

Hiperspektral Görüntülerde Son Elemanların Geliştirilmiş Kısıtlı Optimizasyonla Bulunması

Endmember Detection using Enhanced Constrained Optimization in Hyperspectral Imaging

Seniha Esen Yüksel
Elektrik ve Elektronik Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
Email: eyuksel@ee.hacettepe.edu.tr

Özetçe —Hiperspektral görüntüleme çokça kullanılan yinelemeli kısıtlı son elemanlar ve seyrekliği teşvik eden yinelemeli kısıtlı son elemanlar gibi doğrusal spektral bileşenler analizi yapan algoritmalarda son elemanların sıfırdan küçük veya birden büyük olarak bulunabildiği gözlenmiştir. Fakat yansıma değerleri her zaman 0 ve 1 arasında olmalıdır, dolayısıyla böyle sonuçlar fiziksel olarak mümkün değildir. Bu tür durumları engellemek için son elemanların alabilecekları değerlere kısıt koyan ve son elemanları bu kısıtlar çerçevesinde kuadratik programlama yöntemiyle çözen yeni bir algoritma geliştirilmiştir. Yeni geliştirilen KİDDO algoritması SPICE algoritması ile karşılaştırılmış, son eleman sayısını SPICE'tan daha gürbüz bir şekilde bulabildiği ve bunu yaparken de istenen sınır değerlerinin içinde kaldığı gösterilmiştir.

Anahtar Kelimeler—Hiperspektral görüntüleme; son eleman analizi, kuadratik programlama.

Abstract—A new algorithm is presented for linear spectral mixture analysis that respects the constraints on the endmembers. The results show that it provides a more robust solution as compared to the ICE and SPICE algorithms due to the use of constrained quadratic optimization for endmember detection.

Keywords—Hyperspectral Image Processing, end member detection, quadratic optimization.

I. GİRİŞ

Hiperspektral görüntüleme dijital görüntüleme ile spektroskopinin güçlerini birleştiren bir algılama teknolojisidir. Sadece kırmızı, yeşil, mavi olmak üzere üç banttan görüntü işleyen klasik görüntüleme yöntemlerine kıyasla, hiperspektral kameralar elektromanyetik spektrumu çok sayıda banda böler ve yüzlerce dalga boyundan görüntü alır. Bu görüntüler nesnelerin farklı dalga boylarında ışığı ne kadar yansıtıklarının ölçümüdür. Farklı nesnelere değişik dalga boylarında ışığı farklı şekillerde yansıtır. Bir materyalin ışığı yansıtma özelliği kendisine has spektral imzasını oluşturur [?]. Hiperspektral görüntüleme, içerdiği bu materyale has imza bilgisi sayesinde, malzeme tanımlama ve ayırt etme konularında çok detaylı bilgi sağlar. Son on yıldır hiperspektral görüntü analizi (HGA) çok

ciddi bir ivme kazanmıştır. HGA çalışmaları savunma sanayii başta olmak üzere maden haritalama, gıda denetimi, çevre izleme gibi pek çok alanda uygulama bulmaktadır.

Saf veya saf olduğu varsayılan bir materyale ait spektral imzayı bulma işlemine son eleman analizi denilmektedir. Bir karışımdaki saf materyallerin hangi oranda karıştığını bulma işlemine ise bolluk veya katışım oranı analizi denilmektedir. Hiperspektral görüntüleme en önemli problemlerden bir tanesi son elemanların ve bunların katışım oranlarının otomatik olarak tesbit edilebilmesidir. Yinelemeli kısıtlı son elemanlar algoritması (Iterated Constrained Endmembers - ICE) [?] ve seyrekliği teşvik eden yinelemeli kısıtlı son elemanlar algoritması (Sparsity Promoting ICE - SPICE) [?] ve benzeri birçok çalışma [?], [?], [?], [?], [?] son elemanların bulunması için sözde-ters (pseudo-inverse) hesaplamayı gerektirmektedir. Fakat sözde-ters hesaplayan bu algoritmaların bazen fiziksel olarak gerçekçi olmayan (negatif gibi) değerler verdiği saptanmıştır. Bu problemi gidermek ve reflektans değerleri [0, 1] aralığında olan son elemanlar bulmak için, bu çalışmada kısıtlı ikinci dereceden denklemlerle optimizasyon (KİDDO) algoritması geliştirilmiştir.

