

DİFFERANSİYEL ARAMA ALGORİTMASI TABANLI YENİ BİR KAMERA KALİBRASYON YÖNTEMİ

Pınar ÇİVICİOĞLU¹, Tuba KURBAN², Erkan BEŞDOK³

¹Doç. Dr., Erciyes Üniversitesi, Sivil Havacılık Yüksekokulu, Uçak Elektrik-Elektronik Bölümü, civici@erciyes.edu.tr

²Y. Müh., Arş.Gör., Erciyes Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, tubac@erciyes.edu.tr

³Pof. Dr., Erciyes Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, ebesdok@erciyes.edu.tr

ÖZET

CBS amaçlı veri toplamada yararlanılan en önemli donanımlardan biri optik kamera sistemleridir. Optik kamera sistemlerinden elde edilen verilerden metrik bilgiler çıkartmak için ilgili optik kamera sisteminin kalibre edilmiş olması gerekir. Bu bildiride, yeni bir meta-sezgisel algoritma olan Diferansiyel Arama Algoritması'nın kullanıldığı özgün bir kamera kalibrasyon geliştirme yöntemi tanıtılmıştır. Tanıtılan yöntemin başarısı literatürde yaygın olarak kullanılan meta-sezgisel algoritmalarından ABC, JDE ve PSO yöntemlerinin sağladığı kamera kalibrasyon geliştirme başarıları ile karşılaştırılmıştır. Gerçekleştirilen deneylerden elde edilen sonuçlar, DSA'nın sağladığı kamera kalibrasyon geliştirme başarısının diğer meta-sezgisel yöntemlerin sağladığı geliştirme başarılarından daha iyi olduğunu göstermektedir. Önerilen yöntem kullanılarak elde edilen kamera kalibrasyon parametreleri, tasarlanan bir çizgi lazer tabanlı stereo-görme sistemi üzerinde sınanmıştır.

Anahtar Sözcükler: Kamera Kalibrasyonu, Diferansiyel-Arama Algoritması.

A NEW CAMERA CALIBRATION METHOD BASED ON DIFFERENTIAL SEARCH ALGORITHM

Optic camera systems are one of the most important data collection hardware in GIS. In order to retrieve metric information from camera system they have to be calibrated. In this paper, a new metaheuristic camera calibration boosting method based on DSA has been proposed. The success of the presented method has been compared with the well known metaheuristic methods of ABC, JDE, JADE and PSO. Experimental results exposed that the success of the presented method is better than those of the other metaheuristic methods on camera calibration boosting problem. The camera calibration results obtained from proposed method were also examined in a line-laser based 3d scanning system.

Keywords: Camera Calibration, Differential Search Algorithm.

1. GİRİŞ

Kamera kalibrasyonu, kamera tabanlı bir görüntüleme sisteminin modellenmesini amaçlar (Besdok, E., 2007, Besdok, E., Çivicioğlu, P., 2006(a,b), Tsai, R.Y., 1987, Zhang, ZY., 2000, Heikkila, J., 2000, Fusiello, A., Trucco, E., Verri, A., 2000). Her sistem modelleme probleminde olduğu gibi kamera kalibrasyon probleminde de, sistem girişleri ve sistem çıkışı arasındaki ilişkileri tanımlayan matematiksel modellere ait parametrelerin en iyi değerlerinin elde edilmesi istenir. Kamera kalibrasyon parametreleri, geometrik-anlamı veya geometrik anlamı olmayan formda tasarlanabilir. Bir çok kamera kalibrasyon yöntemi tanıtılmıştır (Tsai, R.Y., 1987, Zhang, ZY., 2000, Heikkila, J., 2000). Literatürde yaygın olarak kullanılan kamera kalibrasyon yöntemleri temel olarak birkaç gruba ayrılabilir; öz-benzerlik tabanlı kalibrasyon yöntemleri, kontrol noktaları kullanan yöntemler, imge veya fiziksel uzay kısıtlarını kullanan kısıtlamalı-optimizasyon yöntemleri (Beşdok, E., 2007, Beşdok, E., Çivicioğlu, P., 2006(a,b)).

Bu bildiride, kamera kalibrasyon parametrelerini elde etmek için, önce kamera projeksiyon matrisi hesaplanmış ardından ilgili matris fiziksel parametrelere ayrıştırılmıştır. Elde edilen kalibrasyon parametrelerine ait değerler, Diferansiyel Arama Algoritması (DSA) (Çivicioğlu, P., 2012) kullanılarak geliştirilmiştir. DSA, Çivicioğlu tarafından tanıtılmış yeni bir metasezgisel algoritmadır ve sayısal kısıtlı, sınırlamalı veya kısıtsız optimizasyon problemlerinin çözümünde kullanılmak üzere önerilmiştir. DSA'nın sayısal problemleri çözme başarısı ayrıntılı olarak (Çivicioğlu, P., 2012)'de incelenmiştir.

Bu bildirinin geri kalan bölümleri şu şekilde organize edilmiştir. Bölüm 2'de DSA tanıtılmıştır, Bölüm 3'te Kamera Kalibrasyonu açıklanmıştır. Bölüm 4'te Sonuçlar verilmiştir.

