# Attention in Autonomous Robotic Visual Search

A. Rasouli, J. K. Tsotsos

*Department of Electrical Engineering and Computer Science, York University, Canada*
*e-mail: {aras, tsotsos}@cse.yorku.ca*

## Abstract

One important element of autonomy for mobile robots, both terrestrial and planetary, is the ability to search for and find targets of interest whether they be important for navigation, for science missions, for localization or for manipulation. In this paper, we introduce a novel method of autonomous visual search that exploits the use of attention in the form of a saliency map that is used to enhance the probability distribution of which areas to look next, increasing the utility of spatial volumes, where objects consistent with the target's visual saliency are observed. Experimental results on a practical mobile robot are presented, which show that our proposed model improves the process of visual search in terms of reducing the time and number of actions to be performed to complete the process.

## 1 Introduction

Exploration on planetary surfaces using mobile robots (rovers) has been one of the main objectives of aerospace missions. Examples of such missions can be found in NASA Mars Exploration Programs, in which a number of rovers have been successfully launched and landed on the surface of Mars with the intention of conducting two types of missions: Science instrument placement for close measurements, and sample acquisition for return to Earth [1]. In these applications, there are constraints that limit the performance of such missions, i.e. they determine how much science can be acquired in any given Martian solar day (or sol). These constraints include human confirmation and intervention in operation, power consumption, data volume and execution time [2].

Despite advances in artificial intelligence and autonomous robotics, traditionally, Mars rovers have been mainly controlled and operated manually until recently, when in NASA's latest work Curiosity, autonomous navigation for hazardous environments was deployed for the first time. However, even in this recent work, a human operator is still a major source of control for undertaking tasks such as search for particular samples, destination selection and environment manipulation. Such human intervention can be the source of major delays during exploration. Due to the limitations mentioned earlier and the importance of maximizing efficiency of missions, it is becoming increasingly important to undertake exploration tasks autonomously by minimizing human involvements. It is estimated that by eliminating each human confirmation step in operation, overall useful activity time could be potentially increased by at least one sol [2].

An important functionality of a fully autonomous agent is the ability to search for a particular object for the purpose of environment manipulation, item detection or sample acquisition. However, the search process for an object in a given image without any attentive processes or knowledge is known to be NP-hard. It has exponential time complexity and the result is independent of its implementation [3]. The common approach to simplify this process is minimization of combinatorial problems of visual search including the relevant size of visual field, the choice of world model, or spatial and feature dimensions of interest. Strategies such as pre-segmentations of region of interest, assuming the values and the ranges of features, and knowledge of objects appearing in scenes are commonly used [4]. Although such approaches simplify the search process significantly, they are not realistic for robotic applications.

In an early version of visual search, Garvey [5] proposed the idea of indirect search in the form of a spatial relationship between an intermediate object and the target. For instance, in order to find a telephone in a room, it is better to look for surfaces e.g. tables that most likely contain the phone. Aydemir et al. [6], in a more recent work, introduced a similar active search approach with the difference of specifying the spatial relationship among objects in the form of a priori knowledge specified by an instruction to the robot such as "find the book in the box on the table". The locations of the intermediate objects are not known at the time of search. Gobelbecker et al. [7] further extended this approach by adding place recognition and defining the relationship between a particular location, e.g. kitchen, and object of interest, e.g. a coffee mug. In this model, the robot first searches for a location previously defined by an instruction, and then continues the search process, if a location of interest is identified. Kunze and Hawes [8] used more detailed descriptions of objects relationships to minimize the search space such as keyboard "*in front of*" monitor and "*left of*" laptop. In their work, only simulation results are presented and possible locations of the target are known in advance.

The major disadvantage of indirect search algorithms is that searching for an intermediate object is not necessarily simpler than finding the actual object. The recognition also is sequential, which means the robot first looks for an intermediate object and then attempts to find the target of interest. Consequently, if the spatial relation between the objects does not hold, indirect search fails to locate the target.

