

ADAPTIVE THRESHOLD UNTUK ALPHA MATTING MENGUNAKAN ALGORITMA OTSU

R. Suko Basuki¹, Moch. Hariadi²

¹Teknik Elektro Institut Teknologi Sepuluh November Surabaya, Indonesia
E-mail : ruri.basuki10@mhs.ee.its.ac.id

² Teknik Elektro Institut Teknologi Sepuluh November Surabaya, Indonesia
E-mail : mochar@ee.its.ac.id

ABSTRAK

Image matting merupakan proses ekstraksi objek foreground dari keseluruhan image. Hal ini memainkan peranan penting dalam proses image editing. Dalam paper ini algoritma Otsu digunakan untuk menghasilkan nilai threshold yang selanjutnya diberikan sebagai nilai alpha dalam "pulling matte". Hasil objek foreground yang dipisahkan selanjutnya diukur kualitasnya dengan menggunakan MSE (Mean Squared Error). Proses pengukuran dilakukan dengan mencari perbedaan diantara objek foreground yang terdapat pada image masukan dengan objek foreground hasil matting.

Kata kunci : Adaptive Threshold, Alpha Matting, Otsu

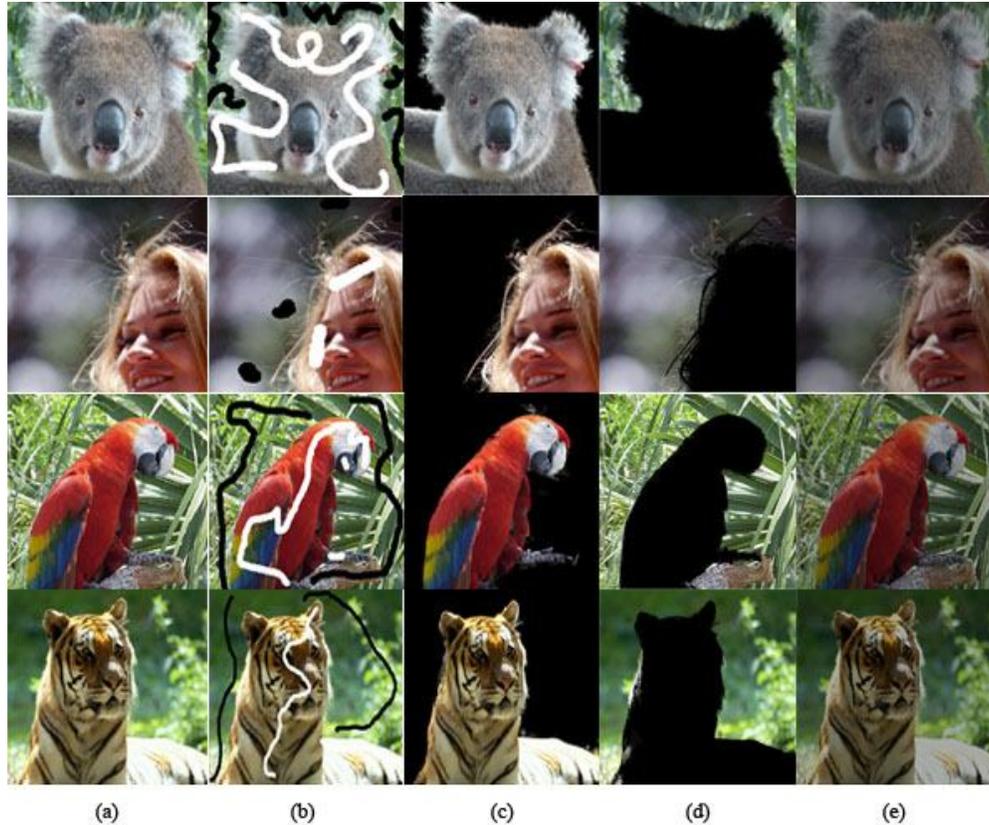
1. PENDAHULUAN

Ekstraksi objek *foreground* dari *image* secara keseluruhan memainkan peranan yang penting dalam proses *image editing*. Akurasi pemisahan objek *foreground* dari *background* ditentukan oleh sebagian atau keseluruhan dari *pixel-pixel* suatu *image*, proses ini disebut "pulling matte" atau "digital matting". Metode *digital matting* diformulasikan sebagai *image* masukan (I) yang diasumsikan merupakan kombinasi dari *image foreground* (F) dan *image background* (B). Warna *pixel* ke- i , diasumsikan sebagai kombinasi linear yang saling terkait diantara warna *foreground* dan *background* [1 2], dimana α - i merupakan komponen *opacity pixel* yang digunakan untuk memadukan secara linear *image foreground* dan *background* seperti dalam persamaan 1.

$$= \quad + (1 - \quad)$$

