuk meg, a pont-modell távolságon kívül más energia tagok is beépíthetőek. A probléma továbbra is globálisan és hatékonyan megoldható marad a gráf-vágás algoritmussal. A pontok térbeli relációinak figyelembevétele egy ismert és használt megközelítés mintavételezésnél [19] vagy multi-modell illesztésnél [12,20,1]. Legjobb tudásunk szerint azonban, az LO-RANSAC féle lokális optimalizációhoz még egy publikációban sem alkalmazták. A Potts-modell megfelelő lenne a páronként energia tagra. Ez a modell azt bünteti, amennyiben az egymással szomszédos pontokhoz különböző címkét rendelünk. Ezzel a jelen esetben azonban probléma merül fel, amikor a keresett modell pontjaihoz közel sok outlier található. Ebben az esetben azt büntetni, ha különbözően vannak címkézve szomszédok a modell osztálytól függetlenül (inlier, outlier) ahhoz vezethet, hogy az az outlierok miatt az inliereket is outliernek címkézi az algoritmus. Ezen probléma megoldásaként azt javasoljuk, hogy különböző költséget rendeljünk a szomszédokhoz annak függvényében, hogy mekkora valószínűséggel inlierek. A javasolt páronkénti energia a következő:

$$E_S(L) = \sum_{(p,q) \in \mathcal{A}} \begin{cases} 1 & \text{ha } L_p \neq L_q \\ \frac{1}{2}(K_p + K_q) & \text{ha } L_p = L_q = 0, \\ 1 - \frac{1}{2}(K_p + K_q) & \text{ha } L_p = L_q = 1 \end{cases} \quad (3)$$

ahol $K_p = K(\phi(p, \theta), \epsilon)$, $K_q = K(\phi(q, \theta), \epsilon)$ és (p, q) egy p és q pontokat összekötő él a \mathcal{A} szomszédsági gráfban. A E_S energiában, amennyiben mindkét pont címkéje outlier, az energia $\frac{1}{2}(K_p + K_q)$, tehát „jutalmazza” a 0 címkét, ha a szomszédos pontok messze vannak a modelltől. A költsége annak, hogy a pontot inlierneket címkézzük $1 - \frac{1}{2}(K_p + K_q)$ ami jutalmazza a címkét, amennyiben a pontok közel vannak a modellhez.

A javasolt energia, amely magában foglalja mind a pont-modell távolságot, mind a térbeli összefüggőséget a következő: $E(L) = E_K(L) + \lambda E_S(L)$, ahol a λ paraméter a térbeli összefüggőség fontosságát hivatott szabályozni. A globálisan optimális címkézés, $L^* = \arg \min_L E(L)$, könnyen és polinomiális időben meghatározható a gráf-vágás algoritmus segítségével.

Algorithm 1 A GC-RANSAC algoritmus.

Input: \mathcal{P} – adatpontok; r – gömb sugár, ϵ – küszöb
 ϵ_{conf} – LO alkalmazási küszöbe, μ – konfidencia;
Output: θ - modell paraméterek; L – címkézés

- 1: $w^*, n_{LO} \leftarrow 0, 0.$
- 2: $\mathcal{A} \leftarrow$ Szomszédosság gráf építése r -el.
- 3: **for** $k = 1 \rightarrow H(|L^*|, \mu)$ **do** ▷ 4.egyenlet
- 4: $S_k \leftarrow$ Minimális minta kiválasztása.
- 5: $\theta_k \leftarrow$ Modell paramétereinek kiszámítása S_k -ből.
- 6: $w_k \leftarrow$ A θ_k modell minőségének kiszámítása. ▷ 5.egyenlet
- 7: **if** $w_k > w^*$ **then**
- 8: $\theta^*, L^*, w^* \leftarrow \theta_k, L_k, w_k$
- 9: **if** $\mu_{12} > \epsilon_{\text{conf}}$ **then** ▷ 6.egyenlet
- 10: $\theta_{LO}, L_{LO}, w_{LO} \leftarrow$ Lokális optimalizáció. ▷ 2.algoritmus
- 11: $n_{LO} \leftarrow n_{LO} + 1.$
- 12: **if** $w_{LO} > w^*$ **then**
- 13: $\theta^*, L^*, w^* \leftarrow \theta_{LO}, L_{LO}, w_{LO}$
- 14: **if** $n_{LO} = 0$ **then**
- 15: $\theta^*, L^*, w^* \leftarrow$ Lokális optimalizáció. ▷ 2.algoritmus
- 16: $\theta^* \leftarrow$ Legkisebb négyzetes illesztés L^* -et felhasználva.

