

УДК 519.6

doi 10.26089/NumMet.v16r334

МЕТОД ИСКЛЮЧЕНИЯ ИЗБЫТОЧНЫХ ОГРАНИЧЕНИЙ В ЗАДАЧЕ ВОССТАНОВЛЕНИЯ ТЕЛА ПО ИЗМЕРЕНИЯМ ЕГО ОПОРНОЙ ФУНКЦИИ

И. А. Палачев¹

Предложен новый алгоритм восстановления тел по измерениям их опорных функций, который представляет собой алгоритм квадратичного или линейного программирования в форме Гарднера–Кидерлена с меньшим числом ограничений. Уменьшение числа ограничений достигается за счет нового метода, который позволяет исключить из исходной системы ограничений часть ограничений как избыточные. Предложен новый подход, позволяющий применять методы восстановления тел по измерениям опорной функции к задаче восстановления тел по теневым контурам. Представлено описание реализации алгоритма, а также результаты его тестирования на реальных промышленных теневых контурах. Предложенный метод в рассмотренном примере позволил сократить число ограничений на 80% и ускорить исходный алгоритм Гарднера–Кидерлена на порядок.

Ключевые слова: опорная функция, восстановление геометрических тел, линейное программирование, квадратичное программирование, теневой контур, преобразование двойственности.

1. Введение. Поводом для данного исследования послужила задача восстановления трехмерных тел по теневым контурам, которая возникает при построении моделей драгоценных камней в ювелирной промышленности. Теневые контуры получаются с помощью фотографирования камня, стоящего на вращающейся подставке. В результате обработки растрового изображения (фотографии) получается теневой контур — многоугольник, являющийся границей тени камня на некоторой вертикальной плоскости.

При изучении этого вопроса было замечено его удивительное сходство с другими задачами, имевшими совершенно иное прикладное происхождение и, как оказалось, изучавшимися в иностранной научной литературе в течение последних почти 30 лет. Ниже будет приведен краткий обзор этих задач и методов их решения.

Настоящая статья преследует двоякую цель. Во-первых, показать что методы восстановления геометрических тел по измерениям опорной функции применимы к исходной прикладной задаче, т.е. что теневые контуры можно интерпретировать как набор измерений опорной функции тела. Во-вторых, доказать свойство одного из методов, позволяющее значительно ускорить алгоритм решения задачи.

2. Основные определения. Как известно (см., например, [1]), всякое выпуклое тело в \mathbb{R}^n однозначно определяется своей опорной функцией.

Определение 1. Опорная функция $h_K : \mathbb{R}^n \rightarrow \mathbb{R}_+$ выпуклого тела $K \subset \mathbb{R}^n$ определяется следующим образом:

$$h_K(u) = \sup_{x \in K} (x, u).$$

Поскольку опорная функция является мультипликативной, ее часто определяют только на единичной сфере $S^{n-1} \subset \mathbb{R}^n$. Если тело K содержит начало координат O в своей внутренности, то его опорная функция положительна.

Допустим, что имеется некоторое реальное физическое тело K , для которого получены приближенные значения h_1, h_2, \dots, h_m опорной функции h_K на *конечном наборе* единичных векторов u_1, u_2, \dots, u_m :

$$h_i = h_K(u_i) + \varepsilon_i, \quad i = 1, 2, \dots, m.$$

Здесь ε_i — погрешность измерения величины h_i . Для краткости здесь и далее будем называть векторы u_i *опорными направлениями*, h_i — *опорными числами*, а плоскости, задаваемые уравнениями $(x, u_i) = h_i$, — *опорными плоскостями*.

Определение 2. Опорные числа h_1, h_2, \dots, h_m по направлениям u_1, u_2, \dots, u_m называются согласованными, если существует такое выпуклое тело $K^* \subset \mathbb{R}^n$, что $h_{K^*}(u_i) = h_i$.

¹ Московский государственный университет им. М. В. Ломоносова, механико-математический факультет, Ленинские горы, 119899, Москва; аспирант, e-mail: palachev.ilya@yandex.ru

Оказывается, не всякий набор вещественных чисел является набором согласованных опорных чисел (рис. 1). Чтобы показать, насколько это важно, определим, следуя [2], так называемый *основной объект* — тело, которое можно восстановить из опорных чисел, если о нем не известны никакие другие априорные данные.

Определение 3. Основным объектом опорных чисел h_1, h_2, \dots, h_m по направлениям u_1, u_2, \dots, u_m называется многогранник K^0 , определяемый как пересечение полупространств, ограниченных опорными плоскостями и содержащих начало координат O :

$$K^0 = \bigcap_{i=1}^m \{ (x, u_i) \leq h_i \}.$$

В случае если одно из опорных чисел было получено с большой погрешностью, то может оказаться, что заметная часть опорных чисел не будут учтена при построении основного объекта. Так возникает вопрос *построения оценки опорной функции* — такого набора согласованных опорных чисел $h^* = (h_1^*, h_2^*, \dots, h_m^*)$, который был бы в некотором смысле ближайшим к набору исходных чисел $h^0 = (h_1^0, h_2^0, \dots, h_m^0)$, т.е. являлся бы решением следующей задачи минимизации:

$$\|h - h^0\|_X \rightarrow \inf, \quad h \in \mathfrak{C}, \tag{1}$$

где X — некоторая метрика в \mathbb{R}^n , а $\mathfrak{C} \subset \mathbb{R}^m$ — множество всех согласованных наборов опорных чисел по направлениям u_1, u_2, \dots, u_m .

3. Исторический обзор. Описанный выше подход для двумерного случая был разработан в статьях Prince и Willsky [2], изучавшими данные из компьютерной томографии, и Lele, Kulkarni и Willsky [3], изучавшими данные, полученные из лазерного радара. В них доказывается так называемая опорная теорема.

Теорема 1. Набор опорных чисел $h = (h_1, h_2, \dots, h_m)$ по направлениям $u_i = (\cos \theta_i, \sin \theta_i) \in \mathbb{R}^2$, $i = 1, \dots, m$, является согласованным тогда и только тогда, когда $Ch \geq 0$, где C — опорная матрица:

$$\begin{pmatrix} -\sin(\theta_2 - \theta_m) & \sin(\theta_1 - \theta_m) & 0 & \dots & 0 & 0 & \sin(\theta_2 - \theta_1) \\ \sin(\theta_3 - \theta_2) & -\sin(\theta_3 - \theta_1) & \sin(\theta_2 - \theta_1) & \dots & 0 & 0 & 0 \\ 0 & \sin(\theta_4 - \theta_3) & -\sin(\theta_4 - \theta_2) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\sin(\theta_{m-1} - \theta_{m-3}) & \sin(\theta_{m-2} - \theta_{m-3}) & 0 \\ 0 & 0 & 0 & \dots & \sin(\theta_m - \theta_{m-1}) & -\sin(\theta_m - \theta_{m-2}) & \sin(\theta_{m-1} - \theta_{m-2}) \\ \sin(\theta_m - \theta_{m-1}) & 0 & 0 & \dots & 0 & \sin(\theta_1 - \theta_m) & -\sin(\theta_1 - \theta_{m-1}) \end{pmatrix}.$$

Таким образом, множество всех согласованных наборов опорных чисел представляет собой конус, называемый *опорным конусом*: $\mathfrak{C} = \{h \in \mathbb{R}^m \mid Ch \geq 0\}$.