Beberapa metode yang digunakan dalam beberapa tahun terakhir ini untuk *digital matting*, "trimap" digunakan sebagai langkah awal untuk proses ekstraksi. *Trimap* merupakan gambaran kasar (dalam bentuk gambar tangan) untuk *image input* yang terbagi dalam tiga area; *foreground* (digambar dengan warna putih), *background* (digambar dengan warna hitam) dan *unknown region* (digambar dengan warna abu-abu). Optimasi *iterative nonlinear* biasanya digunakan untuk *foreground* dan *background* yang nilainya dihitung secara bersamaan dengan α . Untuk menghasilkan nilai yang baik pada *unknown region*, *trimap* harus dibuat sekecil mungkin. Namun, kelemahan dari pendekatan ini adalah kesulitan menangani *image* yang memiliki *pixel* yang kompleks atau *foreground* memiliki banyak lubang [2 7].

Untuk melakukan ekstraksi *alpha matte* dari *image* yang natural digunakan "close-form" yang berkaitan erat dengan metode pewarnaan [3 6]. *Cost function* diperoleh dari asumsi *local smoothness* dalam *foreground* dan *background* yang paling mungkin untuk dieliminasi sehingga menghasilkan *quadratic cost function* pada a . *Global optimum* digunakan pada *cost function* untuk menghasilkan *alpha matte* yang dapat diperoleh dengan menyelesaikan sistem *sparse linear*. Perbedaan *foreground* dan *background* (seperti pada gambar 1.b), digunakan *scribble* (putih untuk *foreground* dan hitam untuk *background*) untuk menghitung nilai *alpha* dari metode *close-form*. Selanjutnya *eigenvectors* akan menguji *sparse matrix* yang memiliki hubungan erat dengan matrik yang digunakan pada algoritma segmentasi *image* spektral. Dalam pendekatan ini, penempatan *image scribble* menjadi petunjuk penting untuk menganalisis *image*. Namun, nilai *threshold* dalam pendekatan ini masih menggunakan nilai konstanta ($t=0,3$), yang menyebabkan pemisahan antara *foreground* dengan *background* kurang alami. Sebagai tindak lanjut dan tambahan dari metode tersebut, dalam paper ini diusulkan algoritma *Otsu Threshold* digunakan untuk mencari nilai ambang yang adaptif untuk diberikan sebagai nilai *alpha*.



Gambar. 1 (a). *Image input*, (b). *Image* dengan coretan: coretan putih menunjukkan *foreground*, hitam menunjukkan *background*, (c, d). Hasil matting, (e). *Image* kombinasi setelah proses matting

