

## TRACKING OBJEK BERGERAK DENGAN ALGORITMA K-MEANS

Dian Rosdiana  
Dosen Teknik Mekanik Industri dan Desain  
Politeknik TEDC Bandung  
dianrosdiana72@gmail.com

### ABSTRAK

Saat ini aplikasi *motion tracking* digunakan secara luas untuk banyak tujuan, seperti menyortir barang yang baik atau tidak dalam sistem otomasi masalah *traffic* kemacetan, menghitung berapa banyak orang yang masuk ke sebuah supermarket atau sebuah mall atau mungkin untuk tujuan *security sistem*. Sebuah metode untuk memisahkan antara background dan obyek yang dideteksi dibutuhkan untuk melakukan *motion tracking*. Penelitian ini dibuat untuk memisahkan background dengan menggunakan metode *K-Means*. Secara detail teknik ini menggunakan ukuran ketidakmiripan untuk mengelompokkan obyek. Penerapan *K-Means*, median filtering dan operasi morfologi terbukti efektif dalam meningkatkan akurasi deteksi objek dengan *precision rate* dan *recall rate* dari metode yang diusulkan untuk video 1 adalah 92.31 % dan 90%.

*Keywords: clustering , objek bergerak, metode k-means,, pengolahan citra.*

### ABSTRACT

*Current motion tracking application used widely for many purposes, such as sorting the good stuff or not in the automation system traffic congestion problems, calculate how many people who go into a supermarket or a mall or perhaps for the purpose of security systems. A method for separating between the background and the object detected it takes to do the motion tracking. This study was made to separate the background by using the method of K-Means. In detail, this technique uses the dissimilarity measures to categorize objects. Application of K-Means, median filtering and morphological operations proved to be effective in improving the accuracy of object detection with precision rate and recall rate to video 1 is 92.31 % and 90 % .*

*Key words: image processing, classification, k-mean, motion tracking.*

### PENDAHULUAN

Dewasa ini aplikasi *motion tracking* semakin banyak dipakai untuk berbagai tujuan, diantaranya dalam sistem otomasi masalah menyortir barang yang baik atau tidak, masalah mendeteksi kemacetan lalu lintas, menghitung berapa banyak orang yang masuk ke sebuah supermarket atau mungkin untuk tujuan *security sistem*. Pada *motion tracking* diperlukan suatu metode untuk memisahkan antara *background* dengan obyek yang di-tracking.

Membuat aplikasi tracking pada background yang statis bukanlah hasil yang sulit, namun apabila tracking dilakukan pada background yang tidak statis akan sulit, dikarenakan perubahan background dapat dikenali sebagai area *tracking*. Apalagi ditambah dengan permasalahan ada bayangan pada objek tersebut. Setelah itu adalah mengenali antara objek yang satu dengan objek yang lain. Oleh karena itu dibutuhkan suatu *background* yang dapat beradaptasi dengan perubahan yang terjadi. Metode *adaptive background* yang umum adalah merata-rata gabungan nilai piksel dari semua input yang berurutan. Namun metode ini tidak dapat berjalan dengan baik jika diberi input yang memiliki banyak obyek bergerak. Pada penelitian

ini metode yang digunakan untuk *background subtraction* adalah *K-Means*. Dengan menggunakan parameter-parameter yang telah disediakan, dapat mengklasifikasikan pixel-pixel sesuai dengan model yang diharapkan.

Perumusan masalah yang dihadapi adalah bagaimana menghasilkan suatu *adaptive background* yang stabil terhadap perubahan cahaya dan perubahan *scene* jangka panjang dan bagaimana pengolahan input yang berasal dari kamera video. Metode *clustering* yang digunakan pada kasus ini adalah *K-means* karena metode ini memiliki kecepatan proses yang lebih cepat dan hasil yang cukup baik.

### METODE PENELITIAN

Pada Penelitian kali ini menggunakan metode penelitian eksperimen. Penelitian eksperimen melibatkan penyelidikan perlakuan pada atribut parameter atau variable tergantung dari penelitiannya dan menggunakan tes yang dikendalikan oleh si peneliti itu sendiri dengan metode penelitian sebagai berikut:

1. Pengumpulan data (*Data Gathering*) Tahap ini dilakukan sebagai langkah awal dari suatu penelitian. Mencari data yang tersedia serta memperoleh data tambahan yang dibutuhkan,

kemudian mengintegrasikan data tersebut di dalam dataset. Data berupa video yang diperoleh dari pihak lain yang biasa disebut data sekunder.

