Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 15-004

Environment and Solar Map Construction for Solar-Powered Mobile Systems

Patrick A. Plonski, Joshua Vander Hook, Volkan Isler

April 16, 2015

Environment and Solar Map Construction for Solar-Powered Mobile Systems

Patrick A. Plonski, Joshua Vander Hook, and Volkan Isler

***Abstract*—Energy harvesting using solar panels can significantly increase the operational life of mobile robots. If a map of expected solar power is available, energy efficient paths can be computed. However, estimating this map is a challenging task, especially in complex environments. In this paper, we show how the problem of estimating solar power can be decomposed into the steps of magnitude estimation and solar classification. Then we provide two methods to classify a position as sunny or shaded: a simple data-driven Gaussian Process method, and a method which estimates the geometry of the environment as a latent variable. Both of these methods are practical when the training measurements are sparse, such as with a simple robot that can only measure solar power at its own position. We demonstrate our methods on simulated, randomly generated environments. We also justify our methods with measured solar data by comparing the constructed height maps with satellite images of the test environments, and in a cross-validation step where we examine the accuracy of predicted shadows and solar current.**

I. Introduction

Mobile robots are useful for outdoor environmental monitoring. However, their lifetimes are usually limited by the energy density of batteries. One method to improve the system lifetime of mobile robots is through the use of solar photovoltaic panels. For example, the solar-powered NASA Mars rover *Opportunity* has remained in operation for more than 10 years, far longer than would have been possible from only the initial energy in its batteries.

In a complex environment, shadows cast by objects or terrain can affect the incoming power available from the panels. To maximize system life, therefore, it can be useful to plan energy-efficient paths along which more insolation can be harvested. As demonstrated in [1], a short deviation in path or sample position can result in dramatically increased insolation. To plan these energy-aware paths, the robot needs a map of available solar power, or insolation, over the operating environment.

In this work we propose a two-step process to estimate the insolation at unobserved future position that a solar powered robot might visit. In the first step, we estimate the magnitude of the solar power collected by the robot with current system and weather conditions. In the second step, we estimate the classification at other future positions for the robot, and combine the likelihood of sun with the estimated power conditioned on class to achieve our final solar power predictions.

If there is an accurate prior map of the shadow-casting objects in the environment, it is straightforward to perform raytracing from the angle of the sun at any time of day and determine if a candidate robot position will be in the sun or the shade. Lacking a prior map, existing approaches use sophisticated sensors such a LIDAR to directly observe the 3d geometry. In this work we are interested in techniques that have minimal hardware overhead, requiring that the robot need measure only its own position and the amount of insolation it receives. An example application where minimal hardware is useful is the use of simple, low-cost robots.

In this work we present two methods which use only position and insolation to estimate solar classification. We have previously used Gaussian Process (GP) regression as a simple data-driven method to estimate the insolation at a position [1]. Here we extend this method to allow shadows that vary in time. Although the maps produced by regression provide accurate estimates of shadows in the short term, the method does not have a model for how shadows should move in response to the known change in sun angle. Even if the correct shadow boundaries are known, it is challenging to determine the future shadow profile without knowledge of the shadow casting objects. Therefore, to accomplish longer-term predictions we present a new classification strategy which infers the underlying geometry of the environment as a latent variable, based only on recorded insolation measurements. Our method for using effectively a single mobile light sensor to build a coarse model of the environment could be of independent interest, beyond its utility for solar estimation.

The authors are with the Department of Computer Science at the University of Minnesota, 200 Union Street SE, Minneapolis MN 55455 USA. {plonski,jvander,isler}@cs.umn.edu

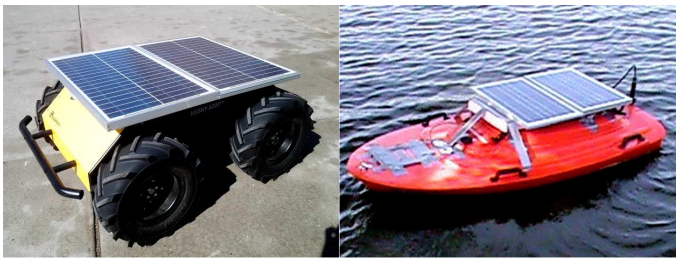


Fig. 1. The two experimental platforms used to validate our algorithm. These platforms were developed for an environmental monitoring application which is presented in [2].

We validate our method in simulation. We also implement our method on two different mobile robotic systems (Figure 1) and verify the algorithm by building an estimate of shadow-casting objects in two representative environments: an urban park and a lake shoreline. The constructed maps are used to predict insolation at withheld positions and times with good accuracy.

II. Related Work

Information about solar energy collected from the environment has been used to inform path planning, usually for long-term mission planning. In open environments (e.g. in Antarctica [3] or on the open ocean [4]) the energy is primarily a function of the robot's orientation. Therefore insolation is constant at a certain time of day if the solar panels are mounted horizontally.

The problem changes in a complex environment that contain objects or terrain which can block the sun. Estimating and avoiding shaded areas can increase system lifetime, but the problem of estimating how collected insolation will vary across a domain has received limited attention. A notable exception is the TEMPEST mission-level path planner [5] which calculates the position of the sun and then performs raytracing on known nearby terrain to build a solar map that can be used to estimate the net energy cost of a path.

Given that sensing the terrain in high detail is a difficult task, in [1] we examined the problem of learning the solar map using only measurements of position and solar insolation. We used Gaussian Process regression to estimate the insolation collected at unobserved locations. When used for short-term prediction, it was possible to achieve energy gains by planning solar-aware paths. However, the assumption of spatially-correlated shadows fails over larger time scales. The method did not have the ability to account for the relative change in sun angle between the time of the training measurements and the time of the planned path.

To obtain a shadow map that intelligently adjusts for time of day, we need to estimate the geometry of the environment. Mapping this geometry is a similar problem to that of estimating occupancy maps for path planning purposes. Existing methods for occupancy mapping (e.g. [6], [7], [8]) require range detection to directly sense the presence or absence of objects. It is unclear if these approaches can be extended

to our sensing model which uses only measurements of solar power associated with positions and times.

In the computer vision literature, shadows are occasionally used to learn about the scene. Abrams et al. [9] used a series of images taken from the same position throughout a clear sunny day to estimate the geometry of an outdoor scene. Their approach, however, requires that the boundaries of a shadow can be accurately estimated for any time slice. This is often not the case when the only solar measurements come from the same position as the robot.

It is possible to start with a conservative estimate of the objects in the environment and use shadow carving techniques (such as [10]) to converge over time upon a correct estimate of geometry of shadow-casting objects in the environment. However, when given sparse information (as from a robot that can only observe insolation at its own position), shadow carving results in an estimate of shade almost everywhere.

Our main contribution in this work is a new method of solar estimation that uses previous insolation measurements along with expectations about the environment to construct a map of the environment geometry. With this map and a calibration step, the estimated insolation can be computed for any position at any time. Assuming the robot can localize itself, the proposed method requires no sensors except a current sensor attached to the solar panel.

III. Approach

In this section we present our problem statement and the outline of our approach. Our goal is to estimate the insolation that will be received by a mobile robot in a workspace at any specified position and time, given that past measures of insolation have been recorded in the same workspace.

Let the prior points visited by the robot be $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where each $\mathbf{x}_i$ is a vector containing the position and time of the visit. The robot records the insolation y_i at each point to form the set of measurements $\mathcal{Y}$. We make the simplifying assumption that the surface is flat, and we do not consider elevation changes or changes in relative orientation of the solar panel with respect to the sun. A variable tilt of the solar panel and elevation changes can be easily accounted for, as long as they are known parameters.

We can formally state our problem as follows:

Problem Statement: Given solar insolation measurements $\mathcal{Y}$ at points $\mathcal{X}$, estimate solar insolation $\mathcal{Y}_*$ at points $\mathcal{X}_*$.