3.. GC-RANSAC

Ebben a szakaszban beépítjük a javasolt energia-minimalizáción alapuló lokális optimalizációt a RANSAC algoritmusba. A fő algoritmus az 1.algoritmusban látható. Az első lépés az \mathcal{A} szomszédossági gráf meghatározása. Ehhez a Fast Approximate Nearest Neighbors módszert [18] alkalmazzuk. Egy adott pontnak a szomszédai az θ körülvevő r sugarú hipergömbben található pontok lesznek. Ez az r egy paramétere az algoritmusnak. Az 1.algoritmusban, a H a következő [7]:

$$H(|L^*|, \mu) = \frac{\log(\mu)}{\log(1 - P_I)}, \quad (4)$$

ahol $P_I = \binom{|L^*|}{m} / \binom{|P|}{m}$. Lényegében ez határozza meg a szükséges iterációk számát egy μ konfidencia, a minimális minta mérete (m) és a az eddigi legjobb modellhez tartozó inlierek számának ($|L^*|$) függvényében. A $|\cdot|$ norma a halmaz számosságát jelöli.

Az algoritmus minden k -edik iterációban kiválaszt egy minimális mintát (mi erre a PROSAC [4] mintavételezést használjuk), majd kiszámítja a θ_k modell paramétereit. Végezetül kiszámítja a modell minőségét az adatpontok függvényében:

$$w_k = \sum_{p \in \mathcal{P}} K(\phi(p, \theta_k), \epsilon) \quad (5)$$

ahol a K egy Gauss-kernel, ahogy azt a 2.egyenletben javasoltuk. Amennyiben w_k értéke nagyobb, mint az addigi legjobb modellé (w^*) ez a modell lesz az új legjobb,

a mentett paramétereket frissítjük, majd a javasolt lokális optimalizációt alkalmazzuk amennyiben szükséges. Ezt a szükségességi feltételt egy későbbi szekcióban tárgyaljuk.

A javasolt lokális optimalizáció az 2.algoritmusban látható. A iteráció két fő lépésből áll: (i) gráf-vágás és (ii) modell illesztést. A G probléma gráf konstrukciója egyelemű (1) és páronkénti (3) energiákkal az 3.algoritmusban látható. Az AddTerm1 és AddTerm2 függvények adják hozzá ezeket a tagokat reprezentáló éleket és csúcsokat a G gráfhoz. [13]-ben látható ilyen gráf-konstrukciós probléma mélységeiben kifejtve (3. fejezet). A G -re alkalmazott gráf-vágás meghatározza az L címkézést, mely figyelembe veszi mind a pontok távolságát a modelltől, mind a térbeli viszonyait. A θ modell-paramétereket ezután kiszámítjuk egy $7m$ méretű véletlenszerűen, az inlierek közül kiválasztott halmazból, ezzel gyorsítva a folyamatot, ahogy azt a [15]-ben javasolják. Itt az m a minimális minta mérete, pl. $m = 2$ egyenesekre. Megjegyezzük, hogy a $7m$ egy kimerítő kereséssel beállított érték és megfelelt nekünk is a vizsgált problémákra. Végezetül kiszámítjuk a θ modell w minőségét, frissítjük az eddigi legjobb modellt és folytatjuk a procedúrát. Ellenkező esetben a lokális optimalizáció terminál. Megjegyezzük, hogy abban az esetben, ha a lokális optimalizáció egyszer sem futott le az algoritmus futása alatt, a kimenetül adott modellre egyszer minden esetben lefuttatjuk azt. Majd a modell paramétereket újra kiszámítjuk az összes inlierre vett legkisebb négyzetes illesztéssel, ahogy azt az eredeti RANSAC is teszi.

Algorithm 2 Lokális optimalizáció.

Input: \mathcal{P} – data points, L^* – labeling,
 w^* – support, θ^* – model;
Output: L_{LO}^* – labeling, w_{LO}^* – support, θ_{LO}^* – model;

```

1:  $w_{LO}^*, L_{LO}^*, \theta_{LO}^*, changed \leftarrow w^*, L^*, \theta^*, 1.$ 
2: while  $changed$  do
3:    $G \leftarrow$  Probléma gráf konstrukciója. ▷ Alg. 3
4:    $L \leftarrow$  Gráf-vágás a  $G$  gráfon.
5:    $I_{7m} \leftarrow 7m$  méretű véletlenszerűen inlier minta.
6:    $\theta \leftarrow$  Modell illesztése a  $I_{7m}$  mintára.
7:    $w \leftarrow$  A  $\theta$  minőségének kiszámítása.
8:    $changed \leftarrow 0.$ 
9:   if  $w > w_{LO}^*$  then
10:     $\theta_{LO}^*, L_{LO}^*, w_{LO}^*, changed \leftarrow \theta, L, w, 1.$ 

```

Az LO lépés alkalmazásnak feltételei [15]-ben: (i) a modell az eddigi legjobb és (ii) egy felhasználó által megadott iteráció számon túl vagyunk. Tesztjeink azonban azt mutatták, hogy ez a megközelítés is túl sok időt tölt olyan modellek optimalizálásával, melyek nem elég ígéretesek. Javasolunk tehát egy egyszerű heurisztikát a felhasználó által meghatározott iteráció limit kicserélésére egy, az input adatoktól függő stratégiára. Ennek segítségével az LO lépést csak néhányszor futtatja az algoritmus anélkül, hogy észlelhető romlás lenne a a módszer pontosságában.

