Problem description and current progress

Ke Yin

July 9, 2014

1 Introduction

Mapping environment variables by robots has drawn a lot of interests these days. In particular, we want to identify the regions of interest, where the values of certain environment variable is significantly different from surrounging area, though almost uniform inside each region. Normally the programmed robots are sent out in the field, moving along designated paths and collecting the measurements of the environment variable on the way. Each robot is equipped with a positioning device and are able to get measurements at a given frequency. It can also send stored data back to the base by wireless communication. If the coordinates at every point on the path can be recorded and the robot can send measurements back in real time, then theoretically the values of the environment variable at every point can be obtained. However, it is impractical due to limited bandwith. To address the bandwith limitation, we propose a different strategy. The robot is programmed to perform on-board summation of the measurements obtained along each path (so-called path integral), and only sending the path integral to the base. Ideally the paths are line segments, so only the coordinates of begin and end of each line segment are needed in order to determine the paths. These path integrals of measurements along with the path information are post-processed using the reconstruction techniques of computed tomography and compressed sensing. Hopefully we are able to reconstruct the environment variables in the field.

2 Experiment settings

To be added.

3 Models and assumptions

Without loss of generality, we assume the area to be explored is a rectangle denoted by Ω, and the interested environment variable $u(x, y)$ is a piece-wise constant function defined on Ω. See Figure 1 for an illustration. Assume the robot travels through n different paths, which are denoted by $C_1, \ldots, C_n$. Along each path C_k, the integral of environment variable u is obtained, which is denoted by g_k. The path integrals are written as

$$g_k = \int_{C_k} u(x, y)\, \mathrm{d}\Gamma,\ k = 1, \ldots, n. \tag{1}$$

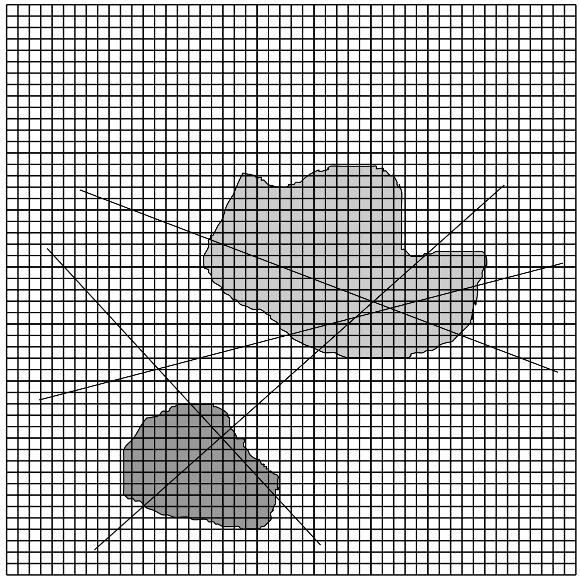

Figure 1: An illustration of the paths of robots. The shaded regions denote the area of interest (unknown), where the values of environment variable u is significantly different from the surroundings. The size of the unit voxel (each small rectangle) is determined by the accuracy of the positioning system.

For computational purpose, the whole domain is discretized into rectangular voxels (See Figure 1). Each path C_k defines a weight $a_{k,ij}$ for each voxel (i, j). If C_k does not intersect with voxel (i, j), then $a_{k,ij} = 0$, otherwise $a_{k,ij}$ is defined to be proportional to the length of the part of C_k that falls within voxel (i, j). It is also assumed that the value of u is a constant within each voxel (i, j), which is denoted by u_{ij}. After discretization, Eq (1) becomes

$$g_k = \sum_i \sum_j a_{k,ij} u_{ij}. \tag{2}$$

Let $g = (g_1, \ldots, g_n)^t$, $u = (u_{ij})$, then g and u have a linear relation, which is written as

$$g = Au, \tag{3}$$

where the linear operator A is specified in Eq (2). Eq (3) is the model equation, which poses an inverse problem. In this equation, g can be obtained from the experiment, A is determined by user-specified paths for the robot, which can be calculated offline. u is the variable that we want to solve. Of course, if the paths form a complete raster scan over the whole domain, then theoretically Eq (3) has a unique solution. For economical reasons, it is desirable to use much fewer paths and still being able to reconstruct a solution for u. Inspired by compresed sensing based image reconstruction techniques, it is hopeful that reconstruction for our under-determined system can be done.