Bildirinin II. bölümünde ICE ve SPICE algoritmaları anlatılacaktır. Bunu takiben, III. bölümde KİDDO algoritması geliştirilecek ve IV. bölümde de deneysel sonuçlar gösterilecektir.

II. ICE VE SPICE ALGORİTMALARI

\mathbf{X} matrisi $d \times N$ boyunda hiperspektral bir görüntü olsun. Burada N pixel sayısını ve d hiperspektral bant sayısını temsil etmektedir. \mathbf{X} matrisinin her bir sütunu, \mathbf{X}_i , görüntüde i . piksele denk gelen $d \times 1$ vektörü temsil etmektedir. Ayrıca, \mathbf{X} matrisinin her bir sırası $N \times 1$ boyundaki \mathbf{x}_j vektörüdür. \mathbf{x}_j , $j = 1, \dots, d$ vektörü, j . spektral banta karşılık gelen N gözlemi belirtmektedir.

Bunun yanısıra görüntüde toplam M tane son eleman bulunuyor olsun. Bu son elemanların yanyana dizilmesinden elde edilen $d \times M$ boyuttaki son eleman matrisi de \mathbf{E} olsun. Bu matrisin her bir sütunu, \mathbf{E}_k , k . son elemanın yer aldığı $d \times 1$ vektördür ve $\mathbf{E}_k^T = (e_{k1}, \dots, e_{kd})$ şeklinde gösterilir.

Burada e_{kj} , j . spektral banttaki k . son elemanın değerine işaret eder. \mathbf{E} matrixinin her bir sırası ise $\mathbf{e}_j^T = (e_{1j}, \dots, e_{Mj})$ vektörünü oluşturur. Burada \mathbf{e}_j , son elemanın j . spektral banta karşılık gelen $M \times 1$ vektördür.

Doğrusal karışım modeline göre, bir hiperspektral görüntü son elemanların katışım katsayılarıyla orantılı olarak eklenmelerinden oluşmaktadır:

$$\mathbf{X}_i = \sum_{k=1}^M p_{ik} \mathbf{E}_k + \epsilon_i \quad i = 1, \dots, N \quad (1)$$

burada ϵ_i bir hata terimi, p_{ik} ise i . pikseldeki k . son elemanın katışım katsayısıdır ve aşağıdaki kısıtları sağlar:

$$\sum_{k=1}^M p_{ik} = 1, \quad p_{ik} \geq 0; \quad k = 1, \dots, M. \quad (2)$$

Bu katışım katsayıları $N \times M$ boyundaki, \mathbf{P} olarak adlandırılan katışım oranı matrisini oluşturur. Bu matrix her bir pikselde her bir son elemanın ne oranda bulunduğunu gösterir.

SPICE algoritması hem son elemanları hem de katışım oranlarını RSS_{reg}^* amaç fonksiyonunu minimize ederek bulur:

$$RSS_{reg}^* = (1 - \mu) \frac{RSS}{N} + \mu V + SPT \quad (3)$$

burada RSS artık kareler toplamını (residual sum of squares), V düzenleme terimini, SPT ise seyrekliği teşvik edici terimi temsil eder. μ ise $(0, 1)$ aralığında ödünleşim (tradeoff) değişkenidir ve tüm veri seti için sabit tutulur. Bu terimler sırasıyla aşağıda açıklanmıştır.

Denklem 3'teki RSS terimi Denklem 1'de verilen hatayı en aza indirmek için artık kareler toplamını hesaplar:

$$RSS = \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^T \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right) \quad (4)$$

$$= \sum_{j=1}^d (\mathbf{x}_j - \mathbf{P} \mathbf{e}_j)^T (\mathbf{x}_j - \mathbf{P} \mathbf{e}_j) \quad (5)$$

Burada birinci ve ikinci satırların eşitliği ICE makalesinde gösterilmiştir [?].

