

非平面に投影された影に基づく形状と光源の同時推定

山下 幸宏[†] 坂上 文彦[†] 佐藤 淳[†]

[†] 名古屋工業大学大学院 情報工学専攻 〒466-8555 愛知県名古屋市昭和区御器所町

E-mail: [†]{sakaue,junsato}@nitech.ac.jp

あらまし 光が物体によって遮られてできる影を用いた 3 次元復元手法は、画像ノイズの影響を受けにくく、また不連続な形状であっても復元できる点で優れている。近年、非平面に投影された影を用いて 3 次元復元を行う手法としてシャドウグラフが提案された。しかし、この手法は計算コストが大きく、また光源位置が既知である必要があった。そこで本稿では、物体形状も光源位置も共に未知という条件下で、その両方を高速に同時推定する手法を提案する。本稿では、シャドウグラフで用いる影に関する拘束に着目し、形状と光源を同時推定する問題をエネルギー最小化の枠組みで定式化する。さらに、コスト関数の最小化に交互最適化や粗密探索を適用したり、不必要な拘束を除去することで、効率的かつ高速に形状と光源位置を同時推定できることを示す。

キーワード Shape from Shadow, 光源推定, シャドウグラフ, エネルギー最小化, 交互最適化, 粗密探索

1. はじめに

単一視点からの 3 次元復元法として陰に基づく復元法 (Shape from Shading) [4] ~ [6] と影に基づく復元法 (Shape from Shadow) [1] ~ [3] が提案されている。陰が光源方向と面の法線方向による輝度の変化で生じるものであるのに対し、影は光源からの光が物体によって遮られて生じるものであることから、その性質は全く異なる。

陰に基づく復元法として、エネルギー最小化法 [4] や局所法 [5]、照度差ステレオ法 [6] などが知られている。これらの手法では、画像の輝度値を直接用いて面の法線を計算し、それを積分することにより 3 次元形状を復元するため、形状復元の際に蓄積誤差が生じやすいという欠点がある。また、対象物体として滑らかな曲面形状を仮定し、かつ特定の反射特性をもつ必要があるなど制約条件が大きい。

一方、影に基づく復元法では、画像の輝度値を直接形状復元に用いるのではなく、影か否かの 2 値情報を基に復元を行うため、画像ノイズの影響を受けにくく、また不連続な形状であっても復元が可能である。さらに、影は反射特性に依存しないため、非等方性物質や複合物質などの複雑な反射特性をもつ物体にも適用できるという利点がある。近年、非平面上の影から 3 次元形状を復元する手法として、Yu ら [1] がシャドウグラフを提案した。シャドウグラフでは、光源位置を変えながら得た影をもとに、シーンの各点の間で成り立つ高さの制約条件を求め、これを単純なグラフ構造で表現していく。この結果、シャドウグラフではグラフの最短経路探索を行うだけで複雑な 3 次元形状を復元することができる。しかし、2 次元画像の全画素をグラフのノードとして扱うため、最短経路探索の計算コストが膨大であるという大きな問題点があった。また、この手法は光源位置が既知であることを前提としていた。

そこで、本稿では光源位置が未知の状況で、影のみを用いて 3 次元形状を復元することを目指す。これまでに、

光源位置が未知という条件下で、影を用いて物体表面の法線を推定する手法を岡部ら [3] や Kriegman ら [2] が提案した。一般に、影は Attached Shadow と Cast Shadow の 2 種類に分類される。Attached Shadow は面の法線が光源の方向を向いていないときに生じる影であり、Cast Shadow は面の法線が光源の方向を向いているにも関わらず光源が物体に遮られてできる影である。岡部らは、物体表面上の各点を様々な光源下で撮影される Attached Shadow により符号化して、その類似度に基づいて面の法線を推定する Attached Shadow Coding を提案した。しかし、この手法では影として Attached Shadow のみを扱うため、Cast Shadow が生じるような凹部分が存在する物体では高精度な復元ができないという問題点があった。また、物体の周りに広く分布する多数の光源位置が必要であり、復元に膨大な画像枚数を要するという問題点もあった。一方、Kriegman らは、物体の遮蔽輪郭線上で計算される法線、および、3 本の Attached Shadow 境界線が一度に交差する点における法線の拘束を組み合わせることで、未知光源下で物体表面の法線を推定する手法を提案した。しかし、同一の点が 3 回以上 Attached Shadow 境界となる必要があるなど制約条件が大きく、また、Attached Shadow 境界を正確に検出するため、実際の画像に適用するのは困難であるという問題点があった。

そこで、本稿では、未知光源下において物体の形状と光源位置を高速かつ高精度に推定する手法を提案する。本稿ではシャドウグラフに着目し、グラフを構成する際に使用する影に関する拘束を用いて、任意の光源配置において形状と光源を同時推定する問題をエネルギー最小化の枠組みで定式化する手法を提案する。さらに、コスト関数の最小化に交互最適化や粗密探索を適用したり、不必要な拘束を除去することで、効率的かつ高速に形状と光源位置を同時推定できることを示す。提案法は、シャドウグラフで用いる拘束に基づいているため、Cast Shadow が生じるような凹部分が存在する物体にも適用

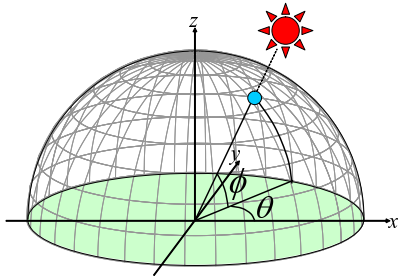


図 1 光源半球面上の無限遠光源座標

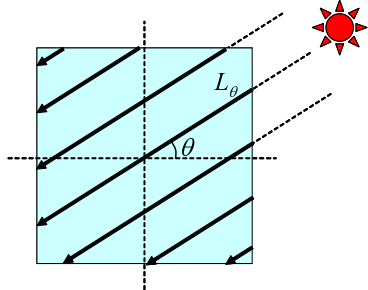


図 2 シーンに平行に照射される光線

でき、任意の光源配置にも対応できる．また、光源配置が物体の周りに一様に分布している必要もなく、少ない画像枚数から形状と光源位置を同時に推定できるという利点がある．