2. DİFFERANSİYEL ARAMA ALGORİTMASI (DSA)

DSA, sayısal optimizasyon problemlerinin çözümünde kullanılmak üzere Çivicioğlu tarafından geliştirilmiş bir yapay-sürü algoritmasıdır (Çivicioğlu, P., 2012). DSA bir süperorganizmanın göç için kullandığı brownian-benzeri rassal-gezinti hareketini simüle eder. Doğada bulunan, doğal besin alanlarının kapasitesi ve verimliliği yıl içi dönemsel iklim koşullarındaki değişiklikler nedeniyle sık-sık değişir. Bu nedenle eusocial, subsocial, veya presocial bir çok canlı türü, yıl içerisinde sezonluk göç davranışı gösterir. Göç davranışı canlıların, sağladığı doğal

kaynakların kapasitesi ve çeşitliliğinde azalma yaşanan bir habitattan daha verimli bir habitata taşınmalarını sağlar. Doğada bir çok canlı türü, dönemsel göç döngüsüne sahiptir. Göç hareketi sırasında, göç eden canlı türleri genellikle çok büyük sayıda birey içeren bir süperorganizma oluşturur. Ardından süperorganizma bulunduğu konumu daha verimli alanlara doğru hareket ederek değiştirmeye başlar. Bir süperorganizmanın hareketi brownian-benzeri bir rassal-gezinti modeli kullanılarak açıklanabilir. Süperorganizmaların davranışlarını modelleyen bir çok Hesaplamalı-Zeka algoritması bulunmaktadır (örnek, PSO (Omran, M.G.H., Clerc, M., 2011), Cuckoo-Search, ABC (Karaboğa, D., Baştürk, B., 2007), ANT Colony). Yağmacı bir çok canlı türü, göç etmeden önce göç etmek istedikleri alanın verimliliğini kontrol eder. Eğer göç edilmek niyetiyle kontrol edilen bir bölgenin sahip olduğu potansiyel, o an süperorganizmanın ihtiyaçlarını karşılama potansiyeline sahip ise, süperorganizma en azından bir süre için yeni bölgeye yerleşir ve yeni verimli bölgeler bulma davranışını tekrarlayarak göçe devam eder. DSA’da, ilgili problemin rastgele çözümlerinden oluşan bir popülasyonun, göç eden bir yapay-süperorganizmaya karşılık geldiği varsayılmıştır.

DSA’da yapay-süperorganizma, probleme ait global minimum değerine doğru göç eder. Bu göç sırasında, yapay-süperorganizma habitat’ta (yani arama-uzayında) rastgele seçilen bazı konumların, göç sırasında geçici olarak konaklama-alanı (yani ‘stopover site’) olarak kullanıma uygun olup olmadığını test eder. Eğer test edilen bir konum, göç sırasında geçici olarak bir süre için konaklamaya uygun ise yapay-süperorganizmanın bu konumu keşfeden üyeleri (yani yapay organizmalar) hemen keşfedilen ilgili konuma yerleşirler ve bu konumdan itibaren göçlerine devam ederler. DSA algoritmasının işleyişine ait sembolik-kod Şekil 1’de verilmiştir.

DSA’da bir yapay-süperorganizma’yı (yani, $Superorganism_g$, $g=1,2,3,\dots,maxgeneration$) oluşturan yapay-organizmalar (yani, X_i , $i=1,2,3,\dots,n,m,\dots,N$), problem boyutu kadar üye (yani, $x_{i,j}$, $j=1,2,3,\dots,D$) içerirler. Burada, N süperorganizmanın eleman sayısını (yani, popülasyon boyutunu) ve D ilgili problemin boyutunu göstermektedir. DSA’da başlangıç-konumda bir yapay-organizma’ya ait bir üye Eşitlik 1 kullanılarak tanımlanır,

$$x_{i,j} = rand \cdot (up_j - low_j) + low_j \quad (1)$$

DSA’da, yapay-organizmalar $X_i = [x_{i,j}]$ ile tanımlanır ve yapay-organizmalar’dan oluşan yapay-süperorganizma $Superorganism_g = [X_i]_g$ ile gösterilir. Yapay-organizmalar’ın rastgele seçilen üyeleri, başarılı bir göç için çok önemli olan ‘stopover site’ları keşfetmek amacıyla $donor = [X_{random_shuffling(i)}]$ hedeflerine doğru hareket eder (random_shuffling fonksiyonu, $i=\langle 1,2,3,\dots,N \rangle$ vektörüne ait elemanların vektör içerisindeki sıralarını rastgele olarak değiştirir). Yapay-organizmalar’a ait üyelerin, habitat’taki konumlarında meydana gelen değişimin büyüklüğü ‘scale’ değeri ile kontrol edilir. DSA’da, ‘Scale’ değeri, şekil-parametre değeri $2 \cdot rand$ ile kontrol edilen bir birim, gamma-random sayı jeneratörü ve [0 1] aralığında çalışan bir uniform-random sayı jeneratörü birlikte kullanılarak üretilir. ‘Scale’ değerinin hesaplanmasında kullanılan yapı (Şekil 1- satır 10’a bakınız) ilgili yapay-süperorganizma’nın habitat’ta radikal şekilde yön değiştirebilmesine olanak sağlar.

Bir ‘stopover site’ın konumu Eşitlik 2 kullanılarak tanımlanır;

$$StopoverSite = Superorganism + Scale \times (donor - Superorganism) \quad (2)$$

DSA’da süperorganizm’a ait yapay-organizma’ların hangi üyelerinin ‘stopover site’ arama sürecine katılmayacağı rassal doğal bir süreç sonunda belirlenir. İlgili rassal sürecin yapısı, Şekil-1 satır 11-33 arasında verilmiştir.

