Vol.2018-CVIM-212 No.35
2018/5/10

Mollifierを用いた非凸最適化手法の制約付き問題への拡張

岩田 大知[1,a)]　鮫島 正樹[1,b)]　菅野 裕介[1,c)]　松下 康之[1,d)]

概要：コンピュータビジョンにおける最適化問題には，非凸最適化問題として定式化されるものがある．多項式時間アルゴリズムが知られている凸最適化問題とは対照的に，非凸最適化問題の最適解を効率よく発見するアルゴリズムは知られていない．非凸最適化問題において性能の良い近似解を得る手法として，非凸な関数を Mollifier によって平滑化して問題を解き，平滑化を緩めながら元の問題の解に近づける段階的最適化が提案されている．しかし，従来の段階的最適化手法は制約の無い最適化問題を対象としており，制約付きの最適化問題に直接適用することができない．本稿では，制約付き凸最適化問題の最適性条件である KKT 条件 (Karush-Kuhn-Tucker condition) にもとづいて，制約付き非凸最適化問題を解けるよう段階的最適化手法を拡張する．さらに，等式制約にもとづいて Mollifier を調整することで，段階的最適化を効率化する手法を提案する．非凸最適化問題として定式化される Shape-from-Shading の問題を対象として実験を行い，従来の最適化手法と比較して提案手法が有効であることを示す．

Daichi Iwata[1,a)]　Masaki Samejima[1,b)]　Yusuke Sugano[1,c)]　Yasuyuki Matsushita[1,d)]

1. 序論

実世界における数多くの問題が非凸最適化問題として定式化されており，コンピュータビジョンにおいても非凸な制約条件を含む Shape-from-Shading [1], [2] や基礎行列の推定 [3] 等の問題が報告されている．非凸最適化問題に対して，勾配を利用した凸最適化手法をそのまま適用すると，局所最適解に陥る可能性がある．従来，局所解を回避するために様々な初期値を用いたり，非凸な関数を凸関数に近似したりして，凸最適化手法を適用する方法が提案されている [4], [5]．これらの方法は，元の非凸最適化問題または近似した凸最適化問題のどちらかを扱っており，非凸性が強く適切な初期値を設定できない問題や，精度よく凸近似ができない問題に対しては良い性能を得られない．これに対し，凸近似した問題から徐々に元の非凸最適化問題に近づけることで，元の問題の非凸性に沿った初期値で探索を行う段階的最適化に関する研究が進められている [6], [7]．段階的最適化では，非凸な関数を平滑化する関数として Mollifier [8] を適用し，凸関数に近似した問題を凸最適化手法によって解く．近似した問題の解を初期解として採用し，平滑化を緩和した問題を繰り返し解くことで，元の問題の最適解を求める．従来の段階的最適化に関する研究では，主に制約条件を持たない非凸最適化問題を対象とする手法が提案されている [9]．しかし，非凸な制約条件をもつ最適化問題が多く存在することから，非凸な制約付き最適化問題に対する段階的最適化を提案する．

本稿では，制約付き問題への拡張のため，平滑化した関数が凸であると仮定し，制約付き凸計画問題に対する KKT 条件を適用する．KKT 条件 (Karush-Kuhn-Tucker condition) にもとづく主双対内点法によって勾配を決定し，準ニュートン法にもとづいて平滑化した関数に対する解を求める．平滑化を緩和しながら解を求めることで，より最適解に近い解を導出する．制約条件として等式制約が与えられる場合，平滑化に用いられる乱数によって等式制約が満足されず，解の収束に時間を要する可能性がある．そこで，等式制約を満たすような乱数の基底を特異値分解によって求め，基底に対する乱数の線型結合によって，関数を平滑化するための乱数を生成する．

1 大阪大学大学院情報科学研究科
Osaka University
a) iwata.daichi@ist.osaka-u.ac.jp
b) samejima@ist.osaka-u.ac.jp
c) sugano@ist.osaka-u.ac.jp
d) yasumat@ist.osaka-u.ac.jp

ⓒ 2018 Information Processing Society of Japan

2. 関連研究

最適化問題は，制約条件を満足しつつ目的関数を最小化または最大化する解 $\boldsymbol{x} \in \mathbb{R}^n$ を求める問題であり，以下のように定式化される．

$$\begin{aligned} \text{Minimize} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad (1 \leq i \leq k) \\ & g_j(\boldsymbol{x}) = 0 \quad (1 \leq j \leq l) \end{aligned} \tag{1}$$

$f_0 : \mathbb{R}^n \to \mathbb{R}$ を目的関数，k 個の不等式制約式と l 個の等式制約式 $f_i, g_j : \mathbb{R}^n \to \mathbb{R}$ を含む条件式を制約条件とよぶ．一つ以上の非凸な目的関数または制約条件をもつ問題を非凸最適化問題とよぶ．

凸最適化問題においては，最適解が満たす必要十分条件として KKT 条件 [10], [11] が知られており，KKT 条件を満たす解を効率よく求めるアルゴリズムとして主双対内点法が知られている．