Следовательно, задача минимизации (1) является задачей квадратичного программирования в метрике L_2 и задачей линейного программирования в метрике L_1 : $\|h - h^0\|_X \rightarrow \inf, \quad Ch \geq 0$.

Авторы публикаций [2, 3] приводят результаты решения этих задач минимизации, проводят анализ их быстродействия, а также предлагают подходы, позволяющие учитывать априорные знания о восстанавливаемых телах (такие как, например, кривизна).

Более сложным является рассмотрение задачи в трехмерном случае. Аналог теоремы 1 для трехмерного случая был доказан Karl, Kulkarni, Verghese и Willsky [4].

Теорема 2. Набор опорных чисел $h = (h_1, h_2, \dots, h_m)$ по направлениям $u_1, u_2, \dots, u_m \in \mathbb{R}^3$ является согласованным тогда и только тогда, когда неравенство

$$\begin{vmatrix} h_i & u_i^T \\ h_j & u_j^T \\ h_k & u_k^T \\ h_l & u_l^T \end{vmatrix} \begin{vmatrix} 1 & u_i^T \\ 1 & u_j^T \\ 1 & u_k^T \\ 1 & u_l^T \end{vmatrix} \geq 0. \tag{2}$$

выполнено для всех четверок индексов $\{i, j, k, l\}$, таких, что сферический треугольник, образованный векторами u_i, u_j, u_k , содержит в своей внутренней точке вектор u_l и не содержит никаких других опорных направлений из u_1, \dots, u_m (такие четверки называются локальными положительными конусами).

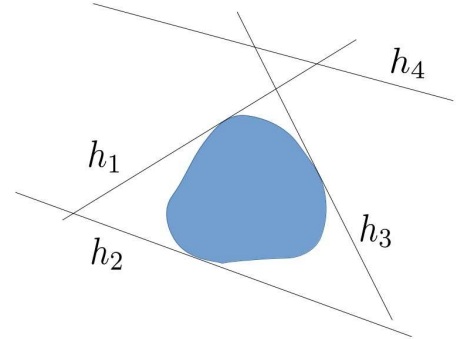


Рис. 1. Пример несогласованных опорных чисел

Таким образом, в трехмерном случае множество \mathfrak{C} всех согласованных опорных чисел представляет собой опорный конус, задаваемый соотношением $Ch \geq 0$, в котором строки опорной матрицы C составлены из коэффициентов уравнения (2).

Авторы не доказали в статье [4] оценок количества условий вида (2), однако провели численный эксперимент по построению условий для случайно выбранных опорных направлений. Из его результатов можно предположить, что для случайно выбранных направлений число уравнений растет примерно как $O(m^2)$ (возможно, с логарифмическим множителем).

Данный алгоритм был реализован Gregor и Rannou для данных из магнитно-резонансной визуализации в статьях [5, 6]. При этом авторы столкнулись с тем, что при больших размерностях (порядка 100 000 опорных направлений) время поиска локальных положительных четверок было на порядок больше, чем время решения самой задачи квадратичного программирования.

Поэтому Gardner и Kiderlen [7] переформулировали исходную задачу в терминах точек касания:

$$\left\| \{(x_i, u_i)\}_{i=1}^m - h^0 \right\| \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_j), \quad 1 \leq i \neq j \leq m, \quad (3)$$

где $x_i \in \mathbb{R}^3$ — точки касания тела с опорной плоскостью, имеющей нормаль u_i . Нетрудно видеть, что если решить задачу оптимизации (3), то по полученным точкам касания x_i^* можно получить согласованные опорные числа $h_i^* = (x_i^*, u_i)$, а по ним — восстановить основной объект.

Преимущество данного подхода состоит в том, что условия согласованности выписываются автоматически, а для их нахождения не требуется искать все локальные положительные конусы. Тем не менее, основная проблема этого подхода, а именно квадратичное число условий, сохраняется в рассмотренной постановке задачи, что делает описанный выше подход неприменимым к задачам большой размерности.

В статьях Gardner, Kiderlen и Milanfar [10] и Guntuboyina [11] были сделаны исследования асимптотической скорости сходимости методов оценки опорной функции, но были ли попытки сократить число условий в описанных выше задачах, нам не известно.

4. Расширение преобразования двойственности. Для того чтобы обосновать предлагаемый нами подход, потребуется понятие *преобразования двойственности*. В соответствии с [8], определим его следующим образом.

Определение 4. Преобразованием двойственности называется преобразование δ , переводящее точки в $\mathbb{R}^3 \setminus \{O\}$ в плоскости: $\delta : (a, b, c) \mapsto \{ax + by + cz = 1\}$, а плоскости, не содержащие начала координат O , — в точки: $\delta : \{ax + by + cz = 1\} \mapsto (a, b, c)$.

Преобразование двойственности также доопределяют на выпуклых многогранниках, содержащих начало координат O в своей внутренности.

Определение 5. Пусть P — выпуклый многогранник, содержащий начало координат O в своей внутренности. Тогда двойственным многогранником $\delta(P)$ называется многогранник, имеющий вершины, двойственные плоскостям граней исходного многогранника, и грани, плоскости которых двойственны вершинам исходного многогранника.

В вычислительной геометрии часто используются следующие свойства преобразования двойственности δ (см., например, [9]).

Теорема 3. Преобразование двойственности δ сохраняет выпуклость для многогранников, содержащих начало координат: если P — выпуклый многогранник, $O \in \text{Int}(P)$, то $\delta(P)$ — выпуклый многогранник, $O \in \text{Int}(\delta(P))$.

Теорема 4. Преобразование двойственности δ обратное самому себе: если P — выпуклый многогранник, $O \in \text{Int}(P)$, то $\delta(\delta(P)) = P$.

Теорема 5. Преобразование двойственности δ сохраняет инцидентность: если P — выпуклый многогранник, $O \in \text{Int}(P)$, то грань f и вершина v многогранника P инцидентны тогда и только тогда, когда инцидентны двойственные им вершина $\delta(f)$ и грань $\delta(v)$ многогранника $\delta(P)$.

Теорема 6. Преобразование двойственности δ сохраняет пересечение многогранников: если P и Q — выпуклые многогранники, $O \in \text{Int}(P)$, $O \in \text{Int}(Q)$, то $P \cap Q \neq \emptyset \Leftrightarrow \delta(P) \cap \delta(Q) \neq \emptyset$.