Eğer, ‘stopover site’ elemanlarından biri bir nedenle habitat sınırlarını aşarsa ilgili eleman rastgele olarak habitat’ta bir başka konuma ötelenir. DSA’da bir ‘stopover site’, o ‘stopover site’i keşfeden bireylerin üyesi olduğu yapay-organizma’nın sahip olduğu kaynaklarından daha verimli ise o yapay-organizma ilgili ‘stopover site’a taşınır. Yapay-organizma’lar yer değiştirdikçe yapay-organizma’ları içeren süperorganizmada global minimuma doğru göçüne devam eder. DSA’nın, DE/best/1, JADE, PSO ve ABC algoritmalarının aksine problemin muhtemel en iyi çözümüne doğru bir şekilde gitme eğilimi yoktur (Çivicioğlu, P., 2012). Bu nedenle multimodal fonksiyonların çözümünde oldukça başarılı bir arama stratejisine sahiptir. DSA’nın sadece iki kontrol parametresi (yani, p_1 ve p_2) bulunmaktadır (Şekil 1, satır 7). p_1 ve p_2 için en uygun değerlerin ne olduğunun belirlenmesi için ayrıntılı deneyler gerçekleştirilmiş ve bu bildiride kullanılan tüm test problemleri için $p_1 = 0.3 \cdot rand$ ve $p_2 = 0.3 \cdot rand$ değerlerinin ilgili test problemlerine en iyi çözümleri sağladığı görülmüştür. DSA, p_1 ve p_2 ’nin başlangıç-değerine aşırı duyarlı değildir ve algoritmik yapısı oldukça basittir. Bu durum DSA’nın farklı mühendislik problemlerine kolayca uygulanabilmesini sağlamaktadır.

DSA'nın, sayısal problemleri çözme başarısının incelenmesi için yaygın olarak kullanılan 50 test probleminden yararlanılmıştır. İlgili test problemleri, Çizelge 1'de verilmiştir. ABC, JDE (Brest, J. ve diğerleri 2007) ve PSO2011, yaygın olarak kullanılan sayısal optimizasyon algoritmaları olduğundan, DSA'nın sayısal-problem çözme başarısını incelemek için ilgili algoritmalar kıyas algoritmaları olarak kullanılmak üzere seçilmiştir.

İlgili test problemleri DSA, ABC, JDE ve PSO2011 kullanılarak 30 farklı başlangıç koşulu için ayrı-ayrı çözülmüştür. Elde edilen sonuçlar, $\alpha = 0.05$ anlamlılık düzeyi için Kruskal-Wallis testi kullanılarak çiftler halinde karşılaştırılmıştır. İstatistiksel testlerde hesaplan p-değerleri bonferroni-holm yöntemi kullanılarak düzeltilmiştir.

DSA, ABC, JDE ve PSO2011'in ilgili test problemlerine sağladığı çözümlerin ortalamaları Çizelge 2'de verilmiştir.

Gerçekleştirilen istatistiksel testler sonucunda elde edilen sonuçlar aşağıda verilmiştir:

- DSA, 34 test probleminin çözümünde ABC'den istatistiksel olarak daha başarılıdır (yani f1-f6, f9, f12, f16-f19, f21-f25, f27, f29-f32, f34, f37-f42, f44, f47-f50).
- DSA'nın ve ABC'nin elde ettiği çözümlerin medyan değeri, 15 test problemi için istatistiksel olarak eşittir (yani f7-f8, f10-f11, f13-f15, f20, f26, f28, f33, f35, f43, f45-f46).
- ABC, 1 test probleminin çözümünde DSA'dan istatistiksel olarak daha başarılıdır (yani f36).

- DSA, 11 test probleminin çözümünde JDE'den istatistiksel olarak daha başarılıdır (yani f2, f13, f17-f18, f23, f33-f36, f42, f45).
- DSA'nın ve JDE'nin elde ettiği çözümlerin medyan değeri, 33 test problemi için istatistiksel olarak eşittir (yani f1, f3-f4, f6-f12, f14-f16, f19-f22, f24-f29, f38-f40, f43-f44, f46-f50).
- JDE, 6 test probleminin çözümünde DSA'dan istatistiksel olarak daha başarılıdır (yani f5, f30-f32, f37, f41).

- DSA, 22 test probleminin çözümünde PSO2011'den istatistiksel olarak daha başarılıdır (yani f1-f6, f13, f16-f18, f22-f23, f27-f28, f30, f33-f34, f36, f42, f45-f46, f49).
- DSA'nın ve PSO2011'in elde ettiği çözümlerin medyan değeri, 24 test problemi için istatistiksel olarak eşittir (yani f7-f12, f14-f15, f19-f21, f24-f26, f29, f31, f35, f38, f40, f43-f44, f47-f48, f50).
- PSO2011, 4 test probleminin çözümünde DSA'dan istatistiksel olarak daha başarılıdır (yani f32, f37, f39, f41).