2. シャドウグラフ

まず、Yu らが提案したシャドウグラフについて説明する．シャドウグラフとは、シーンの各点の間で成り立つ相対的な高さの関係を、光源位置と影情報より計算し、これをグラフ構造で表現したものである．グラフの各ノードはシーン中の各点に対応し、ノード間のコストがこれらの点の間の相対的な高さ（高さの差）を表す．光源は無限遠光源を仮定し、全ての入力画像において、その光源位置は既知とする．また、シーンを撮影するカメラと対象物体の関係は固定されており、その間の距離は十分に離れているものとする．このような条件の下で、光源位置が異なる多数の画像を基にシーンの 3 次元形状復元を行う．

さて、無限遠光源の光源位置は、図 1 に示すシーンの天球（光源半球と呼ぶ）の球面上の座標として、2 つのパラメータ (θ, ϕ) で表現することができる．ここで、 θ はシーンから見た光源の方位角、 ϕ は天頂角を表す．シーンを真上から見ると、図 2 に示すように、光線はシーン中の全ての点において、方位角 θ の方向から平行に照射され、これらの光線を L_θ とする．いま、シーンを L_θ に沿って切り出した断面の 1 つが図 3 である場合を考える．ここで、 h_x は断面上の位置 x における面の高さを表す． L_θ に沿って切り出したシーンの断面上では、全ての点において天頂角 ϕ の方向から光線 L_ϕ が平行に照射される．また、 $[x_c, x_a]$ の領域（斜線部分）は影となる領域である．シーンの各点は入力画像の各画素と対応付いているため、シャドウグラフでは画像の各画素が影であるか否かに着目し、シーンの各点の間に成立する相対的な高さの関係を表す 3 種類の拘束を得る．それら 3 種類の拘束

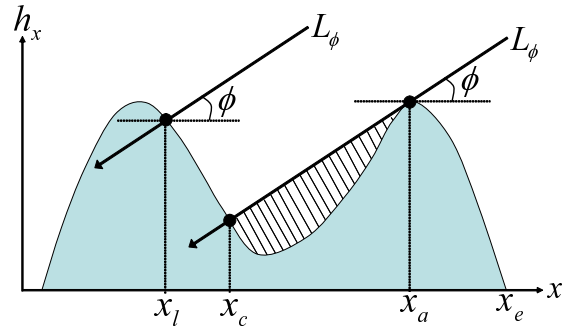


図 3 シーンの断面と 3 種類の拘束

を重み付き有向グラフとして表現することにより、シーンの形状復元に利用する．

(1) シャドウ拘束と絶対拘束

まず、注目する点が影と判定された場合に得られる拘束について考える．影領域 $[x_c, x_a]$ 内の点 x に着目すると、その高さ h_x は点 x_a を通る光線の高さ以下であることから、以下の不等式が成り立つことがわかる．

$$h_x \leq h_{x_a} - (x_a - x) \tan \phi, \quad \forall x \in [x_c, x_a] \quad (1)$$

これをシャドウ拘束と呼ぶ．シャドウ拘束は影領域に位置する点から生じる拘束であり、Attached Shadow 境界（点 x_a ）とそれに関する影領域内の全ての点の高さの関係を表す．特に $x = x_c$ の時には Attached Shadow 境界と Cast Shadow 境界の高さの関係が直接表されており、以下の等式の拘束となる．

$$h_{x_c} = h_{x_a} - (x_a - x_c) \tan \phi \quad (2)$$

これをシャドウ拘束の中でも特に絶対拘束と呼ぶ．これらのシャドウ拘束を、 x_a, x をノードとし、 $-(x_a - x) \tan \phi$ を点 x_a から x へのコストとして重み付き有向グラフで表現する．

(2) アンチシャドウ拘束

次に、注目する点が影でないと判定された場合に得られる拘束について考える．影でない点 x_l に着目すると、この点よりも光源側に位置する点 x の高さは、点 x_l を通る光線の高さより低いことから、以下の不等式が成り立つことがわかる．

$$h_x < h_{x_l} + (x - x_l) \tan \phi, \quad \forall x \in (x_l, x_e] \quad (3)$$

これをアンチシャドウ拘束と呼ぶ．アンチシャドウ拘束は影でない点と、それよりも光源側に位置する全ての点の高さの関係を表す．アンチシャドウ拘束も x_l, x をノード、 $(x - x_l) \tan \phi$ を点 x_l から x へのコストとして重み付き有向グラフで表現する．

光源位置が異なる複数の入力画像から構成されたシャドウグラフは、画像の各画素（シーンの各点）がグラフノードとなっており、各ノード間に与えられたコストはシーンの各点の間の相対的な高さを表している．従って、構成されたシャドウグラフに対して任意の 2 点間の最短経路探索 [7] を行うことにより、これらの 2 点間の相対的な高さの上界値を決定することができる．よって、全

での2点の組合せについて最短経路探索を行うことにより、シーンの3次元形状を復元することができる。

しかし、シャドウグラフでは2次元画像の全画素がグラフノードとなっているため、最短経路探索の計算コストが膨大となる。通常、負のコストを含むグラフに対して全ノード間における最短経路探索を行うには、グラフのノード数に対し3乗の計算オーダーが必要となる。従って、 $N \times N$ 画像から生成されたシャドウグラフの最短経路探索を行うには $O(N^6)$ の計算オーダーが必要である。また、シャドウグラフでは、光源位置が既知でないとグラフのコストが与えられないため、光源位置が未知の場合には適用できないという問題点がある。

3. 形状と光源の同時推定問題の定式化

本節では、2節で述べた影から得られる3種類の拘束を用いて、形状と光源を同時推定する問題をエネルギー最小化の枠組みで定式化する手法を提案する。まず、2節を振り返ると、画像中の影から式(1)~(3)に示したシャドウ拘束、絶対拘束、アンチシャドウ拘束という3種類の拘束が得られるのであった。これらの式を見てわかるように、影から得られる拘束には不等式拘束と等式拘束が混在し、そのままでは定式化をする際に扱いにくい。そこで本節では、これらの不等式拘束と等式拘束を同等の枠組みで扱えるように、各拘束を満たすか否かという2値情報の観点で関数化する。また、それを用いて満たさない拘束の数を表現するコスト関数を構成することにより、形状と光源の同時推定問題をエネルギー最小化問題として定式化する。