2. FUNDAMENTAL

Metode yang ada untuk *image matting* memiliki tujuan penyelesaian *compositing* (persamaan 1) untuk *unknown pixel*. Beberapa metode *image matting* menggunakan trimap [4 8], [5 11], [6 12], [7 15], [8 16] sebagai pendamping *image input* untuk melabelkan *foreground*, *background* dan *unknown pixel*. Biasanya metode ini dilakukan dengan memanfaatkan beberapa asumsi “*local regularity*” untuk menghitung nilai masing-masing *pixel* pada *unknown area*. Dalam algoritma Knockout [Bremen dkk, 2000], setelah dilakukan *segmentasi user*, langkah selanjutnya menghitung kemungkinan nilai warna *foreground* dan *background* yang masuk ke dalam *unknown area*. *Foreground* dihitung dari batas tepi *pixel* pada area *known foreground*. Bobot *pixel* terdekat pada *unknown area* diberi nilai 1 dan selanjutnya menurun secara proporsional terhadap jarak hingga mencapai nilai 0. Prosedur yang sama juga digunakan untuk menghitung bobot *background* berdasarkan *pixel* terdekat pada area *unknown background*. Asumsi beberapa algoritma [7], [8] menyatakan bahwa *foreground* dan *background* berasal dari distribusi warna relatif yang sederhana. Pada algoritma *Bayesian matting* [8] faktor keberhasilan pada algoritma Knockout ditentukan oleh percampuran yang berorientasi pada algoritma Gaussian yang digunakan untuk mempelajari distribusi lokal yang selanjutnya α , *foreground* dan *background* dihitung sebagai distribusi yang paling mungkin.

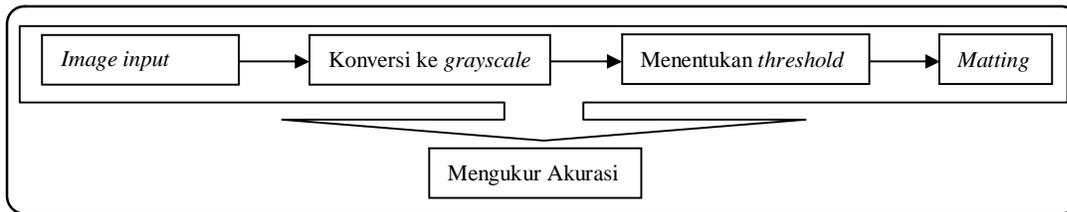
Dalam pendekatan sebelumnya, metode *Poisson matting* dilakukan dengan mengoptimalkan warna *pixel alpha*, *foreground* dan *background* secara statistik. Untuk mengurangi kesalahan yang diakibatkan oleh kesalahan klasifikasi pada kasus warna yang kompleks [5], operasi matte dilakukan secara langsung pada gradient. Perubahan intensitas yang halus dari *foreground* ke *background* merupakan dasar dari formulasi *Poisson matting*. Sun dkk [5], menggunakan *Global Poisson matting* sebagai pendekatan semi otomatis untuk menghitung *matte* dari *gradient image* yang diberikan oleh *trimap*. Perhitungan yang *robust* terhadap *foreground* dan *background* telah dilakukan, namun terjadi kesalahan *matting* yang disebabkan oleh *background* yang kompleks tidak dapat diselesaikan. Untuk mengatasi permasalahan tersebut digunakan *Local Poisson matting* yang digunakan untuk memanipulasi bidang gradien kontinu pada *local region*. Beberapa pendekatan telah berhasil dilakukan untuk menterjemahkan “*user-defined simple constraints*” (seperti *scribble* dan *rectangle*) untuk permasalahan “*min-cut*”. Penyelesaiannya menggunakan segmentasi biner yang selanjutnya ditransformasikan ke dalam *trimap* dengan erosi, namun hasilnya masih samar. *Border matting* [9 10] menggunakan

parametric alpha yang diambil dari strip yang sempit disekitar batas, namun tidak dapat dilakukan untuk kasus yang sama pada objek rambut, karena area yang samar begitu luas, sehingga tidak dapat dilakukan dengan teknik ini. Metode yang diusulkan [3] dan [10] menggunakan *scribble* sebagai batas untuk mengurangi *quadratic cost function* sehingga lebih cocok untuk permasalahan *matting*.