3. KAMERA KALİBRASYONU

Kamera kalibrasyonu problemi, özetle bir kamera sisteminin sistem modelinin çıkartılmasıdır; kamera kalibrasyonu fiziksel uzay ve görüntü düzlemi arasındaki ilişkiyi modelleyen parametreleri elde etmeyi amaçlar. Özet olarak fiziksel uzaydan görüntü düzlemine geçiş, 3x4 boyutlu bir kamera projeksiyon matrisinin (\mathbf{P}) kullanıldığı Eşitlik-3 yardımıyla açıklanabilir;

$$\begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} = \mathbf{P} \cdot \begin{pmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \end{pmatrix} \quad (3)$$

burada,

$$\begin{aligned} x &= \frac{u_i}{w_i} = \frac{m_{11}X_i^w + m_{12}Y_i^w + m_{13}Z_i^w + m_{14}}{m_{31}X_i^w + m_{32}Y_i^w + m_{33}Z_i^w + m_{34}} \\ y &= \frac{v_i}{w_i} = \frac{m_{21}X_i^w + m_{22}Y_i^w + m_{23}Z_i^w + m_{24}}{m_{31}X_i^w + m_{32}Y_i^w + m_{33}Z_i^w + m_{34}} \end{aligned} \quad (4)$$

Algorithm 1: Pseudo-Code of Differential Search Algorithm

```

Input: ObjFun, SizeOfColony, dim, P1, P2, low, up, and maxcycle
Output: globalminimum, globalminimizer
// please see Matlab for 'ones', 'rand', 'randg', 'randi', 'feval' and 'randperm'
1 function algo_ds(SizeOfColony,dim,P1,P2,low,up,maxcycle)
// INITIALIZATION
2 globalminimum=inf
3 if numel(low)==1 then
4     low=low*ones(1,dim)
5     up=up*ones(1,dim)
6 end
7 superorganism=genpop(SizeOfColony,dim,low,up)
8 y=ObjFun(superorganism)
9 for epk form 1 to maxcycle do
// SELECTION-I
10 donor=superorganism(randperm(SizeOfColony),:)
11 scale=randg(2*rand)*(rand-rand)
// MUTATION (Morphogenesis)
12 stopover=superorganism+scale.*(donor-superorganism)
// P1=P2='3*rand' has been used in DSA
13 p1=eval(P1)
14 p2=eval(P2)
15 r=rand(SizeOfColony,dim)
16 if rand<rand then
17     if rand<p1 then
18         for i form 1 to SizeOfColony do
19             | r(i,:)=r(i,:) < rand
20         end
21     else
22         for i form 1 to SizeOfColony do
23             | r(i,randi(dim))=0
24         end
25     end
26 else
27     for i form 1 to SizeOfColony do
28         | d=randi(dim,1,[p2·rand])
29         | for j form 1 to size(d,2) do
30             | r(i,d(j))=0
31         end
32     end
33 end
34 r=r>0
// RECOMBINATION
35 stopover(r)=superorganism(r)
36 stopover=update(stopover,low,up)
37 fit_s=ObjFun(stopover)
38 ind=fit_s<y
// SELECTION-II
39 if numel(ind)>0 then
40     y(ind)=fit_s(ind)
41     superorganism(ind,:)=stopover(ind,:)
42 end
43 yindexbest ← min(y) | indexbest ∈ {1, 2, 3, ..., SizeOfColony}
44 if yindexbest < globalminimum then
45     globalminimum:=yindexbest
46     globalminimizer:=superorganism(indexbest,:)
47     // Export globalminimum and globalminimizer
48 end
49 end
// Sub-Functions
50 function pop=genpop(a,b,low,up)
51 pop=ones(a,b)
52 for i form 1 to a do
53     | for j form 1 to b do
54         | pop(i,j)=rand*(up(j)-low(j))+low(j)
55     end
56 end
57 function p=update(p,low,up)
58 [SizeOfColony,dim]=size(p)
59 for i form 1 to SizeOfColony do
60     | for j form 1 to dim do
61         | if p(i,j)<low(j) then
62             | p(i,j)=rand*(up(j)-low(j))+low(j)
63         end
64         | if p(i,j)>up(j) then
65             | p(i,j)=rand*(up(j)-low(j))+low(j)
66         end
67     end
68 end
69 end

```

Şekil 1 : DSA algoritmasının sembolik-kodu (Çivicioğlu, 2012).

Çizelge 1: DSA, ABC, JDE ve PSO2011'in sayısal problemleri çözme başarılarının karşılaştırılmasında kullanılan test problemleri (Çivicioğlu, 2012, Karaboğa, D., Baştürk, B., 2007).

Test Problemi	Boyut	Alt-Limit	Üst-Limit	
f1	Foxholes	2	-65.536	65.536
f2	Goldsteinprice	2	-2	2
f3	Penalized	30	-50	50
f4	Penalized2	30	-50	50
f5	Ackley	30	-32	32
f6	Beale	5	-4.5	4.5
f7	Bohachevsky1	2	-100	100
f8	Bohachevsky2	2	-100	100
f9	Bohachevsky3	2	-100	100
f10	Booth	2	-10	10
f11	Branin	2	-5	10
f12	Colville	4	-10	10
f13	Dixonprice	30	-10	10
f14	Easom	2	-100	100
f15	Fletcher	2	-pi	pi
f16	Fletcher	5	-pi	pi
f17	Fletcher	10	-pi	pi
f18	Griewank	30	-600	600
f19	Hartman3	3	0	1
f20	Hartman6	6	0	1
f21	Kowalik	4	-5	5
f22	Langermann	2	0	10
f23	Langermann	5	0	10
f24	Langermann	10	0	10
f25	Matyas	2	-10	10
f26	Michalewics	2	0	pi
f27	Michalewics	5	0	pi
f28	Michalewics	10	0	pi
f29	Perm	4	-4	4
f30	Powell	24	-4	5
f31	Powersum	4	0	4
f32	Quartic	30	-1.28	1.28
f33	Rastrigin	30	-5.12	5.12
f34	Rosenbrock	30	-30	30
f35	Schaffer	2	-100	100
f36	Schwefel	30	-500	500
f37	Schwefel_1_2	30	-100	100
f38	Schwefel_2_22	30	-10	10
f39	Shekel10	4	0	10
f40	Shekel5	4	0	10
f41	Shekel7	4	0	10
f42	Shubert	2	-10	10
f43	Camelback	2	-5	5
f44	Sphere2	30	-100	100
f45	Step2	30	-100	100
f46	Stepint	5	-5.12	5.12
f47	Sumsquares	30	-10	10
f48	Trid	6	-36	36
f49	Trid	10	-100	100
f50	Zakharov	10	-5	10