Однако в данной работе нам потребуется доопределить преобразование двойственности также и на двух более широких классах многогранников.

Определение 6. Пусть K — выпуклый многогранник, $O \in \text{Int}(K)$, а p — точка, лежащая в его внешности. Многогранником видимости для K и p будем называть объединение отрезков $[p, q]$, таких, что точка $q \in \partial K$ видна из точки p : $P(K, p) = \bigcup_{q \in \partial K, [p, q] \cap \text{Int}(K) = \emptyset} [p, q]$. Многогранник P называется многогранником видимости, если $P = P(K, p)$ для некоторого многогранника K и точки p .

Определение 7. Многогранником двойственной видимости называется выпуклый многогранник P , являющийся пересечением такого набора полупространств $\{a_i x + b_i y + c_i z + d_i \leq 0\}$, $i = 1, \dots, s$, что ровно одно из них не содержит начало координат O в своей внутренней, а остальные — содержат O в своей внутренней, т.е. $d_1 > 0$ и $d_i < 0$, $i = 2, \dots, s$.

Очевидно, что такой класс многогранников является самым простым способом расширить исходную область определения δ : каждый выпуклый многогранник P , такой, что $O \in \text{Int}(P)$, можно представить в виде пересечения полупространств такого же вида, только при условии, что у всех них свободные коэффициенты отрицательны: $P = \{a_i x + b_i y + c_i z + d_i \leq 0\}$, $d_i < 0$, $i = 1, \dots, s$.

Теперь доопределим преобразование двойственности δ на введенных классах многогранников. Для этого докажем следующую теорему.

Теорема 7. Пусть P — многогранник видимости, причем $P = P(K, p)$. Пусть p, q_1, q_2, \dots, q_s — вершины P , а f_1, \dots, f_r — плоскости граней P . Введем следующие обозначения для их двойственных образов:

$$\delta(p) = \{a_p x + b_p y + c_p z = 1\}, \quad \delta(q_i) = \{a_i x + b_i y + c_i z = 1\}, \quad \delta(f_i) = g_i.$$

Обозначим многогранник $D = \text{Hull}(g_1, \dots, g_r)$. Тогда верны следующие утверждения:

- 1) точки g_1, \dots, g_r находятся в экстремальном положении, т.е. лежат на границе своей выпуклой оболочки: $g_i \in \partial D$;
- 2) многогранник D — это многогранник двойственной видимости, представимый следующим образом:

$$D = \{-a_p x - b_p y - c_p z + 1 \leq 0\} \cap \bigcap_{i=1}^s \{a_i x + b_i y + c_i z - 1 \leq 0\}. \quad (4)$$

Доказательство. Рассмотрим следующие многогранники:

$$\begin{aligned} D_1 &= \bigcap_{i=1}^s \{a_i x + b_i y + c_i z - 1 \leq 0\}, \\ D_2 &= D_1 \cap \{a_p x + b_p y + c_p z - 1 \leq 0\}, \\ D' &= D_1 \cap \{-a_p x - b_p y - c_p z + 1 \leq 0\}. \end{aligned}$$

По определению, D_1 и D_2 являются выпуклыми и содержат начало координат в своей внутренней. Нетрудно видеть, что

$$\delta(D_1) = K, \quad \delta(D_2) = \text{Hull}(K, p),$$

поскольку преобразование двойственности сохраняет отношение инцидентности. Кроме того, из определения этих многогранников следует, что

$$D_1 = D' \cup D_2, \quad \partial D_1 \cup \delta(p) = \partial D' \cup \partial D_2.$$

Фиксируем некоторую плоскость f_i грани многогранника P . Возможны два случая:

- 1) грань P , лежащая в f_i , суть грань многогранника K . Тогда g_i — это вершина многогранника D_1 , не лежащая на границе D_2 . Следовательно, $g_i \in \partial D'$;
- 2) грань P , лежащая в f_i , суть грань многогранника $\text{Hull}(K, p)$, не являющаяся гранью K . Тогда она инцидентна вершине p и, следовательно, $g_i \in \delta(p) \subset \partial D'$.

Таким образом, все $g_i \in \partial D'$, а поскольку D' по определению выпуклый, то g_i находятся в экстремальном положении и их выпуклая оболочка $D = \text{Hull}(g_1, \dots, g_r) \subset D'$. Обратное включение $D' \subset D$ тоже легко доказать: если g — вершина D' , то либо g — вершина D_1 , либо $g \in \delta(p)$. В обоих случаях $g = g_i$ для некоторого i . \square

Данная теорема позволяет естественным образом доопределить преобразование двойственности на двух введенных выше классах многогранников.

Определение 8. Пусть P — многогранник видимости, тогда положим $\delta(P) = D$, где D — многогранник, определяемый соотношением (4).

Определение 9. Пусть D — многогранник двойственной видимости, тогда его нетрудно представить в виде (4). Положим $\delta(D) = P$ из теоремы 7.

Смысл этого доопределения состоит в том, чтобы доказать аналоги теорем 3–6 для расширенного преобразования. Аналоги первых двух из этих теорем вложены в определения, аналог теоремы 5 тоже тривиален. Аналог теоремы 6 можно сформулировать в следующем виде.

Теорема 8. *Многогранники двойственной видимости P и Q пересекаются тогда и только тогда, когда пересекаются двойственные им многогранники видимости $\delta(P)$ и $\delta(Q)$.*

Из этой теоремы вытекает тривиальное следствие.

Лемма 1. *Пусть K – выпуклый многогранник, $O \in \text{Int}(K)$. Тогда многогранники*

$$K_1 = K \cap \{a_1x + b_1y + c_1z + 1 \leq 0\}, \quad K_2 = K \cap \{a_2x + b_2y + c_2z + 1 \leq 0\}$$

являются многогранниками двойственной видимости. Обозначим

$$p_1 = \delta(\{a_1x + b_1y + c_1z + 1 = 0\}), \quad p_2 = \delta(\{a_2x + b_2y + c_2z + 1 = 0\}).$$

Тогда многогранники K_1 и K_2 пересекаются тогда и только тогда, когда пересекаются двойственные им многогранники видимости $P(p_1, \delta(K))$ и $P(p_2, \delta(K))$, или, что то же самое,

$$K_1 \cap K_2 \neq \emptyset \Leftrightarrow [p_1, p_2] \cap \delta(K) = \emptyset.$$

Пересечение двух многогранников видимости $P(p_1, \delta(K))$ и $P(p_2, \delta(K))$, соответствующих одному многограннику $\delta(K)$, очевидно, можно интерпретировать как то, что точки p_1 и p_2 “видят” друг друга, т.е. многогранник $\delta(K)$ “не закрывает” одну точку от другой.