3.1 影から得られる拘束の2値情報化

まず、影から得られる各拘束を2値情報の観点で関数化する。はじめに、式(1)~(3)のシャドウ拘束、絶対拘束、アンチシャドウ拘束をそれぞれ以下の形に変形する。

$$h_x - h_{x_a} + (x_a - x) \tan \phi < 0, \quad \forall x \in (x_c, x_a) \quad (4)$$

$$h_{x_c} - h_{x_a} + (x_a - x_c) \tan \phi = 0 \quad (5)$$

$$h_x - h_{x_l} - (x - x_l) \tan \phi < 0, \quad \forall x \in (x_l, x_e] \quad (6)$$

ここで、式(4),(5)の x_a と式(6)の x_l を拘束の始点と呼び、式(4),(6)の x と式(5)の x_c を拘束の終点と呼ぶ。さて、2節で述べたように、これらの拘束は光源の方位角が θ である場合に、その方向に沿って切り出したシーンの断面上で生成される。そこで、それを明示的に表すために、式(4)~(6)の左辺をそれぞれ以下のように $S_\theta, B_\theta, A_\theta$ と置くことにする。

$$S_\theta = h_x - h_{x_a} + (x_a - x) \tan \phi \quad (7)$$

$$B_\theta = h_{x_c} - h_{x_a} + (x_a - x_c) \tan \phi \quad (8)$$

$$A_\theta = h_x - h_{x_l} - (x - x_l) \tan \phi \quad (9)$$

次に、次式で表されるような関数 $G_1(x)$ と $G_2(x)$ を用意する。

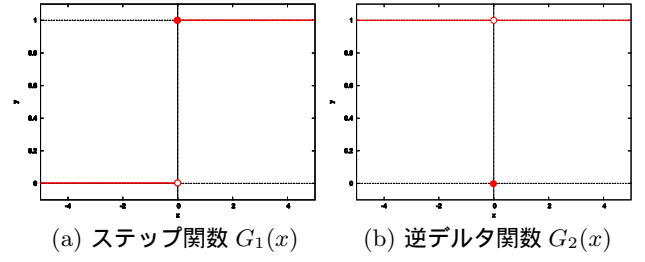


図4 拘束の2値情報化に用いる関数

$$G_1(x) = \begin{cases} 0 & (x < 0) \\ 1 & (x \geq 0) \end{cases} \quad (10)$$

$$G_2(x) = \begin{cases} 0 & (x = 0) \\ 1 & (x \neq 0) \end{cases} \quad (11)$$

$G_1(x)$ は図4(a)に示すように $x < 0$ のときに0、それ以外は1となるステップ関数、 $G_2(x)$ は図4(b)に示すように $x = 0$ のときのみ0、それ以外は1となる逆デルタ関数である。

ここで、 S_θ, A_θ を関数 $G_1(x)$ に、 B_θ を関数 $G_2(x)$ に代入したものを考える。すると、次式に示すように、いずれの拘束も満たしている場合に0、満たしていない場合に1が出力されることがわかる。

$$G_1(S_\theta) = \begin{cases} 0 & (\text{Satisfied}) \\ 1 & (\text{Not Satisfied}) \end{cases} \quad (12)$$

$$G_2(B_\theta) = \begin{cases} 0 & (\text{Satisfied}) \\ 1 & (\text{Not Satisfied}) \end{cases} \quad (13)$$

$$G_1(A_\theta) = \begin{cases} 0 & (\text{Satisfied}) \\ 1 & (\text{Not Satisfied}) \end{cases} \quad (14)$$

以降では、式(12)~(14)をそれぞれシャドウ拘束、絶対拘束、アンチシャドウ拘束と呼ぶことにする。このように、影から得られる各拘束を満たしているか否かという2値情報の出力となるように関数化することで、不等式で表される拘束（シャドウ拘束、アンチシャドウ拘束）と等式で表される拘束（絶対拘束）を同等の枠組みで扱うことができる。

3.2 コスト関数の定義

次に、式(12)~(14)で定義した2値情報化された拘束を用いて、形状と光源の同時推定問題を解くためのコスト関数を定義する。いま、 $N \times N$ 画像が M 枚入力される場合を考えると、求めるべきパラメータはシーン中の各点の高さを表す形状パラメータ h_p ($p = 1, \dots, N^2$) と、各画像における光源位置を表す光源パラメータ θ_k, ϕ_k ($k = 1, \dots, M$) である。すなわち、合計 $N^2 + 2M$ 個の未知パラメータを求める必要がある。ここで、 k 番目の入力画像に着目すると、光源の方位角 θ_k の方向に沿うシーンの断面は多数存在し、それら全ての断面から式(12)~(14)に示す各拘束が生成される。そこで以降で

は、各画像から生成される全ての拘束をまとめて扱うために、各拘束に通し番号を付けることにする。\$k\$ 番目の入力画像における光源の方位角が \$\theta_k\$ である場合に、その画像から生成される各拘束のうち、\$l\$ 番目のシャドウ拘束を \$G_1(S_{\theta_k}^l)\$, \$m\$ 番目の絶対拘束を \$G_2(B_{\theta_k}^m)\$, \$n\$ 番目のアンチシャドウ拘束を \$G_1(A_{\theta_k}^n)\$ と置き、これらを用いて形状と光源の同時推定問題を解くためのコスト関数を次式のように定義する。

$$F(h_1, \dots, h_{N^2}, \theta_1, \dots, \theta_M, \phi_1, \dots, \phi_M) = \sum_k \left\{ \sum_l G_1(S_{\theta_k}^l) + \sum_m G_2(B_{\theta_k}^m) + \sum_n G_1(A_{\theta_k}^n) \right\} \quad (15)$$

ここで、\$S_{\theta_k}^l, B_{\theta_k}^m, A_{\theta_k}^n\$ は、各拘束の始点を \$x_s^l, x_s^m, x_s^n\$、終点を \$x_g^l, x_g^m, x_g^n\$ と置くことで、それぞれ次式のように表される。

$$S_{\theta_k}^l = h_{x_g^l} - h_{x_s^l} + (x_s^l - x_g^l) \tan \phi_k \quad (16)$$

$$B_{\theta_k}^m = h_{x_g^m} - h_{x_s^m} + (x_s^m - x_g^m) \tan \phi_k \quad (17)$$

$$A_{\theta_k}^n = h_{x_g^n} - h_{x_s^n} - (x_g^n - x_s^n) \tan \phi_k \quad (18)$$