In this work we take advantage of the fact that on a clear day the majority of positions can be approximated as entirely sunny or entirely shaded, and split solar prediction into separate calibration and classification problems. Calibration estimates the power collected in the sun or the shade at any time of day given current system and weather conditions, and classification estimates the likelihood that a particular point is in the sun. The proposed method of solar prediction can be described with the following three step process:

Solar Prediction Method:

- *Calibration* — from solar magnitudes $\mathcal{Y}$ and timestamps, estimate calibration parameters that determine the insolation in sun and shade, and classify $\mathcal{Y}$ to the solar label set $\mathcal{Z}$ such that each $z_i \in \{\texttt{sunny}, \texttt{shaded}\}$
- *Environment Reconstruction* — from the label set $\mathcal{Z}$ and the measurement positions $\mathcal{X}$, build a model of the shadow-casting objects in the environment, denoted $\mathcal{H}$.
- *Estimation* — use $\mathcal{H}$ along with the environmental model to estimate the probability $p(z_*)$ that point $\mathbf{x}_*$ can see the sun. From this probability and the calibration parameters, obtain the final solar prediction for that ray y_*. Repeat until all we have all predictions $\mathcal{Y}*$.

The rest of this work is laid out as follows: In Section IV, we address the first step of our method. In Section V we show how estimation can be achieved using existing GP regression techniques without environment reconstruction. In Sections VI and VII we present our new environment construction and estimation methods. In Section VIII we demonstrate the feasibility and suitability of our new reconstruction method for an example simulated application. In Section IX we qualitatively and quantitatively compare the GP regression with our new method using real solar data collected by mobile outdoor robots.

IV. Calibration

In this section we justify our approximation of solar estimation as primarily a classification problem, and we present our method to transition between insolation y (measured in current from the panel) and classification $z \in \{\texttt{sunny}, \texttt{shaded}\}$ for any time of day. This is used as the first step of the pipeline, when $\mathcal{Y}$ is converted to $\mathcal{Z}$, and then it is again used as the last step of the pipeline, when $p(\mathcal{Z}_*)$ is converted to $E[\mathcal{Y}_*]$.

A. Background

First we provide a brief overview of the relevant factors in solar photovoltaic power generation for our system. For a more comprehensive presentation of photovoltaic power as it relates to mobile robotics, see [1].

In this work we focus primarily on clear conditions (with no clouds). This is because these are the conditions where the spatial and temporal variability in solar power is the highest. In fully overcast conditions, it is unimportant to learn the geometry of the environment. In partly cloudy conditions, the techniques described in this paper will need a preprocessing step that can distinguish between the shade of a cloud and the shade of an object on the ground. This is a task beyond the scope of this paper.

Here is how we estimate how the energy from a solar panel will change during a day. There are well-known equations to determine the solar zenith and azimuth angles from GPS position and time of day [11]. When a solar panel is held to a fixed voltage, the current from the panel I can be approximated as proportional to the number of photons that impact the cell [12]. Radiation to a panel can be decomposed according to its source angle in the following manner: If it comes directly from the solar disk it is direct insolation. If it comes from elsewhere in the sky it is diffuse insolation. Threlkeld and Jordan [13] showed that it is possible to estimate for a clear day the degree of attenuation a ray of sun experiences traveling through the atmosphere (this attenuation is greater when the sun is lower in the sky because the radiation has to travel through more atmosphere) from an assumed optical depth. They also showed that the diffuse insolation on a clear day is proportional to the direct radiation, with the parameter that relates them varying with dust and water vapor in the atmosphere.

In the special case where the solar panel is horizontal, we can parameterize the direct current I_r and diffuse current I_f using three constants: the optical depth k and scaling factors C_r and C_f. The formulas are the following:

$$I_r = C_r \cdot e^{\frac{-k}{cos(\theta)}} \tag{1}$$

$$I_f = C_f \cdot cos(\theta) \cdot e^{\frac{-k}{cos(\theta)}}. \tag{2}$$

In the above θ represents the zenith angle of the sun. If the entirety of the solar array can see the sun, the total solar current $I_s = I_r + I_f$. If at least one cell in each series string is fully shaded, $I_s = I_f$. This

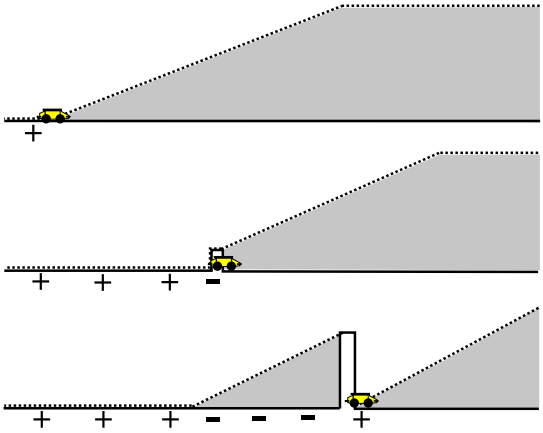


Fig. 2. An illustration of the construction of the upper heights $\mathcal{H}^u$ (dotted lines) and lower heights $\mathcal{H}^\ell$ (solid line) in a one-dimensional environment. The robot moves from left to right. **Top:** The first measurement is sunny indicated by the plus sign. **Middle:** It encounters a shaded region. **Bottom:** It again encounters sun. The lower height map $\mathcal{H}^\ell$ is increased in the least number of cells which will explain the shade measurements. Note that the gray region between $\mathcal{H}^\ell$ and $\mathcal{H}^u$ is considered unknown: it may contain an object or not.

situation is common when the array of solar cells is small compared with the shadow casting objects.

In our previous work [1] we treated the solar field as continuous. In this work we approximate every position in the environment as either fully in the sun or fully shaded. This can be easily justified when the shadows cast by objects are significantly larger than a single solar cell, as was the case for all of our experiments. Furthermore, we approximate diffuse insolation as constant, and neglect the reflected insolation term. These approximations greatly simplify our raytracing task, and, as we show in Section IX, they have only a small effect on the insolation estimate for a sunny day.

If the robot has a tilted panel, or more than one panel, the task of estimating insolation from solar angle becomes more complex, and might depend on the rotation of the robot. This is similarly true if the robot is operating on a slope. However, on a clear day it is easy to tell if a particular panel is in the sun or the shade, as long as the panel faces the sun.

B. Calibration Procedure

To classify the training set we compute θ_t for each measurement time in $\mathcal{X}$ and compute the C_r for all measurements assuming that they are all sunny, and assuming $k = 0.2$ (this is given as a representative default for the United States in [11]). We take the 90th percentile C_r as the representative and classify a measurement as sunny if it has C_r closer to the representative than to zero. Everything else is classified as shaded.

To go from labels back to continuous estimates we perform a numeric optimization to find the C_r, C_f, and k that minimize the mean error between theoretic and measured solar current, both in the sun and in the shade. When there is not a large measurement time window it can be difficult to determine the best k without overfitting, so it can remain set at 0.2.

V. Gaussian Process Classification

Before we present our main method of solar estimation which reconstructs the geometry of the environment we will detail our first approach to estimate sun and shade, which models the solar map as a spatiotemporal Gaussian Process.

Consider the information available to a robot as it moves through the environment and measures its insolation, as shown in Figure 2. We can use well-established astronomical calculations to transform any $\mathbf{x}_i$ into a ray r_i that points from the ground to the position of the sun [11]. If the robot measures direct sunlight ($z_i = \texttt{sunny}$), we know r_i did not intersect any object and we can safely assume no objects lie along r_i. If $z_i = \texttt{shaded}$ it is apparent r_i must have intersected with some object in the environment. However, the location of intersection is unknown. This is an important asymmetry: each measurement of sun specifically eliminates candidate object positions, but a single measurement of shade provides very little information.

One method we could use to represent the environment is the following: Populate the domain with voxels from the ground up to the maximum object height and then attempt to learn which ones cast shadows by eliminating voxels that sunny rays pass through and reinforcing a voxel if we can determine that a shaded ray was specifically shaded by it. This is similar to the shadow carving used in [10]. While voxel-based methods allow complicated objects to be reconstructed, they scale poorly to large environments. Moreover, for most environments, the vast majority of possible voxel arrangements are disconnected and impossible; the voxels allow too much descriptivity.

A common way to infer more from each particular measurement is by using Gaussian Process regression to associate measurements with other nearby positions in space and time.

What if we model the shadow-casting objects directly with a Gaussian Process? O'Callaghan et al. [6] modeled occupancy directly using Gaussian Processes, but it is unclear how this sort of inference can be performed when we only use solar measurements. Regression methods including GP regression rely on defining similarity between measurement positions (constructing a covariance matrix, in GP terminology). If we measure that some ray r_i results in $z_i = \texttt{sunny}$, we can reasonably expect that rays passing close to r_i will result in a similar sunny measurement. However, if $z_i = \texttt{shaded}$, the shadow-casting object could have been anywhere along r_i. Therefore little can be confidently said about a ray r_j that has a different angle than r_i, even r_i and r_j intersect.