Çizelge 2: DSA, ABC, JDE ve PSO2011'in test problemlerine sağladıkları ortalama çözümler.

#	ABC	JDE	PSO2011	DSA
f1	0.998003837794449570	1.064140548428523600	1.331602926487629900	0.998003837794449570
f2	3.000000046542302000	2.9999999999920500	2.9999999999920500	2.9999999999919200
f3	0.00000000000000436	0.003455634008349869	0.127872806239163000	0.000000000000000000
f4	0.00000000000000412	0.000732491055725586	0.004394946334353517	0.000000000000000000
f5	0.00000000000034046	0.081101705642285957	1.521432297372501200	0.000000000000021020
f6	0.00000000000002760	0.000000000000000000	0.000000004192296805	0.000000000000000000
f7	0.000000000000000000	0.000000000000000000	0.000000000000000000	0.000000000000000000
f8	0.000000000000000000	0.000000000000000000	0.000000000000000000	0.000000000000000000
f9	0.00000000000000566	0.000000000000000000	0.000000000000000000	0.000000000000000000
f10	0.00000000000000002	0.000000000000000000	0.000000000000000000	0.000000000000000000
f11	0.397887357729738160	0.397887357729738160	0.397887357729738160	0.397887357729738160
f12	0.071567506072597015	0.000000000000000000	0.000000000000000000	0.000000000000000000
f13	0.00000000000003808	0.666666666666666630	0.666666666666675070	0.400000009128288200
f14	-0.99999999999999890	-0.99999999999999890	-0.99999999999999890	-0.99999999999999890
f15	0.000000000000000000	0.000000000000000000	0.000000000000000000	0.000000000000000000
f16	0.021868849833187230	0.944372865543282590	48.746516444692730000	0.000000000000000000
f17	11.068149625354797000	713.722697462691600000	918.951849278285180000	0.000000000000000000
f18	0.00000000000000048	0.004819357854318506	0.006894369481971326	0.000000000000000000
f19	-3.862782147820753100	-3.862782147820753100	-3.862782147820753100	-3.862782147820753100
f20	-3.321995171584244000	-3.298216547320261100	-3.318032067540246800	-3.321995171584244000
f21	0.000441486635962576	0.000368531813760384	0.000307485987805606	0.000307485987805604
f22	-1.080938442134438100	-1.076428076265744800	-1.080938442134438100	-1.080938442134438100
f23	-1.499999007080076200	-1.343139943257970000	-1.389199220074464500	-1.499999223352495700
f24	-0.840634809650068200	-0.882715279883576230	-0.916620678868023010	-1.190564611591903100
f25	0.00000000000000446	0.000000000000000000	0.000000000000000000	0.000000000000000000
f26	-1.821043683677681300	-1.821043683677681300	-1.821043683677681300	-1.821043683677681300
f27	-4.693468451957112800	-4.689345693261705300	-4.656564639705393900	-4.693468451957112800
f28	-9.660151715641349700	-9.639723098613243800	-8.971733030754931400	-9.660151715641349700
f29	0.083844001403803228	0.015410513005585573	0.011968722456044061	0.001239881585104723
f30	0.000260433001346215	0.000000000000000066	0.000013071891200815	0.000001041051900331
f31	0.007790531109495759	0.002018511626149027	0.000125488283423822	0.000078214195963962
f32	0.025016325252703038	0.001301031618067892	0.000354834551317880	0.006055501521948293
f33	0.000000000000000000	1.127620264705736500	25.636760225867558000	0.000000000000001894
f34	0.285683346590413220	1.063099694480250000	2.675704311426969600	0.265774923620062280
f35	0.000000000000000005	0.003886363951413971	0.000000000000000000	0.000000000000000000
f36	-12569.486618173018000	-12304.974337534099000	-7684.610475778376900	-12569.486618173018000
f37	14.566873412694815000	0.000000000000000000	0.000000000000000000	0.000000000044391259
f38	0.00000000000000489	0.000000000000000000	0.000000000000000000	0.000000000000000000
f39	-10.536409816692048000	-10.313043716242612000	-10.106187362165304000	-10.536409816692050000
f40	-10.153199679058224000	-9.565613576121565500	-9.537393808204546600	-10.153199679058224000
f41	-10.402940566818662000	-9.161581335473734100	-10.402940566818662000	-10.402940566818662000
f42	-186.730908831023980000	-186.730908831023980000	-186.730907356988470000	-186.730908831023980000
f43	-1.031628453489877000	-1.031628453489877000	-1.031628453489877000	-1.031628453489877000
f44	0.00000000000000447	0.000000000000000000	0.000000000000000000	0.000000000000000000
f45	0.000000000000000000	0.900000000000000020	2.299999999999998000	0.000000000000000000
f46	0.000000000000000000	0.000000000000000000	0.133333333333333300	0.000000000000000000
f47	0.00000000000000473	0.000000000000000000	0.000000000000000000	0.000000000000000000
f48	-49.99999999999659000	-50.0000000000206000	-50.0000000000206000	-50.0000000000206000
f49	-209.99999999947050000	-210.0000000002670000	-210.0000000001450000	-210.0000000002730000
f50	0.000000040238042357	0.000000000000000000	0.000000000000000000	0.000000000000000000