Doğrusal karışım modeline göre son elemanlar bir simplexin köşelerini oluşturmaktadır. Fakat gürültülü ortamlarda RSS terimi gürültüden çokça etkilenebilmekte ve gereğinden geniş bir dışbükey sınıra neden olmaktadır. Bunu önlemek için RSS^* 'e bir de düzenleme terimi eklenir. Denklem 3'teki V bir düzenleme terimidir ve simpleksin boyutunu kısıtlamaya yarar. Simpleksin köşelerinin birbirlerine olan uzaklığının karelerinin toplamını küçültür ve son elemanların sayısı M 'den bağımsız olması için standartlaştırılmıştır:

$$V = \frac{1}{M(M-1)} \sum_{k=1}^{M-1} \sum_{l=k+1}^M (\mathbf{E}_k - \mathbf{E}_l)^T (\mathbf{E}_k - \mathbf{E}_l) \quad (6)$$

$$= \frac{1}{M(M-1)} \sum_{j=1}^d \mathbf{e}_j^T (M \mathbf{I}_M - \mathbf{1} \mathbf{1}^T) \mathbf{e}_j \quad (7)$$

burada \mathbf{I}_M , $M \times M$ boyundaki birim matrisi göstermektedir, $\mathbf{1}$ ise $M \times 1$ boyutunda birlerden oluşan vektördür. Birinci ve ikinci satırların eşitliği ICE makalesinde gösterilmiştir [?].

Denklem 3'te eşitliğin sağındaki ilk iki terim ICE algoritmasında geliştirilmiştir; üçüncü terim (SPT) eklendiğinde SPICE algoritması elde edilir. Ortamdaki toplam son eleman sayısı doğru olarak bilindiğinde, ICE algoritması son elemanları bulmak için mükemmel bir yöntemdir. Fakat toplam son eleman sayısı M bilinmediğinde, bu sayıyı otomatik olarak belirlemek için bir mekanizması bulunmamaktadır. Bu da algoritmanın başarısını büyük ölçüde etkilemektedir.

Ortamdaki toplam son eleman sayısını otomatik olarak bulmak için SPICE algoritması seyrekliği teşvik edici (sparsity promoting) SPT terimini ekler. Bu terim gereksiz son elemanların katışım oranlarını sıfıra çekerek bu son elemanların elenmesini amaçlar. Denklem 3'teki, seyrekliği teşvik edici SPT terimi şöyle hesaplanır:

$$SPT = \sum_{k=1}^M \gamma_k \sum_{i=1}^N p_{ik} \quad (8)$$

ve

$$\gamma_k = \frac{\Gamma}{\sum_{i=1}^N p_{ik}} \quad (9)$$

ve Γ katışım oranlarının hangi hızla sıfıra gideceğini ayarlayan bir sabittir. Denklem 9'e göre, eğer bir son elemanın katışım oranlarının toplamı küçük ise γ_k katsayısı daha büyük olur ve katışım oranlarının minimizasyonunu hızlandırır [?], [?].

Bu bilgiler ışığında, Denklem 3'ün hem tüm son elemanlar için hem de her bir pikseldeki katışım oranlarını için minimize edilmesi gerekir. Hem ICE hem de SPICE algoritmaları bu işlemi yinelemeli olarak yaparlar. Birbirinin ardısına, önce son eleman kestirimlerine göre katışım matrisi \mathbf{P} 'yi hesaplarlar, sonra da katışım kestirimleri sabit tutarak son elemanları tekrar hesaplarlar.

Katışım oranlarının bulunması: Denklem 3'teki V terimi katışım oranlarını içermemektedir. Geride kalan terimler denklem 2'deki kısıtlar kullanılarak çözülebilir.

