Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2013.mssvm05

Vol. **6**, No. 1, pp. 95-115
February 2013

Parallel Algorithm and Software for Image Inpainting via Sub-Riemannian Minimizers on the Group of Rototranslations

Alexey P. Mashtakov, Andrei A. Ardentov and Yuri L. Sachkov*

Program Systems Institute, Pereslavl-Zalessky, Russia.

Received 6 December 2011; Accepted (in revised version) 12 July 2012

Available online 11 January 2013

Abstract. The paper is devoted to an approach for image inpainting developed on the basis of neurogeometry of vision and sub-Riemannian geometry. Inpainting is realized by completing damaged isophotes (level lines of brightness) by optimal curves for the left-invariant sub-Riemannian problem on the group of rototranslations (motions) of a plane SE(2). The approach is considered as anthropomorphic inpainting since these curves satisfy the variational principle discovered by neurogeometry of vision. A parallel algorithm and software to restore monochrome binary or halftone images represented as series of isophotes were developed. The approach and the algorithm for computation of completing arcs are presented in detail.

AMS subject classifications: 65M10, 78A48

Key words: Image inpainting, sub-Riemannian geometry, neurogeometry of vision, group of rototranslations of a plane, parallel software.

1. Image inpainting, neurogeometry of vision and sub-Riemannian geometry

The task of restoring damaged or latent images is one of challenging problems in computer graphics, photo restoration, film and painting. A number of methods were suggested to solve this problem, many of which are based on advanced mathematical techniques, in particular, on the application of the calculus of variations and optimal control [1–4, 23, 24, 29].

This paper is based on provisions a new direction of neuroscience — neurogeometry [5, 6], as well as recent results on sub-Riemannian geometry [7–9]. On the basis of results of these studies were developed an algorithm and a set of parallel software to restore monochrome binary or halftone images represented as series of isophotes (level lines of brightness).

*Corresponding author. *Email addresses:* alexey.mashtakov@gmail.com (A. P. Mashtakov), aaa@pereslavl.ru (A. A. Ardentov), sachkov@sys.botik.ru (Y. L. Sachkov)

1.1. Neurogeometry of vision

An important discovery of neurophysiology of the beginning of this century is a geometric structure corresponding to the primary visual cortex of the human brain. The primary visual cortex performs a primary (preceding any treatment) perception of visual information by the human brain. It was established [5,6] that in order to store images, the primary cortex simulates the contact structure $\{(x,y,p)\} = D \times \mathbb{R}P^1$ on the surface of the retina $D \subset \mathbb{R}^2$. Here, the tangent element p is the slope of the curve $y(x)$ at the point x, and $\mathbb{R}P^1$ is the projective line (the space of all lines in $\mathbb{R}^2$ passing through the origin). It turned out that for effective imaging, for the human brain it is profitable to keep a contour not as a set of successive points (x_i, y_i), but as a set of strokes (x_i, y_i, p_i), in the limit — in the form of a continuous curve $(x(t), y(t), p(t))$, $p = dy/dx$. If part of the curve is damaged or hidden from observation, the missing arc is restored on the basis of the following variational principle: the restored arc should have minimum Euclidean length in the space of contact elements (x, y, θ), $\theta = \arctan p$:

$$\int \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{\theta}^2}\, dt \to \min. \tag{1.1}$$

(If the Euclidean length would be computed in the plane (x, y), then the arc would be trivially and erroneously restored by a straight line segment). The variational principle (1.1) is taken in this work as a basis of the method of restoring a hidden arc. The described internal geometry of the visual cortex is one of the main objects of study of neurogeometry of vision — a direction of neurophysiology which studies the geometric structure of the human brain simulating the spatial images of the external world.

1.2. Statement of the problem of image reconstruction and method of solution

We consider the problem of recovering a monochrome (binary of gray-scale) image, some fragments of which are corrupted or hidden from observation. The goal is to restore the damaged parts of the image in an anthropomorphic (natural for a human being) way. Mathematically, the problem can be formalized as follows. Given a domain $D \subset \mathbb{R}^2$, mutually disjoint subdomains

$$O_1, \cdots, O_N \subset D, \tag{1.2}$$

and a function $f : D \setminus (\bigcup_{i=1}^N O_i) \to [0,1]$, one should restore the function f in the domains $O_1, \cdots, O_N$. Here D is the domain of the initial image, O_i are subdomains with corrupted parts of image, and the function f determines the image (brightness for gray-scale image, and for binary image it is a function, whose level lines coincide with the curves constituting the image). We propose to restore the image in subdomains O_i by completing isophotes — level curves of f in these subdomains (in the case of halftone images, the strips between the reconstructed curves are painted according to the brightness values on these curves). The reconstructing curves are calculated via the variational approach (1.1): the constructed curve $(x(t), y(t))$ should minimize the distance in the space (x, y, θ), where (x, y) are coordinates in the plane $\mathbb{R}^2$ and $\theta(t) = \arctan p(t) = \arctan(\dot{y}(t)/\dot{x}(t))$ is the slope of the tangent to the curve $(x(t), y(t))$. In this work we describe an algorithm for solving the corresponding optimal control problem on the basis of papers [7–9]. Then we present parallel software `OptimalInpaiting` developed for testing our method of image reconstruction.

Our approach seems successful under the following assumptions:

1. Isophotes of an image can be represented by level lines of a smooth function $f : D \to [0,1]$.
2. The function f has no critical points in the corrupted subdomains $O_1, \cdots, O_N \subset D$.
3. Information on intersection points of level lines of f with boundaries of O_i can be effectively extracted from the image.
4. During inpainting, the way of pairing isophotes is known.

We believe that under these assumptions the proposed method can be successfully used in combination with other methods of image inpainting. The assumptions may seem rather restrictive, but some of them can be removed in future work; a discussion of this is given in Conclusions section at the end of the paper.

2. Reconstruction of isophotes via variational approach

Consider a smooth flat curve $AB = \{(x(t), y(t)) \mid t \in [a, b]\}$. Suppose that a part of this curve $CD = \{(x(t), y(t)) \mid t \in [c, d]\}$, $a < c < d < b$, is hidden from observation or damaged. To restore the curve CD, construct the tangent T_C to the curve AC at the point C and the tangent T_D at the point D, see Fig. 1. Denote by θ_c, θ_d the slope of the tangents T_C, T_D. The required curve $\overline{CD} = \{(\overline{x}(t), \overline{y}(t)) \mid t \in [c, d]\}$ should start at the point C with the slope θ_c, terminate at the point D with the slope θ_d, and have the minimum Euclidean length in the space (x, y, θ), see (1.1). The boundary conditions imply a smooth conjugation of the restored curve $\overline{CD}$ with the known arcs AC and DB of the original curve. The original and restored curves are shown in Fig. 2. The minimum condition (1.1) formalizes a natural condition for the new curve $\overline{CD}$: in its search, large deviations are penalized both in the coordinates (x, y), and in the angle θ. Thus, there is minimized a certain integral compromise between the linear and angular velocities of the curve. Moreover, one can minimize a more general length functional reflecting the different weights of space variables (x, y) and the angle variable θ:

$$\int \sqrt{\dot{x}^2 + \dot{y}^2 + \alpha^2\, \dot{\theta}^2}\, dt \to \min, \qquad \alpha > 0, \tag{2.1}$$

this extension is easily performed by the change of variables

$$x = \alpha \widetilde{x}, \qquad y = \alpha \widetilde{y}, \qquad \theta = \widetilde{\theta}. \tag{2.2}$$

The parameter α in (2.1) has dimension of length, and it corresponds to the choice of scale in the plane (x, y).

Examples of curves restored via the approach described are presented in Figs. 3 and 4.

