

二色性反射モデルに基づくリアルタイム鏡面反射成分除去

肥後智昭[†] 宮崎大輔[†] 池内克史[†]

[†] 東京大学生産技術研究所

E-mail : {higo, miyazaki, ki}@cvl.iis.u-tokyo.ac.jp

あらまし 近年、コンピュータビジョンの分野において、鏡面反射成分と拡散反射成分の分離に関する研究が盛んに行われている。分離の主な方法として、偏光板を用いる方法や、色情報を用いる方法など様々な方法が提案されているが、道具を必要としたり、計算に時間がかかってしまうなどの問題がある。そこで本論文では、偏光板などの道具を使わず、カメラで取得した画像から、二色性反射モデルに基づく解析を行い、鏡面反射成分を除去し拡散反射成分だけを残した画像を高速に生成する手法を示す。この手法を用いてビデオカメラで取得した画像から、リアルタイムで鏡面反射成分を除去することに成功した。

Realtime Removal of Specular Reflection Component Based on Dichromatic Reflection Model

Tomoaki Higo[†], Daisuke Miyazaki[†], Katsushi Ikeuchi[†]

[†] Institute of Industrial Science, The University of Tokyo

E-mail : {higo, miyazaki, ki}@cvl.iis.u-tokyo.ac.jp

Abstract Today, a study about separating specular and diffuse reflection components has been vigorously conducted in the field of computer vision. Various methods have been proposed to separate these two components; using polarizing filters, needing color information, and so on. But these methods require specific instruments and take much computational time. In this paper, we propose a novel method based on dichromatic reflection model to generate rapidly diffuse images from camera images without any instruments. Using this method, we succeeded in real-time removal of specular reflection from video camera image.

1 はじめに

近年、コンピュータビジョンやパターン認識の分野では、与えられた画像をもとに、写っている物体の形状やテクスチャといった物体固有の特徴を抽出する処理が必要となっている。あるシーンを撮影した画像から物体が持つ固有の情報と、光源による影響とを分離するために、画像から鏡面反射成分と拡散反射成分との分離に関する研究が盛んに行われている。

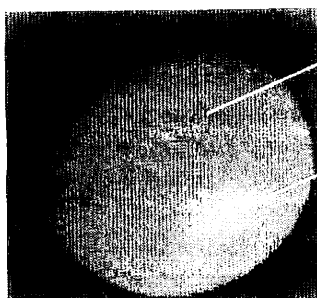


図 1: 拡散反射と鏡面反射

光が物体に当たって反射を起こす場合、大きく分けて 2 種類の反射がある (図 1)。1 つはハイライト成分や *specular* などと呼ばれるもので、物体表面で反射し入射光と同じ色をしている鏡面反射成分、もう 1 つは *diffuse* などと呼ばれるもので、物体内部で複雑な反射をし、物体の色を決めている拡散反射成分である。

分離の主な方法として、偏光板を用いる方法がある [5][6][10]。これは反射光の鏡面反射成分と拡散反射成分の偏光の違いを利用したものであるが、撮影時に偏光板を適切に取り付ける必要があったり、既存の画像については処理できないなどの問題がある。

また他の分離の手法として、複数の画像から色情報を用いるために、光源方向を変化させて分離を行う方法や [11][12]、光源を動かす代わりにカメラを移動させることで分離を行う方法がある [7][13]。これらの方法はある条件下での複数の画像を必要とするため、撮影時の負担が大きく、扱うデータ量が大きくなるなどの問題がある。

同様に色情報を用いて、単一の画像から分離を行う方法がある [1][4][8][9]。これらの手法は一度 *specular-free* 画像を求めておいてから処理を行った

り、何度か画像修復を行う必要があるなど、比較的計算時間がかかる手法が多い。

そこで本論文では、偏光板などの道具を使わず、カメラで取得した単一の画像から、二色性反射モデルに基づく解析を行い、拡散反射成分だけを残した画像を高速に生成する手法を提案する。本手法では、簡便なアルゴリズムを用いているにもかかわらず、非常に有用な結果を得ることができる。さらに計算時間が短いため、ビデオカメラから画像を取得して、リアルタイムで鏡面反射成分の除去を行うことが可能である。