4 Algorithm for solving the inverse problem

Based on the observation that the image u to be reconstructed is piecewise constant, it's gradient ∇u is sparse over the domain. Inspired by the compressed sensing theory, we postulate that the actual solution u should minimize the energy functional

$$J(u) = \frac{\mu}{2}\|Au - g\|_2^2 + \alpha|u|_1 + \beta(|\nabla_x u|_1 + |\nabla_y u|_1), \tag{4}$$

where α, β are some constants to be tuned depending on applications. Here we use Split-Bregman method to solve the problem $\min_u J(u)$. More exactly, we rewrite minimization of (4) as a constraint problem

$$\min_{u,d,d_x,d_y} \frac{\mu}{2}\|Au - g\|_2^2 + \alpha|d|_1 + \beta(|d_x|_1 + |d_y|_1) \quad \text{such that } d = u,\ d_x = \nabla_x u,\ d_y = \nabla_y u.$$

And its associated Augmented Lagrangian functional $L(u,\, d,\, d_x,\, d_y,\, b,\, b_x,\, b_y)$ is

$$\frac{\mu}{2}\|Au-g\|_2^2+\alpha|d|_1+\beta(|d_x|_1+|d_y|_1)+\frac{1}{2}\lambda_1\|d-u+b\|_2^2+\frac{1}{2}\lambda_2(\|d_x-\nabla_x u+b_x\|_2^2+\|d_y-\nabla_y u+b_y\|_2^2).$$

The solution process is iterative, with each iteration consisting of three major steps:

1. Update u by solving the Least-Squares problem

$$\min_u \frac{\mu}{2}\|Au - g\|_2^2 + \frac{1}{2}\lambda_1\|d - u + b\|_2^2 + \frac{1}{2}\lambda_2(\|d_x - \nabla_x u + b_x\|_2^2 + \|d_y - \nabla_y u + b_y\|_2^2). \tag{5}$$

2. Update d, d_x, d_y by soft-thresholding (shrinkage)

$$d = \operatorname{sign}(u - b) \cdot \max(|u - b| - \frac{\alpha}{\lambda_1}, 0).$$

$$d_x = \operatorname{sign}(\nabla_x u - b_x) \cdot \max(|\nabla_x u - b_x| - \frac{\beta}{\lambda_2}, 0).$$

$$d_y = \operatorname{sign}(\nabla_y u - b_y) \cdot \max(|\nabla_y u - b_y| - \frac{\beta}{\lambda_2}, 0).$$

3. Update auxillary variables b, b_x, b_y

$$b = d - u + b.$$

$$b_x = d_x - \nabla_x u + b_x.$$

$$b_y = d_y - \nabla_y u + b_y.$$

In step 1, we need to solve following linear equation

$$(\mu A^T A + \lambda_1 I + \lambda_2 \nabla_x^T \nabla_x + \lambda_2 \nabla_y^T \nabla_y)u = \mu A^T g + \lambda_1(d + b) + \lambda_2 \nabla_x^T(d_x + b_x) + \lambda_2 \nabla_y^T(d_y + b_y). \tag{6}$$

Since it is a positive-definite system, it can be solved by Conjugate Gradient method, with the preconditioner give by the linear operator

$$(\lambda_1 I + \lambda_2 \nabla_x^T \nabla_x + \lambda_2 \nabla_y^T \nabla_y)^{-1}. \tag{7}$$

Because ∇_x, ∇_y are both circulant matrices using finite-difference approximation assuming periodic boundary condition, (7) can be solved by Fourier transform and its inverse.