Ahogy az a RANSAC esetén köztudott, a szükséges iterációk száma, k , az η inlier arány, az m mintaméret és a μ konfidencia függvényében a következő módon számítha-

Algorithm 3 Probléma gráf konstrukciója.

Input: \mathcal{P} – adatpontok, \mathcal{A} – szomszédossági gráf
 θ – modell paraméterek

Output: G – probléma gráf;

```

1:  $G \leftarrow \text{EmptyGraph}()$ .
2: for  $p \in \mathcal{P}$  do
3:    $c_0, c_1 \leftarrow \mathbf{K}(\phi(p, \theta), 1 - \mathbf{K}(\phi(p, \theta), \epsilon))$ 
4:    $G \leftarrow \text{AddTerm1}(G, p, c_0, c_1)$ .
5: for  $(p, q) \in \mathcal{A}$  do
6:    $c_{01}, c_{10} \leftarrow 1, 1$ .
7:    $c_{00} \leftarrow 0.5(\mathbf{K}(\phi(q, \theta)) + \mathbf{K}(\phi(p, \theta)))$ .
8:    $c_{11} \leftarrow 1 - 0.5(\mathbf{K}(\phi(q, \theta)) + \mathbf{K}(\phi(p, \theta)))$ .
9:    $G \leftarrow \text{AddTerm2}(G, p, q, c_{00}, c_{01}, c_{10}, c_{11})$ .

```

tó: $k = \log(1 - \mu) / \log(1 - \eta^m)$. Átrendezve ezt a formulát μ -re a következő egyenletet kapjuk $\mu = 1 - 10^{k \log(1 - \eta^m)}$, amely meghatározza a keresett modell megtalálására vonatkozó konfidenciát, a k -edik iterációban, amennyiben az inlier arány η .

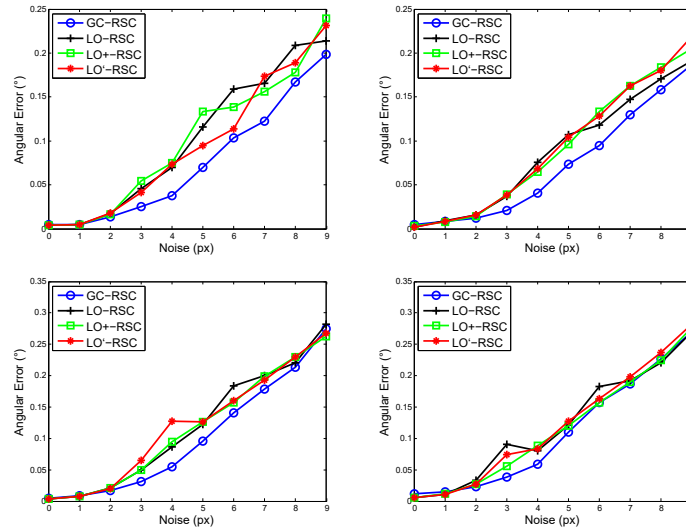
Tételezzük fel, hogy az algoritmus megtalált egy új legjobb modellt η_2 inlier aránnyal a k_2 -edik iterációban, míg az előző legjobb modellt a k_1 -ben találta meg η_1 inlier aránnyal ($k_2 > k_1$, $\eta_2 > \eta_1$). A konfidenciák aránya, μ_{12} , a következő:

$$\mu_{12} = \frac{\mu_2}{\mu_1} = \frac{1 - 10^{k_2 \log(1 - \eta_2^m)}}{1 - 10^{k_1 \log(1 - \eta_1^m)}}. \quad (6)$$

A tesztek során azt figyeltük meg, hogy a modell, amely a terminációhoz vezetett lokális optimalizáció után gyakran eredményezett jelentős növekedést a konfidenciában. A $\mu_{12} > \epsilon_{\text{conf}}$ konfidenciát használjuk tehát feltételként az LO lépés alkalmazására, ahol a ϵ_{conf} egy, a felhasználó által meghatározott küszöb arra, hogy mekkora konfidencia változás számít szignifikánsnak.