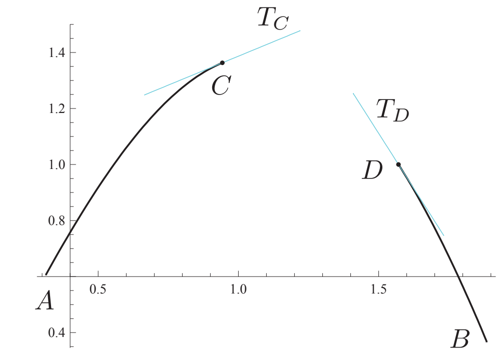


Figure 1: Boundary conditions for restoration of the arc CD.

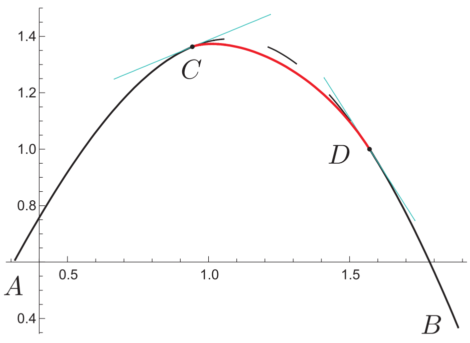


Figure 2: Curve *AB* with original and restored arcs CD.

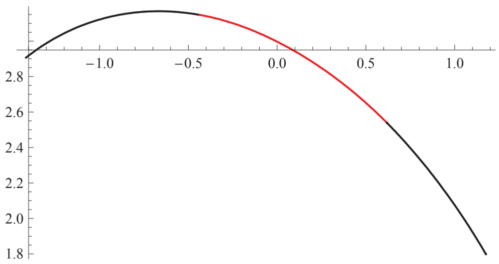


Figure 3: Restored convex isophote.

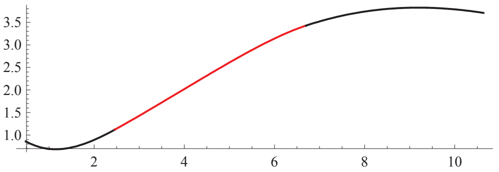


Figure 4: Restored isophote with inflection point.

3. Sub-Riemannian geometry on the group of motions of a plane

Problem (1.1) is formalized as the following optimal control problem [10]:

$$\dot{x} = u_1 \cos\theta, \qquad \dot{y} = u_1 \sin\theta, \qquad \dot{\theta} = u_2, \tag{3.1}$$

$$q = (x, y, \theta) \in M = \mathbb{R}^2_{x,y} \times S^1_\theta, \qquad u = (u_1, u_2) \in \mathbb{R}^2, \tag{3.2}$$

$$q(0) = q_0 = (0,0,0), \qquad q(t_1) = q_1 = (x_1, y_1, \theta_1), \tag{3.3}$$

$$l = \int_0^{t_1} \sqrt{u_1^2 + u_2^2}\, dt \to \min. \tag{3.4}$$

Notice that the state space of problem (3.1)-(3.4) is $\mathbb{R}^2 \times S^1$, not $\mathbb{R}^2 \times \mathbb{R}P^1$ as in Section 1. The two problems on $\mathbb{R}^2 \times S^1$, not $\mathbb{R}^2 \times \mathbb{R}P^1$ have the same dynamics and cost functional, but differ one from another by the range of the variable θ. For the problem on $\mathbb{R}^2 \times S^1$, the angle θ is the angle between the tangent vector to the curve $(x(t), y(t))$ and the positive direction of the axis x, thus $\theta \in S^1 = \mathbb{R}/(2\pi\mathbb{Z})$. And for the problem on $\mathbb{R}^2 \times \mathbb{R}P^1$, the angle θ is the angle between the tangent line to the curve $(x(t), y(t))$ and the axis x, thus $\theta \in \mathbb{R}P^1 = \mathbb{R}/(\pi\mathbb{Z})$. Each solution to the problem on $\mathbb{R}^2 \times \mathbb{R}P^1$ can be obtained by the natural projection $S^1 \to \mathbb{R}P^1$ of solutions of two lifted problems on $\mathbb{R}^2 \times S^1$, for θ and $\theta+\pi$. On the other hand, it turns out that solving the problem on $\mathbb{R}^2 \times S^1$ is easier than on $\mathbb{R}^2 \times \mathbb{R}P^1$.

The state space of problem (3.1)-(3.4) is $M = \mathbb{R}^2_{x,y} \times S^1_\theta$, and it is naturally identified with the group SE(2) of orientation-preserving motions (rototranslations) of a two-dimensional plane, which is represented by 3 × 3 matrices as follows:

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

Then problem (3.1)-(3.4) is obviously reformulated as a left-invariant sub-Riemannian problem on the Lie group SE(2) [10, 11]. Consider a rank 2 nonintegrable left-invariant sub-Riemannian structure on SE(2), i.e., a rank 2 nonintegrable left-invariant distribution Δ on SE(2) with a left-invariant inner product $\langle\cdot,\cdot\rangle$ on Δ. One can easily show that such a structure is unique, up to a constant scalar factor in the inner product. We choose the following model for such a sub-Riemannian structure:

$$\Delta_q = \mathrm{span}(X_1(q), X_2(q)), \qquad \langle X_i, X_j\rangle = \delta_{ij}, \quad i,\ j = 1,2, \tag{3.5}$$

$$X_1(q) = qE_{13}, \qquad X_2(q) = q(E_{21} - E_{12}), \qquad q \in \mathrm{SE}(2), \tag{3.6}$$

(where E_{ij} denotes the 3 × 3 matrix with identity entry in row i and column j, and zero entries elsewhere) and study the corresponding optimal control problem:

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

Vector fields (3.6) are written as differential operators on SE(2) in the following form:

$$X_1 = \cos\theta \frac{\partial}{\partial x} + \sin\theta \frac{\partial}{\partial y}, \qquad X_2 = \frac{\partial}{\partial\theta}.$$

The problem can be reformulated in robotics terms as follows. Consider a mobile robot in the plane that can move forward and backward, and rotate around itself (Reeds-Shepp car) [12]. The state of the robot is described by coordinates (x, y) of its center of mass and by angle of orientation θ. Given an initial and a terminal state of the car, one should find the shortest path from the initial state to the terminal one, when the length of the path is measured in the space (x, y, θ), see Fig. 5.

Problem (3.1)-(3.4) is important for hypoelliptic diffusion equation on SE(2) [13–16, 30–32].

It was shown in work [7] that problem (3.1)-(3.4) has an optimal solution for any terminal point $q_1 \in \mathrm{SE}(2)$.

Notice that sub-Riemannian problem (3.1)-(3.4) was also considered in [2, 14] by using another parameterization of extremal curves. Although, no study of optimality of extremal trajectories was performed in that work.

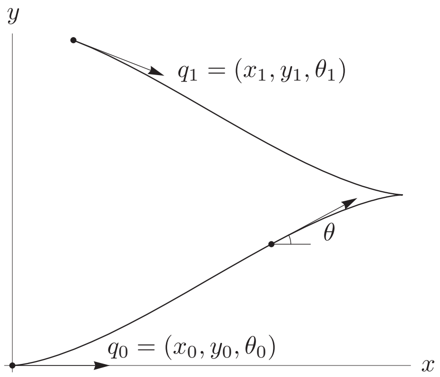


Figure 5: Statement of problem (3.1)-(3.4).

3.1. Reduction of problem (3.1)-(3.4) to solving systems of equations

In this subsection we describe some results of papers [7–9] that allow us to reduce problem (3.1)-(3.4) to solving systems of equations in elliptic functions.

