

Aplikasi Penggabungan Objek Transparan Kedalam Sebuah *Image*

Meliana Luwuk¹, Liliana², Djoni H. Setiabudi³

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: meliana1993@hotmail.com¹, lilian@petra.ac.id², djonihs@petra.ac.id³

ABSTRAK: Dewasa ini manipulasi gambar yang berkaitan dengan pengambilan suatu objek (*matting*) dan penggabungan pada *background* baru (*compositing*) banyak dipergunakan (misal dalam dunia perfilman). Objek yang akan diolah dapat beraneka ragam, tetapi permasalahan akan timbul ketika objek yang diolah adalah objek transparan. Belum banyak riset mengenai permasalahan ini karena sulitnya mendapatkan nilai transparansi serta pembiasan objek, sehingga dalam skripsi ini dikembangkan aplikasi untuk melakukan proses *matting* dan *compositing* objek tranparan.

Proses *matting* akan dilakukan dengan menggunakan metode Grabcut, lalu nilai alpha/transparansi dari objek akan dihitung dengan *alpha matting*. Nilai transparansi akan digunakan dalam *compositing* objek dengan *background* baru, sedangkan teknik pembiasan dilakukan dengan menggunakan rumus pembiasan *Snell's Law*. Aplikasi akan dibuat menggunakan bahasa pemrograman C# dengan Visual C# 2010 sebagai IDE nya.

Hasil pengujian menunjukkan bahwa nilai alpha/transparansi yang didapat dipengaruhi oleh pencahayaan, pembiasan oleh *background* lama, serta motif dari objek. Pembiasan pada objek sudah natural jika dibandingkan dengan kondisi asli. Selain itu posisi mata dapat diatur oleh pengguna untuk mendapatkan hasil sesuai keinginan. Kekurangan dari aplikasi ini adalah hasil gambar tidak dapat menyerupai objek pada gambar awal objek diambil. Selain itu, dikarenakan objek transparan yang diekstrak adalah *white object*, hasil *compositing* pada objek transparan berwarna akan terlihat tidak alami.

Kata Kunci: *Matting, Compositing, Transparent, Refraction, Alpha*

ABSTRACT: Nowadays, image manipulation, which is related to matting and compositing process, is commonly used (ex: filming industry). Processed object may vary, however issues may arise when the object is transparent. Not many research has been conducted about the issue due to the difficulties in extracting transparency value and object refraction, therefore in this thesis, an application to execute matting process and transparent object compositing is developed.

Matting process will be executed using Grabcut method, followed by calculation of transparency value of an object using alpha matting. The transparency value is then used in compositing an object with a new background, while refraction will be executed using refraction formula. The

application is based on C# programming language with Visual C# 2010 as its IDE.

Experiments result showed that extracted transparency value was affected by lighting, refraction of the old background, and object's pattern. Refraction result looks natural compared with the actual condition. Moreover, eye position can be configured by the user to obtain results as desired. The downfall of this application is that the result image cannot look similar to the real object. Furthermore, since the extracted transparent object is a white object, the compositing result on colored transparent object may appear unnatural.

Keywords: *Matting, Compositing, Transparent, Refraction, Alpha*

1. PENDAHULUAN

Tingkat kebutuhan manusia dalam hal manipulasi suatu citra/gambar terus meningkat, terutama di bidang desain. Pembuatan poster, desain ruangan, dsb, cenderung membutuhkan manipulasi seperti menghapus objek yang ada pada suatu gambar dan melakukan kreasi warna pada sebuah gambar. Hal tersebut dilakukan untuk menampilkan hasil yang lebih bagus atau memiliki kesan tertentu. Kebutuhan ini selaras dengan keinginan manusia untuk mendapatkan citra yang kreatif secara praktis/langsung, sehingga bermunculan aplikasi-aplikasi serta riset yang dilakukan berkaitan dengan manipulasi gambar.

Salah satu manipulasi gambar yang ada dan banyak digunakan adalah mengganti latar dari sebuah gambar dimana gambar diambil dengan latar berwarna biru. Dalam hal ini, perlu ditentukan objek (artis, hewan, benda, dan lain-lain) dari sebuah gambar yang ingin dipertahankan sebelum memindahkannya dari gambar awal ke gambar baru yang diinginkan (latar luar angkasa, kerajaan, dan lain-lain). Hal ini dikenal dengan *matting* dan *compositing*. *Matting* itu sendiri adalah metode untuk mengambil *object/region of interest* dari suatu gambar sedangkan *compositing* adalah metode untuk menggabungkan objek yang telah didapat dengan *background* lain.

