

А.А.Ардентов, Ю.Л.Сачков

Антропоморфное восстановление поврежденных изображений на основе методов субримановой геометрии

Аннотация. Согласно результатам нейрогеометрии зрения, человеческий мозг восстанавливает изофоты на поврежденных изображениях с помощью оптимальных кривых для некоторой задачи оптимального управления — субримановой задачи на группе движений плоскости. В статье описывается алгоритм антропоморфного восстановления поврежденных бинарных изображений на основе оптимального синтеза для этой задачи.

Ключевые слова и фразы: восстановление изображений, субриманова геометрия, оптимальное управление.

1. Введение

Задача восстановления поврежденных или скрытых изображений является одной из актуальных проблем компьютерной графики. Для ее решения разработан ряд методов, многие из которых используют технику вариационного исчисления и оптимального управления [1–3]. Данная работа основана на положениях одного из новых направлений нейрофизиологии — нейрогеометрии зрения [4, 5], а также на недавних результатах по субримановой геометрии [6–8]. На основе результатов этих исследований разработан алгоритм для восстановления серии изофот на поврежденных штриховых изображениях.

Работа поддержана Госконтрактом № 07.514.11.4033 по ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы», а также Российским фондом фундаментальных исследований, проект 09-01-00246-а.

2. Постановка задачи восстановления изображения и метод ее решения

Задача восстановления изображения может быть формализована следующим образом. Дана прямоугольная область $D \subset \mathbb{R}^2$, взаимно непересекающиеся подобласти $O_1, \dots, O_N \subset D$, и функция $f : D \setminus (\bigcup_{i=1}^N O_i) \rightarrow [0, 1]$. Требуется восстановить функцию f в областях O_1, \dots, O_N . Здесь D — область исходного изображения, O_i — подобласти с поврежденным изображением; функция f задает доступное наблюдателю изображение (для полутонового изображения яркость, а для штрихового изображения это функция, линии уровня которой совпадают с кривыми, составляющими изображение). Восстановление изображения в областях O_i должно быть антропоморфным, т.е. естественным для глаза человека.

Согласно результатам нейрогеометрии зрения [4,5], человеческий мозг восстанавливает изофоты $(x(t), y(t))$, $t \in [t_0, t_1]$, на поврежденных изображениях на основе вариационного принципа

$$(1) \quad \int_{t_0}^{t_1} \sqrt{\dot{x}^2 + \dot{y}^2 + \alpha^2 \dot{\theta}^2} dt \rightarrow \min, \quad \alpha > 0,$$

где $\theta = \arctg(\dot{y}/\dot{x})$. Таким образом, минимизируется интегральный компромисс между линейной (\dot{x}, \dot{y}) и угловой $\dot{\theta}$ скоростями восстанавливающей кривой. Параметр α задает масштаб на изображении, и должен подбираться в зависимости от изображения. В данной работе вариационный принцип (1) положен в основу восстановления поврежденных изображений.

При математическом моделировании использовались следующие допущения:

- изображение может быть представлено как портрет линий уровня гладкой функции $f : D \rightarrow [0, 1]$,
- функция f не имеет критических точек в областях повреждения $O_1, \dots, O_N \subset D$,
- известна информация о точках пересечения линий уровня функции f с границами областей O_i .

При выполнении этих условий предложенный метод может быть использован в комбинации с другими методами восстановления изображений.

Замена переменных

$$(2) \quad x = \alpha \tilde{x}, \quad y = \alpha \tilde{y}, \quad \theta = \tilde{\theta}$$

преобразует задачу (1) к случаю $\alpha = 1$. Поэтому при математическом рассмотрении задачи будем считать $\alpha = 1$, а в алгоритме восстановления изображений параметр α будет подбираться с целью устранения точек возврата на восстанавливающих кривых. Описанный вариационный метод восстановления изображений рассмотрен в работе [9]. В данной работе представлены следующие новые результаты:

- а) получено явное сведение задачи (1) к решению систем алгебраических уравнений,
- б) использован параметр α для устранения точек возврата на восстанавливающих кривых,
- в) детально описан алгоритм восстановления изофот.