主双対内点法 KKT 条件を満たす解は，ニュートン法 [12], [13] を適用することによって得られ，この解法をまとめて主双対内点法と呼ぶ．ニュートン法を適用するため，以下の修正 KKT 条件 $\boldsymbol{r}_t(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \boldsymbol{0}$ を利用する．

$$\boldsymbol{r}_t(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \begin{bmatrix} \nabla f_0(\boldsymbol{x}) + Df(\boldsymbol{x})^\top \lambda + \boldsymbol{A}^\top \boldsymbol{\nu} \\ -\mathrm{diag}(\boldsymbol{\lambda}) f(\boldsymbol{x}) - (1/t)\boldsymbol{1} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \end{bmatrix} \tag{2}$$

上式では，等式制約を $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ としており，$f(\boldsymbol{x}), Df(\boldsymbol{x})$ は不等式制約 $f_i(\boldsymbol{x})$ を用いて以下のように与えられる．

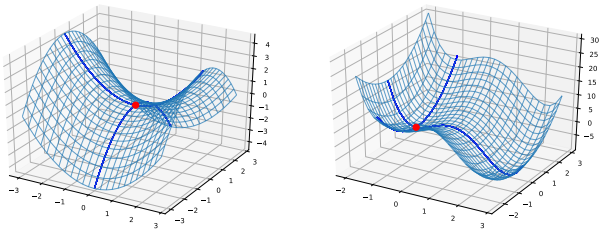
$$f(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_k(\boldsymbol{x}) \end{bmatrix}, \qquad Df(\boldsymbol{x}) = \begin{bmatrix} \nabla f_1(\boldsymbol{x})^\top \\ \vdots \\ \nabla f_k(\boldsymbol{x})^\top \end{bmatrix} \tag{3}$$

式 (2) において，$\boldsymbol{\lambda}$ と $\boldsymbol{\nu}$ は双対変数である．$t\ (\in \mathbb{R}_{++})$ は，目的関数に対して不等式制約を相対的に重視するパラメータであり，ニュートン法の反復にともなって少しずつ増加させる．

$\boldsymbol{r}_t = [\boldsymbol{r}_{dual}, \boldsymbol{r}_{cent}, \boldsymbol{r}_{pri}]^\top$ と置くと，$\boldsymbol{r}_t = \boldsymbol{0}$ を解くためのニュートンステップは $f(\boldsymbol{x}) \prec \boldsymbol{0}$，$\lambda \succ \boldsymbol{0}$ を満たす点 $[\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}]$ において，以下の式を解くことによって得られる．

$$\begin{bmatrix} \nabla^2 f_0(\boldsymbol{x}) + \sum_{i=1}^k \lambda_i \nabla^2 f_i(\boldsymbol{x}) & Df(\boldsymbol{x})^\top & \boldsymbol{A}^\top \\ -\mathrm{diag}(\boldsymbol{\lambda}) Df(\boldsymbol{x}) & -\mathrm{diag}(f(\boldsymbol{x})) & \boldsymbol{0} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \\ \Delta \boldsymbol{\nu} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{r}_{dual} \\ \boldsymbol{r}_{cent} \\ \boldsymbol{r}_{pri} \end{bmatrix} \tag{4}$$

一方，非凸最適化問題においては，KKT 条件を満たす解が必ずしも最適解になるとは限らない．最適解ではないものの，KKT 条件を満たす解として，図 1 に示す鞍点や局所最適解が挙げられる [14]．



[a] 鞍点　　[b] 局所最適解

図 1: 鞍点と局所最適解の例．赤点がそれぞれの図において鞍点と局所最適解を示す．

このため，凸最適化手法をそのまま適用しても，鞍点や局所最適解に陥る解が得られる可能性がある．非凸最適化問題においては，鞍点や局所最適解を回避して，大域的な最適解を得ることが目標となる．

段階的最適化 段階的最適化 [6], [7] は非凸で微分可能な連続的関数に対して，局所最適解の影響を低減しながら大域的最適解を探索する手法である．元の関数を平滑化し，鞍点や局所最適解を回避しやすい関数に変形することによって複雑な非凸関数を扱う．関数を強く平滑化すると凸関数のように見なすことができ，最適解を簡単に決定することができる [15]．平滑化の極端なケースとして凸近似を用いて初期解を定めた後，原問題を解く方法は数多くみられる．しかし段階的最適化では，平滑化を弱めることで徐々に元の関数に近づけ，最適解を追跡することで元の関数の大域的最適解を決定する．段階的最適化の概要を図 2 に示す．

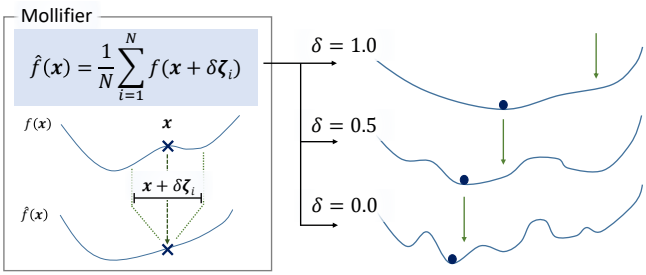


図 2: 1 次元の段階的最適化

段階的最適化において重要なのは，図 2 に示すような段階的な近似が可能な写像 K である．このような写像としてよく用いられるのが以下の Mollifier (軟化子) [8] である．

Mollifier Mollifier は，無限回微分可能であり，積分可能な関数の畳み込みとして振る舞う写像である．関数 $f(\boldsymbol{x})$ が積分可能であるとすると Mollifier K は以下を満たす．パラメータ ϵ は，近似の度合いを調整するために用いられ，ϵ が大きいほど凸に近くなる．

$$\hat{f}(\boldsymbol{x}) = \lim_{\epsilon \to 0} \int \epsilon^{-1} K(\boldsymbol{\theta}/\epsilon) f(\boldsymbol{x} - \boldsymbol{\theta}) d\boldsymbol{\theta} \tag{5}$$