Via Pontryagin Maximum Principle [10, 17], extremal trajectories in problem (3.1)-(3.4) are parameterized by points of the phase cylinder $C = (2S^1_\gamma) \times \mathbb{R}_c$ of the pendulum

$$\dot\gamma = c, \qquad \dot c = -\sin\gamma. \tag{3.7}$$

The family of arc-length parametrized extremal trajectories in problem (3.1)-(3.4) is described by the exponential mapping

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

The equation of pendulum (3.7) has the energy integral $E = c^2/2 - \cos\gamma \in [-1, +\infty)$. Consider the following decomposition of the cylinder C into disjoint invariant sets of the pendulum:

$$C = \bigsqcup_{i=1}^{5} C_i, \tag{3.8}$$

where

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

In work [7] were introduced elliptic coordinates (φ, k) on the domain $C_1 \cup C_2 \cup C_3$ of the cylinder C, where k is a reparametrized energy, and φ is the time of motion of the pendulum (3.7). In the elliptic coordinates the flow of the pendulum (3.7) rectifies: $\dot\varphi = 1$, $\dot k = 0$. Using these coordinates we obtained the following parametrization of extremal trajectories.

If $\lambda = (\varphi, k) \in C_1$, then $\varphi_t = \varphi + t$ and:

$$\begin{aligned}
\cos\theta_t &= \operatorname{cn}\varphi \operatorname{cn}\varphi_t + \operatorname{sn}\varphi \operatorname{sn}\varphi_t,\\
\sin\theta_t &= s_1(\operatorname{sn}\varphi \operatorname{cn}\varphi_t - \operatorname{cn}\varphi \operatorname{sn}\varphi_t),\\
\theta_t &= s_1(\operatorname{am}\varphi - \operatorname{am}\varphi_t) \pmod{2\pi},\\
x_t &= (s_1/k)[\operatorname{cn}\varphi(\operatorname{dn}\varphi - \operatorname{dn}\varphi_t) + \operatorname{sn}\varphi(t + \mathrm{E}(\varphi) - \mathrm{E}(\varphi_t))],\\
y_t &= (1/k)[\operatorname{sn}\varphi(\operatorname{dn}\varphi - \operatorname{dn}\varphi_t) - \operatorname{cn}\varphi(t + \mathrm{E}(\varphi) - \mathrm{E}(\varphi_t))].
\end{aligned}$$

In the domain C_2, it will be convenient to use the coordinate $\psi = \varphi/k$, $\psi_t = \varphi_t/k = \psi + t/k$. If $\lambda \in C_2$, then:

$$\begin{aligned}
\cos\theta_t &= k^2 \operatorname{sn}\psi \operatorname{sn}\psi_t + \operatorname{dn}\psi \operatorname{dn}\psi_t,\\
\sin\theta_t &= k(\operatorname{sn}\psi \operatorname{dn}\psi_t - \operatorname{dn}\psi \operatorname{sn}\psi_t),\\
x_t &= s_2 k[\operatorname{dn}\psi(\operatorname{cn}\psi - \operatorname{cn}\psi_t) + \operatorname{sn}\psi(t/k + \mathrm{E}(\psi) - \mathrm{E}(\psi_t))],\\
y_t &= s_2[k^2 \operatorname{sn}\psi(\operatorname{cn}\psi - \operatorname{cn}\psi_t) - \operatorname{dn}\psi(t/k + \mathrm{E}(\psi) - \mathrm{E}(\psi_t))].
\end{aligned}$$

Here and below we use Jacobi's functions $\operatorname{am}(\varphi, k)$, $\operatorname{cn}(\varphi, k)$, $\operatorname{sn}(\varphi, k)$, $\operatorname{dn}(\varphi, k)$, $\mathrm{E}(\varphi, k)$; moreover, $K(k)$ is the complete elliptic integral of the first kind [18]. If $\lambda \in \cup_{i=3}^{5} C_1$, then extremal trajectories are parameterized by elementary functions [7].

Consider the following decomposition of $M = \mathrm{SE}(2) = \mathbb{R}^2_{x,y} \times S^1_\theta$ depending on values of the functions $R_1 = y\cos\frac{\theta}{2} - x\sin\frac{\theta}{2}$, $R_2 = x\cos\frac{\theta}{2} + y\sin\frac{\theta}{2}$:

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

where each of the sets M_i is determined by constant signs of the functions $\sin\theta$, R_1, R_2 described in Table 1.

Table 1: Definition of domains M_i.

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

The preimage of the exponential mapping is the Cartesian product $N = C \times \mathbb{R}_+$. In [8,9] was the following global description of the cut time is presented:

$$t_{\mathrm{cut}}(\lambda) = \sup\{t_1 > 0 \mid q_s \text{ is optimal for } s \in [0, t_1]\}, \qquad \lambda \in C,$$

along extremal trajectories of problem (3.1)-(3.4). In work [9] were defined the sets

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

Since for any $q_1 \in M$ an optimal control exists, then the mapping $\mathrm{Exp} : \widehat{N} \to \widehat{M}$ is surjective; although, it has multiple (Maxwell) points, thus it is not injective. Further, in works [8,9] were defined an open dense subset $\widetilde{N} \subset \widehat{N}$, and its decomposition into disjoint subsets $\widetilde{N} = \sqcup_{i=1}^{8} D_i$, such that the exponential mapping has the following global structure:

$$\mathrm{Exp} : \widetilde{N} \to \widetilde{M} \ \text{ and all } \ \mathrm{Exp} : D_i \to M_i \ \text{ are diffeomorphisms.} \tag{3.9}$$

4. Algorithm for image inpainting

This section describes the algorithm `GlobalSolve` of restoring images via the variational principle (2.1) on the basis of results described in the previous section. The general structure of the algorithm is presented in Fig. 6. From the top to the bottom, `GlobalSolve` consists of the following subalgorithms:

- `RestoreDomain` restores corrupted isophotes in a subdomain O_k, $k = \overline{1,N}$ (1.2), see Subsection 4.4.

- `FindRoot` numerically evaluates parameters that determine a corrupted isophote $(x(t), y(t))$ from its boundary conditions $q_1 = (x_1, y_1, \theta_1)$ (3.3), see Subsection 4.2.

- `Solver` solves a system of 3 algebraic equations in 3 elliptic functions

  $$\mathrm{Exp}(\nu) = q_1, \qquad q_1 \in M_i, \quad \nu \in D_i \tag{4.1}$$

  for its root $\nu \in D_i \cap C_j$, $j \in \{1,2\}$, see Subsection 4.1.

- `RemoveCusp` aims to remove cusps at optimal trajectories $(x(t), y(t))$ by appropriate change of the parameter α (2.1), see Subsection 4.3.

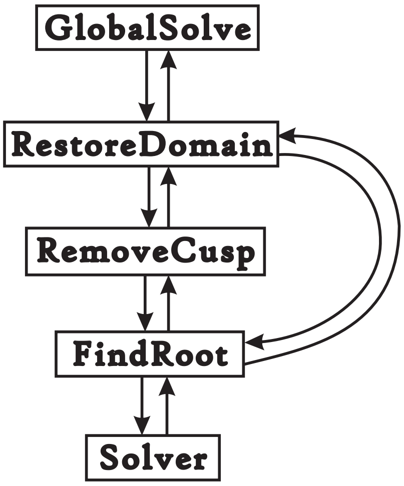


Figure 6: General structure of the algorithm.

The algorithm GlobalSolve was realized by a parallel software OptimalInpaiting described below.

4.1. Algorithm Solver for solving system of Eq. (4.1)

Input: $q_1 \in M_i$, initial $\in$ {true, false}, $j \in \{1,2\}$, $i \in \{1,\cdots,4\}$, $v \in D_i \cap C_j$.
Output: $v \in D_i \cap C_j$.
Performed actions: The algorithm searches numerically for an approximate root $v \in D_i \cap C_j$ of system (4.1) with a prescribed accuracy ε. The variable v is both an input parameter (initial approximation of the root in the case initial = true) and an output parameter (the root found).
Constants of the algorithm: maxiteration, maxiterrnd $\in \mathbb{N}$, $\varepsilon > 0$.