Because rays of different angles have low covariance, we ignore the latent variable of environment geometry, and model the solar map directly as a spatio-temporal Gaussian Process on the surface of the earth. We model the solar map with two spatial dimensions and one temporal dimension for time of day. When the different times of day have been sufficiently sampled, this method can produce an accurate time-varying map, even though it has no knowledge about what exactly causes the shadow movement. For longer timescales another dimension can be added to allow for deviation in solar angles and terrain throughout the year.

Gaussian Processes are usually used for regression, but in this work we are interested in classification. In our previous work [1] we treated the solar field as a continuous GP. Here we show that we can simplify the problem by estimating only the labels sunny and shaded given previously observed labels. We rely on our calibration method to convert observed continuous measurements $\mathcal{Y}$ into labels $\mathcal{Z}$ as a preprocessing step for classification, and to convert the label estimates $\mathcal{Z}^*$ back into continuous estimates $\mathcal{Y}^*$. There are many different methods to use the techniques of GP regression to achieve classification, and we used one of the simplest, called least-squares classification [14]. In this method, the labels are assigned the values ± 1 and regression is performed as usual, treating the field as a Gaussian Process (all points are jointly Gaussian). Then, to obtain $p(z_i = +1|\mathcal{X}, \mathcal{Z})$, we compute the likelihood that the Gaussian random variable described by the mean and variance at x_i will take a positive value. Some sources (e.g. [14], [6]) suggest that the output

mean and variance should be "squashed" through a sigmoid with tunable parameters but we found that the cumulative distribution function of the regression output was sufficiently expressive.

GP regression requires a method to obtain a covariance matrix of all the measurement positions. We used an anisotropic version of the stationary exponential covariance function, so that the covariance between the values of the process at positions $\mathbf{x}_i$ and $\mathbf{x}_j$ was described by the following formula:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(\sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \text{diag}(\theta_s, \theta_s, \theta_t)(\mathbf{x}_i - \mathbf{x}_j)}\right) + \sigma_n^2 \delta_{ij} \quad (3)$$

Where σ_f^2 is the prior variance, σ_n^2 is the noise variance, θ_s is the inverse length parameter in space, and θ_t is the inverse length parameter in time. The only remaining hyperparameter is the prior mean μ_g. We optimize these hyperparameters to maximize the posterior likelihood of the measurements we have received conditioned on the model.

GP regression is a good option when there are very few measurements, and little can be deduced about the environment geometry. In addition, GP regression can be expected to perform well when the environment has been densely sampled in space and time, because in such a situation it is fully descriptive of the real solar maps. However, as we will show, environment reconstruction can provide significantly more accurate solar maps in medium-term situations, particularly when the time for which we want a map is not close to times in the training set.

VI. Environment Reconstruction

In this section we detail our environmental model and the algorithm we use to estimate the geometry of the environment using solar measurements.

Modeling the environment as an unstructured set of voxels allows too much descriptivity. Previously we used the fact that usually shadows are bigger than the robot, and move slowly, so they can be well-estimated without sampling every point in the spatiotemporal process. However, there is an additional piece of structure that we can use. Instead of using voxels we can represent our environment geometry with a heightmap (i.e. the geometry is a terrain). This is a powerful assumption which is reasonable in practice because most shadow-casting objects have decreasing cross sectional area at greater distance from the ground. Even when objects don't satisfy this property (such as arches or trees), knowledge of the upper envelope is often sufficient to plan energy-efficient paths: the robot can plan to wait on the sun-facing side of an object rather than under it.

The terrain-based, or 2.5D, representation provides a convenient method to represent uncertainty: We maintain a continuous maximum height $h^u(s_i) \in \mathcal{H}^u$ and a minimum height $h^\ell(s_i) \in \mathcal{H}^\ell$ for each discrete square s_i with edge length d_ϵ. The set of all squares is $\mathcal{S}$. See Figure 2 for an illustration of the constructed minimum and maximum heights on a simple line. We allow for violations in the terrain assumption and errors from quantization by allowing a proportion of sun rays to pass uninhibited through any particular terrain block, with a likelihood that decreases the farther it must travel.

A. Lower Heights

First we will present our method for constructing the lower height map $\mathcal{H}^\ell$; once we have this constructing the upper height map $\mathcal{H}^u$ is fairly straightforward. Refer to Table I for a list of the parameters we use when constructing the lower height map.

The process of constructing $\mathcal{H}^\ell$ is a selection of the most likely arrangement of lower heights that explains the shadows we have detected. Recall that we can use well-established astronomical calculations to transform any measurement position $\mathbf{x}_i$ into a measurement ray r_i which starts at the spatial position of $\mathbf{x}_i$ and points towards the position of the sun at the time of $\mathbf{x}_i$ We start with a fading function which describes the likelihood that a particular ray will intersect the environment before it reaches the robot on the ground, conditioned on the modeled heights of the blocks in the squares:

TABLE I. Heightmap construction parameters

Name	Description
α	Controls likelihood of sun when a ray is unobstructed by $\mathcal{H}$
β	Controls likelihood a ray will remain sunny when it travels through $\mathcal{H}$
γ	Regularization cost per instances of $h^\ell > 0$
ξ	Regularization cost per unit of h^ℓ
h_{max}	The maximum height of any object in the domain

$$p(z_i = \texttt{sunny} | r_i, \mathcal{H}) = \exp\left(-\alpha - \beta \sum_{\substack{s_j \in \mathcal{S}(r_i) \\ h(s_j) \geq h(r_i \cap s_j)}} ||r_i \cap s_j||\right) \quad (4)$$

In Equation 4, $\mathcal{S}(r_i)$ denotes the set of all squares which r_i passes directly above, $r_i \cap s_j$ denotes the line segment portion of r_i that lies directly above s_j, $h(r_i \cap s_j)$ denotes the mean height of this line segment, and $||r_i \cap s_j||$ is the length of this line segment. The parameters α and β are chosen based on how well our heightmap model matches the environment. If the environment is perfectly described by a heightmap of resolution d_ϵ, α can be taken as 1 and β as 0 (though in practice the algorithm to fit the heights works better if it slightly relaxed). However, if the shadow casting objects do not exactly fill the squares, β should be selected in accordance with the probability of occlusion per unit distance a ray travels through an offending square of the heightmap. Similarly, if we expect there are some shadows which are not explained by the model, α should be set based on the probability of an unexplained shade ray.

Assuming independence between sun and shade measurements, we can express the negative log likelihood of any set of observations as the following cost function f:

$$f(\mathcal{R}, \mathcal{Z}, \mathcal{H}) = \sum_{r_i \in \mathcal{R}} \begin{cases} \alpha + \beta \sum\limits_{\substack{s_j \in \mathcal{S}(r_i) \\ h(s_j) \geq h(r_i \cap s_j)}} ||r_i \cap s_j|| & \\ \qquad \text{when } z_i = \texttt{sunny} & \\ -\log\left(1 - \exp\left(-\alpha - \beta \sum\limits_{\substack{s_j \in \mathcal{S}(r_i) \\ h(s_j) \geq h(r_i \cap s_j)}} ||r_i \cap s_j||\right)\right) & \\ \qquad \text{when } z_i = \texttt{shaded} & \end{cases} \quad (5)$$

Note that in Equation 5, the cost of a ray of sun is linear in the length it travels through offending estimated heights; in contrast, the cost of a ray of shade is very large if it is unoccluded but quickly drops to near zero if it travels through a block. This mathematically captures the asymmetry in the information gained from measurements of sun compared with shade.

For the lower heights, we make a normalizing assumption that the most likely explanation of any set of measurements is a small number of short blocks, rather than the reverse. We express this concept with an energy function g defined as follows:

$$g(\mathcal{H}^\ell) = \sum_{s_i \in \mathcal{S}} \begin{cases} \gamma + \xi h^\ell(s_i) & \text{when } h^\ell(s_i) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Now to find the cost of a lower heightmap we simply sum f and g. This allows us to balance the tasks of minimizing misclassification, and minimizing the complexity of the environment. This framework implicitly models the prior of lower height estimates as independent and identically distributed, with a likelihood that decreases exponentially with height.

The problem of finding the $\mathcal{H}^\ell$ that minimizes $f(\mathcal{R}, \mathcal{Z}, \mathcal{H}^\ell) + g(\mathcal{H}^\ell)$ is challenging because our objective is clearly non-convex. We solve it by initializing all $\mathcal{H}^\ell$ to 0 and iteratively greedily selecting the single height change which results in the greatest cost reduction, until we arrive to a local minimum. To prevent the accretion of several real objects into an estimated single very tall object, we define a maximum allowed object height h_{max}.