対象の関数が積分可能であっても，解析的な積分が困難であることが多い．そこで，Mollifier に無限回微分可能なガウス関数を選択し，モンテカルロ積分を行うことによって近似する方法が提案されている．サンプリング数 N，スムースパラメータ δ，標準正規乱数 $\boldsymbol{\xi}_i$ とすると以下の式で関数を平滑化することができる．

$$\hat{f}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x} + \delta\boldsymbol{\xi}_i) \tag{6}$$

元の関数に正規乱数から得られたノイズを加え，いくつかのサンプルをとり，その平均をとることで関数の平滑化を行っている．スムースパラメータ δ はノイズの大きさを変えることで，平滑化の量を制御している．つまり，δ が大きいほど関数は強く平滑化され，δ が小さいほど元の関数に近いかたちになるように平滑化される．δ を徐々に小さくすることで，段階的最適化を実現することができる．

段階的最適化は，これまで無制約の最適化問題を対象としてきた．制約付きの最適化問題に対しては，制約条件のラグランジュ緩和によって制約なしの最適化問題に近似し，段階的最適化が行われている．制約条件を緩和することなく段階的最適化を行うことができれば，より良い解を得られる可能性がある．

3. 提案手法

本論文では，式 (1) で示す制約付き非凸最適化問題を対象とし，Mollifier による段階的最適化を行う方法を提案する．

3.1 前提条件

Mollifier によって近似された問題を従来の凸最適化手法である主双対内点法 [16], [17] を用いて解くため，以下の前提条件を置く．

関数の微分に関する前提条件 関数 $f_i(\boldsymbol{x})$ および $g_j(\boldsymbol{x})$ は連続 2 回微分可能な関数とする．

非凸制約条件に関する前提条件 主双対内点法では，等式制約 $g_j(\boldsymbol{x}) = 0$ において，$g_j(\boldsymbol{x})$ が非凸である場合に式 (4) を解析的に解くことができない．そこで，非凸な等式制約 $g_j(\boldsymbol{x}) = 0$ は以下の不等式制約に分解して扱う．

$$g_j(\boldsymbol{x}) = 0 \Leftrightarrow g_j(\boldsymbol{x}) \leq 0, g_j(\boldsymbol{x}) \geq 0 \tag{7}$$

以下では，等式制約に含まれる関数がアフィンであるものを特に $g_j(\boldsymbol{x}) = 0$ と表現し，アフィンでない関数を含む非凸な等式制約は，不等式制約に分解し，$f_i(\boldsymbol{x})$ による不等式制約 $f_i(\boldsymbol{x}) \leq 0$ に含めるものとする．

3.2 KKT 条件による制約付き最適化問題への拡張

Mollifier を用いた非凸最適化手法を，KKT 条件を用いて制約付き最適化問題に拡張する手法を説明する．Mollifier を用いることで，非凸な関数 f_i $(i = 0, \cdots, n)$ は凸に近い関数 $\hat{f}_i$ に近似できることから，$\hat{f}_i$ に対して KKT 条件を適用することで制約付き非凸最適化問題へと拡張することができる．そのアルゴリズムを以下に示す．

(1) 式 (3) の $f(\boldsymbol{x}) \prec \boldsymbol{0}$ を満たす初期解 $\boldsymbol{x}_0$，式 (6) のスムースパラメータ δ とサンプル数 N を設定する．Mollifier として正規乱数 $\boldsymbol{\xi}$ を生成する．
(2) 関数 f に対して，式 (6) を用いることで平滑化された関数 $\hat{f}$ を得る．
(3) Mollifier によって平滑化された関数 $\hat{f}$ に対して，KKT 条件の最適解 $\boldsymbol{x}_i$ を求める．
  (a) $\boldsymbol{\lambda} \succ \boldsymbol{0}$ を設定する．
  (b) 関数 $\hat{f}$ に対して式 (4) を解くことで $[\Delta\boldsymbol{x}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\nu}]$ を求める．
  (c) 直線探索を行い，適切なステップ幅 $s > 0$ を決定する．$\boldsymbol{x} = \boldsymbol{x} + s\Delta\boldsymbol{x}$，$\boldsymbol{\lambda} = \boldsymbol{\lambda} + s\Delta\boldsymbol{\lambda}$，$\boldsymbol{\nu} = \boldsymbol{\nu} + s\Delta\boldsymbol{\nu}$ としてそれぞれの変数を更新する．
  (d) $[\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}]$ が十分収束するまでステップ (b)，(c) を繰り返す．
(4) 得られた $\boldsymbol{x}_i$ を次のループの初期解とする．スムースパラメータ δ を小さくすることで Mollifier を弱めながら再び (2) へ戻り，数回繰り返す．

上記のアルゴリズムで示す主双対内点法における直線探索にはバックトラック法を用いる．ある点を $[\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}]$ としたとき，次の探索点 $[\boldsymbol{x}^+, \boldsymbol{\lambda}^+, \boldsymbol{\nu}^+]$ をステップ幅 s を用いて以下のように表現する．

$$\boldsymbol{x}^+ = \boldsymbol{x} + s\Delta\boldsymbol{x}, \boldsymbol{\lambda}^+ = \boldsymbol{\lambda} + s\Delta\boldsymbol{\lambda}, \boldsymbol{\nu}^+ = \boldsymbol{\nu} + s\Delta\boldsymbol{\nu} \tag{8}$$