Son elemanların bulunması: Burada da SPT terimi son eleman bilgisi içermemektedir. Denklem 3'te SPT haricindeki terimler sözde-ters yöntemi ile çözülür [?]:

$$\mathbf{e}_j = \left\{ \mathbf{P}^T \mathbf{P} + \lambda \left(\mathbf{I}_M - \frac{\mathbf{1} \mathbf{1}^T}{M} \right) \right\}^{-1} \mathbf{P}^T \mathbf{x}_j \quad (10)$$

III. KİDDO: KISITLI İKİNCİ DERECE DENKLEMLERLE OPTİMİZASYON

ICE ve SPICE algoritmalarında sözde-ters yöntemlerin kullanılması sıfırdan küçük veya birden büyük son elemanlar bulunmasına yol açmaktadır. Fakat yansıma değerleri her zaman 0 ve 1 arasında olmalıdır, dolayısıyla böyle sonuçlar fiziksel olarak mümkün değildir. Bu tür durumları engellemek için son elemanların alabilecekleri değerlere kısıt koyan ve son elemanları bu kısıtlar çerçevesinde ikinci dereceden programlama yöntemiyle çözen KİDDO algoritması geliştirilmiştir.

j . spektral banttaki son elemanlar e_j olsun, öyle ki $e_j^T = (e_{1j}, \dots, e_{kj}, \dots, e_{Mj})$ ve $j = 1, \dots, d$. Bu değerlerin her biri $0 \leq e_{kj} \leq 1$ kısıtını sağlamalıdır.

Bu kısıtla birlikte, minimizasyon için sözde-ters yöntemler yerine kuadratik programlama kullanılması gerekmektedir. Genel tabiriyle optimizasyon fonksiyonu $e'Ae + e'b$ şeklinde yazılabilirse standart optimizasyon kütüphanelerini kullanılabilir. Bunun için RSS_{new} terimi denklem 12'de verildiği şekilde yeniden düzenlenmiştir (denklem 12 bir sonraki sayfadadır).

Denklem 12'de verilen terimler şöyledir:

$$H = \mathbf{P}^T \mathbf{P} + \lambda \left(\mathbf{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \quad (13)$$

$$f = -2\mathbf{P}^T \mathbf{x}_j \quad (14)$$

$$\lambda = \frac{N\mu}{(M-1)(1-\mu)} \quad (15)$$

Bu denklemler standart optimizasyon kütüphaneleriyle çözülebilmektedir. Optimasyon her bir bant için ayrı ayrı yapılır ve sınır kısıtları olarak da üst limit $ub = 1$ ve alt limit $lb = 0$ verilir. Burada $\mathbf{1}$, $M \times 1$ boyuttaki bir vektörünü, $\mathbf{0}$ ise $M \times 1$ sıfır vektörünü göstermektedir.

IV. DENEYSSEL SONUÇLAR

Bir üçgenin köşelerini oluşturmaları amacıyla $([0+eps, 0+eps]; [0+eps, 1-eps]; [1-eps, 0+eps])$ koordinatlarında üç son eleman belirlenmiştir. Burada eps epsilon komşuluğuna denk gelmektedir. Materyallerin tam olarak sıfır yansıması olamayacağı göz önüne alınarak eklenmiştir. Daha sonra düzgün dağılım kullanılarak rastgele sayılar üretilmiş ve katışım katsayılarını oluşturmak için toplamları bir olacak şekilde normalize edilmiştir. Bu son elemanları, katışım katsayılarını, ve de Denklem 1'deki doğrusal katışım modelini kullanarak 1500 adet veri üretilmiştir. Üretilen veriler şekil 1'de siyah noktalar olarak gösterilmiştir.

Aynı parametreler kullanılarak SPICE ve KİDDO algoritmaları karşılaştırılmıştır. Şekil 1'de SPICE'in bulunduğu son elemanlara karşılık KİDDO'nun sonuçları verilmiştir. İlk olarak iki algoritmaya da ortamda $M = 3$ son eleman bulunduğu söylenmiştir ve iki algoritma da aynı son elemanlarla ilklenmiştir. Şekil 1(a)'da verilen sonuçlara göre, simplexin hacmi sınırlandırılmadığında, yani regularizasyon parametresi $\mu = 0$ kullanıldığında, SPICE algoritması olması gereken üçgenin

dışına taşarak fiziksel olarak imkansız olan negatif değerli son elemanlar bulmuştur. Buna karşılık KİDDO son elemanlarını üçgenin köşelerine doğru olarak yerleştirmiştir.