Steps of the algorithm:

1. If initial = true, then go to step 3.

2. Initial approximation v is selected randomly in the subdomain $D_i \cap C_j$.

3. An iterative algorithm of search for an approximate root of system (4.1) is started from the initial point v. If the number of iterations of this algorithm exceeds the value of maxiteration, then go to step 2. If $v \notin D_i \cap C_j$, then go to step 2. If the total number of iterations of the algorithm Solver exceeds maxiterrnd, then the algorithm terminates (the root was not found).

4. If $|\mathrm{Exp}(v) - q_1| < \varepsilon$, then the algorithm terminates and returns the value of the root v.

4.2. Algorithm FindRoot for computing an optimal trajectory of (3.1)-(3.4)

Input: $q_1 \in M$, initial $\in$ {true, false}, $j \in \{1,2\}$, $v \in C_j$.
Output: $j \in \{1,2\}$, $v \in C_j$.
Performed actions: The algorithm finds numerically a root v of system (4.1) and the number j of the domain C_j such that $v \in C_j$. The parameter α (2.2) is set equal to 1. The variables v and j are both input parameters (initial approximation of the root and the number of the domain which contains this root in the case initial = true) and output parameters (the root found).

Steps of the algorithm:

1. The number i of the domain M_i such that $q_1 \in M_i$ is computed by Table 1.

2. If initial = true, then go to step $(j+2)$, i.e., 3 or 4.

3. The algorithm Solver is applied with the parameters q_1, initial, $i, j = 1, v$. If the root is found, then the algorithm terminates successfully, otherwise initial := false.

4. The algorithm Solver is applied with the parameters q_1, initial, $i, j = 2, v$. If the root is found, then the algorithm terminates successfully, otherwise initial := false and go to step 3.

Given a root $\nu = (\lambda, t_1)$ of system (4.1), the corresponding optimal trajectory of problem (3.1)-(3.4) is $q(t) = \mathrm{Exp}(\lambda, t)$, $t \in [0, t_1]$. Although, optimal trajectories of problem (3.1)-(3.4) may have cusps, which seems not appropriate for isophotes of images being reconstructed. Recall that a curve $(x(t), y(t))$ has cusp at a point $(x(\tau), y(\tau))$ if

$$(\dot{x}^2 + \dot{y}^2)(\tau) = 0, \qquad \lim_{t \to \tau - 0} \frac{(\dot{x}, \dot{y})}{\sqrt{\dot{x}^2 + \dot{y}^2}}(t) = - \lim_{t \to \tau + 0} \frac{(\dot{x}, \dot{y})}{\sqrt{\dot{x}^2 + \dot{y}^2}}(t).$$

An example of a solution to problem (3.1)-(3.4) with a cusp is given at Fig. 5.

One can often remove cusps by changing the parameter α (2.2), see Subsection 4.3.

4.3. Algorithm `RemoveCusp` for removal of cusps on isophotes

`Input:` $q_1 = (x_1, y_1, \theta_1) \in M$.
`Output:` $\alpha > 0$, $j \in \{1, 2\}$, $\nu \in C_j$.
`Performed actions:` The algorithm searches for a value of $\alpha > 0$ such that the optimal trajectory $(x(t), y(t))$ of the problem (3.1)-(3.3) with the cost functional

$$l_\alpha = \int_0^{t_1} \sqrt{u_1^2 + \alpha^2 u_2^2}\, dt \to \min \tag{4.2}$$

has no cusps. The new problem (3.1)-(3.3), (4.2) is reduced to the original problem (3.1)–(3.4) by the change of coordinates (2.2).
`Constants of the algorithm:` $\Delta\alpha = 0.1$,

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

The constant α was chosen empirically on the basis of results of numerical simulations.

`Steps of the algorithm:`

1. Initial values of $\alpha := \alpha_{\mathtt{init}}$, `initial` $:=$ `false` are set.
2. $(x_\alpha, y_\alpha) := (\alpha x_1, \alpha y_1)$.
3. The algorithm `FindRoot` is applied with the parameters $(x_\alpha, y_\alpha, \theta_1)$, `initial`, j, ν.
4. If the root ν computed by `FindRoot` corresponds to a curve $(x(t), y(t))$ without cusps (i.e., $\dot{x}^2(t) + \dot{y}^2(t) \neq 0$), then the algorithm terminates successfully.
5. `initial` $:=$ `true`, $\alpha := \alpha + \Delta\alpha$ and go to step 2.

Examples of recovered images with and without cusps are given in Figs. 7 and 8. These images were obtained respectively without and with change of the parameter α.

The diffeomorphic property (3.9) of the mapping Exp is used in this algorithm: close points in the preimage of the exponential mapping are transformed by Exp to close points in its image. The same property (3.9) is used below in the algorithm `RestoreDomain`, see Subsection 4.4.

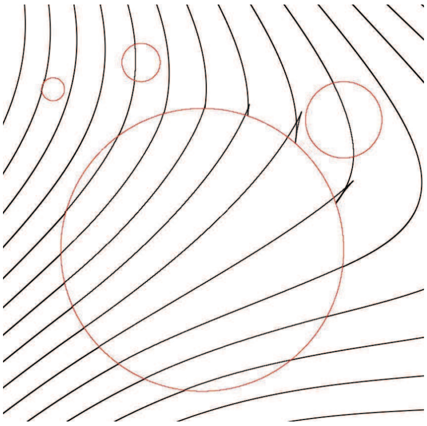


Figure 7: Recovered image with cusps.

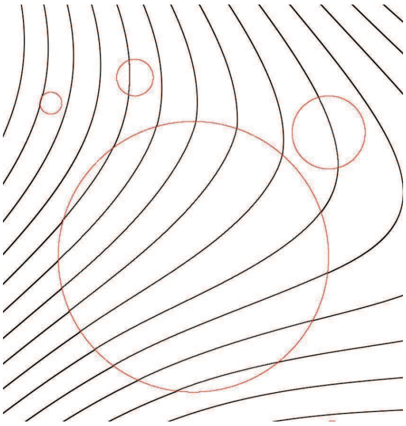


Figure 8: Recovered image without cusps.

4.4. Algorithm RestoreDomain for computing isophotes in a corrupted domain

Input: $k \in \mathbb{N}$.

Output: text file outputs<k>.

Performed actions: The number k corresponds to the input file inputs<k> which contains coordinates of endpoints q_1 of all isophotes for the corrupted domain $O_k \subset D$ (1.2). The algorithm finds all the parameters for restoring these isophotes and writes them to the file outputs<k>.

Steps of the algorithm:

1. initial := false.

2. The input data $q_1 = (x_1, y_1, \theta_1)$ for the current isophote is taken from the file inputs<k>. If the file is empty, then the algorithm terminates.

3. If initial = true, then $\alpha_0 := \alpha + \mathtt{sign}(\alpha - \alpha_{\mathtt{init}})\Delta\alpha$ and FindRoot is applied with the parameters $(\alpha_0 x_1, \alpha_0 y_1, \theta_1)$, initial, v, j. If the root v computed by FindRoot corresponds to a curve without cusps, then the required root is found, go to step 5. Otherwise $\alpha_0 := \alpha_{\mathtt{init}} + \mathtt{sign}(\alpha_{\mathtt{init}} - \alpha)\Delta\alpha$ and the algorithm FindRoot is applied with the parameters $(\alpha_0 x_1, \alpha_0 y_1, \theta_1)$, initial, v, j. If the root v computed by FindRoot corresponds to a curve without cusps, then the required root is found, go to step 5.

4. The algorithm RemoveCusp is applied with the input q_1.

5. The parameters α, v, j are written to the file outputs<k>, go to step 2.

Since the calculation of the parameters for the trajectories from different corrupted domains occurs independently, then it makes sense to evaluate them in parallel. This was realized in a parallel software for image inpainting described below.