Alternatively, Butko et al. [9] proposed the use of saliency to guide the attention of a companion robot toward humans for social interactions. The saliency information is extracted by analyzing temporal data and detecting motion within the environment. This information is then used to move the robot to locations with a higher chance of detecting the human subjects. In this work, the detection of stationary objects was not addressed.

Cantrell et al. [10] used a bottom up approach of generating saliency information based on color distributions of objects. They only showed experimental results of a fixed location camera within a controlled environment and did not demonstrate performance of the proposed model in a cluttered background that contains similar color distributions to the target.

Shubina and Tsotsos [4] introduced a Bayesian algorithm for conducting search in an unknown environment. In this model, a uniform probability distribution is assigned to the search environment. The robot chooses the direction that yields the highest probability of detecting the target. If the object of interest was not found within the robot's effective field of view (the 3D spatial region where the recognition algorithm used in the search can detect the target), the probability of those regions are lowered to zero and redistributed to the regions that are not previously explored by the robot. The process of direction selection and recognition is continued until the target is found.

Saidi et al. [11] improved the probability reallocation of the above model by taking into account the effect of occlusion. Once an obstacle is detected, the probability distribution of the target's locations for the regions behind the obstacle is lowered as the chance of detecting the target beyond that point is smaller. Despite their strong performances, these methods do not efficiently use the information acquired through the early stages of the search. The robot only focuses on the regions within its effective depth of field and discards any information beyond that point, which can be very useful for improving the later stages of the search.

In this paper, an extension to [4] is presented that uses a general framework of saliency to guide the attention of the robot to locations with higher probability of detecting the target. At the end, an example of the proposed model, using a mobile robot, is presented followed by an empirical performance evaluation.

# 2 Searching an Unknown Environment

Assume a robot is required to search an unknown 3D environment with known exterior boundaries for a particular object. The direct approach would be to consider every possible configuration of camera geometry to capture images of previously unseen locations. Although such an exhaustive search approach would suffice for a solution, for the reasons mentioned earlier, it is not computationally or mechanically feasible.

As an alternative approach, Ye and Tsotsos [12] formulated the visual object search as a problem of maximizing the probability of detecting the target within a predefined cost constraint. They characterized the search region by the probability distribution function (PDF) of the target's presence. In this model, the control of the sensing parameters and camera geometry depends on the current search region and the recognition algorithm's ability to detect the target. The massive search space is reduced to a small, finite number of actions to be considered, each in turn, updates the status of the search space.

## 2.1 Problem Statement

A search region $\Omega$ is a 3D space to be searched with known boundaries while its internal configuration is unknown. This region is tessellated into a 3D grid of non-overlapping cubic elements, $c_i$, $i = 1 \dots n$ each holding the probability and solidity (whether or not the cube represents solid or free space). The search agent's action is defined by an operation $\boldsymbol{f}$ on $\Omega$, which consists of taking an image according to the camera configuration $S(\tau)$ and analyzing it to detect the target, where $S(\tau)$ specifies the camera position $(x_c, y_c, z_c)$, direction of viewing axis $(p, t)$, and the width and height of its solid viewing angle $(w, h)$ at time $\tau$. Actions are represented by $\boldsymbol{f} = \boldsymbol{f}(S(\tau), a)$, where $a$ is an algorithm used to analyze the image.

The cost function of action $\boldsymbol{f}$, $t(\boldsymbol{f})$, is the time (or could be extended to other forms of costs such as energy consumption) required for its execution. This cost includes every aspect of operations such as changing the sensors' configurations, acquiring an image and running a recognition algorithm.

The target distribution is specified by PDF $\boldsymbol{p}$, which is a function of both position and time as it is updated after each operation. The probability of detecting the target at location $(x, y, z)$ at time $\tau$ is given by $\boldsymbol{p}((x, y, z), \tau)$ whereas $\boldsymbol{p}(c_{out}, \tau)$ gives the probability of the target to be outside the search region $\Omega$ at time $\tau$.

The detection function $\boldsymbol{b}\big((x, y, z), f\big)$ on $\Omega$ gives the conditional probability of detecting the target by applying action $\boldsymbol{f}$ considering that the target centered at cube $c_i$, whose center is $(x, y, z)$.