Şekil 1(b)'de yine $\mu = 0$ tutulup her iki algoritma $M = 20$ son elemanla ilklenmiştir. Seyreklikleştirici terim sayesinde her ikisi de bu sayıyı azaltmış, fakat KİDDO $M_{yeni} = 3$ bulurken, SPICE $M_{yeni} = 4$ bulmuştur. Bunda SPICE'in bir son elemanı negatif değerli bulmasının payı yüksektir.

Şekil 1(c)'de $\mu = 0.0001$ 'a yapılarak her iki algoritmaya da simplexin hacmi üzerinde düzenleme uygulanmıştır. Buna göre her ikisi de $M_{yeni} = 3$ son eleman olduğunu bulmuş fakat SPICE iki son elemanın yerini yanlış bulmuştur.

Şekil 1(d)'de $\mu = 0.001$ yapıldığında, SPICE'in bulunduğu hacim daralmış fakat hala sol alt köşede bir negatif son eleman gözlenmiştir. Bu karşılık KİDDO algoritması $[0,0]$ bulmuştur.

Sadece Şekil 1(d)'de $\mu = 0.01$ olarak verildiğinde SPICE ve KİDDO aynı son elemanları bulmuşlardır ve $[0,1]$ aralığında kalmışlardır. Fakat bu da çok fazla regularizasyon uygulanması ve üçgenin küçük olmaya zorlanması sonucunda doğru son elemanları bulamamalarına neden olmuştur.