式 (15) は全画像からの全てのシャドウ拘束、絶対拘束、アンチシャドウ拘束の和で表されており、各拘束からは満たしていれば 0、満たしていなければ 1 が出力される。よって、このコスト関数の出力値は満たしていない拘束の数を表すことになる。従って、式 (15) を最小化する形状パラメータ \$h_p\$ (\$p = 1, \dots, N^2\$) と光源パラメータ \$\theta_k, \phi_k\$ (\$k = 1, \dots, M\$) を求めることにより、影から得られる拘束を最も良く満たすシーンの形状と光源位置を同時に推定することができる。また、このようにして推定された形状と光源が作る影は、入力画像の影のでき方を最も良く表現する。

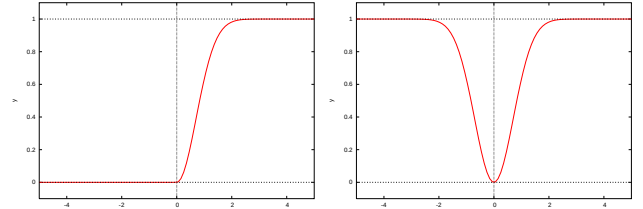
4. エネルギー最小化による形状と光源の同時推定

本節では、3.2 節で定義したコスト関数を効率的に最小化する手法について述べる。特に、最小化の際に交互最適化や粗密探索を用いたり、余分な拘束を除去することで、影から形状と光源を効率的かつ高速に同時推定する手法を提案する。

4.1 不連続関数の非線形近似による最急降下法の適用

まず、式 (15) の最小化に最急降下法 [10] を適用する方法について述べる。最急降下法を適用するためには、各パラメータの勾配成分を算出する必要がある。しかし、式 (15) 中に存在する関数 \$G_1(x), G_2(x)\$ が不連続関数であるため、そのままでは微分不可能である。そこで、式 (15) を微分可能にするため、関数 \$G_1(x), G_2(x)\$ を非線形近似することを考える。

まず、関数 \$G_1(x)\$ は図 4(a) に示したステップ関数であった。ステップ関数の非線形近似にはシグモイド関数が一般によく用いられるが、シグモイド関数を微分する



(a) 片側逆ガウス関数 \$g_1(x)\$ (b) 逆ガウス関数 \$g_2(x)\$

図 5 \$G_1(x), G_2(x)\$ の非線形近似に用いる関数

と、入力が負値の場合においても勾配成分が出現する。一方、式 (15) において関数 \$G_1(x)\$ への入力 \$S_{\theta_k}^l, A_{\theta_k}^n\$ が負値であることは拘束を満たしていることを意味する。よって、関数 \$G_1(x)\$ の近似にシグモイド関数を用いると、満たしている拘束からも勾配成分を出現させることになり、推定に悪影響を及ぼしてしまう。そこで、関数 \$G_1(x)\$ の近似にはシグモイド関数を用いず、代わりに次式に示す片側逆ガウス関数 \$g_1(x)\$ を用いる。

$$g_1(x) = \begin{cases} 1 - \exp(-\beta x^2) & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (19)$$

ここで、\$\beta\$ は関数の広がり具合を決める係数であり、\$\beta\$ の値が大きいほどステップ関数をよく近似する。関数 \$g_1(x)\$ をプロットしたものを図 5(a) に示す。図 5(a) より、関数 \$g_1(x)\$ はステップ関数 \$G_1(x)\$ を滑らかに近似し、さらに、入力が負値の場合には出力が 0 となるため、満たしている拘束からは勾配成分が出現しないことがわかる。

一方、関数 \$G_2(x)\$ は図 4(b) に示した逆デルタ関数であった。そこで、関数 \$G_2(x)\$ の近似には、次式に示す逆ガウス関数 \$g_2(x)\$ を用いる。

$$g_2(x) = 1 - \exp(-\beta x^2) \quad (20)$$

関数 \$g_2(x)\$ をプロットしたものを図 5(b) に示す。図 5(b) より、関数 \$g_2(x)\$ は逆デルタ関数 \$G_2(x)\$ を滑らかに近似していることがわかる。

式 (15) 中の関数 \$G_1(x), G_2(x)\$ を、上で定義した関数 \$g_1(x), g_2(x)\$ で近似することによりコスト関数が微分可能となり、最急降下法を適用できる。

4.2 各パラメータの勾配成分の算出

次に、各パラメータの勾配成分を算出する。まず、式 (15) 中の \$S_{\theta_k}^l, B_{\theta_k}^m, A_{\theta_k}^n\$ の中身は式 (16) ~ (18) のように表されることから、パラメータ \$h_p\$ (\$p = 1, \dots, N^2\$), \$\phi_k\$ (\$k = 1, \dots, M\$) については、以下のように式 (15) を数式的に微分して勾配を計算することができる。

$$\frac{\partial F}{\partial h_p} = \sum_k \left[\sum_l \{ \delta(p, x_g^l) g_1'(S_{\theta_k}^l) - \delta(p, x_s^l) g_1'(S_{\theta_k}^l) \} + \sum_m \{ \delta(p, x_g^m) g_2'(B_{\theta_k}^m) - \delta(p, x_s^m) g_2'(B_{\theta_k}^m) \} + \sum_n \{ \delta(p, x_g^n) g_1'(A_{\theta_k}^n) - \delta(p, x_s^n) g_1'(A_{\theta_k}^n) \} \right] \quad (21)$$

$$\frac{\partial F}{\partial \phi_k} = \sum_l \frac{x_s^l - x_g^l}{\cos^2 \phi_k} g_1'(S_{\theta_k}^l) + \sum_m \frac{x_s^m - x_g^m}{\cos^2 \phi_k} g_2'(B_{\theta_k}^m) - \sum_n \frac{x_g^n - x_s^n}{\cos^2 \phi_k} g_1'(A_{\theta_k}^n) \quad (22)$$

ここで、 $g_1'(x), g_2'(x)$ は、それぞれ関数 $g_1(x), g_2(x)$ の導関数を表す。また、式 (21) の $\delta(i, j)$ は、Kronecker のデルタであり、 $i = j$ の場合のみ 1、それ以外は 0 となる。

一方、パラメータ θ_k ($k = 1, \dots, M$) については、式 (15) を数式的に微分して勾配を計算できないため、数値微分 [11] を用いて以下のように勾配を計算する。

$$\frac{\partial F}{\partial \theta_k} = \frac{1}{12\Delta\theta} (C_{\theta_k-2\Delta\theta} - 8C_{\theta_k-\Delta\theta} + 8C_{\theta_k+\Delta\theta} - C_{\theta_k+2\Delta\theta}) \quad (23)$$