ile tanımlanmıştır. Eşitlik-4'ün çözülmesi için en az 6 noktaya ait fiziksel uzay ve imge koordinatları gereklidir. Yukarıda verilen eşitlikler yeniden düzenlenirse Eşitlik 5 elde edilir;

$$A \cdot p = 0 \quad (5)$$

buradan,

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1 X_1 & -x_1 Y_1 & -x_1 Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1 X_1 & -y_1 Y_1 & -y_1 Z_1 & -y_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2 X_2 & -x_2 Y_2 & -x_2 Z_2 & -x_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2 X_2 & -y_2 Y_2 & -y_2 Z_2 & -y_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -x_N X_N & -x_N Y_N & -x_N Z_N & -x_N \end{bmatrix} \quad (6)$$

olmak üzere,

$$p = [m_{11}, m_{12}, \dots, m_{33}, m_{34}]^T \quad (7)$$

olarak tanımlanabilir. A'nın rankı, 11 olduğundan, p vektörü tekil değerlere ayrıştırma (SVD; $A=U \cdot D \cdot V^T$) yöntemi kullanılarak kolayca çözülebilir; A'nın en küçük tekil değerine karşılık gelen V matrisi kolonu, çözüm olarak kullanılır. Bu durumda elde edilen \mathbf{P} parametreleri belirsiz bir ölçek faktörüne göre belirlenmiş olur. Kamera parametrelerinin hesaplanması için \mathbf{P} 'nin ayrıştırılması gerekir; $\mathbf{P}=\mathbf{K} \cdot [\mathbf{R}|\mathbf{T}]$.

\mathbf{P} 'nin yapısı ayrıntılı olarak yazılırsa,

$$\mathbf{P} = \begin{bmatrix} -f_x \cdot r_{11} + o_x \cdot r_{31} & -f_x \cdot r_{12} + o_x \cdot r_{32} & -f_x \cdot r_{13} + o_x \cdot r_{33} & -f_x \cdot T_x + o_x \cdot T_z \\ -f_y \cdot r_{21} + o_y \cdot r_{31} & -f_y \cdot r_{22} + o_y \cdot r_{32} & -f_y \cdot r_{23} + o_y \cdot r_{33} & -f_y \cdot T_y + o_y \cdot T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \quad (8)$$

elde edilir. \mathbf{P} 'yi normalize etmek için kullanılacak ölçek değeri $\gamma = \sqrt{p_{31}^2 + p_{32}^2 + p_{33}^2} = \sqrt{r_{31}^2 + r_{32}^2 + r_{33}^2}$ kullanılarak hesaplanır ve $\mathbf{P} \leftarrow (\mathbf{P} / \lambda)$ kullanılarak normalize edilir. Ana noktanın koordinatları, Eşitlik 9 kullanılarak ayrıştırılabilir;

$$\begin{aligned} o_x &= \mathbf{P}_{1,13} \cdot \mathbf{P}_{3,13}^T \\ o_y &= \mathbf{P}_{2,13} \cdot \mathbf{P}_{3,13}^T \end{aligned} \quad (9)$$

$\mathbf{K}_{3,13} = [0 \ 0 \ 1]$ olmak üzere, odak uzaklıkları Eşitlik 10 kullanılarak ayrıştırılır;

$$\begin{aligned} f_x &= \sqrt{\mathbf{P}_{1,13} \cdot \mathbf{P}_{1,13}^T - \mathbf{K}_{1,3}^2} \\ f_y &= \sqrt{\mathbf{P}_{2,13} \cdot \mathbf{P}_{2,13}^T - \mathbf{K}_{2,3}^2} \end{aligned} \quad (10)$$

Öteleme vektörü Eşitlik 11 kullanılarak ayrıştırılır;

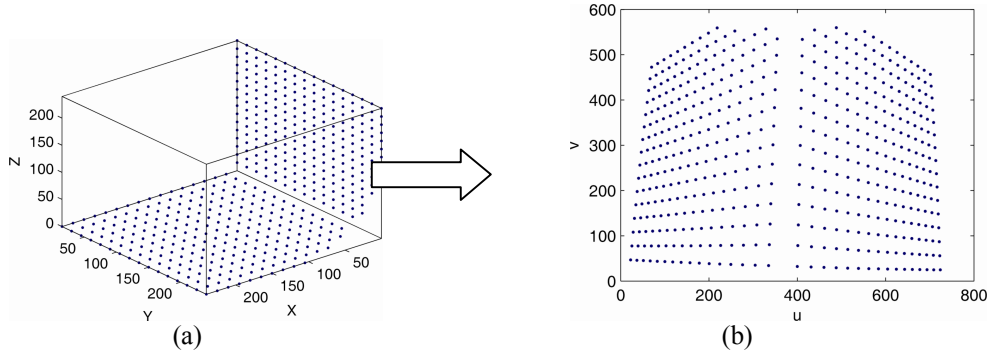
$$\begin{aligned} T_z &= \mathbf{P}_{3,4} \\ T_x &= \frac{P_{1,4} - o_x \cdot T_z}{f_x} \\ T_y &= \frac{P_{2,4} - o_y \cdot T_z}{f_y} \end{aligned} \quad (11)$$