5. Метод исключения избыточных условий. Рассмотрим задачу в терминах точек касания в метрике L_∞ . Если ввести дополнительную переменную ε , то задачу можно переписать в следующем виде:

$$\varepsilon \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_j), \quad |(x_i, u_i) - h_i| \leq \varepsilon. \quad (5)$$

Оказывается, при такой постановке задачи большинство условий вида $(x_i, u_i) \geq (x_j, u_j)$ являются избыточными. В основе предлагаемого метода лежит *достаточное условие избыточности*.

Мы будем рассматривать его вне контекста задачи оптимизации (5), поскольку в дальнейшем метод понадобится и для задач в метриках L_1 и L_2 .

Итак, рассмотрим систему ограничений:

$$(x_i, u_i) \geq (x_j, u_j), \quad |(x_i, u_i) - h_i| \leq \varepsilon. \quad (6)$$

Теорема 9. *Пусть система ограничений (6) разрешима для некоторого ε^0 . Обозначим многогранник*

$$K = \bigcap_{i=1}^m \{(x, u_i) \leq h_i + \varepsilon^0\}. \quad (7)$$

Пусть δ – полярное преобразование двойственности. Тогда если отрезок, который соединяет точки $\delta(\{(x, u_i) = h_i - \varepsilon^0\})$ и $\delta(\{(x, u_j) = h_j - \varepsilon^0\})$, пересекает $\delta(K)$, то условия $(x_i, u_i) \geq (x_j, u_j)$ и $(x_j, u_j) \geq (x_i, u_j)$ избыточны в системе ограничений (6).

Доказательство. Пусть система ограничений (6) разрешима для некоторого ε^0 . Тогда, как всякая задача линейного программирования, она имеет оптимальное решение $x = x^*$, $\varepsilon = \varepsilon^*$, удовлетворяющее следующим условиям: $(x_i, u_i) \geq (x_j, u_j)$, $|(x_i, u_i) - h_i| \leq \varepsilon^0$.

Из ограничения $|(x_i, u_i) - h_i| \leq \varepsilon^0$ можно получить два условия: ограничение на (x_i, u_i) сверху $(x_i, u_i) \leq h_i + \varepsilon^0$ и снизу

$$(x_i, u_i) \geq h_i - \varepsilon^0. \quad (8)$$

Ограничение $(x_i, u_i) \geq (x_j, u_j)$ означает, что точка x_i реализует максимум величины проекции на направление u_i среди всех точек касания. Следовательно, для всех точек касания $x = x_k$ выполнено условие $(x, u_i) \leq h_i + \varepsilon^0$.

Повторяя это утверждение для всех $i = 1, 2, \dots, m$, получим, что все точки касания лежат в многограннике K , введенном в формулировке теоремы (7). Отсюда и из (8) следует, что каждая точка касания x_i лежит в многограннике K_i , определяемом следующим образом:

$$K_i = K \cap \{(x_i, u_i) \geq h_i^0 - \varepsilon^0\}.$$

Докажем, что если многогранники K_i и K_j не пересекаются, то в системе ограничений (6) условия $(x_i, u_i) \geq (x_j, u_i)$ и $(x_j, u_j) \geq (x_i, u_j)$ избыточны.

Действительно, из $K_i \cap K_j = \emptyset$ следует, что плоскость $\{(x_i, u_i) = h_i - \varepsilon^0\}$ не пересекает многогранник K_j , а плоскость $\{(x_j, u_j) = h_j - \varepsilon^0\}$ не пересекает многогранник K_i .

Тогда из условий $x_i \in K_i$ и $x_j \in K_j$ следует, что $(x_i, u_j) < h - \varepsilon^0$ и $(x_j, u_i) \geq h - \varepsilon^0$, откуда имеем условие $(x_i, u_j) \leq (x_j, u_i)$. Аналогично из условий $x_i \in K_i$ и $x_j \in K_j$ можно вывести $(x_j, u_i) \leq (x_i, u_i)$.

Остается доказать, что многогранники K_i и K_j не пересекаются тогда и только тогда, когда отрезок, соединяющий точки $\delta(\{(x, u_i) = h_i^0 - \varepsilon^0\})$ и $\delta(\{(x, u_j) = h_j - \varepsilon^0\})$, пересекает $\delta(K)$. Но это утверждение является прямым следствием леммы 1. \square

В метрике L_1 задачу (3) можно записать в следующем виде:

$$I = \sum_{i=1}^m |(x_i, u_i) - h_i| \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_i), \quad 1 \leq i \neq j \leq m, \quad (9)$$

а в метрике L_2 — в следующем:

$$I = \sum_{i=1}^m ((x_i, u_i) - h_i)^2 \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_i), \quad 1 \leq i \neq j \leq m. \quad (10)$$

Между тем, довольно естественно ожидать, что полученные оптимальные точки касания будут удовлетворять условию $|(x_i^*, u_i) - h_i| \leq \varepsilon^0$ для некоторого фиксированного $\varepsilon^0 > 0$ при условии малости погрешности. Если наложить на задачи (9) и (10) дополнительные ограничения

$$|(x_i, u_i) - h_i| \leq \varepsilon^0,$$

то задачу (9) можно будет переформулировать в следующем виде:

$$I = \varepsilon_1 + \dots + \varepsilon_m \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_i), \quad 1 \leq i \neq j \leq m, \\ |(x_i, u_i) - h_i| \leq \varepsilon_i, \quad i = 1, \dots, m, \quad 0 \leq \varepsilon_i \leq \varepsilon^0, \quad i = 1, \dots, m,$$

а задачу (10) — в следующем:

$$I = \varepsilon_1^2 + \dots + \varepsilon_m^2 \rightarrow \inf, \quad (x_i, u_i) \geq (x_j, u_i), \quad 1 \leq i \neq j \leq m, \\ |(x_i, u_i) - h_i| \leq \varepsilon_i, \quad i = 1, \dots, m, \quad 0 \leq \varepsilon_i \leq \varepsilon^0, \quad i = 1, \dots, m.$$

Строго говоря, такие две задачи не являются эквивалентными исходным. Однако, из соображений локальности, мы считаем, что решения последних двух задач будут в некотором смысле близки к решениям первых.

Итак, приведем алгоритм построения системы избыточных ограничений.

1. $S = \emptyset$.
2. По исходным опорным данным $u_i, h_i, i = 1, \dots, m$, построить тело K^0 как пересечение полупространств:

$$K^0 = \bigcap_{i=1}^m \{(u_i, x) \leq h_i\}. \quad (11)$$

3. Пусть A_1, \dots, A_s — вершины тела K^0 . Для $i = 1, \dots, m$ вычислить опорные числа тела K^0 по направлениям u_i следующим образом: $h_i^0 = \max_{j=1, \dots, s} (u_i, A_j)$.

4. Вычислить L_∞ -расстояние от полученных опорных чисел до исходных: $\varepsilon^0 = \max_{i=1, \dots, m} |h_i - h_i^0|$.