4. Kísérleti eredmények

Ebben a szekcióban a javasolt GC-RANSAC algoritmust validáljuk mind szintetikus, mind publikusan elérhető valós adatokon. Az alábbi módszerekkel hasonlítjuk össze a javasolttal: RANSAC [7], LO-RANSAC [5], LO⁺-RANSAC, LO'-RANSAC [15] és EP-RANSAC [14]. Az EP-RANSAC módszerhez a küszöb paramétert kimerítő kereséssel állítottuk be úgy, hogy a legkisebb átlagos hibát vesse az algoritmus a vizsgált adathalmazokon. A többi paraméter a szerzők által javasolt értéket kapta. Mindegyik módszer a PROSAC [4] mintavételezést használja és az MSAC által javasolt csonkított négyzetes hibát, illetve $\epsilon = 0.3$ küszöböt, ahogy azt [15]-ben javasolták. Az EP-RANSAC algoritmus inlier maximalizációt alkalmaz, tekintve, hogy a költségfüggvénye nem cserélhető fel az MSAC-féle négyzetes hibára. A szomszédos pontok meghatározására használt hipergömb sugara 20 pixel volt és a konkatenált 4D megfeleltetésben operált. A λ paraméter és ϵ_{conf} is 0.1 volt.



1. ábra. A megtalált egyenesek átlagos szöghibája (fokban) a σ zajszint (pixelben; 1000 futtatás mindegyikre) függvényeként. Az első sorban nem szaggatott, míg a második sorban szaggatott egyeneseket generáltunk. Az első oszlopban 100 outlier adtunk az egyenes 100 pontjához, míg a másodikban 500-at.

Szintetikus tesztek 2D egyenesekkel Azért, hogy összehasonlítsuk a GC-RANSAC-et a state-of-the-arttal egy teljesen kontrollált környezetben, két egyszerű tesztet választottunk: szaggatott és nem szaggatott 2D egyenesek illesztését. Minden egyes teszt esetén először egy 600×600 -as ablakot generáltunk, azon belül egy véletlenszerű egyenes 100 pontját vettük, a koordinátáikhoz nulla várható értékű Gauss-zajt adtunk σ szórással. A nem szaggatott egyenes uniform eloszlással választottuk ki a 100 pontot (lásd 2.ábra; bal). A szaggatott egyenes esetén 10 véletlenszerű csomópontot választottunk, majd minden egyes csomópont körül 10 véletlen pontot adtunk az egyeneshez legfeljebb 10 pixelre tőle (lásd 2.ábra; jobb). Végezetül, k darab outlier-t adtunk a színtérhez. Minden egyes zajszinten 1000 tesztet végeztünk. Az 1.ábra az átlagos szöghibát mutatja (fokban) a zaj σ függvényeként. Az első és második sor a nem szaggatott és a szaggatott egyenesek esetére vonatkoznak. Az első oszlopban látható diagrammok esetén 100, a második oszlopban levőknel 500 outlier-t adtunk a színtérhez. Az 1.ábra azt mutatja, hogy a GC-RANSAC pontosabb egyeneseket talál, mint a többi algoritmus.

Fundamentális mátrix becslés. Fundamentális mátrix becslésre a kusvod2 (24 képpár)³, Multi-H⁴ (5 képpár), és AdelaideRMF⁵ (19 képpár) adatbázisokat használtuk (példákat lásd az 3.ábrán). A Kusvod2 adatbázis 24 különböző méretű képpárból áll pont-megfeleltetésekkel és manuálisan kiválasztott inlierekkel. Az AdelaideRMF és

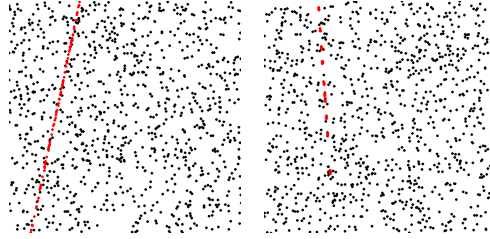
³ <http://cmp.felk.cvut.cz/data/geometry2view/>

⁴ <http://web.eee.sztaki.hu/dbarath/>

⁵ cs.adelaide.edu.au/hwong/doku.php?id=data

	LO	LO ⁺	LO'	GC
L	6%	5%	4%	15%
F	29%	30%	24%	32%

1. táblázat. Azon esetek aránya százalékosan, mikor nem csak inlierekből álló minta vezetett a megoldáshoz egyenes (L) és fundamentális mátrix (F) illesztésre. Egyenesekre 1000 futtatást végeztünk 100, 500 és 1000 outlierrel és $\sigma = 0, \dots, 9$ pixel zajszinteken. Tehát összesen 15000 futtatás eredményeként adódnak az értékek. Fundamentális mátrix esetén 1000 futtatás történt az AdelaideRMF adatbázison.



2. ábra. Példa bemenet nem szaggatott és szaggatott egyenesekre. Az 1000 fekete pont outlier és a 100 piros inliers.