3. Субриманова задача на группе движений плоскости

Задача минимизации функционала $\int \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{\theta}^2} dt$ для кривых $(x(t), y(t), \theta(t))$ с закрепленными концами может быть сформулирована как следующая задача оптимального управления:

- (3) $\dot{x} = u_1 \cos \theta, \quad \dot{y} = u_1 \sin \theta, \quad \dot{\theta} = u_2,$
- (4) $q = (x, y, \theta) \in M = \mathbb{R}_{x,y}^2 \times S_\theta^1, \quad u = (u_1, u_2) \in \mathbb{R}^2,$
- (5) $q(0) = q_0 = (0, 0, 0), \quad q(t_1) = q_1 = (x_1, y_1, \theta_1),$
- (6) $l = \int_0^{t_1} \sqrt{u_1^2 + u_2^2} dt \rightarrow \min.$

Пространство состояний $M = \mathbb{R}_{x,y}^2 \times S_\theta^1$ естественным образом отождествляется с группой $SE(2)$ движений двумерной плоскости, сохраняющих ориентацию. Эта группа представляется матрицами 3×3 следующим образом:

$$SE(2) = \left\{ \left(\begin{array}{ccc} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{array} \right) \mid \theta \in S^1 = \mathbb{R}/(2\pi\mathbb{Z}), x, y \in \mathbb{R} \right\}.$$

Задача (3)–(6) естественно переформулируется как левоинвариантная субриманова задача на группе Ли $SE(2)$ [10, 11]. Рассмотрим неинтегрируемую левоинвариантную субриманову структуру ранга 2 на $SE(2)$, т. е. неинтегрируемое левоинвариантное распределение Δ ранга 2 на $SE(2)$ вместе с левоинвариантным скалярным произведением

$\langle \cdot, \cdot \rangle$ на Δ . Легко показать, что такая структура единственна, с точностью до скалярного множителя в скалярном произведении. Рассмотрим следующую модель такой субримановой структуры:

$$\begin{aligned} \Delta_q &= \text{span}(X_1(q), X_2(q)), & \langle X_i, X_j \rangle &= \delta_{ij}, \quad i, j = 1, 2, \\ X_1(q) &= qE_{13}, & X_2(q) &= q(E_{21} - E_{12}), \quad q \in \text{SE}(2), \end{aligned}$$

(где E_{ij} есть матрица 3×3 с единственным единичным элементом в i -ой строке и j -ом столбце, и нулевыми остальными элементами) и соответствующую задачу оптимального управления:

$$\begin{aligned} \dot{q} &= u_1 X_1(q) + u_2 X_2(q), & q &\in M = \text{SE}(2), \quad u = (u_1, u_2) \in \mathbb{R}^2, \\ q(0) &= q_0 = \text{Id}, & q(t_1) &= q_1, \\ l &= \int_0^{t_1} \sqrt{u_1^2 + u_2^2} dt \rightarrow \min. \end{aligned}$$

Эту задачу можно переформулировать на языке робототехники следующим образом. Рассмотрим мобильный робот на плоскости, который может двигаться вперед и назад, и вращаться вокруг себя (машина Ридса–Шеппа) [12]. Состояние робота описывается координатами (x, y) его центра масс и углом поворота θ . При заданных начальном и конечном состояниях машины требуется найти кратчайшую траекторию из начального состояния в конечное, если длина траектории измеряется в пространстве (x, y, θ) , см. рис. 1.

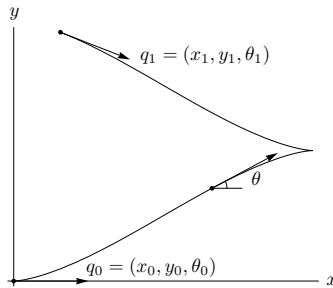


Рис. 1. Постановка задачи (3)–(6)

В работе [6] показано, что для любой конечной точки $q_1 \in \text{SE}(2)$ задача (3)–(6) имеет оптимальное решение.