5. Parallel software `OptimalInpaiting` for image reconstruction

To test the approach for image inpainting described above, we developed a parallel software `OptimalInpaiting` for restoration of corrupted monochrome binary or halftone images represented as series of isophotes (level lines of brightness). Computation of recovered isophotes is based on the algorithm described in Section 4. The following assumptions are adopted in development of the software:

1. Analytical data on corrupted image are known to the user.
2. Original image can be represented by a family of level lines of a smooth function without critical points.
3. Corrupted domains of the image are disks.

These assumptions can be weakened, see discussion in Section 6.

5.1. `OptimalInpaiting` : Logical structure

The structure of `OptimalInpaiting` software is shown in Fig. 9. A user works with the software through an interface. First, the software creates the original image and forms the corrupted subdomains for the image. After that the corrupted subdomains are applied to image. The result is a corrupted image in accordance with the parameters chosen by the user. Then tasks are being created, each of which corresponds to a corrupted isophote

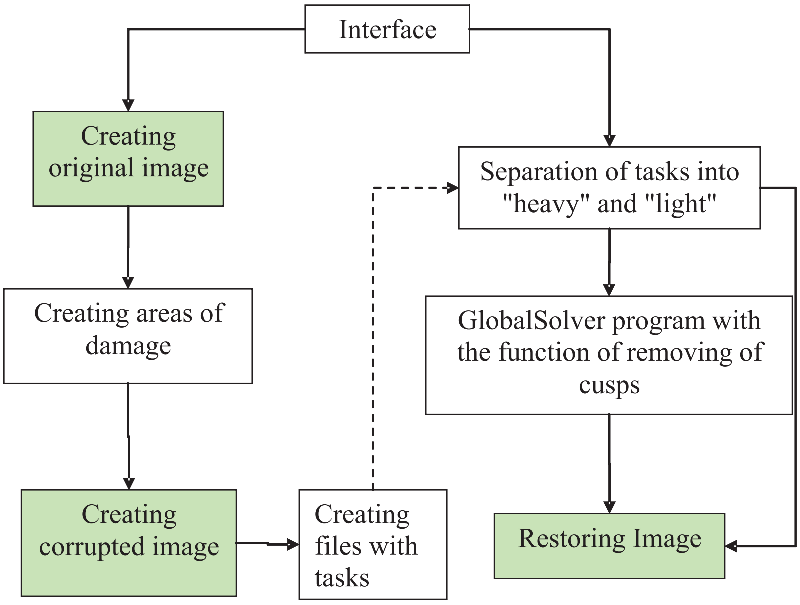

Interface
Creating original image
Separation of tasks into "heavy" and "light"
Creating areas of damage
GlobalSolver program with the function of removing of cusps
Creating corrupted image
Creating files with tasks
Restoring Image

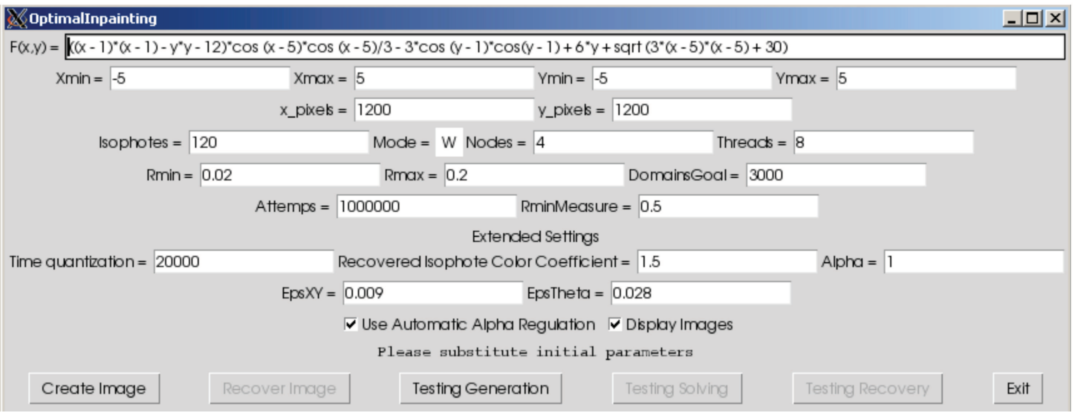
Figure 9: Structure of `OptimalInpaiting` .

(level line of brightness) in corrupted subdomains. Parameters of the tasks are calculated on the basis of information about corrupted subdomains. Tasks are divided into "heavy" and "light": the task is considered light if the boundary conditions define a straight-line isophote (up to a user-specified threshold), otherwise the task is considered heavy. Straight lines are taken as a solution of light tasks. The boundary conditions for heavy tasks are split into files corresponding to the corrupted subdomains. Each problem is solved as an optimal control problem, which was reduced to solving systems of algebraic equations as was shown above (also see [7–9]). The system of equations is solved numerically using the algorithm `Solver` (see Section 4.1). The solution curve corresponding to the parameters found may have cusps (i.e., be not smooth), see Fig. 7. To eliminate the non-smoothness, the parameter α is used, see (4.2). The structure of `GlobalSolve` and the use of the parameter α is described above in Section 4. The output of the program `GlobalSolver` (realization of algorithm `GlobalSolve` in C language) is an output file, which lists the values of the parameters that define the optimal curves for the respective tasks. According to the obtained values, recovering isophotes are constructed and applied to the corrupted image. As a result, the output of `OptimalInpaiting` obtains the restored image.

5.2. Input parameters

User interaction with the application is performed via a graphical interface designed in language Tcl/Tk. Fig. 10 shows the screenshot of input form of the interface that specifies the parameters required for the application. All input data are logically divided into 4 groups:

1. Defining the original image:
   - `F(x,y)` — a function that determines the original image as phase portrait of its level lines,
   - `Xmin`, `Xmax`, `Ymin`, `Ymax` — determine a domain of variables x and y,
   - `x_pixels`, `y_pixels` — width and height of image in pixels,
   - `Isophotes` — number of level lines of the function $F(x,y)$ on the domain $(x,y) \in$ `[Xmin,XMax]` × `[YMin,YMax]`,
   - `Mode` — the value is "W" for black and white images and "G" for greyscale images.
2. Defining the process of calculating the parameters of curves recovering isophotes (`Nodes`, `Alpha`, `EpsXY`, `EpsTheta`, `UseAutomaticAlphaRegulation`).
3. Defining corrupted subdomains (`Rmin`, `Rmax`, `DomainsGoal`, `Attempts`, `RminMeasure`).
4. Advanced (`Time_quantization`, `Recovered_Isophote_Color_Coefficient`, `DisplayImages`,`Threads`).



F(x,y) = ((x - 1)*(x - 1) - y*y - 12)*cos (x - 5)*cos (x - 5)/3 - 3*cos (y - 1)*cos(y - 1) + 6*y + sqrt (3*(x - 5)*(x - 5) + 30)

Xmin = -5 Xmax = 5 Ymin = -5 Ymax = 5

x_pixels = 1200 y_pixels = 1200

Isophotes = 120 Mode = W Nodes = 4 Threads = 8

Rmin = 0.02 Rmax = 0.2 DomainsGoal = 3000

Attemps = 1000000 RminMeasure = 0.5

Extended Settings

Time quantization = 20000 Recovered Isophote Color Coefficient = 1.5 Alpha = 1

EpsXY = 0.009 EpsTheta = 0.028

☑ Use Automatic Alpha Regulation ☑ Display Images

```
Please substitute initial parameters
```

Figure 10: Input form.

5.3. Process of executing

When all input parameters are entered, the user should click `CreateImage`. This creates a text file *fcutcurves.cpp*, in which the problem is written in C++ language: it includes definition of the function $\mathrm{F}(x, y)$ and parameterization of closed curves bounding the corrupted subdomains. Further work of the application is illustrated in Fig. 11.