この技術によって、例えば物体認識やステレオのシステムにおいて、しばしば邪魔になる鏡面反射成分を瞬時に除去することができるので、より高速に処理を行えるなどの応用が期待できるだろう。

章立ては以下の通りである。次の章では、二色性反射モデルなどの本手法に必要な物理的な基礎について述べ、第 3 章では、独自の色空間を用いた本手法の原理とその証明を行う。さらに本システムのアルゴリズムを説明する。第 4 章では、実験によって鏡面反射成分を除去した結果を示す。そして最後の章でまとめを行う。

2 物理的基礎

2.1 二色性反射モデル

二色性反射モデルとは、S.A.Shafer によって提案された反射モデルで [3]、「反射光のスペクトルは鏡面反射成分と拡散反射成分のスペクトルの線形和で表すことができる」というものである。二色性反射モデルは今日でも非常に多くの反射モデルの基礎となっている。本論文では物体が二色性反射モデルに従っていることを利用する。

反射光の輝度を I 、鏡面反射成分の輝度を I_s 、拡散反射成分の輝度を I_d とすると、二色性反射モデルは

$$I = I_s + I_d \quad (1)$$

と表すことができる。

例えば図 1 の鏡面反射の反射光では、鏡面反射成分の影響だけではなく、拡散反射成分の影響も表れており、図 1 の拡散反射の反射光では、鏡面反射成分は 0 で、拡散反射成分の影響のみということになる。つまり鏡面反射成分の除去とは、鏡面反射の反射光における鏡面反射成分を 0 にすることである。

2.2 hue, saturation, intensity

色を直感的に考える際には一般に *hue*, *saturation*, *intensity* の3つの要素の組み合わせで考えることが多い。*hue* とは色相のことで、りんごの色は赤、空の色は青、葉っぱの色は緑というように、それぞれ区別される色あいのことである。*saturation* とは彩度のことで、あざやかさの度合いを示したものである。*intensity* とは明度や輝度と呼ばれるもので、明るさの度合いを表したものである。

本論文ではこれら *hue*, *saturation*, *intensity* の3つの要素を分かりやすく示した独自の色空間の座標系を考えることによって、高速に鏡面反射成分を除去を行う。

3 鏡面反射成分除去

3.1 色空間

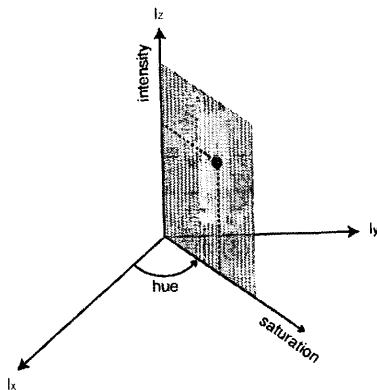


図 2: 色空間

本論文では鏡面反射成分を除去する計算を高速に行うために、D.Miyazaki らによって提案された色空間を用いる [2]。図 2 に示すように、 (I_x, I_y) 平面状で、 I_x 軸とのなす角度で *hue* が決まり、同じく (I_x, I_y) 平面状で、原点から近いほど *saturation* が小さく、遠いほど *saturation* は大きくなっている。また I_z 軸は *intensity* を表している。

この色空間は Bajcsy の S 空間と同様に [4]、*hue*, *saturation*, *intensity* を考えるのに最適である。さらに S 空間よりも RGB 空間からの変換が容易である。同様の性質を持った Ohta の (I_1, I_2, I_3) 空間と比較して [14]、対称性に秀でているため、直感的にわかりやすく、かつ解析の際の計算も容易である。

また HSI 色空間とも非常に似ているが、鏡面反射成分を除去する際に、3.2 の原理で述べるように、直線近似によって求めることができるため、HSI 色空間よりもロバストに計算できるという特長がある。