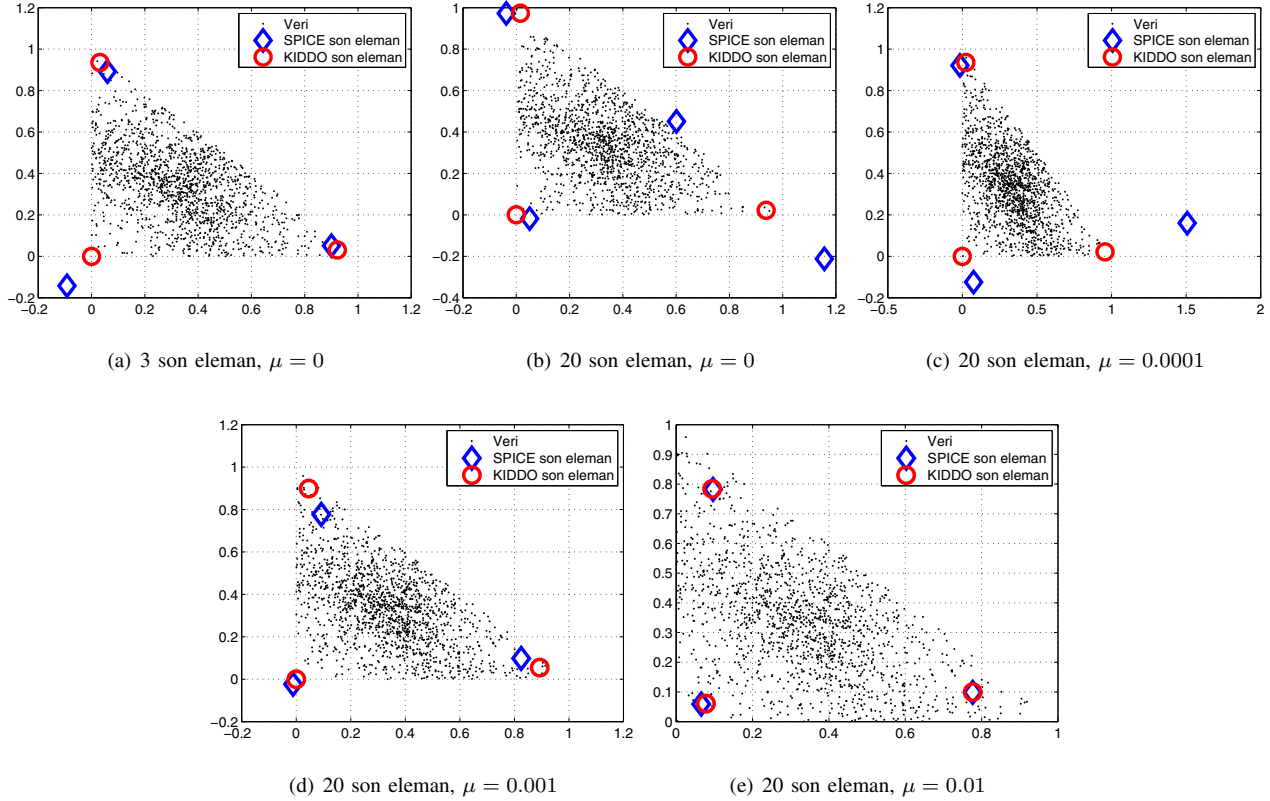
Şekil 1'den görülebileceği üzere SPICE algoritması negatif veya 1'den büyük değerlere ulaşabilmektedir. Bu tür durumları engellemek için regularizasyon katsayısı μ 'nün yükseltilmesi gerekmektedir. Böylece simplex küçülmektedir ve son elemanlar $[0,1]$ içinde kalmaktadır. Fakat bu da daha küçük bir dışbükey zarfa sebebiyet vermekte ve bu yüzden de gerçek son eleman değerlerinin bulunamamasına yol açmaktadır. Buna karşılık, KİDDO algoritması son elemanları her zaman $[0,1]$ aralığında bulmaktadır ve bunun dışındaki fiziksel olarak gerçek olamayacak değerlere izin vermemektedir. Ayrıca, gerektiği takdirde (daha küçük bir dışbükey zarf aranıyorsa) KİDDO algoritması da regularizasyon katsayısından faydalanabilmektedir. Belli bir μ değeri aşıldığı takdirde, hem SPICE hem de KİDDO algoritmaları son elemanlar için benzer sonuçları bulmaktadır.

Buna göre açıdan KİDDO algoritması SPICE'in başarılı olduğu yerlerde aynı başarıyı yakalamış, başarılı olamadığı uç noktalarda ise ona gürbüzlük katmıştır. Bu örnekte KİDDO için deneysel olarak belirlenen μ parametresine ihtiyaç kalmamıştır. Hem son eleman sayısının bulunmasında hem de doğru ve fiziksel olarak gerçekçi son eleman değerlerinin elde edilmesinde başarılı olmuştur. Çalışmalarımız KİDDO'nun gerçek verilere uygulanmasında devam etmektedir.

KAYNAKÇA

- [1] D. Gerçek, D. Çeşmeci, M. K. Güllü, A. Ertürk, and S. Ertürk, "Hiperspektral görüntüleme ve görüntü işlemenin bugünü, yakın geleceği ve mevcut araştırma eğilimleri," *XYZ Dergi*, pp. 18–25, Mart 2013.
- [2] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. Huntington, "Ice: a statistical approach to identifying endmembers in hyperspectral images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 42, no. 10, pp. 2085 – 2095, oct. 2004.
- [3] A. Zare and P. Gader, "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *Geoscience and Remote Sensing Letters, IEEE*, vol. 4, no. 3, pp. 446–450, July 2007.
- [4] —, "Hyperspectral band selection and endmember detection using sparsity promoting priors," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, no. 2, pp. 256–260, April 2008.