まず始めに，$\boldsymbol{\lambda}^+ \succ \boldsymbol{0}$ を満たすような 1 を超えない最大となる正のステップ長 s を計算する．

$$\begin{aligned} s^{max} &= \sup\{s \in [0,1] | \boldsymbol{\lambda} + s\Delta\boldsymbol{\lambda} \succeq \boldsymbol{0}\} \\ &= \min\{1, \min\{-\boldsymbol{\lambda}_i / \Delta\boldsymbol{\lambda}_i | \Delta\boldsymbol{\lambda}_i < \boldsymbol{0}\}\} \end{aligned} \tag{9}$$

$s = 0.99s^{max}$ としてバックトラック法を行う．以下の式を満たすまで s を小さくしながら計算を繰り返す．

$$||\boldsymbol{r}_t(\boldsymbol{x}^+, \boldsymbol{\lambda}^+, \boldsymbol{\nu}^+)||_2 \leq (1 - \alpha s)||\boldsymbol{r}_t(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})||_2 \tag{10}$$

α は一般的に 0.01 から 0.1 の間から選ばれる．s を小さくし，初めて式 (10) を満たすとき，ステップ幅 s として採用する．

式 (4) を用いて KKT 条件を満たす解を求める際，右辺の要素が小さくなるように各ニュートンステップを決定する．イテレーションの終了条件は以下のとおりである．

$$(||\boldsymbol{r}_{pri}||_2^2 + ||\boldsymbol{r}_{dual}||_2^2)^{1/2} \leq \epsilon \tag{11}$$

ϵ は十分小さな値である．主残差と双対残差が十分小さくなったとき，最適解を決定したとしてイテレーションを終了する．

3.3 等式制約にもとづく Mollifier の調整

制約条件のない非凸最適化問題に対する Mollifier として，正規乱数がこれまで用いられてきた．しかし，制約条件付きの非凸最適化問題に拡張した場合，生成した正規乱数が必ずしも等式制約を満たすとは限らない．このため等式制約の主残差 $\boldsymbol{r}_{pri}$ が大きくなり，式 (11) の終了条件が満たされず，収束に時間がかかる可能性がある．

そこで，等式制約を満たすような正規乱数を発生させ，式 (11) から等式制約を満たすような Mollifier を使用することで収束が早くなる可能性が考えられる．最適化問題において等式制約 $\boldsymbol{Ax} = \boldsymbol{b}$ が与えられたとき，Mollifier によって解 $\boldsymbol{x}$ に乱数 $\boldsymbol{\xi}$ が加えられるため，以下の等式制約を見たす必要がある．

$$\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{\xi}) = \boldsymbol{b} \tag{12}$$

変数 $\boldsymbol{x}$ が等式制約 $\boldsymbol{Ax} = \boldsymbol{b}$ を満たすのであれば，乱数 $\boldsymbol{\xi}$ は以下の条件を満たす必要がある．

$$\boldsymbol{A\xi} = \boldsymbol{0} \tag{13}$$

これは同次線形方程式であり，$\boldsymbol{\xi} \neq \boldsymbol{0}$ となる解は特異値分解によって求めることができる．行列 $\boldsymbol{A}$ を特異値分解することで $\boldsymbol{A\xi} = \boldsymbol{0}$ を満たす $\boldsymbol{\xi}$ がいくつか得られる．それらを $\boldsymbol{\xi}_i$ $(1 \leq i \leq n)$ とすると，$\boldsymbol{\xi}_i$ を用いて式 (12) を満たすような Mollifier $\hat{\boldsymbol{\xi}}$ を以下の式のように生成することができる．

$$\hat{\boldsymbol{\xi}} = a_1\boldsymbol{\xi}_1 + a_2\boldsymbol{\xi}_2 + \cdots + a_{n-1}\boldsymbol{\xi}_{n-1} + a_n\boldsymbol{\xi}_n \tag{14}$$

a_i $(1 \leq i \leq n)$ は標準正規乱数である．それぞれの $\boldsymbol{\xi}_i$ は直交しており，それらを線形結合しても式 (13) を満たしている．したがって，式 (14) を用いることで制約を満たすような重みづけされた正規乱数を発生させることができる．

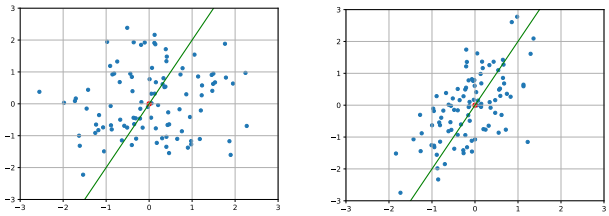
しかし，等式制約を完全に満たすような Mollifier では関数を十分に平滑化することができず，局所解に陥ってしまう可能性がある．局所解を回避するため，提案手法では以下の処理を行う．