RGB とこの色空間との関係は次のように表される。

$$\begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} \quad (2)$$

さらに、この色空間での要素 (I_x, I_y, I_z) を用いると、*hue*, *saturation*, *intensity* の値が次式のように求まる。

$$\text{hue} = \arctan \frac{I_y}{I_x} \quad (3)$$

$$\text{saturation} = \sqrt{I_x^2 + I_y^2} \quad (4)$$

$$\text{intensity} = I_z \quad (5)$$

3.2 原理

本手法を用いるにあたって、まず次の3つを仮定する。

- 光源が白色で一様であること
- 物体の鏡面反射成分は光源のスペクトルと同じであること (NIR 仮定)
- 一つの *hue* には一つの surface color しかないこと

それでは鏡面反射成分を除去する原理について説明する。まず対象となる画像の各ピクセルの RGB の値を 3.1 の色空間上にプロットする。次に図 2 のように *hue* の値ごとに平面を考え、横軸を *saturation*、縦軸を *intensity* として、プロットした点のうち拡散反射の点に着目する。すると、色空間の特性から図 3 のように *saturation* と *intensity* は比例の関係になるので、この性質を利用して鏡面反射成分の除去を行う。

鏡面反射成分を除去するためには、まず *hue* の値で分類することによって、surface color ごとに分類する。さらに鏡面反射と拡散反射の *saturation* が同じ値になるのであれば、鏡面反射成分のベクトルは *intensity* と平行になるため、図 4 の直線の比例定数を用いて、*saturation* の値から拡散反射の *intensity* を求めることができる。これは鏡面反射成分を除去することと同意である。

以上より、証明すべき内容は以下の3つである。

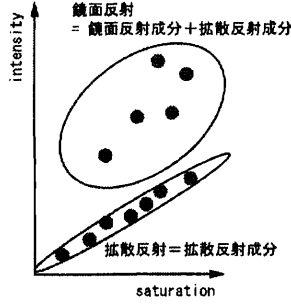


図 3: saturation-intensity 図

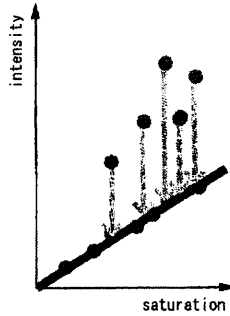


図 4: 鏡面反射成分を除去

(I) surface color が同じであれば、拡散反射の *hue* は全て等しく、鏡面反射の *hue* も全て等しい。かつ、両者は等しい。

(II) surface color が同じであれば、鏡面反射の *saturation* は拡散反射の *saturation* と等しく、鏡面反射の *intensity* は拡散反射の *intensity* よりも大きい。

(III) 拡散反射のピクセルは、*saturation* を横軸・*intensity* を縦軸とした直線となる

ただし一つの *hue* には一つの surface color しかないという仮定から、(II)(III) の証明については一つの *hue* だけで証明を行えば、他の *hue* でも同じことになる。

証明を行うための予備段階として、2.1 の二色性反射モデルを、RGB の輝度値に対して数式で表すことを考える。NIR 仮定から、鏡面反射成分は光源の色と同じであり、また、光源は白色であると仮定しているので、鏡面反射成分 I_s は

$$I_s = s \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (6)$$

と表せる。ここで縦ベクトルのそれぞれの成分は RGB の値を表しており、 s は鏡面反射成分の強さを表すスカラーである。また拡散反射成分 I_d は、ランバート面を仮定すると

$$I_d = \begin{pmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{pmatrix} \cos \theta \quad (7)$$

となる。ここで $\cos \theta$ はランバートの法則に出てくる shading を表す値である。 $(\alpha_r, \alpha_g, \alpha_b)$ は拡散反射の反射率を表しており、この値によって surface color が決まる。よって反射光の RGB の輝度は二色性反射モデルを用いて以下のように表すことができる。