Dewasa ini, sudah terdapat berbagai riset, seperti *matting* menggunakan pendekatan Bayesian [1], untuk melakukan hal tersebut tetapi muncul permasalahan jika objek yang dituju bersifat transparan. Hal ini dikarenakan ketika objek transparan berada di sebuah gambar, maka secara tidak langsung, objek tersebut akan mengandung warna dari *background* lama, sedangkan untuk proses *compositing*, objek tersebut harus terlihat alami dengan menampakan

warna dari *background* yang baru. Untuk menghilangkan warna *background* pada objek diperlukan perhitungan nilai transparansi yang bergantung pada nilai *background*, akan tetapi nilai *background* jarang disediakan secara khusus oleh pengguna/orang awam.

Dari permasalahan yang dijabarkan di atas dapat dilihat manipulasi terhadap gambar banyak dibutuhkan dalam aplikasi nyata. Oleh karena itu, dibuat aplikasi untuk melakukan *matting* serta *compositing* terhadap objek yang transparan.

2. TEORI PENUNJANG

2.1. *Matting* dan *Compositing*

Matting adalah metode untuk mendapatkan suatu objek dari sebuah gambar [2]. Permasalahan utama dalam proses *matting* ialah penentuan warna pada *pixel* yang berada di *edge*/pinggiran objek yang telah terekstrak. Setiap *pixel* pada objek sudah tercampur dengan warna *background* dan tidak diketahui *ratio*/perbandingannya. Terdapat beberapa algoritma yang dapat digunakan untuk proses *matting* diantaranya Alpha *matting*, Bayesian *matting*, Poisson *matting* [3], dan lain-lain. Algoritma yang ada menggunakan *trimap* sebagai sebuah pendekatan untuk mendapatkan *matte*. *Matte* dapat digunakan sebagai filter untuk memisahkan antara *foreground* dan *background* berdasarkan nilai transparansi suatu *pixel*. Sedangkan proses untuk menggabungkan suatu objek dengan *background* lain merupakan proses *compositing*.

Pada proses *compositing*, sebuah gambar $I(x,y)$ bisa didapatkan dari berbagai kombinasi *background* $B(x,y)$ dan *foreground* $F(x,y)$ dengan menggunakan alpha *matte* $\alpha(x,y)$, secara matematis, persamaannya adalah sebagai berikut:

$$I = \alpha F + (1 - \alpha)B \quad (1)$$

dimana nilai dari $\alpha(x,y)$ berada antara [0,1]. Ketika nilai $\alpha(x,y) = 1$ atau 0 dapat diartikan bahwa *pixel* pada (x,y) adalah pasti *foreground* atau pasti *background*, selain itu maka nilai pada *pixel* tersebut merupakan perpaduan keduanya [4].

2.2. Alpha *Matte*

Pada objek yang transparan, beberapa hal yang akan diperhatikan yaitu nilai dari alpha *matte*. Nilai ini akan dipergunakan untuk mengetahui intensitas transparansi setiap *pixel*nya. *Matte* akan digunakan untuk proses *compositing* dan akan menentukan tingkat transparansi/*coverage* objek. Persamaan yang digunakan adalah persamaan pada riset oleh Gong, Liang, Yang R., dan Yang, Y.H. (2010) [5], yaitu sebagai berikut:

$$\alpha = \frac{(I-B).(I-B)}{(F-B).(I-B)} \quad (2)$$

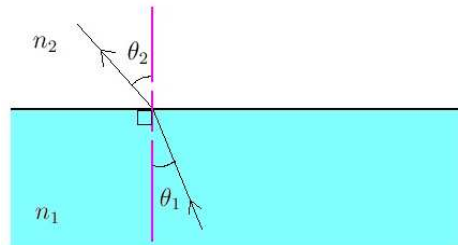
dimana I adalah gambar sembarang dari *user*, lalu F dan B berturut-turut merupakan estimasi *foreground* dan estimasi *background* dari gambar yang diberikan. Dalam persamaan tersebut nilai *alpha* setiap *pixel* dihitung dengan melibatkan seluruh *channel* warna. Estimasi dilakukan dengan *input* berupa *trimap* dan algoritma yang digunakan adalah interpolasi linier dimana nilai dari *pixel* yang akan dihitung diambil dari rata-rata 4-*connectivity pixel* disekelilingnya.