5. Построить многогранник D_{ε^0} как выпуклую оболочку двойственных образов верхних плоскостей:

$$D_{\varepsilon^0} = \text{Hull} \left\{ \delta \left(\{(x, u_i) = h_i + \varepsilon^0\} \right) \right\}_{i=1}^m.$$

6. Построить двойственные образы нижних плоскостей: $p_i = \delta\left(\{(x, u_i) = h_i - \varepsilon^0\}\right)$.
7. Для всех $i, j = 1, \dots, m, i \neq j$, проверить следующее условие: $[p_i, p_j] \cap D_{\varepsilon^0} = \emptyset$. Если оно выполняется, то добавить неравенства $(u_i, x_i) \geq (u_i, x_j)$, $(u_j, x_j) \geq (u_j, x_i)$ в систему ограничений S .

Вообще говоря, программист на свой страх и риск может в шаге 4 приведенного алгоритма уменьшить полученное отклонение на некоторый множитель, для того чтобы как можно значительно сократить число ограничений:

$$\varepsilon' = f\varepsilon^0, \quad 0 < f < 1.$$

Однако, в отличие от ε^0 , для ε' нельзя гарантировать согласованность системы ограничений (6). При этом может оказаться, что $\varepsilon' < \varepsilon^*$, и тогда система (6) не будет согласованной. Полученная при прохождении остальных шагов алгоритма задача не будет эквивалентна исходной, а ее решение не будет удовлетворять всем условиям согласованности.

6. Реализация алгоритма. В качестве входных данных для программы использовались теневые контуры, полученные в результате фотографирования драгоценных камней. Теневые контуры представляли собой проекции вдоль некоторых горизонтальных направлений $\nu_i, i = 1, \dots, n$. Каждый такой контур является многоугольником на плоскости $(\nu_i, x) = 0$. Рассматривавшиеся производственные теневые контуры могли быть незамкнутыми, поскольку сторона контура, соответствующая грани, на которой стоял камень на подставке во время фотографирования, часто не включалась в описание контура. Описание каждого контура представляло собой список пар координат концов сторон контура.

Для упрощения задачи над теневыми контурами производилась следующая предобработка.

1. Если контур являлся незамкнутым, то он замыкался: между соседними сторонами добавлялась дополнительная сторона.
2. Вычислялся центр масс C объединения всех контуров, и из всех координат концов сторон вычитался вектор OC . Это делалось для того, чтобы гарантировать положительность всех опорных чисел (опорные числа положительны тогда и только тогда, когда тело содержит начало координат O в своей внутренней).
3. Невыпуклые контуры заменялись своей выпуклой оболочкой.

Очевидно, что процесс получения теневого контура можно интерпретировать как измерение опорной функции тела на большом круге сферы, т. е. каждый теневой контур по сути представляет собой непрерывное семейство измерений опорной функции тела. Поскольку такие измерения получают экспериментальным путем, то для нескольких контуров может не найтись выпуклого многогранника, им соответствующего.

Чтобы согласовать контуры между собой, из них извлекались нормали всех сторон всех контуров $u_i, i = 1, \dots, n$, и расстояния от начала координат до сторон h_i . Эти данные интерпретировались как измерения опорной функции исходного тела по опорным направлениям u_i . Очевидно, что согласованность опорных чисел h_i является необходимым условием согласованности теневых контуров. Целью работы программы является построение таких согласованных чисел h_i^* , которые бы были наиболее близки к исходным.

Алгоритм построения системы избыточных ограничений был реализован на основе API-библиотеки алгоритмов вычислительной геометрии CGAL [12]. В качестве алгоритма построения выпуклой оболочки был использован стандартный алгоритм Quickhull [15] из пакета трехмерных выпуклых оболочек [13].

Для нахождения пересечения полупространств в уравнении (11) все плоскости, задающие полупространства, сначала отображались в двойственные им точки с помощью преобразования двойственности δ , затем строилась выпуклая оболочка полученных точек, вычислялись уравнения плоскостей граней полученного двойственного многогранника $\delta(K^0)$ и на их основе вычислялись координаты вершин многогранника K^0 из уравнения (11).

Для быстрой проверки пересечения отрезка и многогранника использовалась структура данных AABB-дерево [14], позволяющая автоматически распараллеливать запросы на проверки пересечения.

С помощью алгоритма построения системы избыточных ограничений строилась система ограничений задачи, к которым затем добавлялись условия локальности $|(x_i, u_i) - h_i| < \varepsilon^0$ для L_∞ -задачи либо $|(x_i, u_i) - h_i| < \varepsilon_i$ для L_1 - и L_2 -задач.

Полученные задачи решались с помощью следующих пакетов программ для решения задач оптимизации:

- 1) пакет GLPK [16] использовался для решения L_∞ - и L_1 -задач и для генерации задач в формате MPS, который затем подавался на вход другим решателям;
- 2) пакет CLP [17] использовался для решения L_∞ - и L_1 -задач;
- 3) Iprort [18, 20], решатель на основе метода внутренней точки, использовался для решения L_∞ -, L_1 - и L_2 -задач с помощью C++ API для разделяемой библиотеки;
- 4) пакет IBM ILOG CPLEX [19] использовался для решения L_∞ - и L_1 -задач в формате MPS или LP.

В методе внутренней точки, реализованном в решателе Iprort, требуется указать начальную точку алгоритма. От того, насколько эта точка близка к точке минимума, существенно зависит быстродействие алгоритма. Поэтому в качестве начального положения точек касания x_i были взяты точки касания пересечения полупространств из уравнения (11): $(u_i, x_i^0) = \max_{j=1, \dots, N} (u_i, y_j)$, где $y_j, j = 1, \dots, N$, — координаты вершин многогранника K^0 из уравнения (11).

В результате работы решателей получался набор точек касания $x_i^*, i = 1, \dots, m$, на основе которых строился набор опорных чисел $h_i^* = (x_i^*, u_i)$. По этим числам и происходило восстановление трехмерного тела как пересечения полупространств: $K^* = \bigcap_{i=1}^m \{(x, u_i) \leq h_i^*\}$.

Вычисление этого многогранника осуществлялось точно так же, как и вычисление многогранника K^0 (см. выше).

7. Результаты тестирования алгоритма. Реализованный алгоритм тестировался на ЭВМ с 4-ядерным процессором Intel Core i5-2500 CPU с частотой 3.30 GHz, ОЗУ 16 ГБ и операционной системой Ubuntu 14.04 LTS.

Данная статья не преследует цель сравнить скорость тех или иных решателей, упомянутых выше, поскольку все они использовались со стандартными настройками конфигурации и, возможно, для тех или иных пакетов был использован не самый оптимальный выбор настроек. Тем не менее, чтобы дать читателю понятие о скорости алгоритма, приведем здесь те результаты, которые удалось получить.

Получить решение линейных задач (т. е. L_1 и L_∞) за приемлемое время удалось с помощью пакетов IBM ILOG CPLEX и Iprort. Единственным решателем, который смог справиться с квадратичной задачей, оказался Iprort. По этой причине здесь будут приведены только результаты, полученные с помощью этих двух пакетов.