2. Pengolahan awal data (*Data Pre-processing*)  
Tahap ini merupakan tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data sebelum dilanjutkan ke tahap selanjutnya. Data video diekstrak menjadi image. Ditahap ini dilakukan penyeleksian citra, pengelompokkan citra untuk pelatihan dan pengujian.
3. Metode yang diusulkan (*Proposed Method*)  
Tahap ini dijelaskan tentang metode yang diusulkan untuk digunakan dalam deteksi objek pada pelacakan video. Penjelasan meliputi pengaturan dan pemilihan nilai dari parameter-parameter dan arsitektur melalui uji coba.
4. Eksperimen dan pengujian model (*Model Testing and Experimen*)  
Tahap ini akan menjelaskan tahapan eksperimen penelitian dan teknik pengujian yang akan digunakan.
5. Evaluasi dan validasi hasil (*Result Evaluation*)  
Pada tahap ini dilakukan evaluasi terhadap model yang ditetapkan untuk mengetahui tingkat performa model.

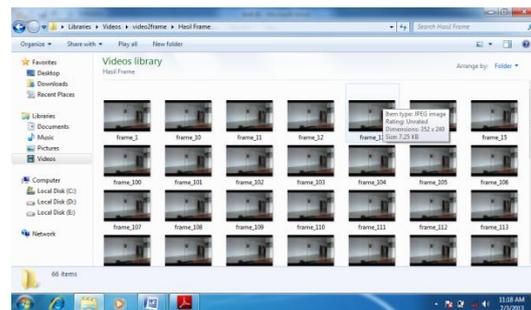
### Pengumpulan Data

Pada penelitian ini menggunakan data sekunder. Data berupa video yang diekstrak menjadi citra. Data video tersebut merupakan objek orang berjalan pada suatu ruangan yang berdurasi 1 menit 44 detik dengan dimensi frame 352 X 240, frame rate 29/detik dan format MPG.



Gambar 1: Citra objek didalam sebuah ruangan  
Konversi video menjadi citra menggunakan *video2pic.m* yang dibuat dari program matlab.

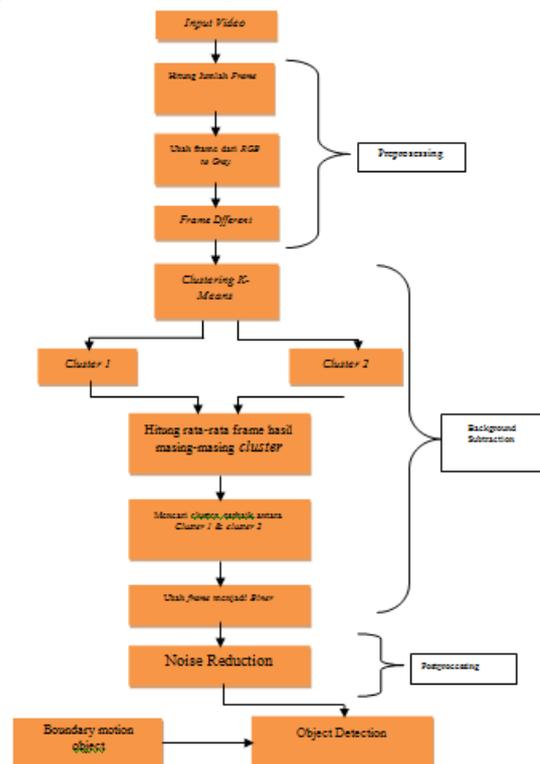
Data pertama (video objek didalam ruangan) diekstrak menjadi 720 buah citra disimpan dengan format JPG. Dari hasil ekstraksi data sekunder akan diambil 40 citra berurutan yang akan digunakan sebagai sampel pada eksperimen yang akan dilakukan.