There exists an optimal $\mathcal{H}^\ell$ such that each nonzero $h^\ell(s_i) \in \mathcal{H}^\ell$ is equivalent to some $h(s_i \cap r_j)$ where $r_j \in \mathcal{R}$. Therefore finding the optimal greedy height change is a matter of iterating through the list of rays that intersect each square. To perform this optimization we first sort the heights of all the rays that intersect each square. As each ray can intersect at most $O\left(\sqrt{|\mathcal{S}|}\right)$ squares, there are a total of $O\left(|\mathcal{R}|\sqrt{|\mathcal{S}|}\right)$ elements to be sorted, so the time complexity is $O\left(|\mathcal{R}|\sqrt{|\mathcal{S}|}\left(\log(|\mathcal{R}|) + \log(|\mathcal{S}|)\right)\right)$. Then the time for each iteration is proportional to the sum of all lists of rays: $O\left(|\mathcal{R}|\sqrt{|\mathcal{S}|}\right)$. In practice we found that the number of iterations was small compared with $|\mathcal{S}|$, because few squares were adjusted more than once. In comparison, GP regression requires a matrix inversion involving all measurements we wish to use, at time cost $O(|\mathcal{R}|^3)$. Therefore, the proposed heightmap method can be expected to scale much better than GP regression to situations involving a large number of training measurements.

In our terrain representation we assume that the robot operates on the ground plane at all times; thus a possible explanation for any shade measurement is that the robot is contained within a square with non-zero height. This may seem counter-intuitive but it is necessary because some of the objects we must reconstruct, such as trees, reach their maximum radius at a height other than their bottom, and so are approximated by their upper envelopes.

B. Upper Heights

Why do we need more than one heightmap? With our map of lower heights, we have an estimate of the smallest, simplest environment that could have cast shadows similar to those which we have observed. This map potentially neglects information that could have been gained from `sunny` measurements: the geometry constructed for an unobserved position is the same as that constructed for a position that has been consistently sunny. Moreover, there are potentially rays of shade which are inconsistent with H^ℓ, because the complexity cost would have been too much to account for them. We improve our model for both of these situations by maintaining an upper bound on square heights.

The upper bound for a particular square is realized in the case when all other squares are at their lower bound, because this leaves the most shade that needs to be explained. Therefore, we define the upper height in each square s_i as the height h^u which minimizes the descriptivity cost function $f(\mathcal{R}, \mathcal{Z}, \mathcal{H}^\ell \cup h^u(s_i))$, where $\mathcal{H}^\ell \cup h^u(s_i)$ is the lower height map but with the block in square s_i raised up to h^u. All upper heights are independent, conditioned on the measurements and the lower height map; therefore they are also easy to compute. We use the same h_{max} as before as the maximum upper height. Our only normalization is the following: when there is a range of heights with the same minimum cost, we select height as the tallest in the range.

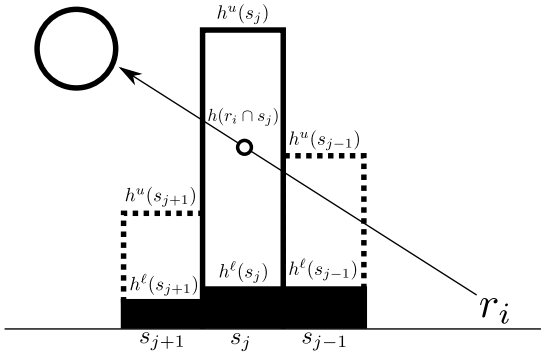


Fig. 3. The probability the ray r_i is unoccluded in square s_j is a function of the upper height in s_j, the lower height in s_j, the mean height of r_i in s_j, and the distance $||r_i \cap s_j||$ that r_i must travel above square s_j. The probability that the ray r_i is unoccluded in all of the squares $\mathcal{S}(r_i)$ that it passes above is the product of independent probabilities.

VII. Estimation

The methods described in the previous section allow us to construct an upper bound $\mathcal{H}^u$ and lower bound $\mathcal{H}^\ell$ heightmap of our environment. In this section we describe how to obtain from these upper and lower heightmaps an estimate of sun probability at any $\mathbf{x}_i$, and a corresponding estimated sun value. As previously, we start by transforming $\mathbf{x}_i$ into r_i, the ray from the same position on earth which points towards the sun.

We define a new fading function, similar to Equation 4. When a ray passes above a square s_i that has $h^\ell(s_i) > 0$, we treat the probability distribution of height in s_i as uniform between $h^\ell(s_i)$ and $h^u(s_i)$. When $h^\ell = 0$, there is a high likelihood that there is no object in the square. We model this by using a different β parameter when the ray passes above squares that have $h^\ell = 0$.

We describe the discontinuity between a ray slightly below h^ℓ and slightly above by using the ramp function $R(\cdot)$. Our full fading function is the following:

$$\begin{aligned} &p(z_i = \mathsf{sunny} | r_i, \mathcal{H}^\ell, \mathcal{H}^u) = \\ &\exp(-\alpha \\ &- \beta_1 \sum_{\substack{s_j \in \mathcal{S}(r_i) \\ h(r_i \cap s_j) < h^u(s_j) \\ h^\ell(s_j) > 0}} \left(1 - R\left(\frac{h(r_i \cap s_j) - h^\ell(s_j)}{h^u(s_j) - h^\ell(s_j)}\right)\right) ||r_i \cap s_j|| \\ &- \beta_2 \sum_{\substack{s_j \in \mathcal{S}(r_i) \\ h(r_i \cap s_j) < h^u(s_j) \\ h^\ell(s_j) = 0}} \left(1 - \frac{h(r_i \cap s_j)}{h^u(s_j)}\right) ||r_i \cap s_j|| \\ &) \end{aligned} \quad (7)$$

See Figure 3 for an illustration of the variables in Equation 7.

Equation 7 has three hyperparameters: α, which controls the prior probability of sun, β_1, which controls the likelihood of occlusion when passing above a square that has a block in it, and β_2, which controls the likelihood of occlusion when passing above a square that does not have a block estimated in it. The relationship between

β_1 and β_2 is an estimate of the likelihood that an unobserved square will contain a block. The three hyperparameters depend on the true descriptivity of the model, so they are ideally chosen through optimizing the likelihood of measurements that have been left out of the training set.

We have presented in this section a method to estimate the likelihood that a ray of sun at any angle will intersect an object before it reaches a solar panel on the ground. This estimate is consistent with the input upper and lower bounds on object heights. It also has the property that the estimated likelihoods are minimally affected when the resolution of the grid d_ϵ is changed, if object heights are held constant (the only change is because of improved height quantization of the sample ray). By combining the likelihood estimate of sun with the sun and shade insolation estimated in Section IV, the expected value of insolation can be obtained for any time and position.

$$E[y^*|x^*] = E[I_f|x^*] + p(z^*|x^*)E[I_r|x^*] \tag{8}$$

The method also performs well in practice, as we will see in Sections VIII and IX,

VIII. Simulations

A particularly interesting application for our solar estimation algorithm is that of a low-complexity robot operating in a cluttered, unknown environment for a long time. The robot can travel along an arbitrary trajectory and it should be possible to estimate the environment and the solar map for any time. From this estimate the robot can plan its future trajectory accordingly.

To show that our approach is suitable for this application we performed simulations where we generated random sample paths in randomly generated environments. Then we compared our algorithms sun likelihood estimates with a ground truth simulated shadowmap across the environment.

We generated our simulated terrain environments in the following manner: Each 40x40 heightmap was constructed out of 50 extruded square objects with edge length between 1 and 5 units and heights between 1 and 20 units. Over these objects were overlayed 25 square "holes" (portions with height zero) with edge lengths between 1 and 10 units. Hole positions as well as object positions and heights were all integers selected uniformly with replacement. An example simulated heightmap is shown in Figure 4.

We generated 200 such ground truth heightmaps and for each one simulated 10 days of observation, using sun angles derived from Minneapolis on the vernal equinox March 20, and the 9 days after. On each day there were 5 cuts from a random position on a border to a random position on one of the remaining 3 borders. Each cut was performed at a random time of day between 8:00 and 16:00 local solar time, and it consisted of solar samples which were 0.31 units apart from each other. After each day, our reconstruction algorithm made an estimate of the 0.5-unit resolution solar map at a random time on the 11th day, March 30, with the time of the final map fixed in the same way as the times of the cuts.. We compared these estimated solar maps with the ground truth and the maps estimated by the Gaussian Process method. An example ground truth map with improving daily estimates is shown in Figure 5, and the performance of the two strategies is shown in Figure 6.