В документации Iprort указано, что его производительность может меняться на порядки в зависимости от того, какой пакет линейной алгебры к нему подключен. Наиболее быструю производительность пакета Iprort для данной задачи удалось получить с использованием метода MA57 из пакета HSL [21] и пакета OpenBLAS [22], использованного в качестве реализации библиотеки основных функций линейной алгебры (BLAS).

В качестве входных данных алгоритма были использован набор из 200 теневых контуров, полученных в результате фотографирования реального камня. Каждый контур содержал в среднем примерно по 15 сторон: общее число сторон во всех контурах равнялось 2912. Для проверки асимптотики скорости работы алгоритма из данного набора контуров равномерно выбирались поднаборы из 10, 20, 30, ..., 180, 190, 200 контуров и подавались в качестве входных данных для алгоритма.

На рис. 2 показан график времени работы алгоритма построения системы избыточных условий. Очевидно, что данный алгоритм осуществляет квадратичный перебор всех пар опорных направлений, и как видно по графику, квадратичная сложность имеет место и на практике.

Какую пользу приносит предлагаемый алгоритм? Сколько условий он позволяет отбросить? На рис. 3 показаны графики числа ограничений до и после применения алгоритма, а на рис. 4 — график отношения этих величин. Таким образом, в рассмотренном примере удалось сократить более 80% ограничений.

Для обоих решателей, как и требовалось ожидать, быстрее всего удастся получить решение для задачи в метрике L_∞ . С помощью CPLEX задача решается за 9.52 секунды на 100 контурах (1476 опорных чисел) и за 71.24 секунды на 200 контурах (2912 опорных чисел), и с помощью Iprort за 32.17 и 278.73 секунды соответственно. В метрике L_1 эти же задачи решаются за 11.37 и 94.19 секунд с помощью CPLEX и за 65.29 и 655.79 секунд с помощью Iprort. В метрике L_2 задачу удалось решить только с помощью Iprort: за 38.42 секунды на 100 контурах и за 360.53 секунды на 200 контурах.

Насколько предложенный новый алгоритм быстрее исходного? Если не исключать избыточные условия, то при прочих равных условиях (решатели, библиотеки линейной алгебры и т. д.) задача на 100 контурах решается примерно за то же время, что и задача с исключенными избыточными условиями на

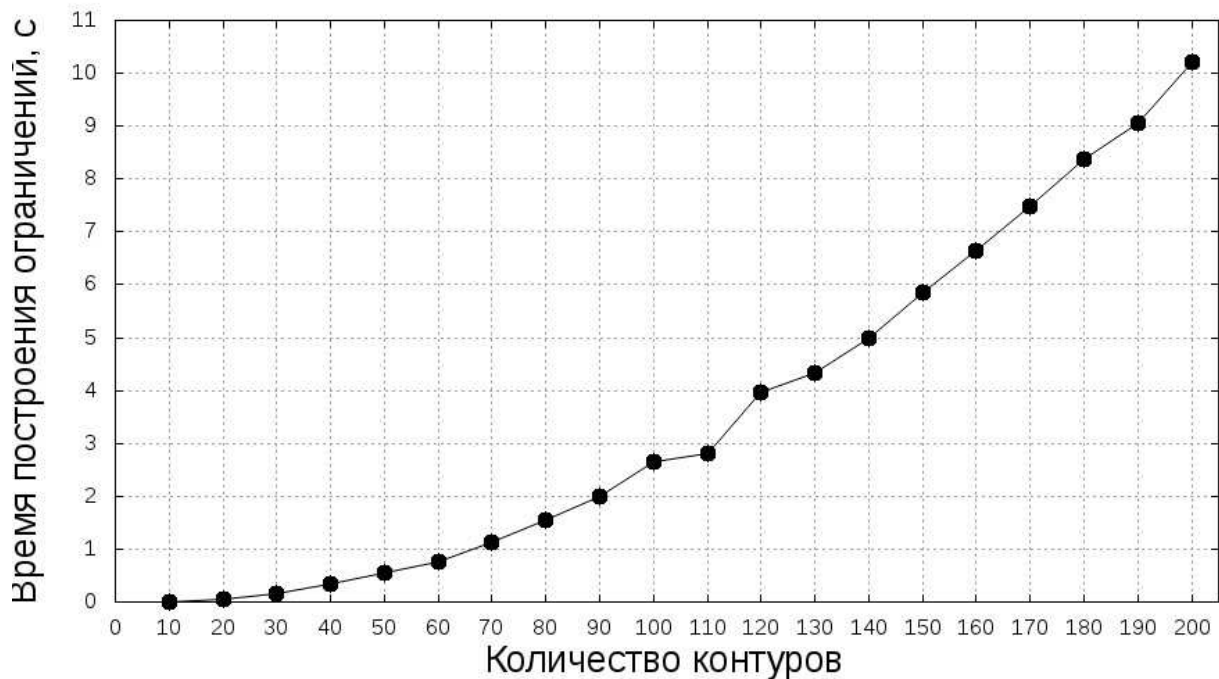


Рис. 2. Время работы алгоритма нахождения системы избыточных условий

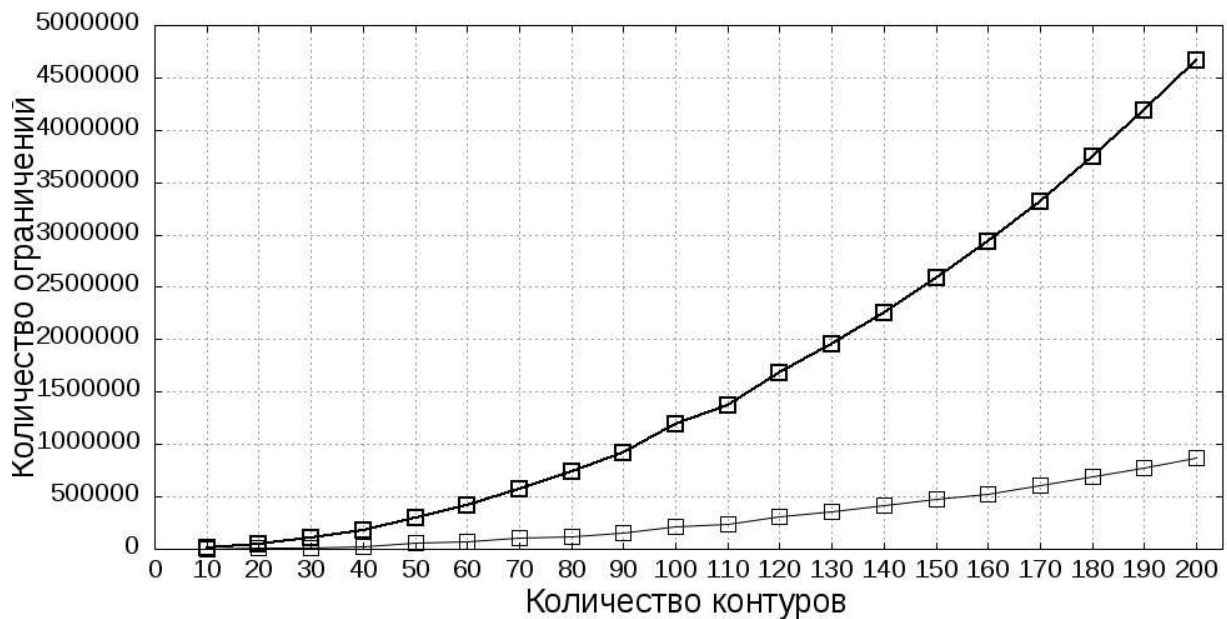


Рис. 3. Количество ограничений. Жирным цветом помечен график общего числа ограничений, обычным цветом — график числа избыточных ограничений