Gambar 2: Citra Hasil Ekstraksi video

### PEMBAHASAN

Metode yang diusulkan untuk penelitian ini menerapkan algoritma *K-Means* untuk tracking objek yang bergerak dengan mengelompokkan pixel pada citra termasuk sebagai *background* atau *foreground*. Ditambahkan operasi *morfologi* dan *median filtering* dalam rangka meningkatkan *performa* dari model yang diajukan. Berdasarkan Gambar 1, maka metode yang diusulkan pada penelitian ini dapat dijelaskan antara lain :



Gambar 3: Proposed Method

#### 1. Input image frame

Data *image*/citra yang telah disiapkan dalam pengolahan data awal. Di-*load* oleh model. Data yang digunakan adalah data pada folder eksperimen.

#### 2. Preprocessing

Image frame hasil ekstraksi mempunyai ruang warna RGB. Setiap image frame akan masuk proses inialisasi. Sebelum proses inialisasi dilakukan, tentukan dulu image yang menjadi model background. Image *frame* pertama akan digunakan sebagai model *background* awal. Tahap preprocessing ini dilakukan sebagai persiapan dalam pendeteksian *foreground* menggunakan *Background Subtraction*.

### 3. Background Subtraction

Pada tahap *background Subtraction* berisi beberapa proses. *Frame* akan dicluster dengan *K-mean* dengan 2 buah *cluster*. Ini dilakukan untuk mencari cluster yang terbaik. Selanjutnya hasil *cluster* tersebut akan diambil yang terbaik. Hasil *cluster* akan dirubah dalam bentuk biner. Nilai 0 adalah *Background* dan 1 adalah *foreground*.

### 4. Postprocessing

Tahapan ini diaplikasikan teknik *Morfologi* sebagai *filtering* untuk memperbaiki hasil segmentasi dari *Background Subtraction*. [4] mengungkapkan bahwa hasil dari *Background Subtraction* dapat ditingkat dengan melakukan *post-processing*. Teknik yang dapat digunakan dalam *post-processing* seperti *Morfologi (Closing, Opening, Dilation, Erosion)*, *Median filter*, atau teknik peningkatan citra lain.

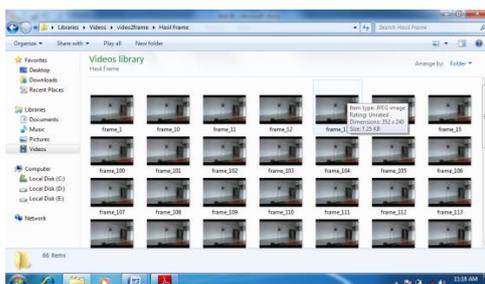
### 5. Object Detection and Marking

Tahap terakhir adalah deteksi objek dan penandaan objek (*foreground*). diaplikasikan algoritma *connected component* yang dilanjutkan dengan teknik *Bounding Box* yang akan memberi batas objek yang terdeteksi pada citra. Eksperimen menggunakan tool matlab R2009a. Dengan Hardware Processor Atom dan Ram 2 GB.

## Hasil Eksperimen dan Pengujian Model

Pengumpulan data, pengolahan awal citra, ekstraksi fitur dan pengujian model. Eksperimen menggunakan tool matlab R2009a. Dengan Hardware Processor Atom dan Ram 2 GB

### 1. Pengumpulan Data



Gambar 4: Citra Hasil Ekstraksi video

## 2. Pengolahan Awal Citra

Langkah pengolahan awal citra yang telah dilakukan yaitu hitung *Frame*, Simpan *frame*, ubah frame RGB menjadi *Frame Gray*.

### a) Hitung Frame

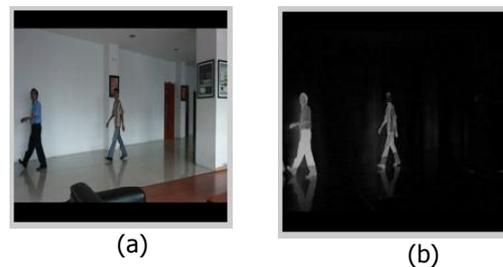
Setelah Video diinput maka video tersebut dihitung berapa jumlah framenya. Ini dimaksudkan untuk memberikan pelabelan/penomoran pada *frame*.