2.3. Pembiasan

Pembiasan adalah proses ketika cahaya melintasi suatu medium ke medium yang lain. Hukum pembiasan membantu untuk memprediksi pembengkokan yang terjadi, hukum ini dikenal dengan Snell's law. Dalam hukum ini, arah cahaya pembiasan bergantung pada indeks medium yang dilewati cahaya dan arah cahaya datang. Persamaan untuk hukum ini dapat dilihat pada persamaan berikut:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (3)$$

n merupakan indeks bias dan θ merupakan sudut yang terbentuk. Pada Gambar 1 diilustrasikan proses pembiasan dimana dari medium n_1 terjadi pembiasan ke medium n_2 .



Gambar 1. Ilustrasi pembiasan Snell's law

Sumber :

<http://www.physicsclassroom.com/Class/refrn/u14l2b.cfm>

Setiap benda transparan memiliki indeks bias sendiri-sendiri tergantung pada kecepatan cahaya dilingkungan vakum dan di material transparan. Cahaya selalu lebih lambat saat melintasi material daripada melintasi vakum, sehingga indeks bias selalu lebih besar dari 1 [6]. Pada Tabel 1 terdapat beberapa material diikuti dengan nilai indeks biasnya.

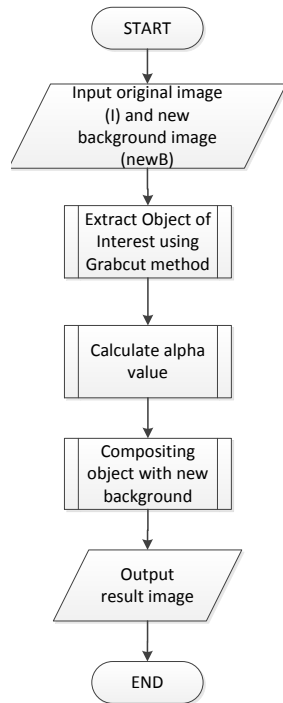
Tabel 1. Tabel Indeks Bias

Material	n (indeks bias)
Vacuum	1.000
Air	1.000277
Water	4/3
Crown glass	1.50 – 1.62
Flint glass	1.57 – 1.75

Sumber : <http://hyperphysics.phy-astr.gsu.edu/hbase/tables/indrf.html>

3. DESAIN SISTEM

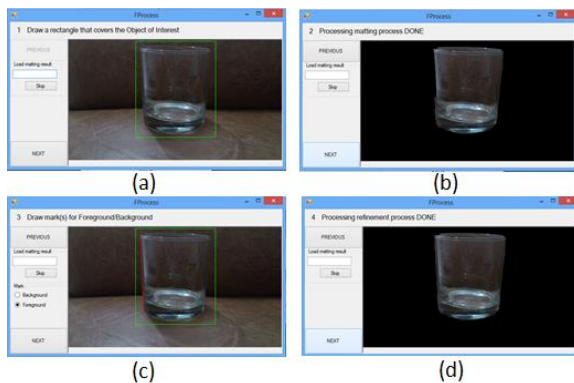
Pada pembuatan aplikasi ini terdapat tahap proses yang akan dilakukan seperti diagram alir pada Gambar 2. Proses-proses yang akan dilakukan antara lain adalah proses *matting*, lalu proses perhitungan alpha, proses perhitungan pembiasan hingga *compositing* objek pada *background* baru.



Gambar 2. Diagram alir sistem secara garis besar

Input pada aplikasi yang dibuat adalah 2 buah gambar yaitu gambar objek asli dan gambar *background* baru sedangkan *output* nya adalah hasil *compositing* dari objek transparan pada *background* baru.