(1) Mollifier の値が，等式制約から外れることを許容するため，ランダムな摂動 $\boldsymbol{\alpha}$ を加える．摂動 $\boldsymbol{\alpha}$ は正規乱数とし，Mollifier の分散が 1 になるように摂動の分散を決定する．

$$\hat{\boldsymbol{\xi}} = a_1\boldsymbol{\xi}_1 + a_2\boldsymbol{\xi}_2 + \cdots + a_{n-1}\boldsymbol{\xi}_{n-1} + a_n\boldsymbol{\xi}_n + \boldsymbol{\alpha} \tag{15}$$

(2) ニュートンステップが極端に小さくなったときは局所解に陥ったとみなして，正規乱数のみ用いた Mollifier を使用する．

図 3 に等式制約に対するサンプリングの例を示す．従来の Mollifier では，式 (6) によって N 点をサンプリングする際，制約条件に関係なくサンプリングがなされていたため，図 3 [a] に示すように等式制約から大きく外れた値もサンプリングされている．しかし，等式制約によって調整された式 (15) の Mollifier を使用することで，図 3 [b] のように制約に沿ったサンプリングをすることが可能になる．制約条件に沿った Mollifier を使用することで等式制約が満たされ，式 (4) に対する準ニュートン法において収束が早まることが期待される．



[a] 通常の正規乱数　　[b] 式 (15) によって重みづけされた乱数

図 3: 等式制約 $y = 2x$ のもとで原点を中心にサンプリングした例

3.4 Mollifier による制約付き非凸最適化アルゴリズム

3.2 節と 3.3 節を踏まえて，Mollifier による制約付き非凸最適化アルゴリズムを Algorithm 1 にまとめる．まず，最適化問題における凸性を判定し，凸関数と非凸関数を分けておく．最適化の初期解は主双対内点法で一般的な等式制約の解を利用するが，等式制約が存在しない場合は解 $\boldsymbol{x}$ の上界値と下界値の間の乱数で決定する．等式制約の関数はアフィンであることを前提としているため，線形システムに対する従来の解法を適用して初期解を求める．大きなスムースパラメータ δ からはじめ，あらかじめ特定した非凸関数の Mollifying を行う．Mollifying 後の最適化問題に対して，修正 KKT 条件にもとづいて KKT 方程式を構築し，ニュートンステップ $\Delta\boldsymbol{x}$ を求める．ニュートンステップによる解の更新を一定回数 T だけ行い，徐々にスムースパラメータ δ を減少させることで解 $\boldsymbol{x}^*$ を求める．

4. 評価実験

提案手法の有効性を示すため，実際の非凸最適化問題である Shape-from-Shading を対象として評価実験を行った．Shape-from-Shading [18], [19] とは図 4 のように一つの輝度画像から元の物体の 3 次元形状を推測する問題である．4.1 節に Shape-from-Shading の定式化，4.2 節に評価結果を示す．

4.1 定式化

ピクセル数 p の画像に対して，観測輝度ベクトル $\boldsymbol{m} \in \mathbb{R}^p$，光源方向ベクトル $\boldsymbol{l} \in \mathbb{R}^3$ が与えられているとし，$\|\boldsymbol{l}\|_2 = 1$ を

Algorithm 1 制約付き非凸最適化アルゴリズム

Input: 最適化問題 $P = \{f_0(\boldsymbol{x}), f_i(\boldsymbol{x}) \leq 0, g_j(\boldsymbol{x}) = 0, 1 \leq i \leq k, 1 \leq j \leq l\}$, Mollifying のサンプリング回数 N, ニュートン法の実行回数 T, スムースパラメータ δ, スムースパラメータの低減速度 $\Delta\delta$
Output: 近似解 $\boldsymbol{x}^*$

```
//凸関数と非凸関数のインデックスの集合 C, D
C ← convex_funcs(P)
D ← non_convex_funcs(P)
//初期解の決定
if l ≥ 1 then
  x* ← solve_LS(g_j(x), 0)
else
  x* ← rand(n)
end if
while δ ≥ 0 do
  for t = 0 : T do
    for d ∈ D do
      if l ≥ 1 then
        //等式制約があるときは Mollifier を調整
        U, Σ, V^⊤ ← SVD(A)
        Γ = [v_1, v_2, ···, v_w]^⊤ ← lower(V)
        r ~ N_w(0, σ^2) ∈ R^w
        ξ̂ ← r^⊤ Γ
      else
        ξ̂ ~ N_n(0, σ^2) ∈ R^n
      end if
      //x = x* における Mollifying
      f̂_d(x*) ← 1/N Σ_{s=1}^N f_d(x* + δξ̂)
    end for
    //Mollifying 後の最適化問題
    P̂ ← {f_c(x*), f_d(x*), g_j(x*), c ∈ C, d ∈ D}
    Δx ← solve_modifiedKKT(P̂)
    //x = x* におけるニュートンステップで更新
    x* ← x* + Δx
    t ← t + 1
  end for
  δ ← δ − Δδ
end while
```

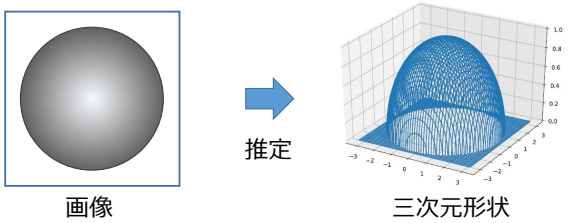


図 4: 半球の輝度画像に対する Shape-from-Shading の例

満たすとする．求める三次元物体の法線マップを $\boldsymbol{N} \in \mathbb{R}^{3\times p}$ とすると，理想的な拡散反射モデルであるランバート反射は以下の式で表現される．

$$\boldsymbol{m}^\top = \boldsymbol{l}^\top \boldsymbol{N} \tag{16}$$

1 枚の輝度画像からは高々 p 本の式しか生成されないため，$3p$ 個の未知変数で構成される法線マップ $\boldsymbol{N}$ を決定することはできず，少なくとも 3 枚の輝度画像が必要となる．