Wang dan Cohen [11 4] mengusulkan “*scribble-based*” untuk *matting* yang interaktif. *Scribble* digunakan sebagai parameter *pixel foreground* dan *pixel background*. Pendekatan ini menghasilkan beberapa hasil yang impresif, namun memerlukan proses yang mahal. Guan dkk [11], mengusulkan pendekatan “*scribble-based*” yang lain dengan menambahkan pendekatan “*random-walk*” [5] dengan menghitung secara iteratif model warna.

3. FRAMEWORK

Tahapan dalam penelitian ini seperti yang ditunjukkan dalam gambar 2.



Gambar 2. Framework Penelitian

3.1 NILAI AMBANG OTSU

Metode Otsu [12] berbasis histogram yang menunjukkan nilai intensitas yang berubah-ubah di setiap *pixel image* satu dimensi. Sumbu x digunakan untuk menyatakan perbedaan level intensitas, sedangkan sumbu y digunakan untuk menyatakan jumlah *pixel* yang memiliki nilai intensitas. Dengan menggunakan histogram dapat dilakukan pengelompokan *pixel image* berdasarkan nilai *threshold* (ambang batas). *Threshold* yang optimal dapat diperoleh ketika *pixel* memiliki perbedaan intensitas sehingga dapat dipisahkan kelompok-kelompoknya. Dua informasi dapat diperoleh dengan memanfaatkan histogram, yaitu jumlah perbedaan tingkat intensitas (dinotasikan dengan L), dan jumlah *pixel* untuk setiap tingkat intensitas (dinotasikan dengan $n(k)$, dengan $k=0 \dots 255$). Tahapan pencarian nilai *threshold* dalam algoritma Otsu adalah sebagai berikut :

1. Menghitung normalisasi histogram *image* yang dinotasikan dengan h_i , dengan $i = 0,1,2,\dots,L-1$.

$$h_i = \frac{n_i}{MN} \quad (2)$$

dimana n_i adalah jumlah *pixel* pada masing-masing intensitas, dan MN adalah jumlah dari n_i yang dimulai dari $i=0$ hingga $i=L-1$.

2. Menghitung jumlah komulatif dari $H(k)$, untuk $k=0,1,2 \dots L-1$.

$$H(k) = \sum_{i=0}^k h_i \quad (3)$$

3. Menghitung rata-rata komulatif $G(k)$, untuk $k=0,1,2 \dots, L-1$.

$$G(k) = \frac{\sum_{i=0}^k i h_i}{\sum_{i=0}^k h_i} \quad (4)$$

4. Menghitung rata-rata intensitas global menggunakan ;

=

(5)

5. Menghitung varian antar kelas, (σ^2) , untuk $k=0,1,2 \dots, L-1$.

$$= \frac{[\sum (x_k - \bar{x})^2]}{(n_k)[1 - \sum (x_k)]}$$

(6)

6. Memilih nilai *threshold* dari k^* di mana nilai index dari varian antar kelas maksimum ($\rightarrow \max$), jika lebih dari satu nilai dari k^* , maka nilai *threshold* ditentukan dari rata-rata nilai k^* .

7. Menghitung ukuran pemisahan η^* dengan $k=k^*$

$$\eta(k) = \frac{(\sigma^2)}{(\mu - \bar{x})}$$

(7)

sedangkan

$$= (1 - \sigma^2)$$

(8)

Catatan : nilai dari k diperoleh ketika (σ^2) maksimum, selanjutnya nilai *threshold* yang dilakukan dengan metode Otsu diberikan untuk α (dalam persamaan 1).