$$\begin{aligned}
RSS_{new} &= \frac{(1-\mu)}{N} \sum_{j=1}^d (\mathbf{x}_j - \mathbf{P}\mathbf{e}_j)^T (\mathbf{x}_j - \mathbf{P}\mathbf{e}_j) + \mu \frac{1}{M(M-1)} \sum_{j=1}^d \mathbf{e}_j^T (\mathbf{M}\mathbf{I}_M - \mathbf{1}\mathbf{1}^T) \mathbf{e}_j \\
&\approx \sum_{j=1}^d \left\{ \frac{(1-\mu)}{N} \mathbf{e}_j^T (\mathbf{P}^T \mathbf{P}) \mathbf{e}_j + \frac{(1-\mu)}{N} (-2) \mathbf{e}_j^T \mathbf{P}^T \mathbf{x}_j + \frac{\mu}{M(M-1)} \mathbf{e}_j^T (\mathbf{M}\mathbf{I}_M - \mathbf{1}\mathbf{1}^T) \mathbf{e}_j \right\} \\
&= \frac{(1-\mu)}{N} \sum_{j=1}^d \mathbf{e}_j^T \mathbf{H} \mathbf{e}_j + \mathbf{e}_j^T \mathbf{f}
\end{aligned} \tag{12}$$



Şekil 1. SPICE ve KİDDO algoritmalarının karşılaştırılması. Tüm deneylerde seyreklik terimi $\gamma = 1$ olarak kullanılmıştır. (a) Hem SPICE hem de KİDDO 3 son elemanla ilklenmiştir. Regularizasyon parametresi $\mu = 0$ kullanılarak göz ardı edilmiştir. Böylece bulunabilecek en geniş üçgen bulunmuştur. Bu durumda SPICE algoritması olması gereken üçgenin dışında taşarak fiziksel olarak imkansız olan negatif değerli son elemanlar bulunmuştur. KİDDO algoritması ise son elemanları doğru olarak bulmuştur. (b) Hem SPICE hem de KİDDO algoritmaları 20 son elemanla ilklenmiş ve regularizasyon parametresi yine $\mu = 0$ şeklinde seçilmiştir. Bu durumda seyreklik terimi SPICE algoritmasında 20 elemanı 4'e indirmiş ve bunlardan ikisi negatif koordinatlı bulunmuştur. Bu karşılık KİDDO hem doğru son eleman sayısı olan 3'ü bulmuş, hem de bu son elemanlar $[0,1]$ aralığında kalmışlardır. (c) Regularizasyon parametresi $\mu = 0.0001$ 'a artırıldığında SPICE'in bulunduğu üçgen küçülmüş ama sonuçları çok iyileştirmemiştir. Negatif son elemanlar hala gözlenebilmektedir. (d) Benzer şekilde regularizasyon parametresi $\mu = 0.001$ yapıldığında dahi negatif son elemanlar gözlenmiş ve en sol alt son elemanın değeri $[-0.0116, -0.0237]$ olarak bulunmuştur. Bu karşılık KİDDO algoritması $[0,0]$ bulmuştur. (e) Sadece $\mu = 0.01$ olarak verildiğinde SPICE ve KİDDO aynı son elemanları bulmuşlardır ve $[0,1]$ aralığında kalmışlardır. Fakat bu da çok fazla regularizasyon uygulanması ve üçgenin küçük olmaya zorlanması sonucunda doğru son elemanları bulamamalarına neden olmuştur.

[5] A. Zare, P. Gader, J. Bolton, S. Yuksel, T. Dubroca, R. Close, and R. Hummel, "Sub-pixel target spectra estimation and detection using functions of multiple instances," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on*, June 2011, pp. 1–4.

[6] C. Ratto, K. Morton, L. Collins, and P. Torriani, "A comparison of principal components and endmember-based contextual learning for hyperspectral anomaly classification," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on*, June 2011, pp. 1–4.

[7] Y. Rezaei, M. Mobasheri, M. Zojey, and M. Schaepman, "Endmember

extraction using a combination of orthogonal projection and genetic algorithm," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 2, pp. 161–165, March 2012.

[8] A. Zare and P. Gader, "Spice: a sparsity promoting iterated constrained endmember extraction algorithm with applications to landmine detection from hyperspectral imagery," pp. 655–659, 2007.

[9] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.