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = s \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \alpha_r \\ \alpha_g \\ \alpha_b \end{pmatrix} \cos \theta \quad (8)$$

式 (8) は、 $s = 0$ のとき拡散反射のモデルであり、 $s > 0$ のとき鏡面反射のモデルとなる。

(I) の証明

拡散反射について *hue* を考えると、式 (8) で $s = 0$ と式 (2) から

$$\left. \begin{aligned} I_x &= \left(\alpha_r - \frac{1}{2}\alpha_g - \frac{1}{2}\alpha_b \right) \cos \theta \\ I_y &= \left(\frac{\sqrt{3}}{2}\alpha_g - \frac{\sqrt{3}}{2}\alpha_b \right) \cos \theta \end{aligned} \right\} \quad (9)$$

である。よって式 (3) から

$$\text{hue} = \arctan \frac{I_y}{I_x} = \arctan \left(\frac{\frac{\sqrt{3}}{2}\alpha_g - \frac{\sqrt{3}}{2}\alpha_b}{\alpha_r - \frac{1}{2}\alpha_g - \frac{1}{2}\alpha_b} \right) \quad (10)$$

ゆえに、 $(\alpha_r, \alpha_g, \alpha_b)$ により一意に値が決まる。

一方鏡面反射について *hue* を考えると、式 (8)(2) から、 s の成分は打ち消されて同様に式 (9) が導かれるので、*hue* の値も式 (10) と同じになり、 $(\alpha_r, \alpha_g, \alpha_b)$ により一意に値が決まる。以上から、surface color が等しければ、拡散反射と鏡面反射の *hue* は一意に求まり、かつ両者は等しい。

(II) の証明

鏡面反射の *saturation* を考える。式 (2)(4)(8) から

$$\begin{aligned} \text{saturation} &= \sqrt{I_x^2 + I_y^2} \\ &= \frac{\cos \theta}{\sqrt{2}} \sqrt{\{(\alpha_r - \alpha_g)^2 + (\alpha_g - \alpha_b)^2 + (\alpha_b - \alpha_r)^2\}} \end{aligned} \quad (11)$$

よって *saturation* は *hue* と同じく *s* によらずに値が決まる。つまり *saturation* は *s* に無関係なので、拡散反射と鏡面反射の *saturation* は同じ値を取ることがわかる。

次に鏡面反射の *intensity* について考えると、式 (5) から

$$\text{intensity} = I_z = \frac{\alpha_r + \alpha_g + \alpha_b}{3} \cos \theta + s \quad (12)$$

また拡散反射の *intensity* は

$$\text{intensity} = I_z = \frac{\alpha_r + \alpha_g + \alpha_b}{3} \cos \theta \quad (13)$$

ゆえに鏡面反射の *intensity* は拡散反射の *intensity* よりも *s* だけ大きい。

(III) の証明

拡散反射について *saturation* と *intensity* を考える。証明 (II) から鏡面反射と拡散反射の *saturation* は等しいので、式 (11)(13) から拡散反射の *saturation* と *intensity* は以下のように比例の関係にあることがわかる。

$$\text{intensity} = A \times \text{saturation} \quad (14)$$

ただし

$$A = \frac{\sqrt{2}}{3} \frac{\alpha_r + \alpha_g + \alpha_b}{\sqrt{(\alpha_r - \alpha_g)^2 + (\alpha_g - \alpha_b)^2 + (\alpha_b - \alpha_r)^2}} \quad (15)$$

よって傾き *A* は $(\alpha_r, \alpha_g, \alpha_b)$ により一意に値が決まっているので、(I) の証明とあわせて、傾き *A* は *hue* ごとに決まる。

以上より、(I)(II)(III) は全て証明されたので、RGB の情報だけを用いて鏡面反射成分を除去できることが示された。さらに式 (11)(13) で $\cos \theta$ の影響により、シェーディングの効果はそのまま残っていることになる。