3.2 MEAN SQUARED ERROR

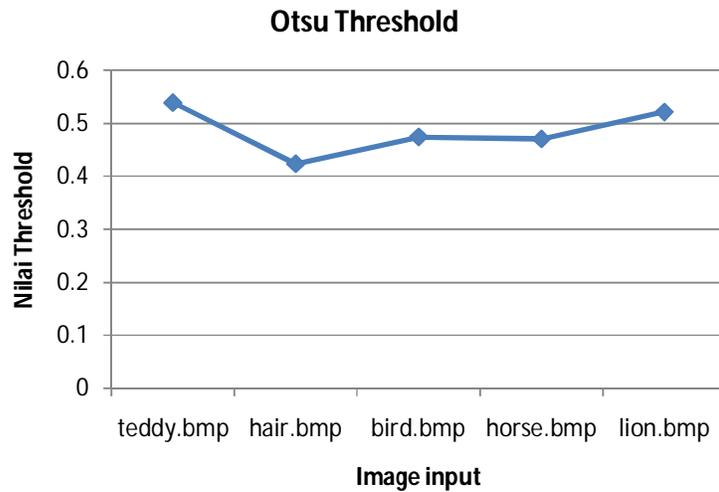
Untuk mengukur kualitas *image* setelah dilakukan proses *matting*, dalam penelitian ini digunakan *MSE (Mean Squared Error)*. *Image* masukan dikomparasi dengan *image* hasil proses *matting* dengan nilai *threshold* yang diperoleh dengan menggunakan algoritma Otsu.

4. EXPERIMENT DAN EVALUASI.

Dalam makalah ini, awalnya pencarian nilai *threshold* dilakukan pada *image input* dengan menggunakan algoritma Otsu (persamaan 2 – 8). Hasil pencarian *threshold* ditunjukkan dalam tabel 1 dan grafik pada gambar 3.

Table 1. Hasil *Threshold*

| <i>Image masukan</i> | <i>Nilai Threshold</i> |
|----------------------|------------------------|
| teddy.bmp | 0,539215686 |
| hair.bmp | 0,423529412 |
| bird.bmp | 0,474509804 |
| horse.bmp | 0,470588235 |
| lion.bmp | 0,521568627 |

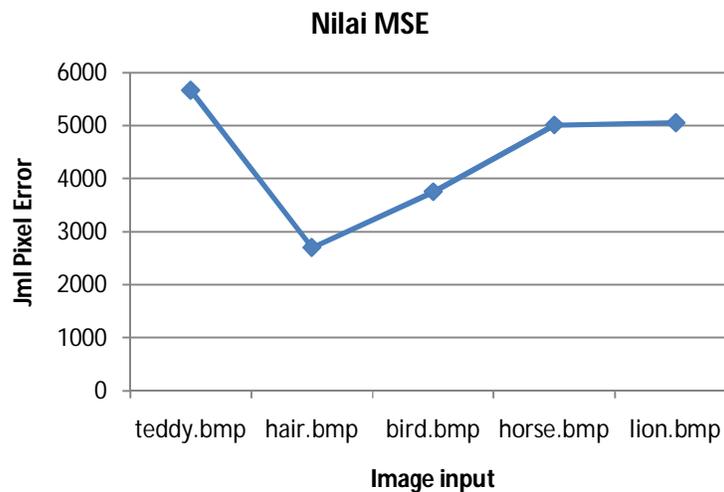


Gambar 3. Nilai *threshold* Otsu

Setelah nilai *threshold* diperoleh, hasilnya diberikan sebagai nilai α untuk keperluan *matting* [1], sehingga objek *foreground* dapat dipisahkan (gambar 1.c). Selanjutnya objek *foreground* yang telah dipisahkan dikombinasikan dengan *image input*. Untuk mengukur kualitas *image* hasil *matting* digunakan MSE dengan membandingkan *image input* (seperti 1.a) dengan *image* hasil kombinasi (gambar 1.e). Hasil pengukuran ditunjukkan dalam tabel 2 dan grafik pada gambar 4.