Proses pertama yang dilakukan adalah proses *matting*. Pada proses ini objek didapatkan menggunakan metode grabcut. Metode ini diambil berdasarkan riset yang dilakukan oleh Yeung, S., Tang, C.K., Brown, M.S., dan Kang, B.S. pada jurnal [7]. Dengan proses inilah, bentuk objek yang diinginkan akan didapat dari suatu gambar (Lihat gambar 3).



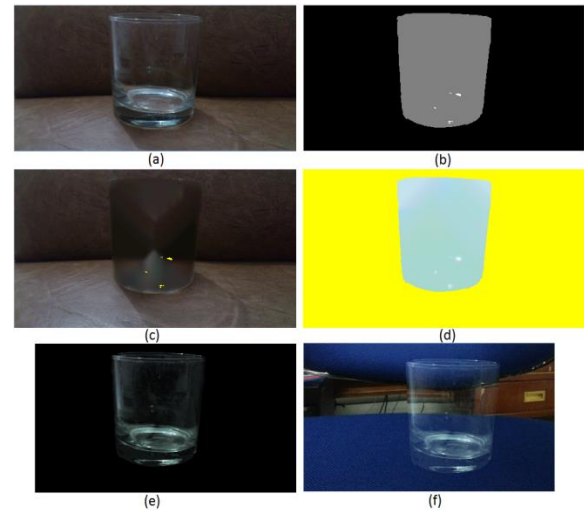
Gambar 3. Penerapan grabcut pada aplikasi (a) penandaan area objek (b) hasil proses (c) penandaan tambahan untuk *desired result* (d) objek hasil proses

Proses kedua merupakan proses perhitungan nilai alpha/transparansi. Nilai ini merupakan *matte* yang akan digunakan untuk proses *compositing* dan akan menentukan tingkat transparansi/*coverage* objek untuk tiap *pixel*-nya.

Pada proses *calculate* alpha dilakukan beberapa langkah sebagai berikut:

- Melakukan perbaikan *trimap* dari *trimap* hasil *thresholding*
- Melakukan estimasi *foreground* dan *background* dengan melakukan interpolasi
- Melakukan perhitungan alpha *matte* dengan persamaan (2)

Pada gambar 3 dapat dilihat hasil estimasi *background* (3c) dan *foreground* (3d) dengan *input trimap* gambar 3a.



Gambar 3. Proses perhitungan nilai alpha (a) gambar asli (b) *trimap* (c) hasil estimasi *background* (d) hasil estimasi *foreground* (e) hasil transparansi dengan alpha *matte* (f) hasil *compositing*

Proses terakhir merupakan penambahan efek pembiasan pada objek dengan melakukan perhitungan dengan rumus pembiasan (lihat persamaan (3)). Setiap *pixel* pada gambar akan dianggap sebagai vektor sehingga bisa diasumsikan bahwa adanya jarak antara objek, *background* baru dan mata. Proses pertama pada perhitungan pembiasan adalah dihitungnya sudut bias setiap *pixel* jika diasumsikan mata diletakkan pada tengah objek. Medium yang digunakan untuk sudut datang adalah medium udara dengan indeks bias 1.000277 sedangkan, untuk sudut bias pada medium kaca dengan indeks bias 1.57. Proses setelah mendapat sudut datang dan sudut bias adalah mencari tahu arah vektor berdasarkan sudut bias yang telah didapat dari persamaan (3). Arah vektor dicari untuk mendapatkan lokasi *pixel* yang akan diambil dari *background* baru.





4. PENGUJIAN

Pengujian terhadap aplikasi dibagi menjadi beberapa bagian diantaranya adalah sebagai berikut:

- Pengujian terhadap objek dengan *background* awal yang bervariasi.
Suatu objek yang ingin diambil oleh *user* tidak dapat dikontrol oleh aplikasi, bisa saja objek berada pada posisi/latar apapun tergantung *input* dari *user*. Pengujian ini dilakukan untuk melihat hasil aplikasi jika objek diambil pada *background* sembarang mulai dari *background* satu warna hingga rumit. Dalam

pengujian ini, (lihat Tabel 2) diambil sebuah objek polos/tidak bermotif agar terlihat perbandingan hasilnya.