ここで、 $\Delta\theta$ は補間に用いる点列の刻み幅であり、 $\Delta\theta$ の値が小さいほど微分の近似精度が高くなる。また、 $C_{\theta_k+\alpha}$ は k 番目の入力画像における光源の方位角が $\theta_k + \alpha$ である場合に、その画像から生成される全てのシャドウ拘束、絶対拘束、アンチシャドウ拘束の和を表す。

以上のように各パラメータの勾配成分を計算し、式 (15) を最急降下法により最小化することで、シーンの形状と光源位置を推定できる。

4.3 交互最適化と粗密探索による効率的な推定

4.1 節で述べたように、不連続関数を非線形近似することで、式 (15) のコスト関数に対して最急降下法を適用可能となる。しかし、形状パラメータと光源パラメータは互いに性質が大きく異なるため、共通のステップ幅で両者を同時に更新し、式 (15) を最小化することは不可能である。そこで、次式に示すように、形状パラメータ h_p ($p = 1, \dots, N^2$) と光源パラメータ θ_k, ϕ_k ($k = 1, \dots, M$) を交互に最適化する。

$$h_p^{(t+1)} = \arg \min_{h_p} F(h_1, \dots, h_{N^2} \parallel \theta_1^{(t)}, \dots, \theta_M^{(t)}, \phi_1^{(t)}, \dots, \phi_M^{(t)}) \quad (24)$$

$$\theta_k^{(t+1)}, \phi_k^{(t+1)} = \arg \min_{\theta_k, \phi_k} F(\theta_1, \dots, \theta_M, \phi_1, \dots, \phi_M \parallel h_1^{(t+1)}, \dots, h_{N^2}^{(t+1)}) \quad (25)$$

ここで、 $F(\cdot)$ は式 (15) で定義したコスト関数を表し、“ $\parallel$ ” の右側は固定するパラメータ、左側は更新するパラメータを表す。また、交互最適化のステップ t における推定値を、パラメータの右上に “ (t) ” を付けて表す。交互最適化の流れは以下ようになる。まず、式 (24) に示すように、未知パラメータのうち光源パラメータを固定し、形状パラメータのみを更新して最小化を行う。これにより、現段階で推定されている光源位置 $\theta_k^{(t)}, \phi_k^{(t)}$ に対して最も相応しい形状 $h_p^{(t+1)}$ を推定できる。続いて、式 (25) に示すように、形状パラメータを固定し、光源パラメータのみを更新して最小化を行う。これにより、式 (24) で推定された新たな形状 $h_p^{(t+1)}$ に対して最も相応

しい光源位置 $\theta_k^{(t+1)}, \phi_k^{(t+1)}$ を推定できる。このように、式 (24) と式 (25) の操作を、初期値 $h_p^{(0)}, \theta_k^{(0)}, \phi_k^{(0)}$ から始めて交互に繰り返すことで、形状パラメータと光源パラメータを段階的に最適な値に近づけることができる。

次に、推定の際に局所解を防ぐ方法について述べる。3.2 節を振り返ると、推定すべきパラメータの数は、 $N \times N$ 画像を M 枚入力する場合、形状パラメータと光源パラメータを合わせて $N^2 + 2M$ 個であった。また、このうち N^2 個は形状パラメータであるため、入力画像の解像度が高いほど、推定すべきパラメータの数が膨大となり、局所解に陥る可能性が高くなる。そこで、初期の推定では低解像度な入力画像を用い、上述した交互最適化により粗い形状と光源位置を推定する。次に、推定された形状と光源位置を初期値として、前回よりも解像度の高い入力画像を用いて、より密な形状と光源位置を推定する。このように、入力画像の解像度を段階的に変更し、粗密探索 [8] を行うことで、局所解を防ぎつつ、形状と光源を効率良く同時推定できる。

4.4 余分な拘束の削減による高速な推定

4.3 節では、式 (15) の最小化の際に、交互最適化や粗密探索を用いることにより、局所解を防ぎつつ効率的に形状と光源を同時推定できることを述べた。しかし、段階的に入力画像の解像度を高くするにつれ、拘束の数が膨大となり、計算量が膨大になってしまうという問題がある。1 枚の入力画像の影から得られる拘束の数について考えると、式 (1) より、シャドウ拘束は 1 つの影領域に対して、その影領域に含まれる点の数だけ拘束が生成される。また、式 (2) より、絶対拘束は 1 つの影領域に対して 1 つだけ拘束が生成される。よって、画像中の影領域に位置する点の数を N_s とすると、シャドウ拘束と絶対拘束を合わせた数は $O(N_s)$ となる。一方で、式 (3) より、アンチシャドウ拘束は 1 つの影でない点に対して、その点より光源側に位置する全ての点の数だけ拘束が生成される。よって、画像中の影でない点の数を N_l とすると、アンチシャドウ拘束の数は $O(N_l^2)$ となる。このように、アンチシャドウ拘束は影でない点の数の 2 乗のオーダーで生成されるため、入力画像の解像度が高くなるにつれて、その数が膨大になってしまう。

そこで、アンチシャドウ拘束の数を最小限に減らすことを考える。まず、あるシーンの断面において生成される拘束を図 6(a) を用いて考える。図 6(a) において、青色はシャドウ拘束と絶対拘束、赤色はアンチシャドウ拘束、白色は拘束の始点を表す。さて、2 節を振り返ると、影から得られる拘束はいずれも、シーン中の 2 点間の高さの上界値を光線の高さとして表すのであった。ここで、図中の点 x_s に関する拘束（緑色の囲い部分）に着目すると、拘束 A における光線の高さは、拘束 B, C, D における光線の高さよりも低いいため、拘束 A が表す点 x_s の高さの上界値は、拘束 B, C, D が表す上界値よりも低くなる。これは、拘束 A を満たせば拘束 B, C, D は必然的に満たすことを意味しており、拘束 A を採用すれば拘束 B, C, D は除去することができる。同様の議論により、