Multi-H adatbázisok összesen 24 képpárból állnak pont-megfeleltetésekkel, melyek közül mindegyiket egy-egy homográfiahoz vagy az outlier osztályhoz rendeltek. A tesztek során minden nem az outlier osztályból származó pontot inlierek tekintettünk. Összesen tehát a javasolt módszert 48 képpáron (három publikuson elérhető adatbázisból) teszteltük fundamentális mátrix illesztésre. Mindegyik módszer a 7-pontos módszert [10] használta, ezért a minimális minták hét pont-megfeleltetésből álltak. Legkisebb négyzetes illesztésre nem minimális mintára a normalizált 8-pontos algoritmust [11] használtuk. Megjegyezzük, hogy mindent olyan fundamentális mátrixot eldobtunk, melyre az irányított epipoláris megkötés [6] nem érvényesül.

Az 2.táblázat első három blokkja (mindegyik négy sorból áll) a becült epipoláris geometriák átlagos pontosságát mutatja (1000 futtatás átlaga minden képpáron) az egyes adatbázisokon. A megvizsgált tulajdonságok: **(1)** LO: annak a száma, hogy mennyiszer alkalmaztuk a javasolt lokális optimalizációt a futás alatt (a gráf-vágás lépések zárójelk között láthatóak) **(2)** \mathcal{E} a becült modell geometriai hibája (pixelben) manuálisan annotált inlierekhez viszonyítva. Fundamentális mátrixok és homográfiaik esetén ez a hiba az átlagos Sampson-hiba és a visszavetítési hiba. Esszenciális mátrix esetén a hiba az átlagos Sampson hiba, abból a fundamentális mátrixból számolva, amit a becült esszenciális mátrix indukál. **(3)** \mathcal{T} az átlagos futási idő ezredmásodpercekben mérve. **(4)** \mathcal{S} a terminációig kiválasztott minimális minták száma. Alapvetően ez az iterációk száma. Tisztán látható, hogy a GC-RANSAC fundamentális mátrix esetén mindig a legpontosabb modellhez vezetett kevesebb iteráció alatt, mint a többi módszer.



3. ábra. A GC-RANSAC eredményei példa képpárokon és problémákon. Az inlier megfeleltetéseket színes egyenesekkel és körökkel jelöljük, míg az outliereket fekete kereszttel.

Homográfia becslés. Annak érdekében, hogy teszteljük milyen pontossággal lehet a javasolt módszerrel homográfákat becsülni a *homogr*⁶ (16 képpár) és az *EVD*⁷ (15 képpár) adatbázisokat használtuk (lásd az 3.ábrát példáért). Az adatbázisokban a képek méretei 329×278 és 1712×1712 között mozognak, illetve manuálisan kiválasztott inlier megfeleltetések is tartoznak a képpárokhoz. A *homogr* adatszett jellemzően kis bázistávolságú képekből áll, míg a *EVD* képei extrém nézetváltozáson esnek át, pl. széles bázistáv. A módszereket a normalizált 4-pontos algoritmus [10] használták mind minimális mintára való illesztés esetén, mind legkisebb négyzetes optimalizációra. A 2. táblázat 4. és 5. blokkja az átlagos eredményeket mutatja a két adatbázison. Látható, hogy a GC-RANSAC vezet a legpontosabb eredményekhez egy esetet kivéve.

Esszenciális mátrix becslése. Esszenciális mátrix becslésére a *strecha* adatbázist [24] használtuk, mely elsősorban épületfotókból áll. A képek mérete 3072×2048 . Az adatbázis a ground truth projekciós mátrixokat is tartalmazza. A módszereket az összes lehetséges képpárra alkalmaztuk minden képszekvenciában. A SIFT detektorral találtunk pont-megfeleltetéseket. Minden képpárhoz kinyertünk egy referencia inlier halmazt olyan módon, hogy először, a megadott projekciós mátrixokból kiszámoltuk [10] a fundamentális mátrixot. Azokat a megfeleltetéseket tekintettük ground truth inliereknek, amelyekre a szimmetrikus epoláris hiba kisebb volt 1 pixelnél. Azokat a képpárokat, ahol összesen kevesebb, mint 20 inlier volt nem használtuk fel a kiértékelésben. A módszereket összesen 467 képpáron teszteltük. Az eredmények a 2.táblázat hatodik blokkjában láthatóak. A módszerek magas futási idejének oka az, hogy az inlie-

⁶ <http://cmp.felk.cvut.cz/data/geometry2view/>

⁷ <http://cmp.felk.cvut.cz/wbs/>

rek aránya alacsony volt (27%) és, mindemellett, több ezer megfeleltetést (átlagosan 2323) detektált a SIFT algoritmus így súlyosan lelassítva a verifikációs folyamatot. A GC-RANSAC minden esetben a legpontosabb esszenciális mátrixokat becsülte. Megjegyezzük, hogy egy lényeges romlás figyelhető meg minden módszer esetén, amennyiben egy idő limitet használtunk. Ennek az oka az, hogy a sok pont-megfeleltetés miatt már egyetlen verifikációs lépés is túl sok időt emészt fel.