Tabel 2. Pengujian terhadap objek dengan background awal yang bervariasi

No	Gambar Asli	Hasil Compositing
1		
2		

Dari pengujian sebuah objek pada *background* bervariasi dapat dilihat bahwa *background* awal sangat mempengaruhi hasil transparansi suatu objek. Pada Tabel 2 gambar 2 dapat dilihat bahwa *background* awal ikut serta pada hasil transparansi dan efek ini akan sangat terlihat jika objek diletakkan pada latar baru dengan motif polos. Sedangkan gambar 1 cenderung lebih bagus karena objek tersebut memiliki *background* awal dengan mayoritas satu warna. Hal ini dikarenakan tidak adanya *background* awal tanpa objek sehingga pada proses perhitungan nilai alpha, nilai *background* diambil dari hasil estimasi dan belum tentu akurat untuk mengekstrak nilai *foreground*.

- Pengujian terhadap objek bermotif dan tidak. Pengujian ini dilakukan untuk mengetahui seberapa jauh aplikasi dapat menangani kasus/permasalahan untuk objek yang polos serta objek bermotif. Pada Tabel 3 ditunjukkan hasil aplikasi terhadap 2 objek dimana *background* baru yang digunakan adalah *background* satu warna.


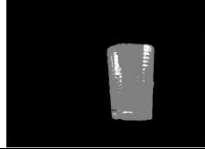







Tabel 3. Pengujian terhadap objek bermotif dan tidak

No	Gambar Asli	Hasil Compositing
1		
2		

Dari pengujian pada Tabel 3 dapat dilihat bahwa terdapat perbedaan antara objek bermotif dan tidak bermotif dalam hal partisipasi *background* awal. *Background* pada hasil objek polos (gambar 1) akan terlihat dihasil akhirnya sedangkan pada objek yang bermotif (gambar 2), *background* awal tersamarkan oleh motif gelas.

- Pengujian pengaruh *trimap* pada hasil aplikasi. *Trimap* merupakan hal utama yang mempengaruhi hasil estimasi baik estimasi *foreground* maupun *background*. Hasil estimasi inilah yang akan mempengaruhi perhitungan nilai alpha, oleh karena itu dilakukan pengujian untuk membandingkan hasil dari sebuah objek dengan 2 buah *trimap* berbeda.

Tabel 4. Pengujian pengaruh trimap pada hasil aplikasi

Ket.		
Trimap		
Est. Back		
Est. Fore		
Hasil		

Trimap yang digunakan pada Tabel 4 gambar sebelah kiri merupakan *trimap* yang didapatkan dari hasil *thresholding* akan tetapi hasil aplikasi menggunakan *trimap* tersebut kurang baik (bandingkan hasil gambar sebelah kiri dan kanan). Hal ini dikarenakan adanya bagian kecil *pixel* yang dianggap *background* padahal sebenarnya merupakan *foreground* (efek dari proses *matting*) atau sebaliknya dan hal ini mempengaruhi estimasi untuk perhitungan alpha. Oleh karena itu, dilakukan perbaikan *trimap* seperti pada sebelah kanan untuk mendapatkan hasil yang lebih baik. Dari pengujian ini dapat disimpulkan *trimap* yang dibentuk akan mempengaruhi hasil akhir dari aplikasi.

- Pengujian terhadap hasil aplikasi pada *background* baru yang bervariasi. Pada Tabel 5 dapat dilihat terdapat sejumlah objek yang digabungkan kepada berbagai *background* baru. Pada aplikasi ini, warna objek bergantung pada warna dari estimasi *foreground* sehingga jika estimasi *foreground* berwarna putih maka warna objek saat *compositing* akan putih. Hal ini merupakan permasalahan karena jika objek berwarna putih

diletakkan pada *background* berwarna terang/rumit maka warna objek akan 'kalah' atau dalam arti warna *background* menjadi dominan dan bentuk gelas tidak terlihat. Berbanding lurus dengan warna hitam, jika diletakkan pada *background* berwarna gelap maka objek menjadi tidak terlihat. Selain itu warna estimasi *foreground* yang diinterpolasi dengan informasi warna yang cenderung diambil dari warna dasar gelas atau pantulan cahaya tidak memungkinkan untuk mendapatkan warna seperti kondisi asli objek.

Tabel 5. Pengujian terhadap hasil aplikasi pada *background* baru yang bervariasi



- Pengujian terhadap objek transparan berwarna
Pengujian dilakukan menggunakan objek transparan yang berwarna atau bukan berwarna putih. Tujuan pengujian ini adalah untuk mengetahui keberhasilan metode yang digunakan untuk mengekstrak *non-white object*.