4. Сведение задачи оптимального управления к решению системы уравнений

В этом разделе мы опишем некоторые результаты работ [6–8], которые позволяют свести задачу (3)–(6) к решению систем уравнений в функциях Якоби.

Из принципа максимума Понтрягина следует, что экстремальные траектории в задаче (3)–(6) параметризуются точками фазового цилиндра $C = (2S_\gamma^1) \times \mathbb{R}_c$ математического маятника

$$(7) \quad \dot{\gamma} = c, \quad \dot{c} = -\sin \gamma.$$

Семейство параметризованных длиной дуги нормальных экстремальных кривых в задаче (3)–(6) описывается экспоненциальным отображением

$$\begin{aligned} \text{Exp} : N &\rightarrow M, & N &= C \times \mathbb{R}_+, \\ \text{Exp}(\nu) &= \text{Exp}(\lambda, t) = q(t), & \nu &= (\lambda, t) = (\gamma, c, t) \in N. \end{aligned}$$

Уравнение маятника (7) имеет интеграл энергии $E = \frac{c^2}{2} - \cos \gamma \in [-1, +\infty)$. Рассмотрим следующее разбиение цилиндра C на взаимно непересекающиеся инвариантные подмножества фазового цилиндра маятника:

$$(8) \quad C = \bigcup_{i=1}^5 C_i,$$

$$\begin{aligned} C_1 &= \{\lambda \in C \mid E \in (-1, 1)\}, \\ C_2 &= \{\lambda \in C \mid E \in (1, +\infty)\}, \\ C_3 &= \{\lambda \in C \mid E = 1, c \neq 0\}, \\ C_4 &= \{\lambda \in C \mid E = -1\} = \{(\gamma, c) \in C \mid \gamma = 2\pi n, c = 0\}, \quad n \in \mathbb{N}, \\ C_5 &= \{\lambda \in C \mid E = 1, c = 0\} = \{(\gamma, c) \in C \mid \gamma = \pi + 2\pi n, c = 0\}. \end{aligned}$$

В работе [6] были введены эллиптические координаты (φ, k) в области $C_1 \cup C_2 \cup C_3$ цилиндра C , где k — перепараметризованная энергия, а φ — время движения маятника (7). В эллиптических координатах поток маятника (7) выпрямляется: $\dot{\varphi} = 1$, $\dot{k} = 0$, и получена следующая параметризация экстремальных траекторий. Далее используются функции Якоби $\text{am}(\varphi, k)$, $\text{sn}(\varphi, k)$, $\text{cn}(\varphi, k)$, $\text{dn}(\varphi, k)$, $E(\varphi, k)$; кроме того, $K(k)$ есть полный эллиптический интеграл первого рода [13].

Если $\lambda = (\varphi, k) \in C_1$, то $\varphi_t = \varphi + t$ и

$$\cos \theta_t = \operatorname{cn} \varphi \operatorname{cn} \varphi_t + \operatorname{sn} \varphi \operatorname{sn} \varphi_t,$$

$$\sin \theta_t = s_1(\operatorname{sn} \varphi \operatorname{cn} \varphi_t - \operatorname{cn} \varphi \operatorname{sn} \varphi_t),$$

$$(9) \quad \theta_t = s_1(\operatorname{am} \varphi - \operatorname{am} \varphi_t) \pmod{2\pi},$$

$$(10) \quad x_t = (s_1/k)[\operatorname{cn} \varphi(\operatorname{dn} \varphi - \operatorname{dn} \varphi_t) + \operatorname{sn} \varphi(t + \operatorname{E}(\varphi) - \operatorname{E}(\varphi_t))],$$