Rotasyon matrisinin ayrıştırılması için Eşitlik 12 kullanılır;

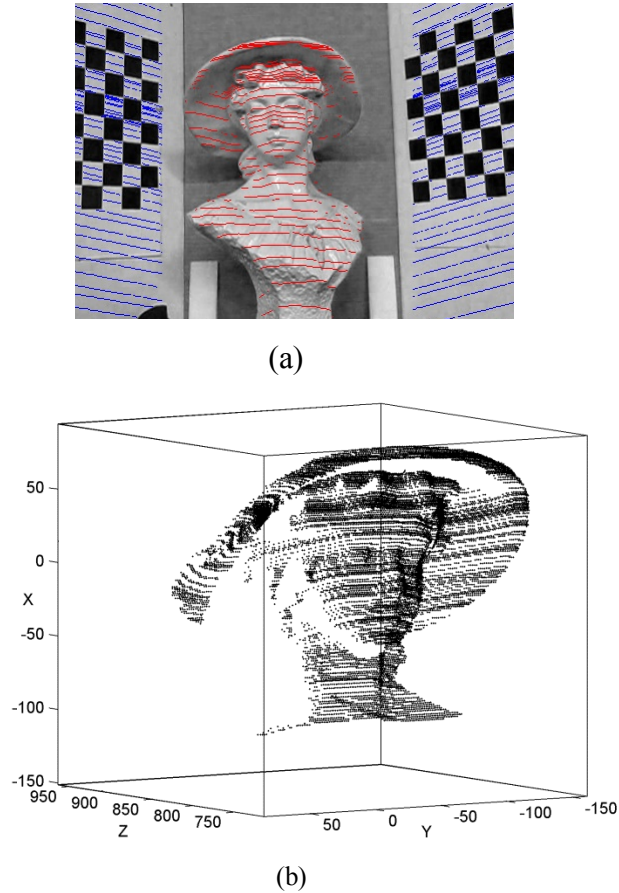
$$\begin{aligned} r_{3,1} &= \mathbf{P}_{3,1} ; r_{3,2} = \mathbf{P}_{3,2} ; r_{3,3} = \mathbf{P}_{3,3} ; r_{1,13} = \frac{\mathbf{P}_{1,13} - o_x \cdot \mathbf{P}_{3,13}}{f_x} ; r_{2,13} = \frac{\mathbf{P}_{2,13} - o_y \cdot \mathbf{P}_{3,13}}{f_y} \\ [\mathbf{U}, \mathbf{D}, \mathbf{V}] &= \text{svd}(\mathbf{R}) \\ \mathbf{D} &= \text{eye}(3,3) \\ \mathbf{R} &= \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^T \end{aligned} \quad (12)$$

Eğer $T_z < 0$ ise $\mathbf{T} := -\mathbf{T}$ ve $\mathbf{R} := -\mathbf{R}$ güncellemesi yapılmalıdır.

Bu makalede kamera kalibrasyon değerlerinin geliştirilmesi için ABC, JDE, PSO2011 ve DSA kullanılarak gerçekleştirilen testlerde kullanılan arama uzayına ait sınır değerleri, DLT (Hartley, H., Zisserman, A., 2003) ile elde edilen P 'nin ilgili parametrelerinin, $\pm \%10$ sınımlarıyla tanımlanmıştır. Amaç fonksiyonu $\arg \min \|P \cdot X_{homogen} - u_{homogen}\|$ olarak belirlenmiştir. Testlerde kullanılan veriler Şekil 2'de gösterilmiştir. metasezgisel algoritmalar için popülasyon büyüklüğü 20, parametre boyutu 12, ve en-büyük iterasyon sayısı 2.000.000 olarak belirlenmiştir. Hesaplamalar 20 defa farklı başlangıç popülasyonları kullanılarak tekrarlanmıştır. Hesaplamalar sonunda ABC 0.45835, JDE 0.45772, PSO2011 0.45979 ve DSA 0.45737 en-iyi global minimum değerlerine ulaşmıştır. Elde edilen sonuçlar Çizelge 3'de gösterilmiştir. DSA ile elde edilen kalibrasyon çözümleri kullanılarak, düzleştirme amaçlı basit bir çizgi lazer kaynağı ve bir web-kamera yardımıyla 3D bir model ölçülmüştür. Elde edilen nokta bulutu Şekil 3'te gösterilmiştir. Bu uygulamada 3D konumsal doğruluk $\sim \pm 2.4$ mm. olarak hesaplanmıştır.



Şekil 2: Kamera kalibrasyon testinde kullanılan veriler; (a) Fiziksel uzaya ait cisim noktaları (X,Y,Z), (b) İlgili cisim noktalarına karşılık gelen resim-uzayına ait nokta deseni (u,v).



Şekil 3: DSA tabanlı kamera kalibrasyon geliştirme tekniğinin uygulaması; (a) 3D model, (b) Modele ait nokta bulutu.