Affin transzformációk becslése. Ehhez a problémához a SZTAKI Earth Observation adatbázist⁸ [2] használtuk (83 darab 320×240 méretű képpár). Az adatbázis képein utcai jelenetek láthatóak egy ballonról nézve. A ballon magassága miatt a képpárok közötti reláció jól közelíthető egy affin transzformációval. Az egyes képpárok esetén a pont-megfeleltetéseket a SIFT detektorral generáltuk, közülük 20 inliert manuálisan kiválasztottunk, hogy egy ground truth affin transzformáció kapjunk. A ground truth affin transzformációval 1 pixeles küszöböléssel megkaptuk az összes inlier megfeleltetést. Ezeket használtuk a becsült modellek minőségének mérésére. A becslése eredmények a 2.táblázat hetedik blokkjában láthatóak. A számolt hiba az alábbi: $|\mathbf{A}\mathbf{p}_1 - \mathbf{p}_2|$, ahol \mathbf{A} a becsült affin transzformáció és \mathbf{p}_k a pont a k -edik képen ($k \in \{1, 2\}$). A módszerek meglehetősen hasonló eredményeket produkáltak, azonban a GC-RANSAC egy hajszálnyival pontosabb volt, és lassabb is a szomszédossági gráf számolása miatt. Ellenben így is gyorsabb a valós időnél elvártnál.

Konvergencia nem csak inliereket tartalmazó mintából. A 1.táblázat azon esetek gyakoriságát mutatja, amikor az eredményül kapott modellt egy olyan kezdeti mintából kaptuk meg, ami nem csak inliereket tartalmazott. Egyenesek esetén (\mathbf{L}) ezt az értéket 1000 futtatás átlagaként kaptuk meg 100, 500 és 1000 outlierrel és $\sigma = 0.0, \dots, 0.9$ zajszintre. Tehát összességében 15000 teszt eredményeinek átlaga látható. Abban az esetben tekintettünk egy minimális mintát nem csak inlierekből állónak, amennyiben legalább egy pontja messzebb volt a ground truth modelltől, mint a ground truth zaj σ . Fundamentális mátrix esetén (\mathbf{F}) ezek az értékek 1000 futtatás átlagai az AdelaideRMF adatbázison. Ebben az adatbázisban az inliereket manuálisan választották ki, ami lényegesen leegyszerűsíti annak vizsgálatát, hogy egy minta csak inlierekből áll-e.

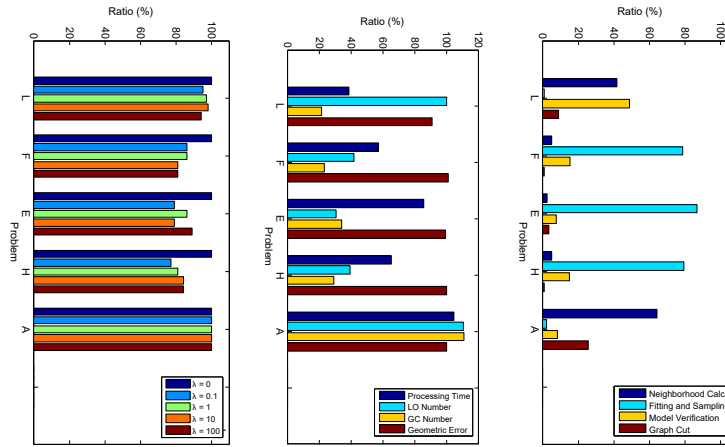
A λ paraméter hatása. A javasolt módszert lefuttattuk az összes problém különböző λ paraméterekkel annak érdekében, hogy lássuk milyen hatást gyakorol a térbeli összefüggőség figyelembevételének súlya (λ) az eredményekre. A megvizsgált értékek: (i) $\lambda = 0$, ami kikapcsolja térbeli összefüggőség figyelembevételét lényegében, (ii) $\lambda = 0.1$, (iii) $\lambda = 1$, (iv) $\lambda = 10$, and (v) $\lambda = 100$. Az 4.ábra felső diagramja a geometriai hibát mutatja a $\lambda = 0$ eset hibájához viszonyítva százalékosan. Mint látható, $\lambda = 0.1$ vezetett a legpontosabb eredményekhez, ezért választottuk ezt a tesztekhez.