Shape-from-Shading では，対象物体に関していくつかの仮定を設けることで 1 枚の輝度画像からの法線マップの推定を可能にする．以下では，仮定を定式化した制約条件について述べる．

輝度の制約 ランバート反射モデルを仮定したとき，観測輝度 $\boldsymbol{m}$，光源方向ベクトル $\boldsymbol{l}$，法線マップ $\boldsymbol{N}$ は以下の制約条件に従う．

$$\boldsymbol{l}^\top \boldsymbol{N} - \boldsymbol{m}^\top \to \boldsymbol{0} \tag{17}$$

スムース性の制約 物体の表面では形状が急激に変化することはなく，それぞれの部分においてスムースである必要がある．ラプラシアン行列を $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ とすると，スムース性の制約は以下のように表せる．

$$\boldsymbol{N}\boldsymbol{D} \to \boldsymbol{0} \tag{18}$$

境界の制約 物体の境界上において，法線は境界に対して垂直であり外側に向いていると仮定する．境界に位置するピクセルを示す行列を $\boldsymbol{F} \in \mathbb{R}^{p\times p}$ とする．境界上のピクセルの法線をもつ行列を $\boldsymbol{G} \in \mathbb{R}^{3\times p}$ とする．行列 $\boldsymbol{F}$ の要素が 1 であればそのピクセルが境界上であることを意味し，それ以外では 0 となる．i 番目のピクセルが境界上であるとき $F_{i,i} = 1$ となり，$\boldsymbol{G}$ の列ベクトル $\boldsymbol{g}_i = \boldsymbol{n}_i$ となる．i 番目のピクセルが境界上にない時 $F_{i,i} = 0$，$\boldsymbol{g}_i = \boldsymbol{0}$ となる．境界の制約は以下のように表せる．

$$\boldsymbol{N}\boldsymbol{F} - \boldsymbol{G} \to \boldsymbol{0} \tag{19}$$

単位ベクトルの制約 表面上の法線はそれぞれ単位ベクトルである必要がある．各ピクセルの法線ベクトルを $\boldsymbol{n} \in \mathbb{R}^3$ とすると以下の制約を持つ．

$$\|\boldsymbol{n}_i\|_2^2 = 1, \forall i \in \{1, \cdots, p\} \tag{20}$$

式 (20) の等式制約は非凸であるため，式 (7) で示すように不等式制約に分解する．

$$\|\boldsymbol{n}_i\|_2^2 \leq 1, \|\boldsymbol{n}_i\|_2^2 \geq 1, \forall i \in \{1, \cdots, p\} \tag{21}$$

これらの制約を用いて，Shape-from-Shading の最適化問題を以下のように定式化できる．

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|\boldsymbol{N}\boldsymbol{D}\|_F^2 \\ \text{s.t} \quad & \boldsymbol{l}^\top \boldsymbol{N} - \boldsymbol{m}^\top = \boldsymbol{0} \\ & \boldsymbol{N}\boldsymbol{F} - \boldsymbol{G} = \boldsymbol{0} \\ & \|\boldsymbol{n}_i\|_2^2 - 1 \leq 0, \|\boldsymbol{n}_i\|_2^2 - 1 \geq 0, \forall i \in \{1, \cdots, p\} \end{aligned} \tag{22}$$

4.2 評価結果

半球状の物体を撮影した 20×20 ピクセルと 30×30 ピクセルの画像に対する Shape-from-Shading において，以

下の手法を実装し，実験結果を比較した．

- 提案手法
- Mollifier において $\boldsymbol{\alpha} = \boldsymbol{0}$ とした提案手法 (摂動なし)
- 等式制約で Mollifier を調整しない提案手法 (調整なし)
- 非凸な等式制約を凸な不等式制約とする凸近似手法

20×20 ピクセルの画像を対象としたときの結果を表 1 に，30×30 ピクセルの画像を対象としたときの結果を表 2 にまとめる．表 1 と表 2 が示す各数値は，それぞれの制約に対するフロベニウスノルムを表す．真値誤差は真値の法線マップ $\boldsymbol{N}_{true}$ と得られた解のフロベニウスノルムとした．

表 1: Shape-from-Shading 最適化問題を解いて得られた解の各制約と実行時間 (20×20 ピクセル)

	提案手法	提案手法 (摂動なし)	提案手法 (調整無し)	凸近似手法
$\boldsymbol{l}^\top \boldsymbol{N} - \boldsymbol{m}^\top$	3.67e-3	3.65e-3	3.91e-3	6.05e-5
$\boldsymbol{ND}$	0.644	0.643	0.644	0.483
$\boldsymbol{NF} - \boldsymbol{G}$	2.97e-4	3.13e-4	3.36e-4	5.51e-5
$\sum_N (\lVert \boldsymbol{n}_i \rVert_2 - 1)^2$	9.99e-7	1.14e-6	1.01e-6	1.27
実行時間 $[s]$	223	210	221	0.03