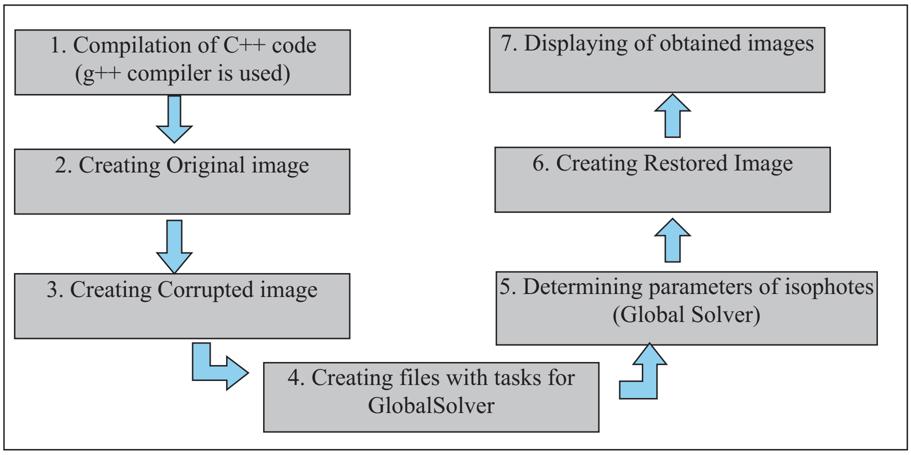

1. Compilation of C++ code (g++ compiler is used)
2. Creating Original image
3. Creating Corrupted image
4. Creating files with tasks for GlobalSolver
5. Determining parameters of isophotes (Global Solver)
6. Creating Restored Image
7. Displaying of obtained images

Figure 11: Process of executing `OptimalInpaiting` .

5.4. Output

As a result, `OptimalInpaiting` displays 6 windows that contain:

1. the original image (Original),
2. the corrupted image (Corrupted),

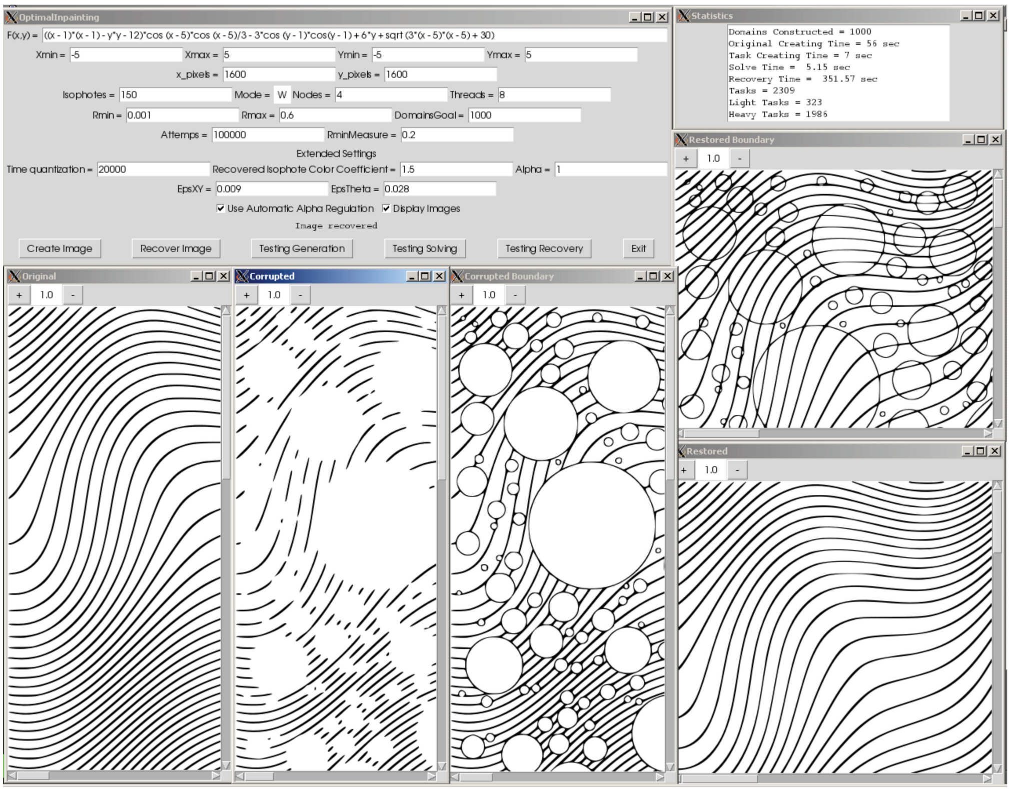
3. the corrupted image with traced boundaries of the corrupted subdomains (Corrupted Boundary),

4. the restored image (Restored),

5. the restored image with traced boundaries of the corrupted subdomains (Restored Boundary),

6. statistics collected during execution.

In the case of testing (buttons `TestingGeneration` and `TestingRecovery`), the plots demonstrating efficiency of parallelization are additionally displayed (plot of time and plot of acceleration). Fig. 12 shows a screenshot of `OptimalInpaiting` application.



F(x,y) = ((x - 1)*(x - 1) - y*y - 12)*cos (x - 5)*cos (x - 5)/3 - 3*cos (y - 1)*cos(y - 1) + 6*y + sqrt (3*(x - 5)*(x - 5) + 30)

Xmin = -5 Xmax = 5 Ymin = -5 Ymax = 5

x_pixels = 1600 y_pixels = 1600

Isophotes = 150 Mode = W Nodes = 4 Threads = 8

Rmin = 0.001 Rmax = 0.6 DomainsGoal = 1000

Attemps = 100000 RminMeasure = 0.2

Extended Settings

Time quantization = 20000 Recovered Isophote Color Coefficient = 1.5 Alpha = 1

EpsXY = 0.009 EpsTheta = 0.028

☑ Use Automatic Alpha Regulation ☑ Display Images

```
Image recovered
```

```
Domains Constructed = 1000
Original Creating Time = 56 sec
Task Creating Time = 7 sec
Solve Time =  5.15 sec
Recovery Time =  351.57 sec
Tasks = 2309
Light Tasks = 323
Heavy Tasks = 1986
```

Figure 12: `OptimalInpaiting` application.

5.5. Technical information

`OptimalInpaiting` was developed in Linux environment (Alt Linux, kern. 2.6.27 - hpc-std-alt2) with the compiler gcc-4.4. The Graphical User Interface (GUI) was written in Tcl/Tk interpreted language. Tcl/Tk program is executed with the use of wish interpreter. The computational module was written in C++ with the use of tsim, libgomp, libgsl, libpng libraries. Images in pgm format are created with the use of Potrace utility. `OptimalInpaiting` was installed on cluster *blade.botik.ru* running under Linux. User interaction with the application occurs via GUI. Connection to the server is implemented on SSH.

5.6. Formulation and solution of the problem of image restoration

In the field `F(x,y)` the user should enter a smooth function of two arguments (the level curves of this function determine the original image). The function `F(x,y)` is written in C-style (in accordance with math.h), i.e.,

- the basic arithmetic operations: $+$, $-$, $*$, $/$,
- x to the power n: `pow(x,n)`,
- trigonometric functions: `sin()`, `cos()`, `tan()`,
- inverse trigonometric function: `acos()`, `asin()`, `atan()`,
- exponential and natural logarithm: `exp()`, `log()`,
- hyperbolic functions: `cosh()`, `sinh()`, `tanh()`.

After entering a correct input, the user can launch the solution process. To do this, click the button `Create Image` in the bottom of the Input form. This starts the process of creating original and corrupted images. The images obtained are displayed on the screen. To start the recovery process, press button `RecoveryImage`. The problem will be solved on the number of nodes prescribed by the used in the field `Nodes` (0 means execution in sequential mode). As a result, the program displays 6 windows, see Fig. 12, which contain the original image, the corrupted image, the corrupted image with traced boundaries of the corrupted subdomains, the restored image, the restored image with traced boundaries of the corrupted subdomains, and statistics. Statement of the problem of image recovery is saved in a file $fcutcurves.cpp$. It contains notation of functions that define the contours of the corrupted subdomains in the parametric form (parameter t), the number of corrupted subdomains, the initial and final value of t for each contour, the function `F(x,y)` (whose level lines determine the original image), the coefficient of smoothing for binary images and number of discretization steps of t. The images obtained are stored in directory `\pict` in two formats (png and pgm). The file $Statistic.txt$ contains collected statistics about process of execution of the application.