Table 2. Hasil Pengukuran

| <i>Image</i> masukan | Nilai MSE |
|----------------------|-------------|
| teddy.bmp | 5669,397258 |
| hair.bmp | 2697,183302 |
| bird.bmp | 3751,389982 |
| horse.bmp | 5015,051846 |
| lion.bmp | 5055,08185 |

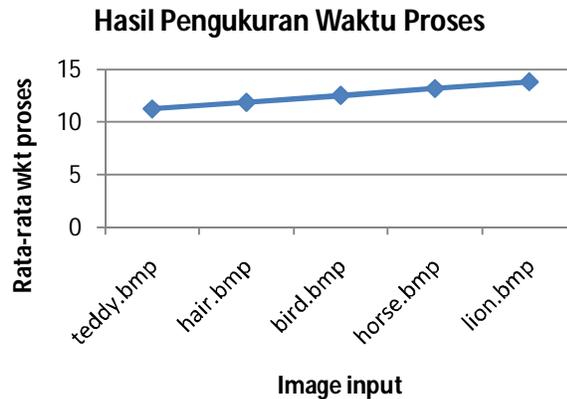


Gambar 4. Nilai pengukuran MSE

Selain pengukuran terhadap akurasi *image*, waktu proses untuk mendapatkan nilai *threshold* juga diukur dengan lima kali iterasi, nilai rata-rata pengukuran ditunjukkan dalam tabel 3 dan grafik dalam gambar 5.

Table 3. Hasil rata-rata waktu proses

| Image Masukan | Rata-rata Waktu Proses |
|---------------|------------------------|
| teddy.bmp | 10,59543472 |
| hair.bmp | 11,24068757 |
| bird.bmp | 11,88594042 |
| horse.bmp | 12,53119328 |
| lion.bmp | 13,17644613 |



5. KESIMPULAN

Dalam makalah ini disajikan *adaptive threshold* untuk *alpha matting* dengan menggunakan algoritma Otsu. Nilai α dalam *alpha matting* yang sebelumnya diberikan nilai konstanta (0,3), dengan *threshold Otsu* dapat dilakukan secara adaptatif terhadap beberapa kasus, meskipun masih memiliki nilai varian yang tinggi dan jumlah *pixel error* yang belum optimal. Oleh karena itu, dalam penelitian selanjutnya perlu dipertimbangkan penggunaan algoritma yang lain seperti *K-Means* atau *Fuzzy C-Means* agar diperoleh nilai *threshold* yang optimal.

DAFTAR PUSTAKA

- [1]. Levin A, Lischinski D, Weiss Y. *A Closed-Form Solution to Natural Image Matting. Analysis*. 2008;30(2):1-15.
- [2]. Wang J, Cohen M. *An Iterative Optimization Approach for Unified Image Segmentation and Matting. Proc. 10th IEEE Int'l Conf. Computer Vision*. 2005.
- [3]. Grady L, Schiwietz T, Aharon S, Westermann R. *Random Walks for Interactive Alpha-Matting. Proc. Fifth IASTED Int'l Conf. Visualization, Imaging, and Image Processing*. 2005.
- [4]. Apostoloff NE, Fitzgibbon AW. *Bayesian Video Matting Using Learnt Image Priors. Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2004.
- [5]. Sun J, Jia J, Tang C, Shum H. *Poisson Matting. Image (Rochester, N.Y.)*. 2004;1(212):315-321.
- [6]. Chuang YY, Agarwala A, Curless B, Salesin DH, Szeliski R. *Video Matting of Complex Scenes. ACM Trans. Graphics*. 2002;vol:21no3pp243-248.
- [7]. Ruzon MA, Tomasi C. *Alpha Estimation in Natural Images. Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2000.
- [8]. Chuang Y, Curless B, Salesin DH, Szeliski R. *A Bayesian Approach to Digital matting. Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2001
- [9]. Rother C, Kolmogorov V, Blake A. *"Grabcut": Interactive Foreground Extraction Using Iterated Graph Cuts. ACM Trans. Graphics*. 2004;vol:23no3pp309-314.
- [10]. Levin A, Lischinski D, Weiss Y. *Colorization Using Optimization. ACM Trans. Graphics*. 2004.
- [11]. Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng, "Easy Matting," *Proc. Ann. Conf. European Assoc. for Computer Graphics*, 2006.
- [12]. Rafael C. Gonzalez., Richard E. Woods. *Digital Image Processing* 3rd ed. Pearson Prentice Hall. 2007