Az új LO feltétel kiértékelése. A javasolt feltételt (6) vizsgáljuk ebben a paragrafusban, mely azt hivatott elérni, hogy a lokális optimalizációt csak a legígéretesebb modellekre alkalmazzuk. A GC-RANSAC-et futtattuk az összes megvizsgált problémán az új és a javasolt feltételekkel. Az standard technika egy iteráció limitet használ (alapértelmezett érték: 50) az első LO futtatás előtt. Az 4.ábra (középső) az egyes tulajdonságok (futási idő – sötétkék; LO – kék, és GC lépések száma – sárga; geometriai hiba – barna) arányát mutatja a standard technikához viszonyítva. A javasolt feltétel lényeges gyorsulást hoz a futási időben észrevehető pontosságbeli romlás nélkül.

⁸ <http://mplab.sztaki.hu/remotesensing>

			Megközelítőleg 60 FPS					95% konfidencia					
			RSC	LO	LO ⁺	LO'	GC	RSC	LO	LO ⁺	LO'	EP-RSC	GC
kusvod2	F, #24	LO		2	2	2	1 (3)		1	1	1		2 (3)
		\mathcal{E}	5.01	4.95	4.97	5.02	4.65	5.18	5.08	5.03	5.22	7.87	4.69
		\mathcal{T}	6.2	6.1	6.3	5.9	4.6	4.9	5.2	5.1	4.9	439.9	3.6
		\mathcal{S}	117	96	99	111	70	93	76	78	87		53
Adelaide	F, #19	LO		2	2	2	1 (3)		2	2	3		2 (4)
		\mathcal{E}	0.55	0.53	0.52	0.55	0.50	0.44	0.45	0.43	0.44	0.71	0.43
		\mathcal{T}	14.2	14.8	14.9	14.1	18.9	262.7	194.2	210.9	237.1	2 121.9	227.1
		\mathcal{S}	124	113	113	122	116	1 363	1 126	1 205	1 305.00		1 115
Multi-H	F, #4	LO		1	1	1	1 (3)		2	1	2		1 (3)
		\mathcal{E}	0.35	0.34	0.34	0.34	0.32	0.33	0.33	0.33	0.34	0.44	0.32
		\mathcal{T}	10.3	11.5	11.1	10.3	14.6	12.8	15.1	14.1	12.4	2 371.8	36.0
		\mathcal{S}	83	76	76	82	74	107	89	90	100		78
EVD	H, #15	LO		2	2	2	2 (2)		4	4	4		3 (6)
		\mathcal{E}	1.53	1.63	1.51	1.58	1.53	0.96	0.95	0.95	0.96	1.17	0.92
		\mathcal{T}	16.8	18.3	18.0	16.8	19.2	247.3	248.0	251.3	247.0	> 10 ⁴	249.9
		\mathcal{S}	320	298	301	318	301	4 303	4 203	4 248	4 291		4 204
homogr	H, #16	LO		2	2	2	1 (3)		2	2	2		1 (4)
		\mathcal{E}	0.53	0.53	0.53	0.53	0.51	0.50	0.50	0.49	0.50	0.58	0.47
		\mathcal{T}	7.1	10.4	9.8	7.1	7.6	17.1	10.1	9.9	8.5	3 339.7	7.9
		\mathcal{S}	193	175	175	189	159	450	212	214	226		165
strecha	E, #467	LO		1	1	1	1 (1)		7	7	7		7 (7)
		\mathcal{E}	11.81	12.34	12.07	12.12	11.6	3.03	2.95	2.94	2.87	3.32	2.83
		\mathcal{T}	11.6	17.3	17.2	17.2	17.3	3 581.9	3 638.5	3 648.4	3 570.0	> 10 ⁶	3 466.4
		\mathcal{S}	31	30	31	31	30	3 654	3 646	3 634	3 653		3 651
SZTAKI	A, #52	LO		1	1	1	1 (3)		1	1	1		1 (3)
		\mathcal{E}	0.41	0.41	0.41	0.41	0.40	0.45	0.46	0.44	0.45	0.48	0.41
		\mathcal{T}	3.5	3.2	3.2	3.2	10.3	1.7	1.7	1.7	1.7	4 718.2	10.2
		\mathcal{S}	26	26	26	26	26	9	9	9	9		9

2. táblázat. Fundamentális mátrix becslés a kusvod2 (24 képpár), AdelaideRMF (19 képpár) és Multi-H (4 képpár) adatbázisokon; homográfia becslés a homogr (16 képpár) és EVD (15 képpár) adatszeteken; esszenciális mátrix becslés a ó strecha adatbázison (467 képpár); és affin transzformáció becslése a SZTAKI Earth Observation adatszeten (52 képpár). A módszereket tehát 597 képpáron teszteltük. Az adatbázisok nevei, a problémák (F/H/E/A), képpárok számai (#) és a megjelenített tulajdonságok az első három oszlopban láthatóak. A következő öt esetén a módszereket egy idő limit (60 FPS = 1/60 másodperc) után megszakítottuk. Itt az EP-RANSAC-et nem mutatjuk, mert az nem alkalmazható valós időben. A többi oszlopban levő eredményekhez nem használtunk idő limitet, hanem az elvárt konfidenciát állítottuk 95%-ra. Az értékek 1000 futtatás átlagai. Az LO a futtatott lokális optimalizációk száma, zárójelben pedig a gráf-vágások száma látható a javasolt módszernél. A második sorok (\mathcal{E}) a becslött modellek geometriai hibáját tartalmazzák pixelben egy manuálisan kiválasztott inlier halmazból számítva. A futási idő (\mathcal{T} , ezredmásodpercekben) és a szükséges iterációk számát (\mathcal{S}) a harmadik és negyedik sorokban írtuk. A geometriai hiba a Sampson-hiba **F** és **E** esetén, illetve a visszavetítési hiba **H** és **A** esetén.