5.7. Effectiveness of parallelizing

PC `OptimalInpaiting` includes parallel computation at 3 stages: creating of original image, determining the parameters of isophotes (GlobalSolver), creating of recovered image. TSim and OMP are used for organization of parallel computation. The application includes functions for testing effectiveness of parallelizing. To start the testing process of creating original and corrupted images with different numbers of threads (`Threads`), click `TestingGeneration`. To start testing the module determining the parameters of isophotes for different numbers of nodes (`Nodes`), click `TestingSolving`. To start the testing process of creating recovered image with different numbers of threads (`Threads`), click `TestingRecovery`. Figs. 19 and 20 show examples of testing of effectiveness of parallelizing.

5.8. `OptimalInpaiting` : Presentation of some results

In this section we present results of output of `OptimalInpaiting` software for the test problem described as follows:

$$F(x,y) = 1.5(x\cos y\cos y\sin x\sin x + y\sin y\sin y\cos x\cos x) \qquad (5.1)$$
$$+x^2 - y^2 - xy - x + 2y,$$
$$\texttt{Xmin} = -10, \quad \texttt{Xmax} = 10, \quad \texttt{Ymin} = -10, \quad \texttt{Ymax} = 10, \qquad (5.2)$$
$$\texttt{x\_pixels} = \texttt{y\_pixels} = 2500, \quad \texttt{Isophotes} = 100, \qquad (5.3)$$
$$\texttt{Rmin} = 0.05, \quad \texttt{Rmax} = 0.4, \quad \texttt{DomainsGoal} = 10^4, \qquad (5.4)$$
$$\texttt{Nodes} = 4, \quad \texttt{Threads} = 8, \quad \texttt{Attempts} = 10^6. \qquad (5.5)$$

Figs. 13-17 present respectively the original, corrupted, and restored images for problem (5.1)-(5.5) in the halftone version (`Mode` $= G$), and Figs. 14-18 present the same in the binary version (`Mode` $= W$). Plots of time and acceleration vs number of threads for the process of creating of images are shown respectively in Fig. 19 and Fig. 20.

6. Conclusion

This paper describes in detail an approach to monochrome image inpainting via completing damaged isophotes by sub-Riemannian length minimizers for the left-invariant sub-Riemannian problem on the group of motions of a plane SE(2). The approach and the algorithm presented in this work were realized by a set of parallel software. The results of work of `OptimalInpaiting` software prove efficiency of our approach to inpainting.

The approach to image inpainting chosen in this work is based on the following assumptions:

1. Isophotes of an image can be represented by level lines of a smooth function $f : D \to [0,1]$.
2. The function f has no critical points in the corrupted subdomains $O_1, \cdots, O_N \subset D$.
3. Information on intersection points of level lines of f with boundaries of O_i can be effectively extracted from the image.
4. During inpainting, the way of pairing isophotes is known.
5. Corrupted domains of the image are disks.

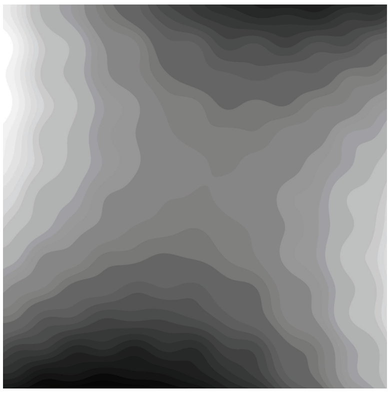


Figure 13: Original halftone image.

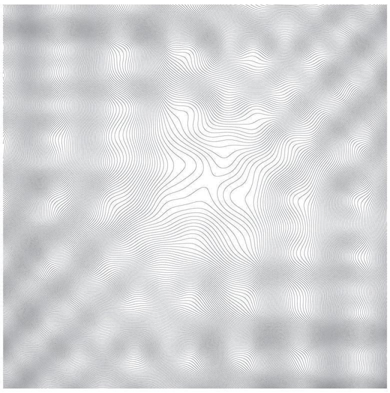


Figure 14: Original binary image.

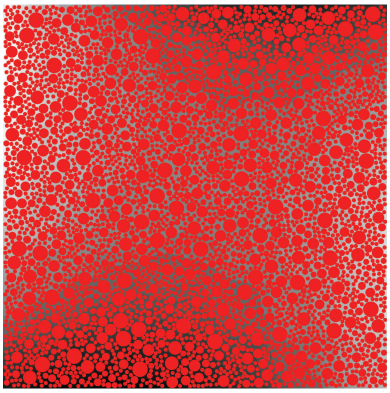


Figure 15: Corrupted halftone image.

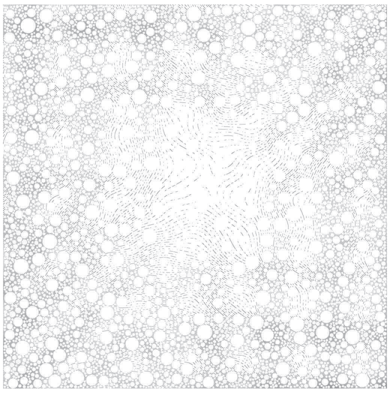


Figure 16: Corrupted binary image.

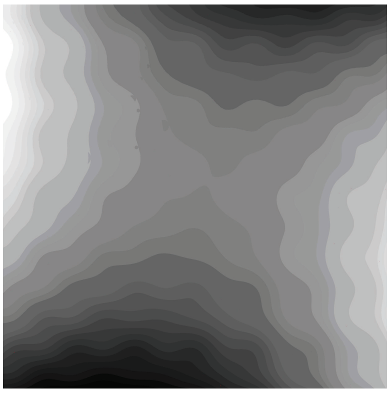


Figure 17: Restored halftone image.

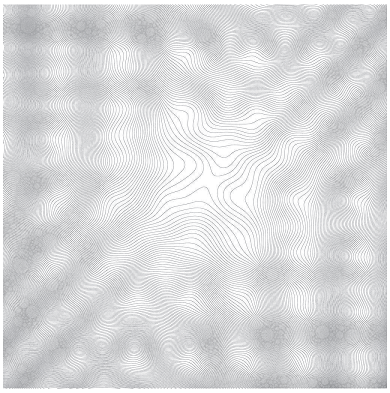


Figure 18: Restored binary image.

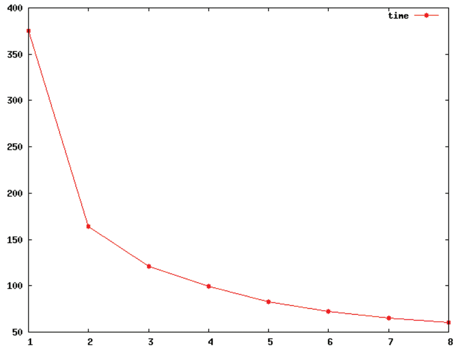


Figure 19: Time vs number of threads.

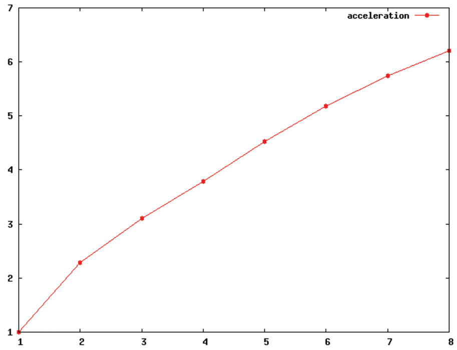


Figure 20: Acceleration vs number of threads.