$$(11) \quad y_t = (1/k)[\operatorname{sn} \varphi(\operatorname{dn} \varphi - \operatorname{dn} \varphi_t) - \operatorname{cn} \varphi(t + \operatorname{E}(\varphi) - \operatorname{E}(\varphi_t))],$$

$$s_1 = \operatorname{sgn} \cos(\gamma/2).$$

В области C_2 удобно использовать координату $\psi = \varphi/k$, $\psi_t = \varphi_t/k = \psi + t/k$. Если $\lambda \in C_2$, то

$$(12) \quad \cos \theta_t = k^2 \operatorname{sn} \psi \operatorname{sn} \psi_t + \operatorname{dn} \psi \operatorname{dn} \psi_t,$$

$$(13) \quad \sin \theta_t = k(\operatorname{sn} \psi \operatorname{dn} \psi_t - \operatorname{dn} \psi \operatorname{sn} \psi_t),$$

$$(14) \quad x_t = s_2 k[\operatorname{dn} \psi(\operatorname{cn} \psi - \operatorname{cn} \psi_t) + \operatorname{sn} \psi(t/k + \operatorname{E}(\psi) - \operatorname{E}(\psi_t))],$$

$$(15) \quad y_t = s_2[k^2 \operatorname{sn} \psi(\operatorname{cn} \psi - \operatorname{cn} \psi_t) - \operatorname{dn} \psi(t/k + \operatorname{E}(\psi) - \operatorname{E}(\psi_t))],$$

$$s_2 = \operatorname{sgn} c.$$

Если $\lambda \in \cup_{i=3}^5 C_1$, то экстремальные траектории параметризуются элементарными функциями [6].

Рассмотрим следующее разбиение пространства $M = \operatorname{SE}(2) = \mathbb{R}_{x,y}^2 \times S_\theta^1$, определяемое значениями функций $R_1 = y \cos \frac{\theta}{2} - x \sin \frac{\theta}{2}$, $R_2 = x \cos \frac{\theta}{2} + y \sin \frac{\theta}{2}$:

$$\widetilde{M} = \{q \in M \mid R_1(q)R_2(q) \sin \theta \neq 0\}, \quad \widetilde{M} = \sqcup_{i=1}^8 M_i,$$

$$M' = \{q \in M \mid R_1(q)R_2(q) \sin \theta = 0\},$$

где каждое из множеств M_i определяется знаками функций $\sin \theta$, R_1 , R_2 , описанными в Таблице 1.

M_i	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
$\operatorname{sgn}(\sin \theta)$	-	-	-	-	+	+	+	+
$\operatorname{sgn}(R_1)$	+	+	-	-	-	-	+	+
$\operatorname{sgn}(R_2)$	+	-	-	+	+	-	-	+

ТАБЛИЦА 1. Определение областей M_i

В работах [7, 8] было получено глобальное описание времени разреза

$$t_{\text{cut}}(\lambda) = \sup\{t_1 > 0 \mid q_s \text{ оптимальна при } s \in [0, t_1]\}, \quad \lambda \in C,$$

вдоль экстремальных траекторий. В работе [8] были определены множества

$$\widehat{M} = M \setminus \{q_0\}, \quad \widehat{N} = \{(\lambda, t) \in N \mid t \leq t_{\text{cut}}(\lambda)\}.$$

Так как для любого $q_1 \in \widehat{M}$ существует оптимальное управление, то отображение $\text{Exp} : \widehat{N} \rightarrow \widehat{M}$ сюръективно; так как оно имеет кратные точки (Максвелла), то отображение не является инъективным. Далее, в работах [7, 8] были определены такие плотные подмножества $\widetilde{N} \subset \widehat{N}$ и их разбиение $\widetilde{N} = \sqcup_{i=1}^s D_i$, что экспоненциальное отображение имеет следующую глобальную структуру:

$$(16) \quad \text{Exp} : \widetilde{N} \rightarrow \widetilde{M} \text{ и все } \text{Exp} : D_i \rightarrow M_i \text{ — диффеоморфизмы.}$$

Поэтому поиск оптимальной кривой в задаче (3)–(6) для конечной точки $q_1 \in M_i$ сводится к решению системы уравнений $\text{Exp}(\lambda, t) = q_1$, имеющей единственный корень $(\lambda, t) \in D_i$.