Çizelge 3: DLT, ABC, JDE ve DSA kullanılarak elde edilen kamera kalibrasyon parametreleri ve doğruluk değeri.

#	K	R	T	MSE
DLT	$\begin{bmatrix} 969.9421 & 0 & 372.0507 \\ 0 & 962.9763 & 276.7439 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.7117 & 0.0304 & -0.7018 \\ -0.1873 & -0.1873 & -0.1477 \\ 0.6770 & 0.6770 & 0.6968 \end{bmatrix}$	$\begin{bmatrix} -2.0671 \\ -91.4093 \\ 297.5354 \end{bmatrix}$	0.3057
ABC	$\begin{bmatrix} 970.5029 & 0 & 371.8879 \\ 0 & 963.4719 & 276.6568 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.7118 & 0.0305 & -0.7017 \\ -0.1873 & -0.9711 & -0.1477 \\ 0.6770 & 0.2366 & 0.6970 \end{bmatrix}$	$\begin{bmatrix} -2.0106 \\ -91.3872 \\ 297.7743 \end{bmatrix}$	0.3052
JDE	$\begin{bmatrix} 970.4977 & 0 & 371.8839 \\ 0 & 963.4699 & 276.6533 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.7118 & 0.0305 & -0.7017 \\ -0.1873 & -0.9711 & -0.1477 \\ 0.6770 & 0.2366 & 0.6970 \end{bmatrix}$	$\begin{bmatrix} -2.0099 \\ -91.3859 \\ 297.7741 \end{bmatrix}$	0.3049
PSO 2011	$\begin{bmatrix} 970.5033 & 0 & 371.9012 \\ 0 & 963.4802 & 276.5570 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.7116 & 0.0305 & -0.7018 \\ -0.1873 & -0.9712 & -0.1479 \\ 0.6769 & 0.2365 & 0.6968 \end{bmatrix}$	$\begin{bmatrix} -2.0102 \\ -91.3863 \\ 297.7738 \end{bmatrix}$	0.3056
DSA	$\begin{bmatrix} 970.4977 & 0 & 371.8839 \\ 0 & 963.4699 & 276.6533 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.7118 & 0.0305 & -0.7017 \\ -0.1873 & -0.9711 & -0.1477 \\ 0.6770 & 0.2366 & 0.6970 \end{bmatrix}$	$\begin{bmatrix} -2.0671 \\ -91.4093 \\ 297.5354 \end{bmatrix}$	0.3038

4. SONUÇLAR

Bu bildiriye Differansiyel Arama Algoritmasının, bir kamera kalibrasyon uygulamasından elde edilen sonuçlarının geliştirilmesinde nasıl kullanılabileceğine ilişkin ayrıntılı bir uygulama sunulmuştur. Elde edilen sonuçlar DSA'nın multimodal kamera kalibrasyon problemine kısmi iyileştirme getirebilme potansiyeli olduğunu göstermiştir.

Teşekkür

Bu çalışma, 110Y309 nolu Tübitak projesi kapsamında desteklenmiştir.

KAYNAKLAR

Beşdok, E. 2007, Neurovision With Resilient Neural Networks, 9th International Conference On Visual Information Systems, Advances In Visual Information Systems, Lecture Notes In Computer Science, 4781, 438-444.

Beşdok, E., Çivicioğlu, P., 2006 (a), Adaptive Implicit-Camera Calibration in Photogrammetry Using Anfis, 10th International Conference On Knowledge-Based And Intelligent Information And Engineering Systems, Lecture Notes In Artificial Intelligence, 4251, 606-613.

Brest, J., Boskoviç, B., Greiner, S., Zumer, V., Maucec, M.S., 2007, Performance comparison of self-adaptive and adaptive differential evolution algorithms. Soft Computing, 11, 617-629.

Çivicioğlu, P., 2012, Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, Computers & Geosciences, 46, 229-247.

Çivicioğlu, P., Beşdok, E., 2006 (b), Implicit Camera Calibration By Using Resilient Neural Networks, 13th International Conference On Neural Information Processing, Lecture Notes In Computer Science, 4233, 632-640.

P. Çiviciođlu, T. Kurban, E. Beşdok, DSA Tabanlı Yeni Bir Kamera Kalibrasyon Yöntemi

Fusiello, A., Trucco, E., Verri, A., 2000, A Compact Algorithm For Rectification Of Stereo Pairs, Machine Vision And Applications, 12 (1), 16-22.

Hartley, H., Zisserman, A., 2003 Multiple View Geometry in computer vision. Cambridge University Press. ISBN 978-0-521-54051-3

Heikkila, J., 2000, Geometric Camera Calibration Using Circular Control Points, IEEE Transactions On Pattern Analysis And Machine Intelligence, 22 (10), 1066-1077.

Karabođa, D., Baştürk, B., 2007. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm, Journal of Global Optimization 39, 459–471.

Omran, M.G.H., Clerc, M., 2011, /<http://www.particleswarm.info>, [accessed 02 October, 2011].

Tsai, RY., 1987, A Versatile Camera Calibration Technique For High-Accuracy 3d Machine Vision Metrology Using Off-The-Shelf TV Cameras And Lenses, IEEE Journal Of Robotics And Automation, 3 (4), 323-344.

Zhang, ZY., 2000, A Flexible New Technique For Camera Calibration, IEEE Transactions On Pattern Analysis And Machine Intelligence, 22 (11), 1330-1334.