200 контурах. Таким образом, новый алгоритм дает возможность за прежнее время решить задачу в 2 раза большей размерности. Кроме того, следует отметить, что в статье Гарднера–Кидерлена рассматривается реализация алгоритма на Matlab, и оригинальный алгоритм решал задачу на 50 опорных числах примерно за 100 секунд. За это же время новый алгоритм успевает решить задачу на 130 теневого контурах, что соответствует 1882 опорным числам.

8. Выводы и направления дальнейших исследований. Из всего вышесказанного можно сделать следующие выводы.

Представленный метод позволяет отбросить значительную часть условий в задаче восстановления выпуклого тела по измерениям его опорной функции (около 80% в рассмотренном примере).

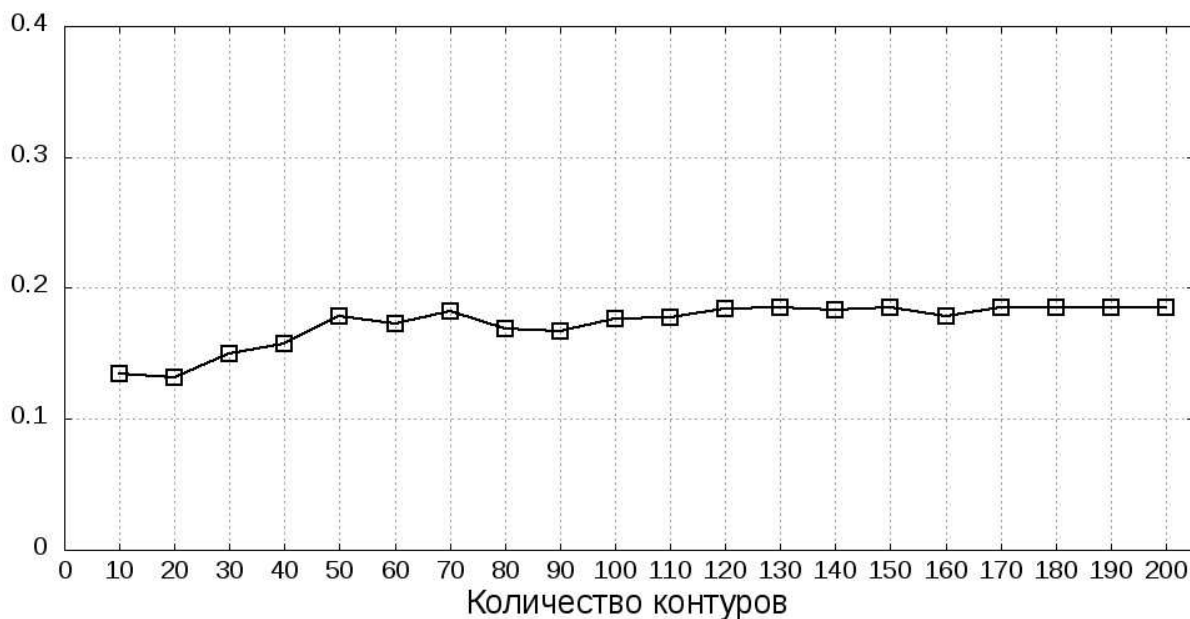


Рис. 4. Отношение числа избыточных ограничений к общему числу ограничений

Благодаря этому можно значительно ускорить решение задачи оптимизации и увеличить допустимые размеры задачи, сохранив возможность ее решения за разумное время.

Сам по себе алгоритм построения множества избыточных ограничений требует гораздо меньшего времени, чем задача оптимизации.

Представленный метод восстановления тел с помощью решения задачи оптимизации, описанной выше, позволяет построить тело, наилучшим образом (с математической точки зрения) соответствующее заданному набору теневых контуров.

В полученном теле будет столько же граней, сколько сторон во всех теневых контурах вместе взятых. В этом заключается основной недостаток получаемых тел.

Следующие направления представляются возможными для дальнейших исследований.

(i) В L_∞ -метрике представленную задачу можно решить и без решения задачи оптимизации: задача существования согласованного набора для заданного ε решается за линейное время, поэтому с помощью деления отрезка пополам можно с любой заданной точностью найти такое минимальное ε , для которого существует согласованное решение.

(ii) С точки зрения практики довольно ценно было бы рассмотреть задачу построения тела с заданным числом граней. Очевидно, что эту задачу можно сформулировать в виде задачи нелинейного программирования на линейном числе кусочно-линейных условий: условие того, что точка реализует максимум проекции на заданном направлении среди конечного набора точек, может быть записано как неравенство между двумя (а не всеми) проекциями.

СПИСОК ЛИТЕРАТУРЫ

1. Ghosh P.K., Kumar K.V. Support function representation of convex bodies, its application in geometric computing, and some related representations // Computer Vision and Image Understanding. 1998. **72**, N 3. 379–403.
2. Prince J.L., Willsky A.S. Reconstructing convex sets from support line measurements // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1990. **12**, N 4. 377–389.
3. Lele A.S., Kulkarni S.R., Willsky A.S. Convex-polygon estimation from support-line measurements and applications to target reconstruction from laser-radar data // Journal of the Optical Society of America A. 1992. **9**, N 10. 1693–1714.
4. Karl W.C., Kulkarni S.R., Verghese G.C., Willsky A.S. Local tests for consistency of support hyperplane data // Journal of Mathematical Imaging and Vision. 1996. **6**, N 2–3. 249–267.
5. Gregor J., Rannou F. Least-squares framework for projection MRI reconstruction // Proceedings of SPIE. Vol. 4322. 2001. doi: 10.1117/12.431168. 2001. Vol. 4322.
6. Gregor J., Rannou F.R. Three-dimensional support function estimation and application for projection magnetic resonance imaging // International Journal of Imaging Systems and Technology. 2002. **12**, N 1. 43–50.