4. ábra. (Felső) A különböző λ (térbeli összefüggőség súlya) értékek hatása. Az geometriai hibát jelenítettük meg a $\lambda = 0$ eset (nincs térbeli összefüggőség; sötétkék) hibájához viszonyítva százalékosan az egyes problémákhoz (**L** – egyenes, **F** – fundamentális mátrix, **E** – esszenciális mátrix, **H** – homográfia, **A** – affin transzformáció). (Középső) Az javasolt LO alkalmazására vonatkozó feltétel (LO, amikor a konfidencia szignifikánsan megugrik) hatása. A megjelenített értékek a javasolt feltétel az eredeti feltétellel összehasonlítva százalékosan. (Alsó) A teljes futási idő egyes algoritmikus lépésekre lebontva. Az eredmények az összes tesztre vett futások átlagai.

Futási idő. A 4. ábra alsó diagramja futási időt mutatja algoritmikus részekre lebontva. A szomszédosságok meghatározása megközelítőleg lineárisan függ a pontszámtól. A világoskék oszlop a mintavételezés és modell-illesztés idejét mutatja. A sárga és barna oszlopok a modell-verifikációt és a javasolt lokális optimalizációt mutatja. Elsősorban a mintavételezéstől és a modell illesztéstől függ az algoritmus futási ideje.

5. Konklúzió

A GC-RANSAC algoritmust mutattuk be, mely geometriailag pontosabb eredményeket szolgáltat, mint a state-of-the-art. Számos problémára valós időben fut megközelítőleg olyan sebességgel, mint a kevésbé pontos alternatívák. Egyszerűbben implementálható, mint a többi lokális optimalizációt alkalmazó RANSAC. A javasolt LO lépés globálisan optimális az aktuális modell paraméterekre nézve. Ezen felül javasoltunk egy feltételt az LO alkalmazására, mely lényeges javulást eredményezett a futási időben anélkül, hogy észrevehetően rontotta volna a pontosságot. A GC-RANSAC algoritmus könnyen beilleszthető a USAC [22] által nyújtott keretbe, mely tartalmazza például a PROSAC mintavételezést, degeneráció tesztelést, gyors kiértékelést korai terminációval.

Hivatkozások

1. D. Barath, J. Matas, and L. Hajder. Multi-H: Efficient recovery of tangent planes in stereo images. In *BMVC*, 2016.

2. C. Benedek and T. Szirányi. Change detection in optical aerial images by a multilayer conditional mixed markov model. *TGRS*, 2009.
3. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR*. IEEE, 1998.
4. O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *CVPR*.
5. O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*. Springer, 2003.
6. O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In *ICPR*, 2004.
7. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 1981.
8. V. Fragoso, P. Sen, S. Rodriguez, and M. Turk. EVSAC: accelerating hypotheses generation by modeling matching scores with extreme value theory. In *ICCV*, 2013.
9. D. Ghosh and N. Kaabouch. A survey on image mosaicking techniques. *JVCIR*, 2016.
10. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
11. R. I. Hartley. In defense of the eight-point algorithm. *PAMI*, 1997.
12. H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2012.
13. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 2004.
14. H. Le, T.-J. Chin, and D. Suter. An exact penalty method for locally convergent maximum consensus. In *CVPR*. IEEE, 2017.
15. K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized ransac. In *BMVC*, 2012.
16. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 2004.
17. D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *CVIU*, 2015.
18. M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP*, 2009.
19. D. Nasuto and J. M. B. R. Craddock. NAPSAC: High noise, high dimensional robust estimation - it's in the bag. 2002.
20. T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. *TIP*, 2014.
21. P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV*. IEEE, 1998.
22. R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. USAC: a universal framework for random sample consensus. *PAMI*, 2013.
23. C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. *PAMI*, 2005.
24. C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *CVPR*. IEEE, 2004.
25. P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. ISOP, 1993.
26. P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 2000.
27. P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *CVIU*, 1998.
28. R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *CVPR*, 2004.
29. M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *ICIP*. IEEE, 2005.