Given the above definition, if the center of the cube $c_i$ falls outside of the current image, $\boldsymbol{b}(c_i, \boldsymbol{f}) = 0$ (this is also true for $c_{out}$ as it is outside of the search region $\Omega$). For those cubes within the image, the value of $\boldsymbol{b}(c_i, \boldsymbol{f})$ is determined by factors such as detection algorithm used and distance between the camera and $c_i$.

The probability of detecting the target by operation $\boldsymbol{f} = \boldsymbol{f}(S(\tau), a)$ is calculated by

$$\boldsymbol{P}_{\Psi_\mathbf{f}}(\boldsymbol{f}) = \sum_{\mathrm{c_i}\in\Psi_{\boldsymbol{f}}} \boldsymbol{p}(c_i,\tau_{\boldsymbol{f}})\boldsymbol{b}(c_i,\boldsymbol{f}), \quad (1)$$

where $\tau_{\boldsymbol{f}}$ denotes the time just before $\boldsymbol{f}$ is applied and $\Psi_{\boldsymbol{f}}$ is the influence range of the action $\boldsymbol{f}$, i.e. those parts of $\Omega$ that are visible to the search agent with the current camera's setting $S(\tau)$.

Let $\boldsymbol{O}_\Omega$ be the set of all possible operations on region $\Omega$, then the effort allocation $\boldsymbol{F} = \{\boldsymbol{f}_1, \dots, \boldsymbol{f}_k\}$, $\boldsymbol{f_i} \in \boldsymbol{O}_\Omega$, is an ordered set of operations over time applied throughout the search.

Given the above expressions, in [13,14], the problem of object search is defined as follows. Let $\boldsymbol{K}$ be the total time available for search. Then for any effort allocation $\boldsymbol{F}$, the probability of detecting the target by this allocation is,

$$P[\boldsymbol{F}] = P(f_1) + \cdots + \left\{\prod_{j=1}^{k-1}\left[1 - P(f_j)\right]\right\} P(f_k),$$

and the total time required to apply this allocation is given by,

$$T[\boldsymbol{F}] = \textstyle\sum_{f\in F} \boldsymbol{t}(\boldsymbol{f}). \quad (2)$$

According to the above definition, the task of object search is to find an allocation $\boldsymbol{F} \subset \boldsymbol{O}_\Omega$ that satisfies $T[\boldsymbol{F}] \leq \boldsymbol{K}$ while maximizing $\boldsymbol{P}[\boldsymbol{F}]$.

### 2.2 Conducting the Search

The problem of object search is proven to be NP-hard and a "greedy" algorithm would suffice as a good approximation to the solution [12]. Based on this, one action at a time can be considered, which is selected along all possible actions given the cost and effect of each one. This idea was further simplified by Ye and Tsotsos who divided the process of search into two stages of "where to look next" and "where to move next".

At the "where to look next" stage, a 'best-first' strategy is employed to examine all possible actions for the search agent at the current location. The goal is to select an operation $\boldsymbol{f} = (p, t, w, h, a)$ that yields the highest utility given by

$$\boldsymbol{E}_{\Psi_\mathrm{f}}(\boldsymbol{f}) = \frac{\sum_{c_i\in\Psi_{\boldsymbol{f}}} \boldsymbol{p}(c_i,\tau_{\boldsymbol{f}})\boldsymbol{b}(c_i,\boldsymbol{f})}{t(\boldsymbol{f})}, \quad (3)$$

where $\Psi_{\boldsymbol{f}}$ is the influence range of operation $\boldsymbol{f}$ and $t(\boldsymbol{f})$ is the time that action $\boldsymbol{f}$ takes. Since the cost of each operation at a stationary location is similar, only the numerator portion of the utility is considered.