### b) Simpan frame

Setelah *frame* dihitung dan diberi pelabelan maka frame-frame tersebut disimpan.

### c) Ubah warna *frame* dari RGB menjadi Gray

Setelah Video menjadi frame dan maka frame-frame tersebut warnanya dari RGB diubah menjadi *Gray*. Ini untuk memudahkan dalam membedakan antara *frame*. Dengan frame menjadi *Gray* maka akan terdapat dua buah warna yaitu hitam dan putih. Seperti terlihat pada gambar dibawah ini :



Gambar 5: Merubah frame RGB to Gray (a) Frame RGB (b) Frame Gray

### d) *Frame Different*

Setelah frame menjadi gray maka frame akan dibedakan antara frame background dengan frame yang telah ada objeknya. Sebelumnya kita telah menentukan dulu frame backgroundnya dengan Code :

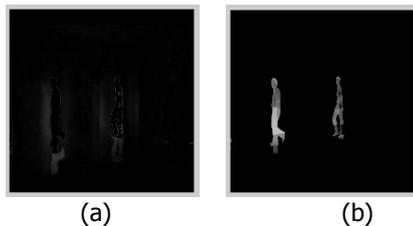


Gambar 6: (a) *Frame* sebagai *Background*, (b) *Frame* yang terdapat objeknya

### 3. Clustering Objek/ Background Subtraction

#### a) Clustering Objek dengan K-Means

Tahap selanjutnya adalah *clustering* dengan *K-Means*. Pada *clustering* akan dicluster antara objek dengan background. Disini akan *di-cluster* sebanyak 2 kali. Ini dimaksudkan untuk mencari yang terbaik antara hasil *cluster* tersebut.



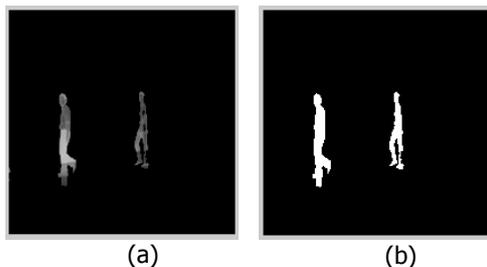
Gambar 8: (a) Frame Hasil Cluster ke-1, (b) Frame Hasil Cluster ke-2 objeknya,

#### b) Memilih *cluster* yang terbaik

Setelah dicluster 2 kali maka hasil masing-masing *cluster* tersebut akan dibandingkan dengan mencari hasil *cluster* yang terbaik.

#### c) Ubah *frame* menjadi *Biner*

Setelah mendapatkan hasil *cluster* yang terbaik maka tahapan selanjutnya *frame* tersebut diubah menjadi *biner* sehingga akan terlihat perbedaan background akan berwarna hitam dan *foreground/objek* akan berwarna putih.



Gambar 9: (a) Frame Hasil *Cluster* yang terbaik, (b) Frame Hasil diubah ke *Biner*.

### 4. Noise Reduction

Setelah diubah kedalam *citra biner*, dilakukan teknik *morphology* untuk penebalan. Fungsinya untuk mengantisipasi hasil *biner* yang kurang baik.

#### PENGUJIAN

Untuk mengetahui hasil akurasi dari deteksi objek dapat diukur menggunakan *precision rate* dan *recall rate* [1] [22]. *precision rate* adalah rasio jumlah *frame* objek yang benar terdeteksi

sebagai sebagai objek (*true positive*) dengan *frame* objek yang benar terdeteksi sebagai sebagai objek (*true positive*) dan jumlah *frame* objek yang tidak benar terdeteksi sebagai sebagai objek (*false positive*). *recall rate* adalah rasio *frame* objek yang benar terdeteksi sebagai sebagai objek (*true positive*) dengan jumlah *frame* objek yang benar terdeteksi sebagai *plat nomor* (*true positive*) dan jumlah *frame* objek yang tidak terdeteksi sebagai objek (*false negative*). Adapun rumus untuk menghitung *precision rate* dan *recall rate* adalah sebagai berikut :

$$\text{Precision rate} = \frac{TP}{TP + FN}$$

$$\text{Recall rate} = \frac{FP}{TP + FP}$$

Dimana :

- TP (*true positive*) = sering juga disebut *hit rate*, menunjukkan jumlah *frame* objek yang benar terdeteksi sebagai objek.
- FN (*false negative*) = jumlah *frame* objek yang tidak terdeteksi sebagai objek
- FP (*false positive*) = Jumlah *frame* bukan objek yang terdeteksi sebagai objek.