For heightmap reconstruction we used the fading parameters $\alpha = -\log(0.95) = 0.05129$ and $\beta = -\log(0.005) = 5.2983$. These could be set strictly because the simulation environment matched the reconstruction assumptions. α was set less strict than β because, for reconstruction purposes, unexplained shade rays were much less bad than unexplained sun rays. The regularization parameters were set with respect to the neg log likelihood of an unexplained shade measurement: $\gamma = -5\log(1-exp(-\alpha)) = -5\log(0.05) = 14.979$, and $\xi = \gamma/20$. This ensured that it would require at least 5 shade measurements to create a block, and an additional 5 shade squares to raise it up to the h_{max} which was 20 units. For classification we fixed the fading parameters as $\alpha = -\log(0.99) = 0.01005$, $\beta_1 = -\log(0.05) = 2.9957$, and $\beta_2 = -\log(0.6) = 0,5105$. These were fixed a priori because for the first few days there were insufficient measurements to fit them properly with cross validation. The GP hyperparameters however were optimized to the data after each day.

We quantitatively examined the performance of our two solar estimation methods by plotting the Receiver Operator Characteristics (ROC) of the estimates for the 11th day. Each ROC curve describes the proportion of sun and shade correctly found as the classification threshold is adjusted. As the number of days in the training set is increased, the estimates become more accurate. See Figure 7 for a chart of the daily improvement of the estimates.

We found that in a cluttered urban-type environment, reconstructing the geometry of the environment was highly beneficial over modeling the shadow map directly with a time-varying Gaussian Process. After three observation days, the reconstruction method performed on average better than GP regression for every confidence threshold. After only one observation day, when very few objects could be accurately placed, the observed constraints on object heights from `sunny` measurements allowed raytracing to predict future sunny positions with much greater accuracy than with GP regression. Next we will see how the methods performed in a challenging real-world situation.

IX. Experiments

To validate our heightmap approach we used measurements of solar current associated with time and GPS coordinates to construct heightmaps for two different environments. Then we used these heightmaps to estimate full solar maps at various times. We qualitatively compared the heightmaps with the true environments, and we performed cross validation to show that our approach could predict solar measurements that were withheld from the training set. We also performed the same cross validation with GP regression, and with a combination classifier which uses both of the methods as inputs.

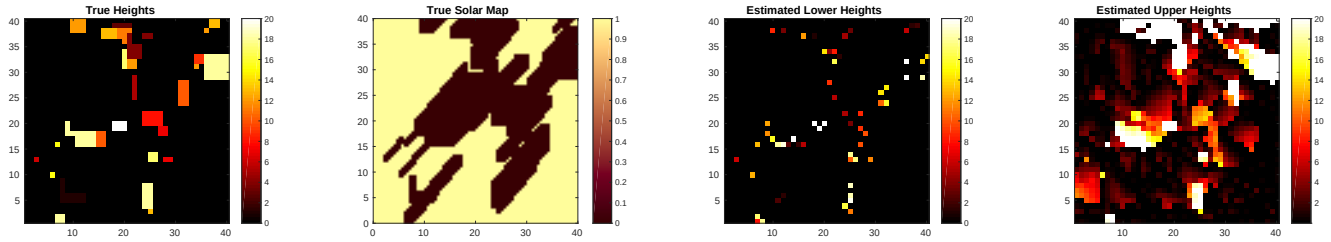
We used solar data collected at the McNamara Alumni Center on June 9, 2012 and solar data collected at Lake Como on September 16, 2012. Both environments are near Minneapolis, MN, USA, and in both environments the primary shadow-casting objects were trees. Solar maps were constructed by measuring the short-circuit current of two 20 watt Solartech Power[1] photovoltaic panels connected in parallel; this quantity is proportional to the amount of power that can be obtained from the panels.

A. Parameters Used

We selected $d_\epsilon = 2.5m$ based on the tradeoff between the size of the trees we encountered and GPS error. We selected $h_{max} = 30m$ at the Alumni Center as an upper limit on the height of most trees in Minnesota. At Lake Como we used $h_{max} = 40m$ to allow for the fact that the trees on the shore were elevated from the surface of the lake.

For reconstruction, we used fading parameters $\alpha = -\log(0.9) = 0.1054$, $\beta = \frac{-\log(0.5)}{d_\epsilon} = 0.2773/m$. This ensured that, for reconstruction purposes, a ray would be considered more likely shade than sun after passing through a single shading square, it it passed through it at near its greatest width. The regularization parameters γ and ξ

[1] www.solartechpower.com



(a) Ground truth heights (b) Ground truth solar map (c) Estimated $\mathcal{H}^{\ell}$ (d) Estimated $\mathcal{H}^{u}$

Fig. 4. A comparison of the ground truth simulated heightmap with the reconstructed estimate generated by the proposed algorithm after 10 days of measurements, where on each day has 5 random cuts across the solar map are observed. Also shown is the ground truth solar map at 13:32 CDT on the 11th simulated day.

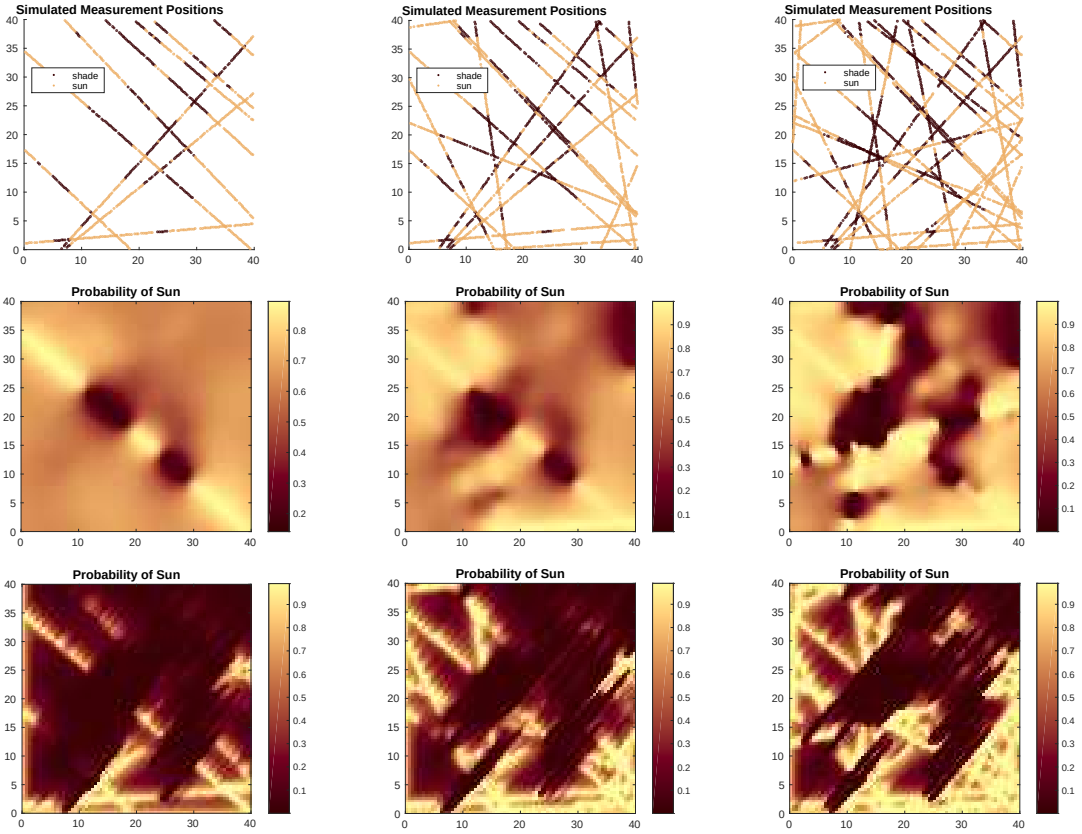


Fig. 5. Top row: the simulated measurements received after 2, 4, and 6 days of observation Middle row: the GP estimates of 11th day solar after 2, 4, and 6 days of observation. Bottom row: the reconstruction estimates of 11th day solar after 2, 4, and 6 days of observation. Each simulated cut across the environment was performed at a random time of day. This was the same environment as in Figure 4, and the ground truth solar map is the same as in that figure. The GP method can only make use of information at times of day close to the time we are interested in (13:32), but the reconstruction method uses all measurements. After 6 days the constructed solar map is close to the ground truth, even though the constructed lower heightmap is conservative in placing objects. This demonstrates that it is unimportant to learn the precise heights, as long as our estimated geometry casts shadows which are similar to those cast by the true environment.