7. Gardner R.J., Kiderlen M. A new algorithm for 3D reconstruction from support functions // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009. **31**, N 3. 556–562.
8. Preparata F.P., Shamos M.I. Computational Geometry: An Introduction. New York: Springer, 1985.
9. Chazelle B. An optimal algorithm for intersecting three-dimensional convex polyhedra // Society for Industrial and Applied Mathematics Journal on Computing. 1992. **21**, N 4. 671–696.
10. Gardner R.J., Kiderlen M., Milanfar P. Convergence of algorithms for reconstructing convex bodies and directional measures // The Annals of Statistics. 2006. **34**, N 3. 1331–1374.
11. Guntuboyina A. Optimal rates of convergence for convex set estimation from support functions // The Annals of Statistics. 2012. **40**, N 1. 385–411.
12. Computational Geometry Algorithms Library (<http://www.cgal.org>).
13. Hert S., Schirra S. 3D Convex Hulls. CGAL User and Reference Manual. 2015 (<http://doc.cgal.org/4.5.2/Manual/packages.html>).
14. Alliez P., Tayeb S., Wormser C. 3D Fast Intersection and Distance Computation (AABB Tree). CGAL User and Reference Manual. 2015 (<http://doc.cgal.org/4.5.2/Manual/packages.html>).
15. Barber C.B., Dobkin D.P., Huhdanpaa H. The Quickhull algorithm for convex hulls // ACM Transactions on Mathematical Software. 1996. **22**, N 4. 469–483.
16. GLPK, GNU Linear Programming Kit (<http://www.gnu.org/software/glpk/glpk.html>).
17. CLP, COIN-OR Linear Programming solver (<https://projects.coin-or.org/Clp>).
18. Ipopt, COIN-OR Interior Point OPTimizer (<https://projects.coin-or.org/Ipopt>).
19. IBM ILOG CPLEX Optimizer (<http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>).
20. Wächter A., Biegler L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming // Mathematical Programming. 2006. **106**, N 1. 25–57.
21. The HSL Mathematical Software Library (<http://www.hsl.rl.ac.uk/>).
22. OpenBLAS. An Optimized BLAS Library (<http://xianyi.github.io/OpenBLAS>).

Поступила в редакцию
29.04.2015

A Method of Redundant Constraint Elimination in the Problem of Body Recovery Based on Support Function Measurements

I. A. Palachev¹

¹ *Lomonosov Moscow State University, Faculty of Mechanics and Mathematics; Leninskie Gory, Moscow, 119899, Russia; Graduate Student, e-mail: palachev.ilya@yandex.ru*

Received April 29, 2015

Abstract: A new body recovery algorithm based on support function measurements is proposed. The proposed algorithm represents a linear or quadratic programming problem in Gardner–Kiderlen form with smaller number of constraints. The reduction of constraint number is based on a new method that allows one to eliminate a part of initial constraints as redundant. A new approach of body recovery based on shadow contours is proposed. It allows one to reuse body recovery methods based on support function measurements. The implementation of the algorithm is described and some results of its testing on real industrial contours are discussed. The proposed method ensures the reduction of constraint number by 80% Gardner–Kiderlen algorithm by an order of magnitude.

Keywords: support function, geometric bodies recovery, linear programming, quadratic programming, shadow contour, duality transformation.

References

1. P. K. Ghosh and K.V. Kumar, “Support Function Representation of Convex Bodies, Its Application in Geometric Computing, and Some Related Representations,” *Comput. Vis. Image Underst.* **72** (3), 379–403 (1998).
2. J. L. Prince and A. S. Willsky, “Reconstructing Convex Sets from Support Line Measurements,” *IEEE Trans. Pattern Anal. Mach. Intell.* **12** (4), 377–389 (1990).

3. A. S. Lele, S. R. Kulkarni, and A. S. Willsky, "Convex-Polygon Estimation from Support-Line Measurements and Applications to Target Reconstruction from Laser-Radar Data," *J. Opt. Soc. Am. A* **9** (10), 1693–1714 (1992).
4. W. C. Karl, S. R. Kulkarni, G. V. Verghese, and A. S. Willsky, "Local Tests for Consistency of Support Hyperplane Data," *J. Math. Imaging Vis.* **6** (2–3), 249–267 (1996).
5. J. Gregor and F. Rannou, "Least-Squares Framework for Projection MRI Reconstruction," *Proc. SPIE*, Vol. 4322 (2001). doi: 10.1117/12.431168
6. J. Gregor and F. R. Rannou, "Three-Dimensional Support Function Estimation and Application for Projection Magnetic Resonance Imaging," *Int. J. Imaging Syst. Technol.* **12** (1), 43–50 (2002).
7. R. J. Gardner and M. Kiderlen, "A New Algorithm for 3D Reconstruction from Support Functions," *IEEE Trans. Pattern Anal. Mach. Intell.* **31** (3), 556–562 (2009).
8. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction* (Springer, New York, 1985).
9. B. Chazelle, "An Optimal Algorithm for Intersecting Three-Dimensional Convex Polyhedra," *SIAM J. Comput.* **21** (4), 671–696 (1992).
10. R. J. Gardner, M. Kiderlen, and P. Milanfar, "Convergence of Algorithms for Reconstructing Convex Bodies and Directional Measures," *Ann. Statist.* **34** (3), 1331–1374 (2006).
11. A. Guntuboyina, "Optimal Rates of Convergence for Convex Set Estimation from Support Functions," *Ann. Statist.* **40** (1), 385–411 (2012).
12. The Computational Geometry Algorithms Library. <http://www.cgal.org>. Cited June 8, 2015.
13. S. Hert and S. Schirra, "3D Convex Hulls," CGAL User and Reference Manual. <http://doc.cgal.org/4.5.2/Manual/packages.html>. Cited June 8, 2015.
14. P. Alliez, S. Tayeb, and C. Wormser, "3D Fast Intersection and Distance Computation (AABB Tree)," CGAL User and Reference Manual. <http://doc.cgal.org/4.5.2/Manual/packages.html>. Cited June 8, 2015.
15. C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Trans. Math. Softw.* **22** (4), 469–483 (1996).
16. GLPK (GNU Linear Programming Kit). <http://www.gnu.org/software/glpk/glpk.html>. Cited June 8, 2015.
17. CLP, COIN-OR Linear Programming Solver. <https://projects.coin-or.org/Clp>. Cited June 8, 2015.
18. Ipopt, COIN-OR Interior Point OPTimizer. <https://projects.coin-or.org/Ipopt>. Cited June 8, 2015.
19. IBM ILOG CPLEX Optimizer. <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>. Cited June 8, 2015.
20. A. Wächter and L. T. Biegler, "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Math. Program.* **106** (1), 25–57 (2006).
21. The HSL Mathematical Software Library. <http://www.hsl.rl.ac.uk/>. Cited June 8, 2015.
22. OpenBLAS. An Optimized BLAS Library. <http://xianyi.github.io/OpenBLAS>. Cited June 8, 2015.