Assumption 1 is not very restrictive. It is known that the retina smoothes images by making the convolution with a Gaussian function [25–27]. It was proved in [28] that convolution of a function from L_2 with a Gaussian is generically a C_∞ and generically Morse function, i.e. a smooth function having as critical points only non-degenerate maxima, minima and saddles. (See also a related work [33].) Thus one can assume that the image being inpainted is given by a smooth (and generically even Morse) function.

Assumption 2 seems rather restrictive, both for our approach, and for other approaches to image inpainting.

Assumption 3 seems not very restrictive since information on intersections of (smooth) level lines with domains with corrupted image can be extracted by modern image processing techniques. Another way to overcome this assumption is the use of a field of orientations instead of isophotes. Such orientation field could be obtained in a more robust way when using SE(2)-convolutions (e.g. with Green's functions of hypo-elliptic PDE's).

The feasible way of pairing of isophotes (see Assumption 4) might be reconstructed on the basis of the following ideas:

- The isophotes cannot intersect, thus pairings of isophotes admitting intersections should be rejected.

- Among the remaining pairings of isophotes, one can choose the optimal one w.r.t. the total length functional (3.4) for all isophotes.

Assumption 5 can be weakened to allow for convex corrupted domains.

The above extensions of our approach are planned to be implemented in future work.

The domain of image inpainting is a vast and actively developing domain of research and applications, and we make no attempt to describe it here. There exist approaches close to that considered in this work. In paper [24], the corrupted image is reconstructed by the minimal surface for the sub-Riemannian structure (3.5), (3.6) on the rototranslations group SE(2). This approach is obviously different from our approach based on sub-Riemannian length minimizers. We believe that the both approaches should be developed, and may be useful for different applications. A similar approach was taken in papers [1,23] with the use of Euler elastica [22]. This approach may be complemented by an approach similar to that used in this work (complementing corrupted arcs by energy-minimizing Euler elasticae [19–21].

One more possible extension of this work can be obtained by applying the above results to incomplete directional fields that are more robustly obtained than isophotes.

The authors believe that our approach should be further developed in order to be applied to natural images, and they are looking for possible applications.

Acknowledgments The authors acknowledge support by Russian Foundation for Basic Research, Project No. 12-01-00913-a, and by the Ministry of Education and Science of Russia within the Federal program "Scientific and Scientific-Pedagogical Personnel of Innovative Russia," Agreement No. 8209 of August 6, 2012.

The authors are grateful to anonymous reviewers whose comments were taken into account during revision of this paper.

References

[1] T.F. Chan, S.H. Kang and J. Shen, Euler's elastica and curvature based inpainting, SIAM J. Appl. Math., 63 (2002), 564–592.

[2] G. Citti and A. Sarti, A cortical based model of perceptual completion in the roto-translation space, J. Math. Imaging Vis., 24 (2006), 307–326.

[3] S. Esedoglu and J. Shen, Digital image inpainting by the Mumford-Shah-Euler image model, Europ. J. Appl. Math., 13 (2002), 353–370.

[4] B. B. Kimia, I. Frankel and A.-M. Popescu, Euler spiral for shape completion, Int. J. Comp. Vision, 54 (2003), 159–182.

[5] J. Petitot, The neurogeometry of pinwheels as a sub-Riemannian contact structure, J. Physiology - Paris, 97 (2003), 265–309.

[6] J. Petitot, Neurogeometrie de la vision — Modeles mathematiques et physiques des architectures fonctionnelles, Editions de l'Ecole Polytechnique, 2008.

[7] I. Moiseev and Yu. L. Sachkov, Maxwell strata in sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV, ESAIM: COCV, 16 (2010), 380–399, available at arXiv:0807.4731v1, 29 July 2008.

[8] Yu. L. Sachkov, Conjugate and cut time in the sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV, 16 (2010), 1018–1039.

[9] Yu. L. Sachkov, Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane, ESAIM: COCV, 17 (2011), 293–321.

[10] A. A. Agrachev and Yu. L. Sachkov, Control Theory from the Geometric Viewpoint, Springer-Verlag, Berlin, 2004.

[11] R. Montgomery, A Tour of Subriemannian Geometries, Their Geodesics and Applications, American Mathematical Society, 2002.

[12] J. P. Laumond, Nonholonomic motion planning for mobile robots, Lecture notes in Control and Information Sciences, 229, Springer, 1998.

[13] R. Duits and M. van Almsick, The explicit solutions of linear left-invariant second order stochastic evolution equations on the 2D Euclidean motion group, Quarterly of Applied Mathematics, 66(1) (2008), 27–67.

[14] R. Duits and E. M. Franken, Left-invariant parabolic evolutions on SE(2) and contour enhancement via invertible orientation scores. Part II: nonlinear left-invariant diffusions on invertible orientation scores, Quarterly of Applied Mathematics, 68(2) (2008), 293–331.

[15] R. Duits and E. M. Franken, 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, 68(2) (2008), 255–292.

[16] A. A. Agrachev, U. Boscain, J. P. Gauthier and F. Rossi, The intrinsic hypoelliptic Laplacian and its heat kernel on unimodular Lie groups, Journal of Functional Analysis, 256(8) (2009), 2621–2655.

[17] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko, The Mathematical Theory of Optimal Processes, Wiley Interscience, 1962.

[18] E. T. Whittaker and G. N. Watson, 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 University Press, Cambridge, 1996.

[19] Yu. L. Sachkov, Maxwell strata in Euler's elastic problem, Journal of Dynamical and Control Systems, 14(2) (2008), 169–234.

[20] Yu. L. Sachkov, Conjugate points in Euler's elastic problem, Journal of Dynamical and Control Systems, 14(3) (2008), 409–439.

[21] A. Ardentov and Yu. L. Sachkov, Solution of Euler's elastic problem (in Russian), Avtomatika i Telemekhanika, 2009, No. 4, 78–88. (English translation in Automation and remote control.)

[22] L. Euler, Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes, sive Solutio problematis isoperimitrici latissimo sensu accepti, Lausanne, Geneva, 1744.

[23] D. Mumford, Elastica and computer vision, Algebraic Geometry and Its Applications, pp. 491–506, 1994.

[24] R. K. Hladky and S. D. Pauls, Minimal surfaces in the roto-translation group with applications to a neuro-biological image completion model, Journal of Mathematical Imaging and Vision, 36(1) (2010), 1–27.

[25] M. Langer, Computational Perception, Lecture Notes, 2008,
`http://www.cim.mcgill.ca/Ĩanger/646.html`

[26] D. Marr and E. Hildreth, Theory of edge detection, Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 207, No. 1167, 1980, pp. 187–217.

[27] L. Peichl and H. Wässle, Size, scatter and coverage of ganglion cell receptive field centres in the cat retina, J. Physiol., 291 (1979), 117–141.

[28] U. Boscain, J. Duplaix, J. P. Gauthier and F. Rossi, Anthropomorphic image reconstruction via hypoelliptic diffusion. To appear on SIAM Journal of Control and Optimization.

[29] F. Cao, Y. Gousseau, S. Masnou and P. Perez, Geometrically guided exemplar-based inpainting, SIAM Journal on Imaging Sciences, 4(4) (2011), 1143–1179.

[30] L. P. Rothschild and E. M. Stein, Hypoelliptic differential operators and nilpotent groups, Acta Math., 137 (1976), 247–320.

[31] A. Nagel, E. Stein and S. Wainger, Balls and metrics defined by vector fields I: Basic Properties, Acta Math., 155 (1985), 130–147.

[32] L. Hormander, Hypoelliptic second order differential equations, Acta Math., 119 (1968), 147–171.

[33] J. Damon, Local Morse theory for solutions to the heat equation and Gaussian blurring, JDE, 115 (1995), 368–401.