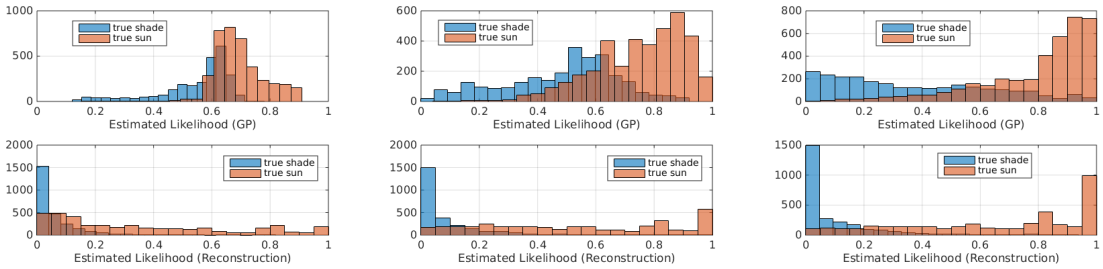


Fig. 6. Histograms of the estimates from GP and reconstruction, after 2, 4, and 6 simulated days. These were generated by estimating the entire simulated solar map in Figure 4 and examining the distribution of estimated solar likelihoods conditioned on the true label. After 2 days the reconstruction method predicts very little sun, so its shade predictions are not very informative. However, after 6 days, almost all of its predictions are confident and accurate.

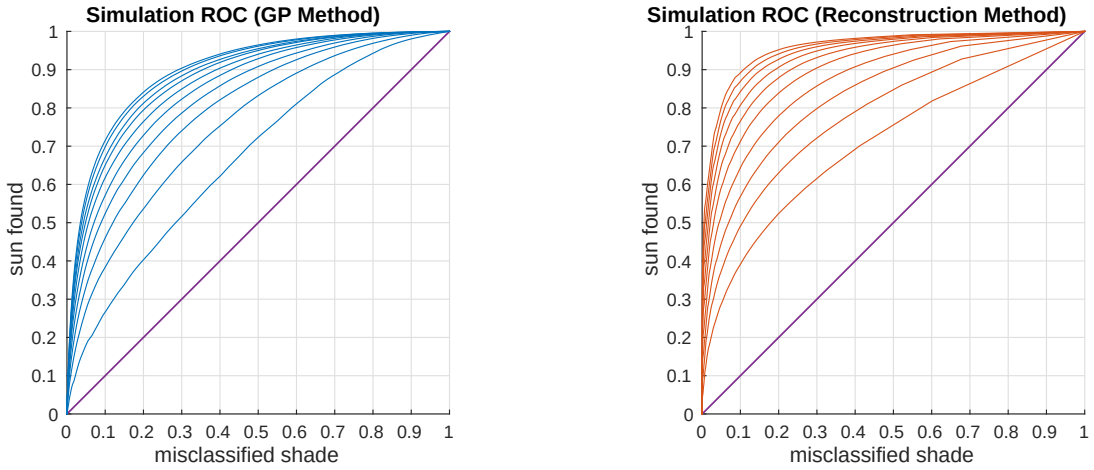


Fig. 7. Receiver Operator Characteristic plots for the two estimation methods, showing the proportion of true sun and shade that are correctly classified after 1 through 10 observation days, across all 200 simulations. These plots show that after three days, on average, the reconstruction method outperforms GP regression for every threshold.

were set with respect to the neg log likelihood of an unexplained shade measurement, $g_u = -\log(1-\exp(-\alpha)) = -\log(0.1) = 2.303$. At the Alumni Center we used $\gamma = 15g_u$ and $\xi = \frac{15g_u}{h_{max}}$. At Lake Como we used $\gamma = 5g_u$ and $\xi = \frac{5g_u}{h_{max}}$. We used less regularization at Lake Como because we were able to restrict the location of shadow casting objects to occur near the shoreline which could be estimated well from satellite imagery.

To obtain the classification fading parameters α, β_1, and β_2, we sought to maximize the cross-validation performance on a set that was held out of the heightmap construction step. We used gradient descent starting from the reconstruction fading parameters. Generally the method was overly confident in its predictions, because none of the held out measurements were very far away from measurements used in construction.

In general, the heightmap construction method is robust to a range of parameters. Adjusting α has little effect. Adjusting β changes whether a square will be raised for an object that doesn't block all rays that pass through the square: when β is increased fewer squares are used to fit any particular shadow casting object. When regularization is too strict, heightmap construction gracefully degrades towards shadow carving, with the upper heights shouldering more of the predictive power. The danger comes when regularization is too lax, and objects are confidently predicted at incorrect positions.

The GP regression hyperparameters were optimized with gradient descent, using the full data set (even for cross validation). This is because the time length parameter $1/\theta_t$ was challenging to observe when there were only two measurement times.

As a preprocessor step to both reconstruction algorithms, we combined nearby measurements if they were closer than 0.3 meters from each other and also adjacent in time, as in [1]. This helped compensate for GPS error, and it helped measurements to be more independent from each other, which is an assumption that the heightmap construction method relies on. In cross validation we used an additional post-processing step to compensate for GPS error: When we wished to know the estimated solar likelihood at a position, we sampled the estimate at 13 nearby positions and used these samples to construct a solar likelihood conditioned on a GPS error described by a 2d independent Gaussian with $\mu = 0, \sigma^2 = 1m^2$.

The classification fading parameters and GP hyperparameters that were fit to our test environments are shown in Table II.

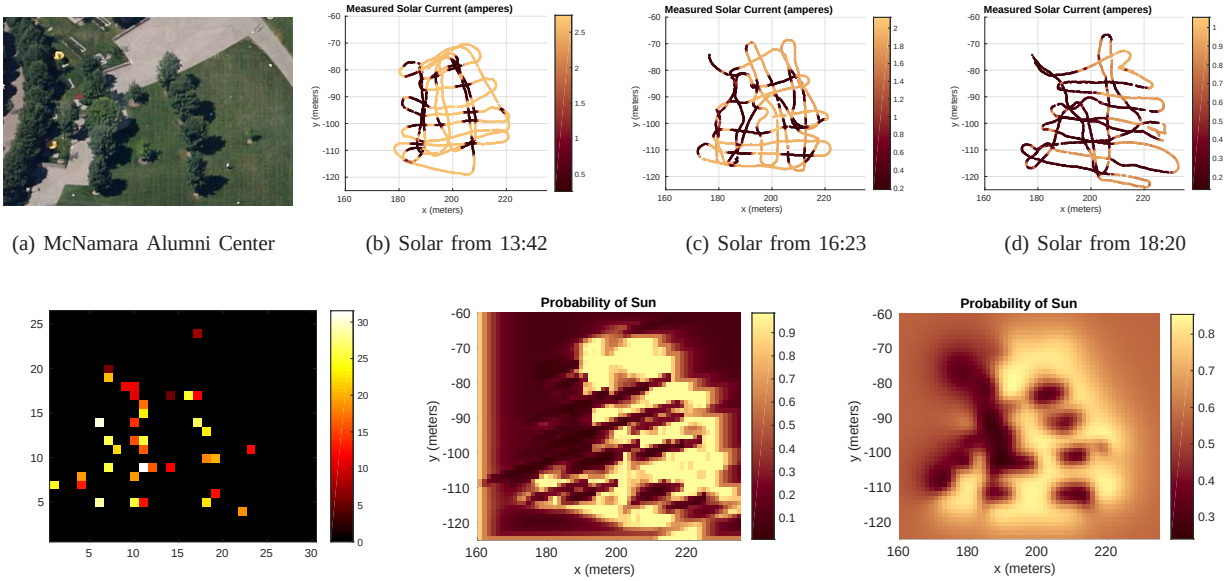
B. McNamara Alumni Center

The McNamara Alumni Center is a roughly 80m by 80m field at the University of Minnesota in Minneapolis that contains within it a few scattered trees. We mounted photovoltaic panels horizontally on a Clearpath Husky A200[2] and densely sampled the solar field of the environment on June 9th, 2012. We sampled the environment at three times of day, starting at 13:42, 16:23, and 18:20 CDT (Central Daylight Time, which has solar noon close to 13:00). Then we used our greedy heightmap construction algorithm with all of the data we had collected and estimated $\mathcal{H}^\ell$ and $\mathcal{H}^u$ for the McNamara Alumni Center. See Figure 8 for a comparison of the satellite image of the environment with the estimated lower heights, and an estimated solar map for 15:05 using both raytracing and GP regression. Observe that the two north-south lines of trees in the center of the satellite image were correctly placed in the heightmap, as were the two stray trees in between and the stray tree to the east. There were two extra trees placed in the eastern part; these were shade measurements that could not be affiliated with any of the real trees so they were inferred to come from small nearby objects.