表 2: Shape-from-Shading 最適化問題を解いて得られた解の各制約と実行時間 (30×30 ピクセル)

	提案手法	提案手法 (摂動なし)	提案手法 (調整無し)	凸近似手法
$\boldsymbol{l}^\top \boldsymbol{N} - \boldsymbol{m}^\top$	1.43e-3	3.15e-3	1.46e-3	5.07e-5
$\boldsymbol{ND}$	0.437	0.449	0.437	0.313
$\boldsymbol{NF} - \boldsymbol{G}$	9.96e-5	1.87e-4	9.81e-5	3.97e-5
$\sum_N (\lVert \boldsymbol{n}_i \rVert_2 - 1)^2$	3.46e-6	1.02e-3	3.06e-6	3.62
実行時間 $[s]$	3993	3752	4146	0.09

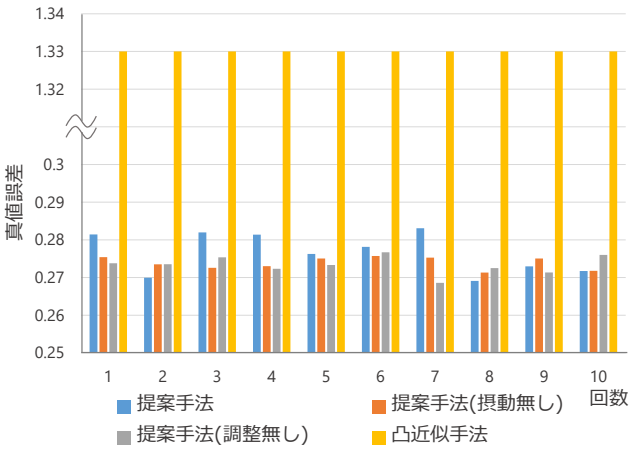


図 5: 10 回の試行における真値誤差 (20×20 ピクセル)

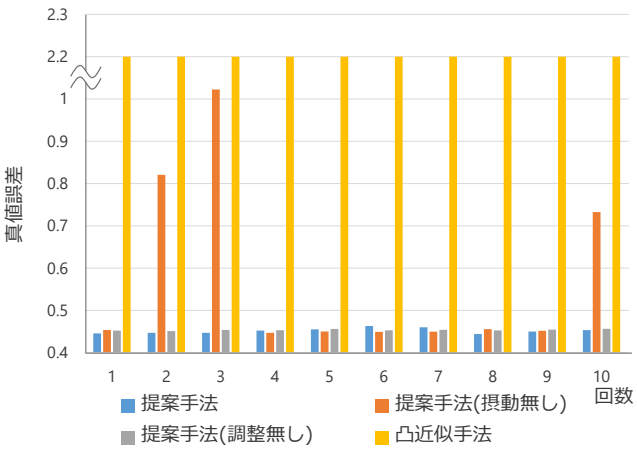


図 6: 10 回の試行における真値誤差 (30×30 ピクセル)

表 3: 真値誤差の平均，中央値と分散 (20×20 ピクセル)

	提案手法	提案手法 (摂動なし)	提案手法 (調整無し)	凸近似手法
平均	0.275	0.273	0.274	1.33
中央値	0.275	0.274	0.274	1.33
分散	2.18e-5	4.56e-06	6.99e-5	0.00

表 4: 真値誤差の平均，中央値と分散 (30×30 ピクセル)

	提案手法	提案手法 (摂動なし)	提案手法 (調整無し)	凸近似手法
平均	0.453	0.573	0.453	2.19
中央値	0.453	0.452	0.450	2.19
分散	8.57e-6	3.93e-2	4.17e-5	0.00

図 5 と図 6 は，20×20 ピクセルと 30×30 ピクセルの画像を対象として 10 回解いたときの真値誤差の分布である．これらの平均，中央値，分散を表 3，表 4 に示す．

まず，提案手法と凸近似手法の比較を行う．表 1 より，20×20 ピクセルにおいて両手法ともに輝度の制約条件 (17)，境界の制約条件 (19) を十分に満たしていることがわかる．提案手法では，非凸制約である単位ベクトルに関する制約条件 (20) を精度よく満たす解を得られており，凸近似手法に比べ真値誤差も小さくなっている．スムース性の制約条件 (18) においては，凸近似手法が良い結果を得られているが，凸近似手法では非凸な制約条件である式 (20) を $\lVert \boldsymbol{n}_i \rVert_2^2 \leq 1$ $(1 \leq i \leq p)$ として緩和しているため，非凸制約を満たさない解が式 (18) の残差を小さくしていると考えられる．

実行時間に関して比較すると，凸近似法は解析的に問題を解くため非常に高速である一方，反復的に解く提案手法では多くの計算時間がかかっている．具体的には，各イテレーションにおいて，ニュートンステップを決定するために KKT 条件式 (4) を解く時間ならびに関数の微分に時間

ⓒ 2018 Information Processing Society of Japan

を要する．特に，扱う次元数が大きくなるにつれ，他の手法よりも多くの時間が必要である．表 2 より，30×30 ピクセルにおいても 20×20 ピクセルのときと同様の結果が確認される．