Итак, в случае общего положения $q_1 \in \widetilde{M}$ решение задачи (3)–(6) сведено к решению систем уравнений

$$x_t(\lambda) = x_1, \quad y_t(\lambda) = y_1, \quad \theta_t(\lambda) = \theta_1, \quad (\lambda, t) \in D_i,$$

где функции x_t, y_t, θ_t — компоненты оптимальных траекторий q_t , описанные формулами (9)–(15), а $(x_1, y_1, \theta_1) = q_1$ — конечная точка искомой траектории.

5. Алгоритм восстановления изображения

В этом разделе описывается алгоритм `GlobalSolve` восстановления изображения на основе вариационного принципа (1), базирующийся на результатах, описанных в предыдущем разделе.

Алгоритм `GlobalSolve` состоит из следующих подалгоритмов:

- `RestoreDomain` восстанавливает поврежденные изофоты в подобласти $O_k, k = 1, \dots, N$, см. подраздел 5.4.
- `RemoveCusp` удаляет точки возврата у оптимальной траектории $(x(t), y(t))$ при помощи соответствующего подбора значения параметра α (1), см. подраздел 5.3.

- **FindRoot** по граничным условиям $q_1 = (x_1, y_1, \theta_1)$ (5) вычисляет параметры, определяющие соответствующую поврежденную изофоту $(x(t), y(t))$, см. подраздел 5.2.
- **Solver** решает систему из трех алгебраических уравнений с тремя неизвестными

$$(17) \quad \text{Exp}(\nu) = q_1, \quad q_1 \in M_i, \quad \nu \in D_i,$$

где $\nu \in D_i \cap C_j$, $j \in \{1, 2\}$, см. подраздел 5.1.

5.1. Алгоритм Solver решения системы уравнений (17)

Вход: $q_1 \in M_i$, $\text{initial} \in \{\text{true}, \text{false}\}$, $j \in \{1, 2\}$, $i \in \{1, \dots, 4\}$, $\nu \in D_i \cap C_j$.

Выход: $\nu \in D_i \cap C_j$.

Выполняемые действия: Алгоритм вычисляет значение корня $\nu \in D_i \cap C_j$ системы (17) с заданной точностью ε . Переменная ν является одновременно входным параметром (начальное приближение для корня при условии, что $\text{initial} = \text{true}$) и выходным параметром (искомый корень).

Константы алгоритма: $\text{maxiteration}, \text{maxiterrnd} \in \mathbb{N}$, $\varepsilon > 0$.

Шаги алгоритма:

- (1) Если $\text{initial} = \text{true}$, то выполняется переход к шагу 3.
- (2) В подобласти $D_i \cap C_j$ случайно выбирается начальное приближение ν .
- (3) Из начальной точки ν запускается итерационный алгоритм поиска приближенного значения корня системы (17). Если число итераций этого алгоритма превышает maxiteration , то выполняется переход к шагу 2. Если $\nu \notin D_i \cap C_j$, то выполняется переход к шагу 2. Если общее число итераций алгоритма **Solver** превышает maxiterrnd , то алгоритм завершается (корень не найден).
- (4) Если $|\text{Exp}(\nu) - q_1| < \varepsilon$, то алгоритм завершается и возвращается значение корня ν . Иначе выполняется переход к шагу 2.

5.2. Алгоритм FindRoot вычисления оптимальной траектории задачи (3)–(6)

Вход: $q_1 \in M$, $\text{initial} \in \{\text{true}, \text{false}\}$, $j \in \{1, 2\}$, $\nu \in C_j$.