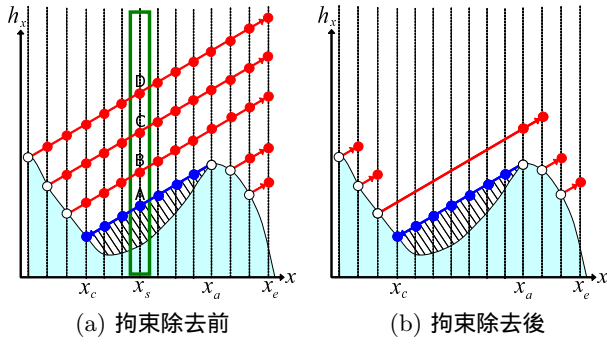


図 6 余分なアンチシャドウ拘束の削減

図 6(a) から除去可能な拘束を全て削減すると図 6(b) のようになる．図 6(b) より，ある点を始点とするアンチシャドウ拘束のうち，以下に示す拘束のみを採用すればよいことがわかる．まず，始点の 1 つ光源側の点が影でない場合，その点を終点とする拘束のみを採用すればよい．また，始点の 1 つ光源側の点が影である場合，その影領域の Attached Shadow 境界と，さらに 1 つ光源側の点を終点とする拘束のみを採用すればよい．上記以外のアンチシャドウ拘束を全て除去することで，1 つの点から生成されるアンチシャドウ拘束の数が最大で 2 つとなり，全体の数を $O(N_l^2)$ から $O(N_l)$ に削減することができる．以上のように，余分なアンチシャドウ拘束を大幅に削減することで，各拘束の効力を維持しつつ，計算量の大幅な削減が可能となる．

5. 実験

本節では，提案法を用いた実験結果を示す．

5.1 実画像実験

まず，実画像を用いて提案法の有効性を評価した結果を示す．本実験では，任意の曲面形状を持つシーンと，複数の物体が隣接するシーンの 2 種類を用いた．各シーンの周りで光源を任意に動かし，それぞれ 24 枚の画像を撮影し，それらの画像を用いて提案法により形状と光源を推定した．提案法における繰り返しの初期値は次のように与えた．まず，形状パラメータの初期値は，図 7(a) に示すように，全ての高さが 0 の平面とした．次に，光源パラメータの初期値は，図 7(b) に示すように，天頂角については全ての画像について $\phi_k = 45^\circ$ とし，方位角については各画像について，シーンの右方向 ($\theta_k = 0^\circ$)，上方向 ($\theta_k = 90^\circ$)，左方向 ($\theta_k = 180^\circ$)，下方向 ($\theta_k = 270^\circ$) の 4 択のうち，最も近いものとした．

また，提案法による推定結果には GBR 不定性 [9] が含まれるため，推定形状と真の形状との間に成立する GBR 変換を計算し，それを用いて推定された形状と光源位置を変換することで GBR 不定性を除去した．その後，形状と光源位置それぞれについて以下の方法により誤差評価を行った．まず，光源位置については，光源半球の原点に対する真値と推定値の平均角度誤差を計算することにより行った．次に，形状については，真の形状の高さの最大値を h_{max} ，最小値を h_{min} ，画素 (i, j) における

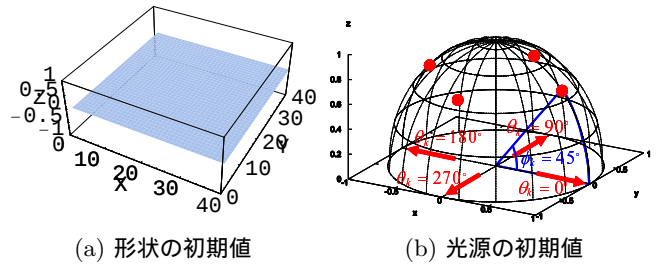


図 7 繰り返しの初期値

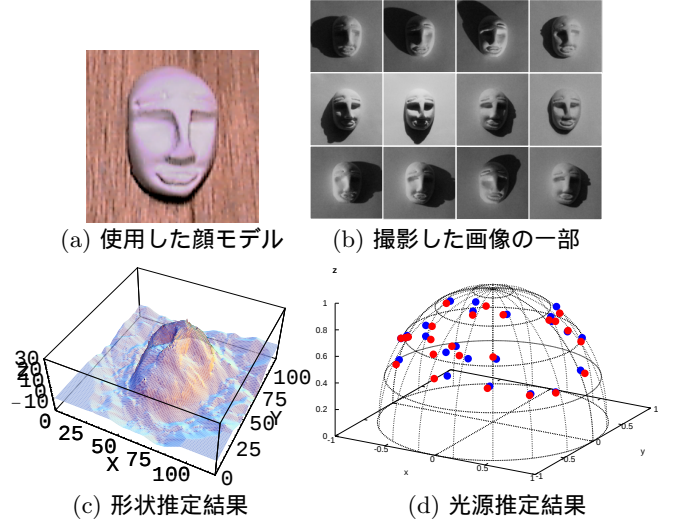


図 8 任意曲面を持つシーンを用いた実画像実験

真の形状の高さを h_{ij} ，推定形状の高さを h'_{ij} ，画像の画素数を N としたとき，次式により平均誤差 $D(\%)$ を計算して評価した．

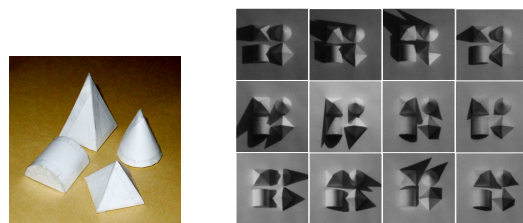
$$D = \frac{1}{N} \sum_{i,j} \frac{\|h_{ij} - h'_{ij}\|}{h_{max} - h_{min}} \times 100 \quad (26)$$

(1) 任意曲面を持つシーンを用いた実画像実験

まず，任意の曲面形状を持つシーンを用いた実画像実験結果を示す．本実験では，図 8(a) に示す顔モデルを用いた．光源を任意に動かして撮影した画像の一部を図 8(b) に示す．図 8(b) の撮影画像を見ると，曲面形状の中で影が複雑に曲がって投影されている様子が確認できる．提案法における最終的な形状と光源の推定結果をそれぞれ図 8(c) と (d) に示す．光源位置については，赤色が推定値，青色が真値を表す．図 8(c) の形状と真の形状との誤差は 4.24%，図 8(d) において推定値と真値との平均角度誤差は 3.75° であった．以上より，提案法は任意曲面形状に対して有効に働き，任意の光源配置において撮影された画像から高精度に 3 次元形状と光源位置を同時に推定できることがわかる．