しかしながら、この手法では白、灰色、黒といったグレースケールのピクセルを扱うことはできない。なぜなら、グレースケールでは $r \approx g \approx b$ であるから、式 (4) より *saturation* ≈ 0 となってしまう、グレースケールのピクセルは、例外的に図 3 に示すような直線上にのらないからである。

また当然のことながらカメラの感度を超えてしまったピクセルは処理できない。例えばビットマップの場合、RGB の値が実際には 255 以上の値であるはずなのに、255 として処理することになるので、当然 *hue* や *saturation* の値が本来の値とは異なり、結果も誤ったものとなる。よって撮影する際にはカメラの感度を超えないように注意する必要がある。

3.3 ホワイトバランス

3.2 の仮定で、光源は白色であるとした。しかしながら、あらかじめホワイトバランスを補正して、光源が白色になるように画像を修正しておくことによって、光源の色が白色でなくても本手法を用いることができる。

その方法は、まず測定したい色の光源の下でホワイトリファレンス (白色拡散標準板) を撮影し RGB の値を計測する。その RGB の値を正規化した値を (l_r, l_g, l_b) とすると、対象となる画像の各ピクセルの RGB の値 (L_r, L_g, L_b) を次式のように (l_r, l_g, l_b) で割り算することによって、光源を白色としたときの RGB の値 $(\hat{L}_r, \hat{L}_g, \hat{L}_b)$ を得ることができる。

$$\begin{pmatrix} \hat{L}_r \\ \hat{L}_g \\ \hat{L}_b \end{pmatrix} = \begin{pmatrix} L_r/l_r \\ L_g/l_g \\ L_b/l_b \end{pmatrix} \quad (16)$$

よって、有色光源の場合には式 (16) によって求めた $(\hat{L}_r, \hat{L}_g, \hat{L}_b)$ に対して、光源が白色である場合と同様の処理を行ってやればよい。その処理方法について次節で説明する。

3.4 アルゴリズム

鏡面反射成分を除去する本システムは 5 つの Step からなる。

本システムのアルゴリズム

Step1: 静止画をキャプチャする。

Step2: 画像の各ピクセルを RGB から色空間に射影することで、*hue*、*saturation*、*intensity* を計算する。

Step3: *hue* の値によりピクセルを分類し、*hue* が一定であるそれぞれの *saturation-intensity* 平面で、拡散反射のピクセルを選んでプロットする。

Step4: 分類した *hue* ごとに拡散反射がプロットされた直線の傾き *A* を計算。

Step5: 傾き *A* を用いて *intensity* を計算しなおし、元の画像と結果の画像を表示する。

具体的には、(Step1) まず取得した静止画の各ピクセルを、(Step2) RGB の値によって色空間に射影する。そして式(3)(4)(5)に従って *hue*、*saturation*、*intensity* を計算する。(Step3) 次に *hue* の値により分類して、*saturation* を横軸、*intensity* を縦軸とした平面を考える。図4のように、*saturation* の値ごとに最小の *intensity* をプロットする。最小の値をとることによって、拡散反射のピクセルのみを選び出すことができ、これらのプロットは(III)の証明より、原点を通る直線上にのる。よって(Step4) 直線の最小二乗法を用いて傾き *A* を各 *hue* ごとに求めてやればよい。鏡面反射は拡散反射と *saturation* が同じで *intensity* のみが違うので、(Step5) 求めた傾き *A* を用いて、式(14)より、全てのピクセルで *saturation* から出力画像の *intensity* を決めてやれば、それらは全て拡散反射の *intensity* と同じ値となり、結果として鏡面反射成分は除去されたことになる。

リアルタイムで連続して処理を行う場合は、Step1~5 で求めた結果を表示したら、Step1 へ戻り、その次の時点での画像をキャプチャして、同様の処理を繰り返せばよい。直線の最小二乗法は線形であるため $O(N)$ で計算ができ、本手法ではこの Step1~5 の処理を高速に行うことができるので、リアルタイムに結果の画像を表示することが可能である。さらに傾き *A* を求める際に用いるサンプル点の数はある程度減らすことが可能なので(例えば、プロットに用いるピクセルを一つ飛ばしにするなど)、さらに高速化することもできる。