Выход: $j \in \{1, 2\}$, $\nu \in C_j$.

Выполняемые действия: Алгоритм численно находит корень ν системы (17) и номер j такой области C_j , что $\nu \in C_j$. Параметр α (2) приравнивается 1. Переменные ν и j являются одновременно входными параметрами (начальное приближение корня и номер области, содержащей этот корень при условии $\text{initial} = \text{true}$) и выходными параметрами (искомый корень).

Шаги алгоритма:

- (1) С помощью Таблицы 1 вычисляется номер i такой области M_i , что $q_1 \in M_i$.
- (2) Если $\text{initial} = \text{true}$, то выполняется переход к шагу $(j+2)$, т. е. 3 или 4.
- (3) Запускается алгоритм Solver с параметрами q_1 , initial , i , $j = 1$, ν . Если корень найден, то алгоритм завершается успешно, иначе $\text{initial} := \text{false}$.
- (4) Запускается алгоритм Solver с параметрами q_1 , initial , i , $j = 2$, ν . Если корень найден, то алгоритм завершается успешно, иначе $\text{initial} := \text{false}$ и выполняется переход к шагу 3.

В итоге для корня $\nu = (\lambda, t_1)$ системы (17) вычисляется оптимальная траектория $q(t) = \text{Exp}(\lambda, t)$, $t \in [0, t_1]$, задачи (3)–(6). Однако оптимальная траектория задачи (3)–(6) может иметь точки возврата (см. рис. 1), которые непригодны при восстановлении изофот поврежденного изображения. Такие точки возврата в большинстве случаев могут быть удалены подбором подходящего значения параметра α (2), см. подраздел 5.3.

5.3. Алгоритм RemoveCusp удаления точек возврата на изофотах

Вход: $q_1 = (x_1, y_1, \theta_1) \in M$.

Выход: $\alpha > 0$, $j \in \{1, 2\}$, $\nu \in C_j$.

Выполняемые действия: Алгоритм находит такое значение $\alpha > 0$, что соответствующая оптимальная траектория $(x(t), y(t))$ для задачи (3)–(5) с функционалом качества

$$(18) \quad l_\alpha = \int_0^{t_1} \sqrt{u_1^2 + \alpha^2 u_2^2} dt \rightarrow \min$$

не имеет точек возврата. С помощью замены координат (2) новая задача (3)–(5), (18) сводится к изначальной (3)–(6).

Константы алгоритма: $\Delta\alpha = 0.1$,

$$\alpha_{\text{init}} = 0.6 \frac{\pi - |\theta_1 - \pi| + 0.7 \frac{|y_1|}{|x_1 + 0.2|}}{0.05 + \sqrt{x_1^2 + y_1^2}} + 0.2.$$

Шаги алгоритма:

- (1) Задаются начальные значения $\alpha := \alpha_{\text{init}}$, $\text{initial} := \text{false}$.
- (2) $(x_\alpha, y_\alpha) := (\alpha x_1, \alpha y_1)$.
- (3) Запускается алгоритм `FindRoot` с параметрами $(x_\alpha, y_\alpha, \theta_1)$, initial , j , ν .
- (4) Если корень ν , вычисленный с помощью `FindRoot`, соответствует кривой $(x(t), y(t))$ без точек возврата (т.е. $\dot{x}^2(t) + \dot{y}^2(t) \neq 0$), то алгоритм завершается успешно.
- (5) $\text{initial} := \text{true}$, $\alpha := \alpha + \Delta\alpha$ и выполняется переход к шагу 2.

Примеры восстановленных изображений с точками возврата и без точек возврата приведены на рис. 2 и 3 соответственно. Эти изображения получены с помощью алгоритмов `FindRoot` и `RemoveCusp`.

В этом алгоритме используется свойство диффеоморфности (16) отображения `Exp`: близкие точки прообраза экспоненциального отображения переходят под действием `Exp` в близкие точки его образа. Это свойство используется также далее в алгоритме `RestoreDomain`, см. подраздел 5.4.