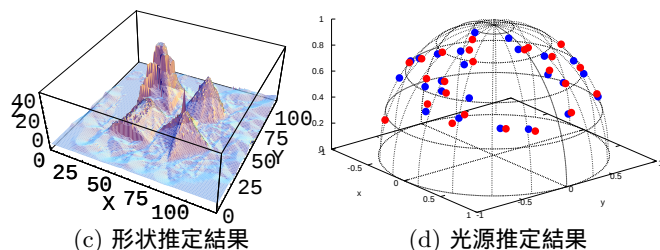
(2) 複数の物体が隣接するシーンを用いた実画像実験

次に，複数の物体が隣接したシーンにおける実画像実験結果を示す．本実験では，図 9(a) に示すシーンを用いた．光源を任意に動かして撮影した画像の一部を図 9(b) に示す．図 9(b) の撮影画像を見ると，シーン中の異なる物体同士で影が複雑に投影し合っている様子が確認できる．また同時に，シーン中の物体が滑らかな曲面形状を



(a) 使用したシーン

(b) 撮影した画像の一部



(c) 形状推定結果

(d) 光源推定結果

図 9 複数の物体が隣接するシーンを用いた実画像実験

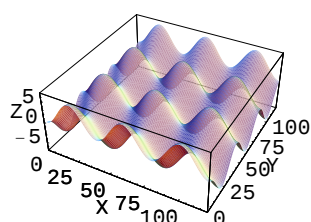


図 10 精度評価に用いた形状

持たないため、投影される影の曲がり方も滑らかでないことが確認できる．提案法における最終的な形状と光源の推定結果をそれぞれ図 9(c) と (d) に示す．図 9(c) の形状と真の形状との誤差は 4.77%，図 9(d) において推定値と真値との平均角度誤差は 5.22° であった．このことから，シーン中の物体が滑らかな曲面形状を持たない場合でも提案法は有効に働き，異なる物体間で相互に影を投影し合う状況において形状と光源が高精度に推定できることがわかる．

5.2 形状と光源の推定精度評価

次に，合成画像を用いて提案法の推定精度評価を行った結果を示す．本精度評価では図 10 に示す曲面形状を対象とした．この形状をカメラに投影することで合成画像を生成し，生成した画像に標準偏差 $\sigma = 2$ の輝度ノイズを与えたものを入力画像として提案法により形状と光源の推定を行った．提案法における繰り返しの初期値および推定された形状と光源の誤差の計測方法は，5.1 節の実画像実験と同様にした．

(1) 入力画像枚数の増減と推定精度

まず，入力画像枚数の増減による形状と光源の推定精度の変化を調べた．本実験では，入力画像枚数を 36 枚, 24 枚, 18 枚, 12 枚, 8 枚の 5 通りに変化させ，それぞれの場合において提案法により形状と光源を推定した．図 11 に入力画像枚数が 36 枚, 18 枚, 8 枚の場合における形状と光源の最終的な推定結果を示す．また，入力画像枚数の増減による推定形状と推定光源位置の誤差の推移をそれぞれ図 12 の (a) と (b) に示す．図 12 より，入力画像枚数が少なくなるに連れて，推定される形状と光源の誤差

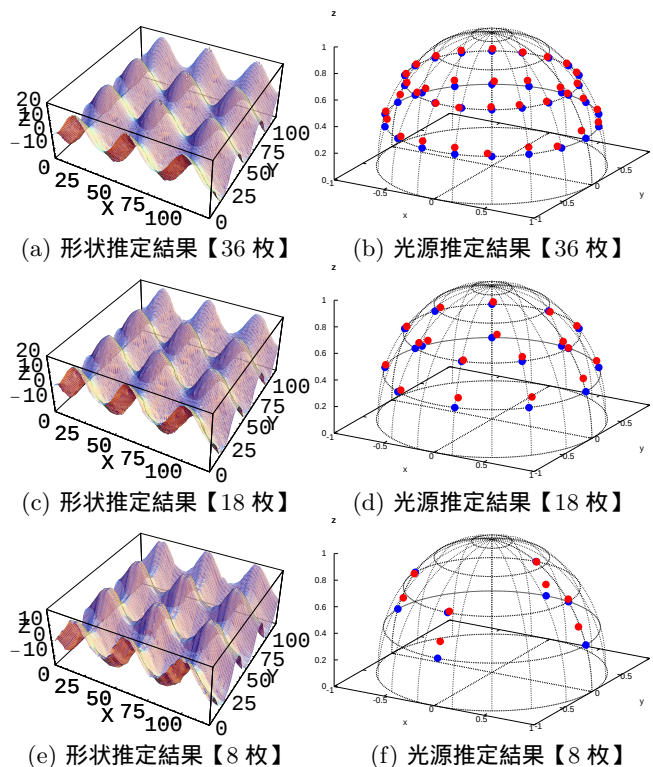
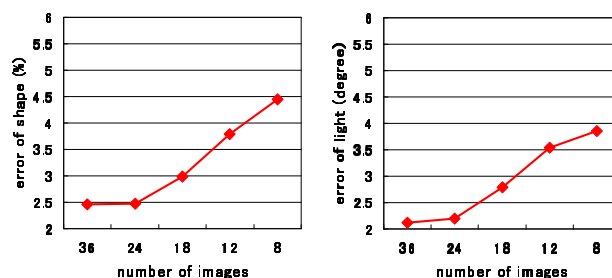


図 11 様々な入力画像枚数に対する推定結果



(a) 形状の誤差

(b) 光源の誤差

図 12 入力画像枚数の増減による誤差の推移

が大きくなることがわかる．しかし一方で，図 11 を見ると，提案法では入力画像枚数が 18 枚程度の少ない枚数でも高精度に形状と光源が推定できることがわかる．

(2) 光源配置に偏りがある場合の推定精度

次に，光源配置に偏りがある場合における形状と光源の推定精度を評価する．本実験では，光源半球の右側半分の領域のみに光源を配置して合計 18 枚の入力画像を作成し，提案法により形状と光源を推定した．提案法により推定された最終的な形状と光源をそれぞれ図 13 の (a) と (b) に示す．図 13(a) の形状と真の形状との誤差は 4.48%，図 13(b) において推定値と真値の平均角度誤差は 3.71° であった．以上より，光源半球の左側半分の領域には全く光源を配置していないにもかかわらず，提案法により高精度に形状と光源が推定できており，提案法が光源配置が偏っている場合にも有効であることがわかる．