Pada gambar 7 ditunjukkan hasil deteksi dari algoritma *K-Means* terhadap objek yang bergerak dengan berbagai *frame* dan pada gambar ditunjukkan langkah per langkah hasil deteksi metode yang diusulkan, yaitu integrasi algoritma *K-Means* dan *metoda morphology*.

	Frame 1	Frame 220	Frame 232	Frame 238	Frame 600
Video sumber					
RGB to Gray					
Cluster 1					
Cluster 2					
Cluster Terbaik					
Frame Biner					
Hasil cluster & background					
	(a) Frame 1	(b) Frame 220	(c) Frame 232	(d) Frame 238	(e) Frame 600

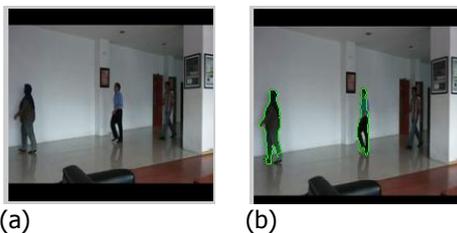
Gambar 10: langkah per langkah hasil deteksi metode yang diusulkan, yaitu integrasi algoritma *K-Means* dan *metoda morphology*.

Pada gambar 11 ditunjukkan jumlah *frame* objek yang benar terdeteksi sebagai objek (*true positive*).



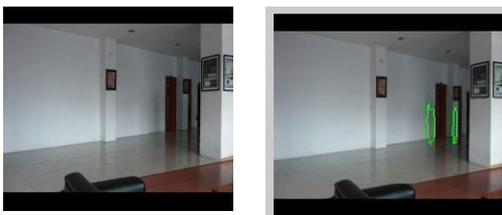
Gambar 11: *Frame* objek yang benar terdeteksi sebagai objek (*true positive*). (a) *Frame* input, (b) *Frame* dengan objek yang di-tracking.

Dan pada gambar 11 ditunjukkan jumlah *frame* objek yang tidak terdeteksi sebagai objek (*false negative*).



Gambar 12: *Frame* objek yang tidak terdeteksi sebagai objek (*false negative*). (a) *Frame* input, (b) *Frame* Output.

Pada gambar 13 ditunjukkan jumlah *frame* bukan objek yang terdeteksi objek (*false positive*).



Gambar 13: *Frame* bukan objek yang terdeteksi objek (*false positive*).

	Frame 211	Frame 459	Frame 561
Video			
RGB to Gray			
Cluster 1			
Cluster 2			
Cluster			
Frame Output			
Objek			
Cluster 2			
	(A) Frame 211	(B) Frame 459	(C) Frame 561

Gambar 14 : *Frame* objek yang tidak terdeteksi sebagai objek (*false negative*) maupun *Frame* bukan objek yang terdeteksi objek (*false positive*).

Sekarang kita akan menghitung *Precision rate* dan *Recall rate* berdasarkan hasil pengujian deteksi pada data uji pertama dan data kedua dengan kita akan mengomparasi dan membandingkan *Precision Rate* dan *Recall Rate* masing-masing maka didapat :

Tabel 1 : *precision rate* dan *recall rate*

Hits (Tp)	Missed (FN)	False (FP)	precision rate	recall rate
36	4	3	92.31 %	90 %

Keterangan:

Hits = *true positive*

Missed = *false negative*

False = *false positive*

Tabel 2 : *precision rate* dan *recall rate* untuk video 1 dan video 2

Context	Video 1
<i>precision rate</i>	92.31 %
<i>recall rate</i>	90 %

Dari hasil algoritma *K-Means* dan metode *morphology* berdasarkan dari parameter *precision rate* dan *recall rate* menunjukkan bahwa metode yang diusulkan, yaitu integrasi algoritma *K-Means* dan metode *morphology* memiliki kinerja deteksi yang lebih akurat. Nilai *precision rate* dan *recall rate* dari metode yang diusulkan untuk video 1 adalah 92.31 % dan 92,31%.