5.4. Алгоритм `RestoreDomain` восстановления одной поврежденной области

Вход: $k \in \{1, \dots, N\}$.

Выход: текстовый файл `outputs<k>`.

Выполняемые действия: Номер k соответствует входному файлу `inputs<k>`, который содержит координаты конечных точек q_1 всех

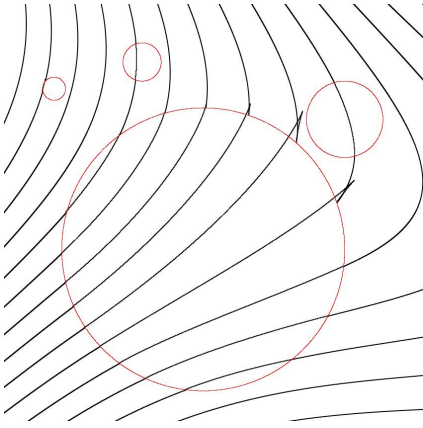


Рис. 2. Кривые с
точками возврата

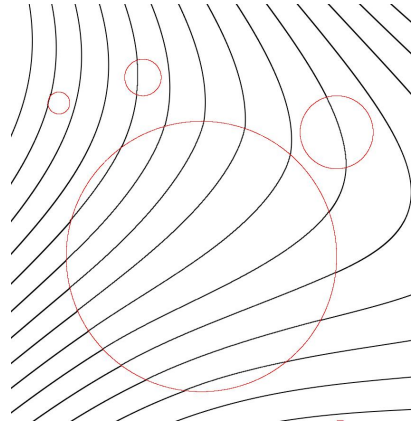


Рис. 3. Кривые
без точек возврата

изофот в поврежденной области $O_k \subset D$. Алгоритм находит все параметры для восстановления этих изофот и записывает их в файл `outputs<k>`.

Шаги алгоритма:

- (1) `initial := false`.
- (2) Из файла `inputs<k>` считываются входные данные с координатами точек $q_1 = (x_1, y_1, \theta_1)$ для текущей изофоты. Алгоритм завершается, если файл пустой.
- (3) Если `initial = true`, то $\alpha_0 := \alpha + \text{sign}(\alpha - \alpha_{\text{init}})\Delta\alpha$ и запускается алгоритм `FindRoot` с параметрами $(\alpha_0 x_1, \alpha_0 y_1, \theta_1)$, `initial`, ν, j . Если корень ν , вычисленный с помощью алгоритма `FindRoot`, соответствует кривой без точек возврата, то искомым корень найден, выполняется переход к шагу 5. Иначе полагается $\alpha_0 := \alpha_{\text{init}} + \text{sign}(\alpha_{\text{init}} - \alpha)\Delta\alpha$ и запускается алгоритм `FindRoot` с параметрами $(\alpha_0 x_1, \alpha_0 y_1, \theta_1)$, `initial`, ν, j . Если корень ν , найденный с помощью `FindRoot`, соответствует кривой без точек возврата, то искомым корень найден, выполняется переход к шагу 5.
- (4) Запускается алгоритм `RemoveCusp` с входным значением q_1 .
- (5) Параметры α, ν, j записываются в файл `outputs<k>`, выполняется переход к шагу 2.

Так как параметры для траекторий из разных поврежденных областей вычисляются независимо, то имеет смысл вычислять их параллельно. Это было реализовано в параллельном программном комплексе для восстановления изображений `OptimalInpainting`.

6. Заключение

В работе описан метод восстановления изофот штрихового изображения с помощью оптимальных кривых в левоинвариантной субримановой задаче на группе движений плоскости $SE(2)$. Метод и алгоритм, представленные в этой статье, были реализованы в параллельном программном комплексе `OptimalInpainting`. Описание этого программного комплекса и демонстрация эффективности подхода будут приведены в последующей работе.