5.3 計算時間評価

最後に，提案法の計算時間について触れる．まず，入力画像枚数を増減させた場合の提案法における計算時間の変化を計測する．次に，4.4 節で述べた余分な拘束の

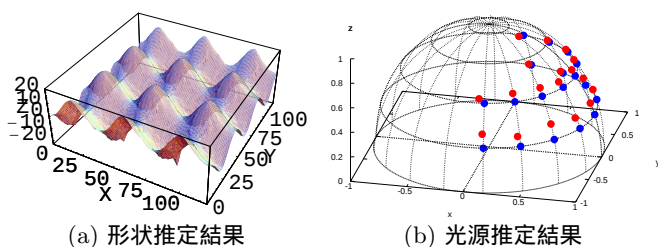


図 13 光源配置に偏りがある場合の推定結果

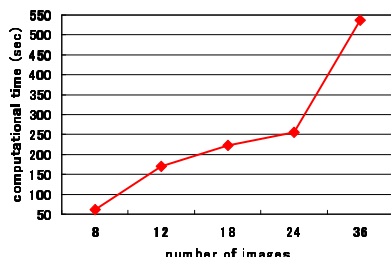


図 14 入力画像枚数の増加に伴う計算時間の推移

削減の有効性を確認する．なお，本実験には Quad Core Xeon 3.0GHz の CPU を搭載した計算機を使用した．

(1) 入力画像枚数の増減と計算時間

まず，入力画像枚数を増減させた場合の提案法における収束までの計算時間を計測した結果を示す．本実験では，5.2 節の (1) の実験と同様に，入力画像枚数を 8 枚, 12 枚, 18 枚, 24 枚, 36 枚の 5 通りに変化させ，提案法によって最終的な推定結果を得るまでの計算時間を計測した．図 14 に，それぞれの入力画像枚数における計算時間を示す．この図より，入力画像枚数が増加するに連れて，計算時間が増加することがわかる．一方で，提案法では 36 枚程度の画像枚数を用いても計算時間は約 9 分であり，入力画像枚数が比較的多い場合においても提案法により形状と光源が高速に推定できることがわかる．

(2) 余分な拘束の削減の有効性評価

次に，4.4 節で述べた余分な拘束の削減により，計算時間がどの程度削減されるのかを評価した結果を示す．本実験では，入力画像枚数を 24 枚とし，画像の解像度を 40×40 , 80×80 , 120×120 の 3 通りに変化させた．それぞれの解像度において，余分な拘束の削減前と削減後における全拘束数および最急降下法の 1 ステップ当たりの平均計算時間を計測した．画像の解像度を变化させたときの全拘束数の推移を図 15(a) に示し，最急降下法の 1 ステップ当たりの平均計算時間の推移を図 15(b) に示す．いずれの図も青色が拘束削減前，赤色が拘束削減後を表す．図 15(a) より，余分な拘束の削減前は，画像の解像度が高くなるに連れて，拘束数が指数的に増加するのに対し，拘束削減後は解像度が高くなっても拘束数は緩やかにしか増加しないことがわかる．また，図 15(b) より，計算時間についても拘束削減前は画像の解像度が高い場合，1 ステップの処理を行うのに膨大な時間を要するのに対し，拘束削減後は解像度が高くても，わずか 2 秒程で処理できることがわかる．

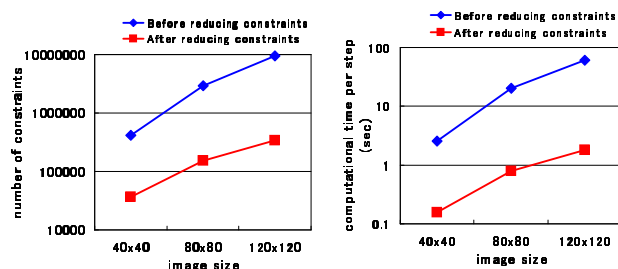


図 15 余分な拘束の削減前と削減後の比較

6. む す び

本稿では，シーンの形状と光源位置が共に未知という条件下で，その両方を影のみから同時に推定する手法を提案した．影に関する 3 種類の拘束に着目し，形状と光源を同時推定する問題をエネルギー最小化の枠組みで定式化した．さらに，コスト関数の最小化に交互最適化や粗密探索を適用したり，余分な拘束を除去することで，効率的かつ高速に形状と光源を同時推定できることを示した．実験により，提案法が様々なシーンに対して有効に働き，形状と光源を高速かつ高精度に同時推定できることを示した．一方，提案法では，影の判定は正しく行われることを仮定しているため，今後の課題としては，影の判定誤りを防ぎつつ，形状と光源を同時推定する手法に拡張することが挙げられる．

文 献

- [1] Y. Yu, J.T. Chang, "Shadow Graphs and 3D Texture Reconstruction," International Journal of Computer Vision, vol.62, no.1-2, pp.35-60, 2005.
- [2] D.J. Kriegman and P.N. Belhumeur, "What shadows reveal about object structure," JOSA A, vol.18, no.8, pp.1804-1813, 2001.
- [3] 岡部 孝弘, 佐藤いまり, 佐藤 洋一, "陰に基づく符号化による未知の反射特性・光源方向における法線推定," Proc. MIRU2009, OS2-1, pp.31-38, 2009.
- [4] K. Ikeuchi, B.K.P. Horn, "Numerical shape from shading and occluding boundaries," Artificial Intelligence, vol.17, no.1-3, pp.141-184, 1981.
- [5] A.P. Pentland, "Local shading analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.6, pp.170-187, 1984.
- [6] R. Woodham, "Photometric method for determining surface orientation from multiple images," Optical Engineering, vol.19, no.1, pp.139-144, 1980.
- [7] T. Corman, C. Leiserson, R. Rivest, "Introduction to Algorithms," MIT. Press, 1990.
- [8] M. Atiquzzaman, "Coarse-to-Fine Search Technique to Detect Circles in Images," The International Journal of Advanced Manufacturing Technology, vol.15, no.2, pp.96-102, 1999.
- [9] P.N. Belhumeur, D.J. Kriegman, A.L. Yuille, "The Bas Relief Ambiguity," International Journal of Computer Vision, pp.1040-1046, 1999.
- [10] 中野 良平, "ニューラル情報処理の基礎数理," 数理工学社, 2005.
- [11] Dahlquist. G, A. Bjorck, "Numerical Methods," Englewood Cliffs, NJ: Prentice-Hall, 1974.