## KESIMPULAN DAN SARAN

### 1. Kesimpulan

Background Subtraction terbukti efektif dalam mendeteksi objek bergerak dari sekuen citra. Permasalahan umum dari *background Subtraction* seperti perubahan pencahayaan, bayangan, dan background bergerak juga dapat diatasi. Penerapan algoritma *K-Means* dan tahap *post-processing* yang diisi dengan *median filtering* dan operasi *morfologi* dapat membantu meningkatkan akurasi dari segmentasi objek bergerak. Dalam uji coba yang dilakukan pada dua dataset metode *Tracking Objek bergerak* dengan algoritma *K-Means* terbukti menghasilkan akurasi segmentasi citra yang lebih baik. Dari hasil algoritma *K-Means* dan metode *morphology* berdasarkan dari parameter *precision rate* dan *recall rate* menunjukkan bahwa metode yang diusulkan, yaitu integrasi algoritma *K-Means* dan metode *morphology* memiliki kinerja deteksi yang akurat. Nilai *Precision rate* dan *Recall rate* dari metode yang diusulkan untuk pada video tersebut adalah 92.31 % dan 92,31%. Dengan demikian terbukti bahwa penerapan algoritma *K-Means* dan *post-processing* yang berisi *median filtering* dan operasi *morfologi* berpengaruh dalam peningkatan akurasi deteksi objek yang dalam penelitian ini diukur dengan kualitas segmentasi citra yang dihasilkan

### 2. Saran

*Tracking Objek bergerak* dengan algoritma *K-Means* dapat mendeteksi objek bergerak dari dataset yang digunakan berupa objek orang bergerak didalam ruangan. Metode algoritma *K-Means* memang terbukti akurat untuk objek yang bergerak. Namun, di penelitian yang akan datang, ada beberapa pekerjaan yang perlu dilakukan, yaitu:

1. Beberapa gambar input yang gelap atau kurang terang tidak terdeteksi sebagai objek. Ini dikarenakan kurangnya penerangan pada ruangan tersebut

2. Menambah jumlah video yang lebih bervariasi baik didalam ruangan maupun diluar ruangan.

Jadi alangkah baiknya apabila dataset berupa video lebih dari satu video. Mialnya video yang diambil pada luar ruangan Namun diperlukan pengaturan parameter (threshold dan learning rate) yang tepat dalam menghasilkan deteksi yang akurat. Dalam penelitian selanjutnya dapat ditambahkan metode penentuan *threshold* atau sering disebut *threshold adaptif* atau metode clustering lainnya.

## DAFTAR PUSTAKA

- [1] Shih-Chieh Wang<sup>1</sup>, Te-Feng Su<sup>2</sup>, and Shang-Hong Lai<sup>1</sup>, "Detecting moving objects from dynamic background with shadow removal", IEEE, 978-1-4244-7874-3, 2010
- [2] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," CVPR, Vol. 2, pp. 246-252, 1999.
- [3] N. Paragios and V. Ramesh, "A MRF-based approach for realtime subway monitoring," CVPR, vol. 1, pp. 1034-1040, 2001
- [4] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models," IEEE Trans. PAMI, Vol.26, No.5, pp. 651-656, 2004.
- [5] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," ECCV, 2000.
- [6] T. Matsuyama, T. Ohya, H. Habe, "Background subtraction for non-stationary scenes," ACCV, pp. 662-667, 2000.
- [7] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," IEEE Trans. PAMI, Vol. 28, pp. 657-662, 2006.
- [8] L. Li, W. Huang, I.-Y.H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," ACM Intern. Conf. Multimedia, 2003
- [9] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," IEEE Trans. PAMI, Vol. 27, pp. 1778- 1792, 2005
- [10] S. Zhang, H. Yao, and S. Liu, "Spatial-temporal nonparametric background subtraction in dynamic scenes," ICME, 2009
- [11] L. Cheng and M. Gong, "Real-time background subtraction from dynamic scenes," ICCV, 2009
- [12] S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, "Region-level motionbased background modeling and subtraction using mrfs," IEEE Trans. Image Processing, vol. 16, no. 5, pp. 1446 -1456, 2007.