C. Lake Como

Lake Como is an urban lake in Como Park in St. Paul, Minnesota, with a ring of trees around its perimeter. We mounted photovoltaic panels horizontally on an Oceanscience Q-Boat[3] and densely sampled the solar field of Lake Como's northeastern edge in the morning of September 16th, 2012. We sampled at three times of day starting at 08:32, 09:49, and 11:04 CDT. Then we used our greedy heightmap construction algorithm on all of the data we had collected and we estimated $\mathcal{H}^\ell$ and $\mathcal{H}^u$ for the part of Lake Como we had observed. We took advantage of our prior knowledge of the shoreline and made the approximation while constructing the lower heightmap that all shadow-casting objects were within 5m of the shoreline as manually placed from satellite imagery. See Figure 9 for a comparison of the satellite image of the environment with the estimated lower heights, and an estimate of the 08:30 solar map using both raytracing and GP regression. Observe that the correct number of trees were placed at approximately the correct positions.

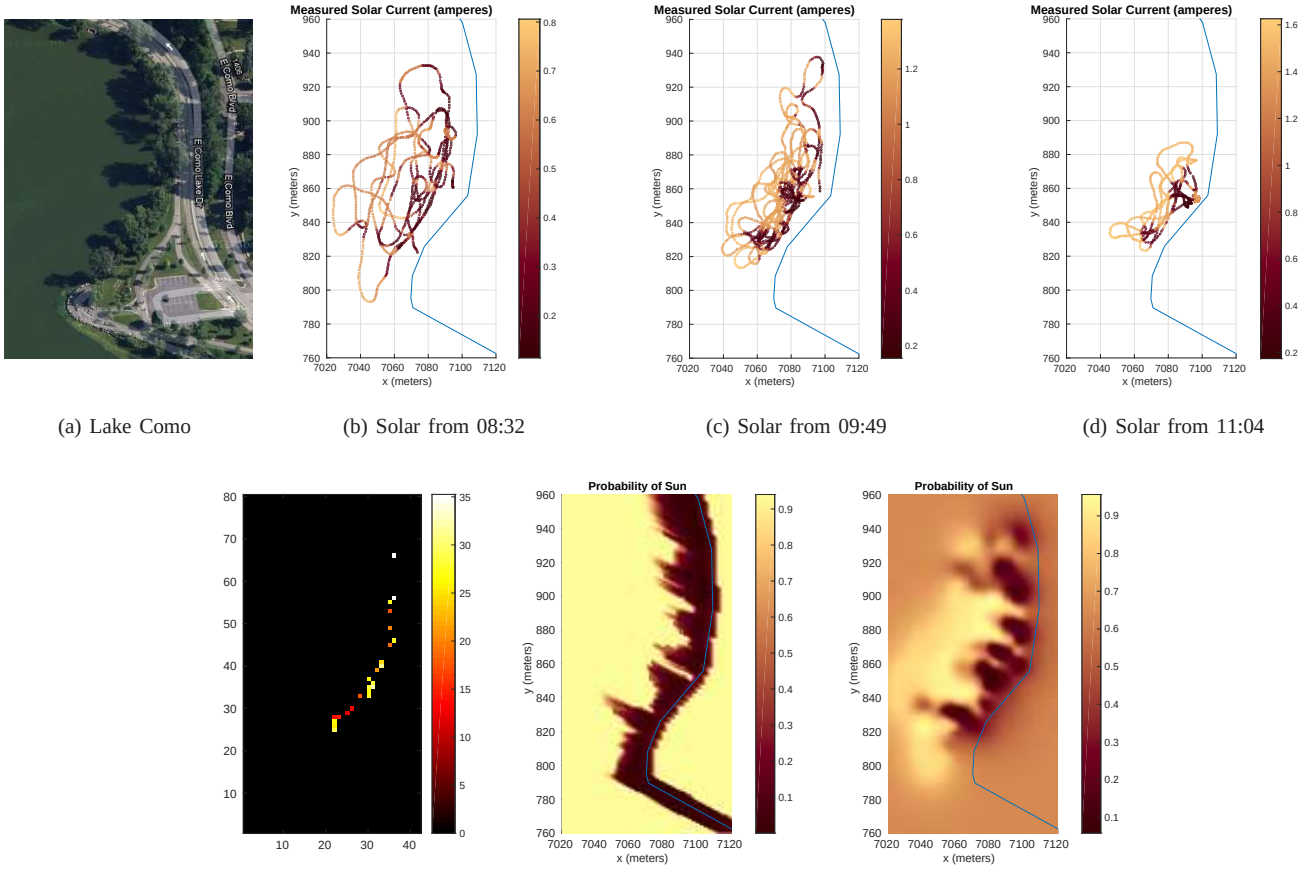
[2]www.clearpathrobotics.com
[3]www.oceanscience.com



(a) McNamara Alumni Center (b) Solar from 13:42 (c) Solar from 16:23 (d) Solar from 18:20

(e) $\mathcal{H}^{\ell}$ in meters (f) Reconstruction estimate for 15:05 (g) GP estimate for 15:05

Fig. 8. The McNamara Alumni Center testing environment, three solar observation training sets, the estimated lower heightmap constructed from these training sets, and an example solar map estimated from the constructed heightmaps compared with the estimate using GP regression. All times are Central Daylight Time.



(a) Lake Como (b) Solar from 08:32 (c) Solar from 09:49 (d) Solar from 11:04

(e) $\mathcal{H}^{\ell}$ in meters (f) Reconstruction estimate for 09:30 (g) GP estimate for 09:30

Fig. 9. The Lake Como testing environment, three solar observation training sets, the estimated lower heightmap constructed from these sets, and an example solar map estimated from the constructed heightmaps compared with the estimate using GP regression. All times are Central Daylight Time.

TABLE II. HYPERPARAMETERS FIT TO TEST ENVIRONMENTS

Trial Environment	σ_f	σ_n	$1/\theta_s$	$1/\theta_t$	μ_g	α	β_1	β_2
McNamara Alumni Center	1.0100	0.0339	4.3619m	0.4537 hours	0.2343	0.0070/m	0.8460/m	0.3313/m
Lake Como	0.9341	0.2989	8.0595m	0.8144 hours	0.3136	0.0698/m	0.8903/m	0.5472/m

D. Cross Validation

We desired a quantitative evaluation of our approach so we performed cross validation on each data set in turn, for both environments. For each cross validation trial we left out a third of the data set, constructed height maps from the other two sets, and then compared the predicted labels $\hat{\mathcal{Z}}$ with the observed labels $\mathcal{Z}_*$. We also compared the projected solar magnitudes in the sun and the shade with the observed magnitudes.

As in our simulations, we examined the ROC plots using environment reconstruction and GP regression. We also plotted the performance of a combination method which simply takes the mean of the outputs of the two methods. This combination method can be expected to perform well because its two inputs use completely different approaches to regularization and therefore can be considered almost independent, conditioned on the input. See Figures 10 for the plots. We can conclude from these that our environment reconstruction method performs better than GP regression on the time-varying solar field, even when our terrain assumption is violated. This is because it allows shadows to be intelligently interpolated and extrapolated. We can also conclude that the combination method performs better than either of its two inputs.

Finally, in Figure 11 we compare the test insolation magnitudes with the curves of estimated insolation throughout the day, as projected from the data set. We found that almost all measurements were close to either the sun or shade curve. The mean cross validation magnitude error (distance to whichever class was closest) was 0.0660 amperes at the McNamara Alumni Center and 0.1301 amperes at Lake Como. The projected solar magnitudes were far more accurate at the Alumni Center than at Lake Como; this could be because at Lake Como the sun angle was generally lower. In general this demonstrates the utility of using a classification step instead of attempting to estimate the full continuous insolation field.