Представленный метод восстановления может применяться при обработке изображений, содержащих окклюзию одного объекта другим, например: фотоснимков, содержащих частично скрытые контуры дорог, рек, других криволинейных объектов.

Список литературы

- [1] Chan T.F. Kang S.H. Shen J. *Euler's elastica and curvature based inpainting* // SIAM Journal of Applied Math. 2002. **63**, no. 2, p. 564–592 ↑1
- [2] Citti G. Sarti A. *A cortical based model of perceptual completion in the roto-translation space* // J. Math. Imaging Vision, 2006. **24**, no. 3, p. 307–326 ↑
- [3] Duits R. Franken E.M. *Left-invariant parabolic Evolutions on $SE(2)$ and Contour Enhancement via Invertible Orientation Scores. Part I: Linear Left-invariant Diffusion Equations on $SE(2)$* // Quarterly of Applied Mathematics, June 2008. **68**, no. 2, p. 255–292 ↑1
- [4] Petitot J. *The neurogeometry of pinwheels as a sub-Riemannian contact structure* // J. Physiology — Paris, 2003, no. 97, p. 265–309 ↑1, 2
- [5] Petitot J. *Neurogeometrie de la vision. Modeles mathematiques et physiques des architectures fonctionelles* : Editions de l'Ecole Polytechnique, 2008. ↑1, 2
- [6] Moiseev I. Sachkov Yu.L. *Maxwell strata in sub-Riemannian problem on the group of motions of a plane* // ESAIM: Control, Optimisation and Calculus of Variations, 29 July 2010. **16**, p. 380–399 ↑1, 3, 4, 4, 4
- [7] Sachkov Yu.L. *Conjugate and cut time in the sub-Riemannian problem on the group of motions of a plane* // ESAIM: Control, Optimisation and Calculus of Variations, 2010. **16**, p. 1018–1039 ↑4
- [8] Sachkov Yu.L. *Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane* // ESAIM: Control, Optimisation and Calculus of Variations, 2011, no. 17, p. 293–321 ↑1, 4, 4

- [9] Сачков Ю. Л., Ардентов А. А., Маштаков А. П. *Параллельный алгоритм и программа восстановления изофот для поврежденных изображений* // Программные системы: теория и приложения, 2010. **1**, № 1, с. 3–20 ↑[2](#)
- [10] Аграчев А. А., Сачков Ю. Л. *Геометрическая теория управления*. М. : Физматлит, 2005. ↑[3](#)
- [11] Montgomery R. *A Tour of Subriemannian Geometries, Their Geodesics and Applications* : American Mathematical Society, 2002. ↑[3](#)
- [12] Laumond J.P. *Nonholonomic motion planning for mobile robots. Lecture notes in Control and Information Sciences*, Vol. **256** : Springer, 1998. ↑[3](#)
- [13] Whittaker E.T. Watson G. N. *A Course of Modern Analysis. An introduction to the general theory of infinite processes and of analytic functions; with an account of principal transcendental functions*. Cambridge : Cambridge University Press, 1996. ↑[4](#)

A. A. Ardentov, Yu. L. Sachkov. *Antropomorphic recovery of corrupted images via methods of sub-Riemannian geometry*.

АБСТРАКТ. According to neurogeometry of vision, human brain recovers isophotes of corrupted images via optimal curves for a certain optimal control problem — sub-Riemannian problem on the group of motions of the lane. The paper describes an algorithm for antropomorphic inpainting of corrupted binary images on the basis of optimal synthesis for this problem.

Key Words and Phrases: image inpainting, optimal control, sub-Riemannian geometry.

Образец ссылки на статью:

А.А.Ардентов, Ю.Л.Сачков. *Антропоморфное восстановление поврежденных изображений на основе методов субримановой геометрии* // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 4(8), с. 3–15. URL: http://psta.psisiras.ru/read/psta2011_4_3-15.pdf