Tabel 6. Tabel pengujian pada objek berwarna

No	Gambar Asli	Hasil <i>Compositing</i>
1		
2		

Tabel 6. Tabel pengujian pada objek berwarna (sambungan)



Pada Tabel 6 dapat dilihat bahwa ketiga objek yang diuji memiliki hasil yang secara kasat mata tidak sesuai dengan objek asli yang ingin diambil. Hal ini dikarenakan estimasi *foreground* yang merupakan informasi warna objek tidak mendapatkan komposisi warna objek sesungguhnya. Berbeda dengan *white object* yang penentuan estimasi warna menggunakan cahaya pantulan pada objek, pada objek transparan berwarna hal tersebut tidak dapat diterapkan. Selain itu warna objek dan *background* sudah tercampur dan komposisi masing-masing bagian sulit ditentukan. Dapat disimpulkan bahwa aplikasi ini tidak dapat mengekstrak objek transparan berwarna/*non-white object*.

5. KESIMPULAN

Berdasarkan hasil pengujian didapatkan beberapa kesimpulan yaitu:

- *Trimap* digunakan sebagai *input* dalam proses estimasi *foreground* dan *background*. *Trimap* yang didapat dari hasil *thresholding* perlu diperbaiki sesuai dengan kondisi nyata suatu objek karena ketika ada warna *foreground* yang masuk pada proses estimasi *background*, maka hasil estimasi akan meleset dan sebaliknya sehingga mempengaruhi hasil estimasi. Hasil estimasi *foreground/background* akan baik jika *area foreground* dan *background* yang ditandai benar.
- Proses perhitungan transparansi suatu objek dipengaruhi oleh gambar estimasi *foreground* dan *background*.
- Objek bermotif dapat diekstrak dengan perhitungan nilai alpha. Motif tersebut bisa terekstrak karena motif merupakan bagian transparansi hanya saja dengan tingkat transparansi yang tinggi atau cenderung tidak memantulkan *background*.
- Pencahayaan dan pemantulan saat objek difoto/diambil secara tidak langsung akan memberikan efek motif pada objek transparan dan hasil ini akan mempengaruhi nilai alpha sehingga jika dilakukan *compositing* objek, motif akan tetap ada.
- Warna dari *foreground* pada hasil akhir akan mengikuti hasil dari estimasi *foreground* karena tidak adanya informasi lain mengenai warna *foreground* sehingga akan sulit untuk mendapatkan warna seperti warna asli objek.
- Objek transparan yang berwarna tidak dapat diekstrak oleh aplikasi karena adanya keterbatasan dalam estimasi warna objek.
- Posisi mata akan mempengaruhi hasil pembiasan pada suatu objek. Perubahan posisi X dan Y akan memberikan efek translasi sedangkan perubahan posisi Z akan memberikan efek mendekat/menjauh.

- Hasil akhir aplikasi akan terlihat tidak natural (secara objek bukan pembiasan) jika objek yang diekstrak berwarna terang digabungkan dengan *background* yang terang pula, dan sebaliknya.

6. REFERENSI

- [1] Chuang, Y.-Y., Curless, B., Salesin, D., Szeliski, R. (2001). *A Bayesian Approach to Digital Matting*. In Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR 2001), Vol. II, 264-271, December 2001.
- [2] Wang, J. and Cohen, M. (2007). *Image and Video Matting: A Survey*. Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 2.
- [3] Sun, J., Jia, J., Tang, C.-K., dan Shum, H.Y. (2004). *Poisson Matting*. ACM Trans. Graph. 23, 3, 315-321.
- [4] Porter, T. dan Duff, T. (1984). *Compositing Digital Images*. ACM Siggraph '84.
- [5] Gong, M., Wang, L., Yang, R., dan Yang, Y.-H. (2010). *Real-Time Video Matting using Multichannel Poisson Equations*.
- [6] Serbanescu, R.M.. (2003). Refraction of Light.
- [7] Yeung, S.-T., Tang, C.-K., Brown, M.S., dan Kang S.B. (2010). *Matting and Composition of Transparent and Refractive Objects*. ACM Trans. Graph., 13 pages.