提案手法，提案手法 (摂動無し)，提案手法 (調整無し) を比較すると，図 6 の 2 回目と 3 回目の試行において，30×30 ピクセルにおいて提案手法 (摂動無し) が局所解に陥っていることがわかる．一方で，提案手法では摂動を加えたことで局所解を回避できている．このことから，扱う変数の規模が大きくなると，等式制約だけで調整した Mollifier では十分に平滑化することができず，局所解に陥いる可能性があると考えられる．実行時間に注目すると，30×30 ピクセルにおいて，提案手法と提案手法 (摂動無し) の収束が速くなっている．提案手法において等式制約を満たすように Mollifier を調整したことで，KKT 条件を解くための時間とイテレーション回数が減少し実行時間が短縮されたと考えられる．

5. 結論

本稿では，制約付きの非凸最適化問題を取り扱った．従来の Mollifier を用いた段階的な非凸最適化手法では，制約なしの最適化問題を取り扱っていた．Mollifier を用いて非凸な関数を平滑化することで凸関数のように扱うことができ，凸最適化手法を応用できる可能性がある．

そこで，制約付きの凸最適化問題の最適解を決定できる KKT 条件を用いることで，Mollifier を用いた非凸最適化手法を拡張し制約付き問題も扱えるような手法を提案した．また，単純に正規乱数を Mollifier として利用する方法では制約条件を満たさない領域からのサンプリングが行われ，効果的な平滑化が行われない可能性があった．そこで等式制約を用いて Mollifier を調整する方法を提案した．

提案手法の有効性を評価するために，一枚の輝度画像から三次元形状を推測する Shape-from-Shading のタスクを行った．評価実験では，20×20 ピクセルと 30×30 ピクセルの三次元形状の推測を行った．どちらにおいても，得られた解の精度から KKT 条件によって拡張した提案手法の有効性を確認することができた．等式制約で Mollifier を調整することによる収束速度の向上は，20×20 ピクセルにおいてあまり確認できなかったが，次元数の大きくなった 30×30 ピクセルでは有効性が確認できた．また，等式制約を完全に満たすような Mollifier では必ずしも良い結果を得られないことも確認された．

今後の課題として，より効果的な Mollifier の調整方法を検討し，提案手法の改善させることや，提案手法が有効な問題の特性を調べることが挙げられる．

参考文献

[1] Ecker, A. and Jepson, A. D.: Polynomial shape from shading, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 145–152 (2010).

[2] Gallardo, M., Collins, T. and Bartoli, A.: Dense Non-rigid Structure-from-Motion and Shading with Unknown Albedos, *Proceedings of IEEE International Conference on Computer Vision*, pp. 3904–3912 (2017).

[3] Chesi, G., Garulli, A., Vicino, A. and Cipolla, R.: Estimating the fundamental matrix via constrained least-squares: A convex approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 397–401 (2002).

[4] Bertsekas, D. P.: *Nonlinear programming*, Athena scientific, Nashua, NH, USA (1999).

[5] Poljak, S., Rendl, F. and Wolkowicz, H.: A recipe for semidefinite relaxation for (0, 1)-quadratic programming, *Journal of Global Optimization*, Vol. 7, No. 1, pp. 51–73 (1995).

[6] Allgower, E. L. and Georg, K.: *Introduction to Numerical Continuation Methods*, SIAM, Philadelphia, PA, USA (2003).

[7] Mobahi, H. and Fisher III, J. W.: A Theoretical Analysis of Optimization by Gaussian Continuation., *Proceedings of AAAI conference on Artificial Intelligence*, pp. 1205–1211 (2015).

[8] Gulcehre, C., Moczulski, M., Visin, F. and Bengio, Y.: Mollifying Networks, *Proceedings of International Conference on Learning Representation* (2017).

[9] Hazan, E., Levy, K. Y. and Shalev-Shwartz, S.: On Graduated Optimization for Stochastic Non-Convex Problems, *Proceedings of International Conference on Machine Learning*, pp. 1833–1841 (2016).

[10] Vanderbei, R. J. et al.: *Linear programming*, Springer (2015).

[11] Boyd, S. and Vandenberghe, L.: *Convex Optimization*, Cambridge University Press, New York, NY, USA (2004).

[12] Kelley, C. T.: *Solving nonlinear equations with Newton's method*, SIAM, Philadelphia, PA, USA (2003).

[13] Hoffman, J. D. and Frankel, S.: *Numerical methods for engineers and scientists*, CRC Press (2001).

[14] Fiacco, A. V. and McCormick, G. P.: *Nonlinear programming: sequential unconstrained minimization techniques*, SIAM, Philadelphia, PA, USA (1990).

[15] Chen, X.: Smoothing methods for nonsmooth, nonconvex minimization, *Mathematical Programming*, Vol. 134, No. 1, pp. 71–99 (2012).

[16] Yamashita, H.: A globally convergent primal-dual interior point method for constrained optimization, *Optimization Methods and Software*, Vol. 10, No. 2, pp. 443–469 (1998).

[17] Ye, Y.: *Interior Point Algorithms: Theory and Analysis*, John Wiley & Sons, Inc., New York, NY, USA (1997).

[18] Zhang, R., Tsai, P.-S., Cryer, J. E. and Shah, M.: Shape-from-shading: a survey, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 21, No. 8, pp. 690–706 (1999).

[19] Ramachandran, V. S.: Perception of shape from shading, *Nature*, Vol. 331, No. 6152, pp. 163–166 (1988).

ⓒ 2018 Information Processing Society of Japan