4 実験

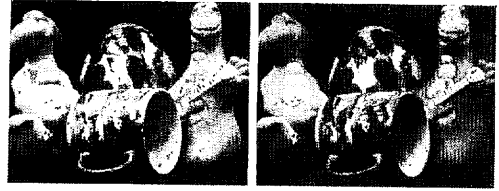


図 5: 白色光源下での鏡面反射成分除去

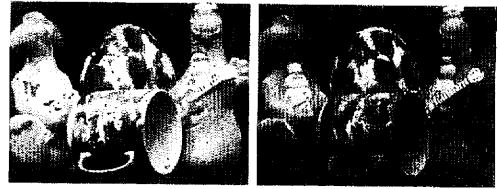


図 6: 赤色光源下での鏡面反射成分除去

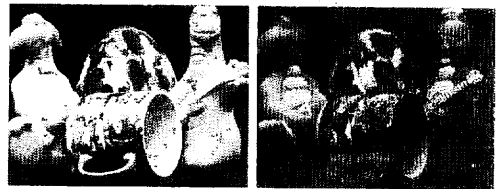


図 7: 緑色光源下での鏡面反射成分除去

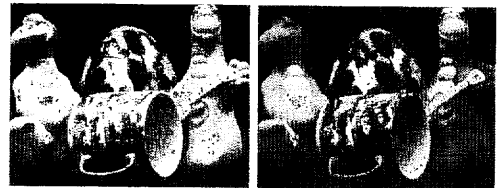


図 8: 青色光源下での鏡面反射成分除去

カメラは SONY の DXC9000(解像度 640×480 、 γ 値は 1)、PC の CPU は Intel PentiumIII(1GHz) を用いて室内で実験を行った。連続的にリアルタイムの処理を行ったところ、およそ 3[fps] で鏡面反射成分を除去した画像を表示することができた。

あるフレームにおける結果の画像を図5に示す。左の図はカメラで撮影した画像、右の図は提案手法により鏡面反射成分を除去した画像である。鏡面反射成分がうまく除去されている。またシェーディングの効果も残っていることがわかる。

ただし、グレースケールのピクセルは提案手法では処理できないので、何も手を加えずそのまま入力

と同じ色を表示するようにした。

また反射光がカメラの感度を超えないように撮影を行い処理を行ったため、得られた結果画像そのままでは全体的に暗くなってしまうので、結果の図5は見やすいようにPhotoshopを用いて明るさのレベル補正を行っている。

次にカラーフィルタを用いて光源の色を白色以外に変えて実験を行った。ホワイトリフレンスでのRGBを10点取って、その平均からホワイトバランスを補正した。その結果を図6、7、8に示す。ただし、これらの結果の図も見やすいように明るさのレベル補正を行っている。光源の色が白色でなくても、ホワイトバランスを補正することによって、鏡面反射成分を除去することができた。

しかしながら、問題点としては3.2の原理で述べた3つの仮定のうち、「一つのhueには一つのsurface colorしかない」という仮定が成り立たない場合がいくつか見つかり、その場合は鏡面反射成分の除去には問題ないが、元の画像と比べて、結果画像の色が異なるという問題が起きてしまう。例えば図9はマクベスチャートの有色部分に対して本手法で処理した結果(右)と元の画像(左)であるが、それぞれの画像の右下の肌色と茶色の部分に注目してみると、これらはsurface colorが異なるのに同じhueの値を持ってしまうため、3.4のアルゴリズムで示したStep4で直線の傾きAがうまく求められず、結果画像が元の画像とは色が異なっている。