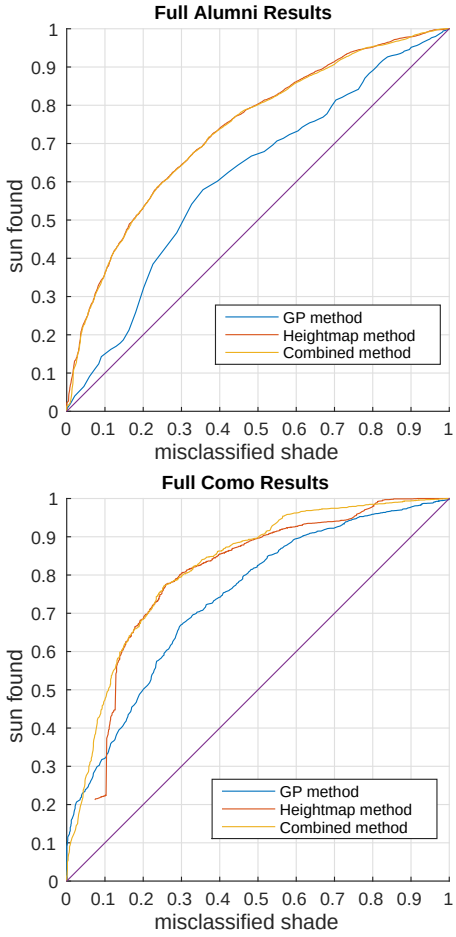


Fig. 10. Receiver Operator Characteristic plots for the two estimation methods, showing the proportion of true sun and shade that are correctly classified on the withheld third data set. At the Alumni Center and Lake Como, the reconstruction method outperformed GP regression. At the Alumni Center we gained little by using the combination method; this tells us that there were few test points that GP predicted better than reconstruction. However at Lake Como GP regression was better than reconstruction at the ends of the ROC curve, where we are seeking for points that can be estimated with very high confidence. The combined method allowed generally more of these points to be found than with either GP regression or by using the heightmaps alone.

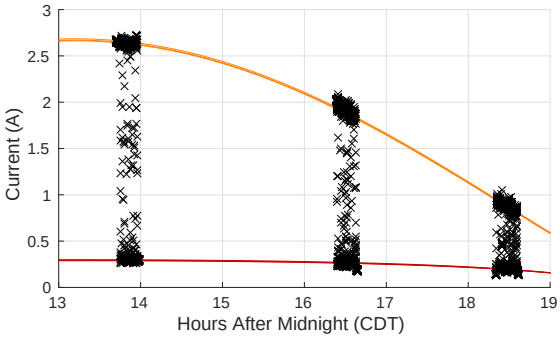
X. CONCLUSION

In this work we presented a new method to predict the solar power received by a photovoltaic panel in an uncertain and complex environment. Our method uses only previous measurements of insolation obtained from the environment. It does not require a detailed three-dimensional environment model as input or any sophisticated sensors to build such a model. The method constructs a probabilistic heightmap of the environment and then performs raytracing to estimate the probability that any point at any time will receive sunlight. We tested the method in simulation and in two environments, and demonstrated that it qualitatively performs better than spatiotemporal Gaussian Process regression.

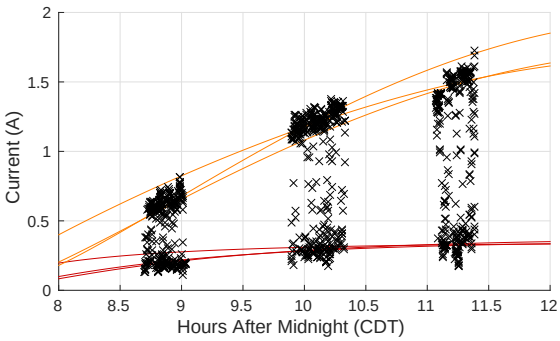
GP regression and heightmap approximation can be considered two different approaches to enforce structure: the former assumes that shadow casting objects are large, and the latter assumes that they are well-approximated by blocks of varying height sitting on the ground. A natural avenue of future work is examining whether these two forms of normalization can be used simultaneously, such as by enforcing smoothness in the heightmap estimates, in an efficient manner.

When constructing the heightmaps, all measurement rays are considered independent observations, and therefore equally informative about the position of objects. However, in reality, measurement rays from different angles are more jointly informative about the positions of objects than measurement rays from the same angle. This could potentially be modeled in f by using a covariance function between different measurement rays so that the cost when a particular square misclassifies rays from the same angle is lessened.

Other future work includes extending the proposed method to handle tilted solar panels, weather, and non-constant robot elevation. It could also be interesting to examine the use of additional sensors that fit in the simple robot framework, such as bump sensors, or inexpensive cameras. Currently we can account for localization error in classification by sampling a set of nearby points; it could be interesting to see if we gain by accounting for localization error at all stages of the algorithm, in a principled way. Finally, now that we have addressed the estimation component, it could be useful to plan exploration paths to the most information about the true object positions and sizes. In [15], we address a simple version of this exploration problem, where the task is to find the height of n objects



(a) Alumni Center



(b) Lake Como

Fig. 11. The measured solar current for our trials plotted against time of day, with best fit curves estimating the current that can be collected in the sun and in the shade. There are three curves showing the estimate for leaving out each of the data sets. At the Alumni Center the three curves agree almost perfectly, but this is not the case at Lake Como. The average cross validation error at the Alumni Center was 0.0660 amperes, and at Lake Como it was 0.1301 amperes. For planning purposes, the estimated magnitude is close to the true magnitude.

using the shortest exploration path, and assuming a constant sun angle. We present an online exploration algorithm that has an $O(\log n)$ competitive ratio with the optimal offline solution.

Acknowledgements

This report is based upon work supported by the National Science Foundation under grant numbers 1111638 and 0917676.

References

[1] P. A. Plonski, P. Tokekar, and V. Isler, "Energy-efficient path planning for solar-powered mobile robots," *Journal of Field Robotics*, 2013. [Online]. Available: http://dx.doi.org/10.1002/rob.21459

[2] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for monitoring invasive fish," *Robotics Automation Magazine, IEEE*, vol. 20, no. 3, pp. 33–41, Sept 2013.

[3] L. Ray, J. Lever, A. Streeter, and A. Price, "Design and Power Management of a Solar-Powered Cool Robot for Polar Instrument Networks," *Journal of Field Robotics*, vol. 24, no. 7, pp. 581–599, 2007.

[4] C. Sauze and M. Neal, "Long term power management in sailing robots," in *OCEANS, 2011 IEEE - Spain*, june 2011, pp. 1 –8.

[5] P. Tompkins, A. Stentz, and D. Wettergreen, "Mission-level path planning and re-planning for rover exploration," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 174–183, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889005001570

[6] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, Jan. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364911421039

[7] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, no. 1998, pp. 111–127, 2003. [Online]. Available: http://link.springer.com/article/10.1023/A:1025584807625

[8] M. Paskin and S. Thrun, "Robotic mapping with polygonal random fields," *arXiv preprint arXiv:1207.1399*, 2012. [Online]. Available: http://arxiv.org/abs/1207.1399

[9] A. Abrams, K. Miskell, and R. Pless, "The Episolar Constraint: Monocular Shape from Shadow Correspondence," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, no. d. Ieee, June 2013, pp. 1407–1414. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6619029

[10] S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini, and P. Perona, "3d reconstruction by shadow carving: Theory and practical evaluation," *International journal of computer vision*, vol. 71, no. 3, pp. 305–336, 2007.

[11] D. Y. Goswami, F. Kreith, and J. F. Kreider, *Principles of Solar Engineering*, 2nd ed. Taylor & Francis, 1999.

[12] E. Lorenzo, G. Araujo, A. Cuevas, M. Egido, J. Minano, and R. Zilles, *Solar Electricity: Engineering of Photovoltaic Systems*, first english ed. Earthscan Publications Ltd., 1994.

[13] J. Threlkeld and R. Jordan, "Direct radiation available on clear days," *ASHRAE Trans.*, vol. 64, p. 45, 1958.

[14] C. Rasmussen and K. Williams, *Gaussian processes in machine learning*. the MIT Press, 2006. [Online]. Available: www.GaussianProcess.org/gpml

[15] P. A. Plonski and V. Isler, "A competitive online algorithm for exploring a solar map," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.