また本手法では仮定でも述べたように、光源が一樣である必要があるので、相互反射の影響が大きい場合や、光源に揺らぎがある場合は、うまく鏡面反射成分を除去できない。

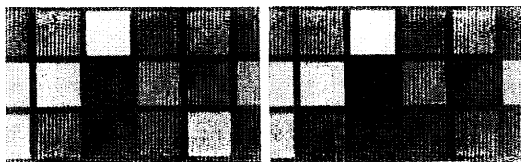


図9: 一つのhueに二つのsurface colorがある例

5 まとめ

本論文では、偏光板や複数の画像を用いず、一樣な光源の下での単一の画像だけを使って、鏡面反射

成分を除去した画像を高速に生成する手法を示した。このアルゴリズムでは独自の色空間に射影することによって非常に簡単な処理を行うだけで、有用な結果を得ることができる。また撮影時に特別な装置を用いたり、特別な処理をしていなくても、画像のRGBデータだけから鏡面反射成分を除去することができるため、既存の画像に対しても適用できるなどの利点がある。

しかし、「一つのhueには一つのsurface colorしかない」という仮定が成り立たない場合の結果画像における色の変化や、相互反射の影響が大きい場合に処理できないといった問題がある。これらの問題については、今後改良していく余地があるだろう。

参考文献

- [1] Robby T.Tan and Katsushi Ikeuchi, "Reflection Components Decomposition of Textured Surfaces Using Linear Basis Functions," *Proc. of Computer Vision and Pattern Recognition*, pp.1:125-131, 2005.
- [2] Daisuke Miyazaki, Robby T.Tan, Kenji Hara, Katsushi Ikeuchi, "Polarization-based Inverse Rendering from a Single View," *Proc. of International Conference on Computer Vision*, pp.982-987, 2003
- [3] S.A.Shafer, "Using color to separate reflection components," in *COLOR Research and Application*, Vol.10, No.4, pp.210-218, 1985
- [4] R.Bajcsy, S.W.Lee, A.Leonardis "Detection of Diffuse and Specular Interface Reflections and Inter-Reflections by Color Image Segmentation," in *International Journal of Computer Vision*, Vol.17, No.3, pp.241-272, 1996.
- [5] Lawrence B. Wolff, "Using polarization to separate reflectance components," *Proc. of Computer Vision and Pattern Recognition*, pp.363-369, 1989.
- [6] 梅山伸二, "物体の見えからの拡散/鏡面反射成分の分離-偏光フィルタを介した多重観測と確率的独立性を用いて-", *Proc. of Meeting on Image Recognition and Understanding*, pp.1:469-476, 2002.
- [7] 阪野貴彦, 池内克史, "EPI 解析に基づく動画からの鏡面反射成分の分離," *Proc. of Meeting on Image Recognition and Understanding*, pp.II:459-464, 2004.
- [8] P.Tan, S.Lin, L.Quan, H.Shum, "Highlight Removal by Illumination-Constrained Inpainting," in *Proc. of IEEE International Conference on Computer Vision*, pp.I:164-169, 2003.

- [9] Robby T.Tan and Katsushi Ikeuchi, "Separating Reflection Components of Textured Surfaces Using a Single Image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, No.2, pp.178-193, 2005.
- [10] T.Shibata, T.Takahashi, D.Miyazaki, Y.Sata and K.Ikeuchi, "Creating photorealistic virtual model with polarization based vision system," *Proc.SPIE*, Vol.5888, pp.25-35, 2005.
- [11] Y.Sato and K.Ikeuchi, "Temporal-color space analysis of reflection," *Journal of the Optical Society of America*, Vol.11, No.11, pp.2990-3002, 1994.
- [12] S.Lin and H.Y.Shum, "Separation of diffuse and specular reflection in color images," *Proc.of Computer Vision and Pattern Recognition*, Vol.1, pp.341-346, 2001.
- [13] R.Swaminathan, S.B.Kang, R.Szeliski, A.Criminisi and S.K.Nayar, "On the motion and appearance of specularities in image sequences," *Proc.of European Conference on Computer Vision*, Vol1, pp.508-523, 2002.
- [14] Y.Ohta, T.Kanade, T.Sakai, "Color Information for Region Segmentation," *Computer Graphics and Image Proceeding*, Vol.13, pp.222-241, 1980