Once an operation is applied, the target's location probabilities are updated as follows:

$$\boldsymbol{p}(c_i,\tau_{\boldsymbol{f}+}) = \frac{\boldsymbol{p}(c_i,\tau_{\boldsymbol{f}})\left(1 - \boldsymbol{b}(c_i,\tau_{\boldsymbol{f}})\right)}{\boldsymbol{p}(c_{out},\tau_{\boldsymbol{f}}) + \sum_{j=1}^{n}\boldsymbol{p}(c_j,\tau_{\boldsymbol{f}})\left(1 - \boldsymbol{b}(c_j,\tau_{\boldsymbol{f}})\right)},$$

$$i = 1, \dots, n, out, \quad (4)$$

where $\tau_{\boldsymbol{f}+}$ is the time after $\boldsymbol{f}$ is applied and $\boldsymbol{p}(c_{out},\tau_{\boldsymbol{f}+})$ is the probability that the target is outside the search region $\Omega$ at the time $\tau_{\boldsymbol{f}+}$. Intuitively, if the target is not found after operation $\boldsymbol{f}$, the probability of the influence range decreases as the other regions' probabilities increase.

After the "covering probability" of all remaining operations, $Prob_{\Psi_\mathrm{f}} = \sum_{c_i\in\Psi_{\boldsymbol{f}}}\boldsymbol{p}(c_i)$, goes below some threshold, $\Theta_{move}$, the robot goes to the next stage, "where to move next" in which the robot chooses a location to move based on two criteria, it must be reachable and have the highest probability of detecting the target. The probability of each location $j$ is calculated by $Prob_{\Psi_\mathrm{j}} = \sum_{c_i\in\Psi_j}\boldsymbol{p}(c_i)$, where $\Psi_j$ is the region within the union of all effective fields of view at position $j$.

## 3 Saliency as Look-Ahead in Visual Search

An implementation of the above approach can be found in the work of Shubina and Tsotsos [4]. They assumed a uniform probability distribution of the target's locations at the beginning of the search. Once an operation is performed by the robot, the probability of the locations are updated according to the influence range of the recognition algorithm and successes of recognition. Regions beyond the range of recognition, but within the camera field of view, are updated similarly to the rest of the environment.

The recognition algorithm used in their experiments is capable of identifying the target up to maximum range of 3 meters. Nevertheless, a typical stereo camera, similar to the one used in [4], is capable of detecting disparity within at least twice as long a range as the proposed recognition algorithm (this can even be more for other types of stereo cameras with larger base line and higher resolutions).

This difference means that a large portion of information acquired by the sensory inputs at each stage of the search is simply discarded due to the limited scope of the recognition algorithm. This information could also be analyzed to guide the search more efficiently.

In light of such potential improvement, we propose an algorithm that dynamically extracts visual clues from regions beyond the effective range of each recognition action in the form of a saliency map. This map then is used to refine the probability distribution of the potential target's locations and, as a result, direct the robot's attention to those regions with greater chance of detecting the target.

## 4 Generating the Saliency Map

The common approach to generate saliency information within an image is the use of characteristics of interest such as orientation, color, shape, motion, etc. (see [13, 14]). Despite the popularity of such methods, they would not suffice for our application. For the purpose of visual search, a more general framework is needed that not only pinpoints the possible locations of the target within an environment but also leads the attention of the search agent to those locations with a higher chance of containing the object we are looking for.

### 4.1 Attention based on Information Maximization

To develop our saliency map, we first employed the work of Bruce and Tsotsos [15], Attention based on Information Maximization (AIM). In this algorithm, an image is decomposed into independent features by applying a basis function previously trained by an Independence Component Analysis (ICA) model [16] over a large number of natural image samples. Then, the joint likelihood of these features is calculated over the entire image using a Gaussian window

$$p(w_{i,j,k} = v_{i,j,k}) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{\forall s,t \in \Psi} \omega(s,t) e^{-(v_{i,j,k} - v_{i,s,t})^2 / 2\sigma^2}, \quad (5)$$

with $\sum_{s,t} \omega(s,t) = 1$ , where $w_{i,j,k}$ denotes set of independent coefficients based on neighborhood centered at $j$ and $k$, $v_{i,j,k}$ is the local statistic value and $\Psi$ is the context on which the probability estimate of the coefficients of $\omega$ is based. Given the assumption that the ICA generated features are independent, the overall probability density function of features is given by

$$p(w_1 = v_1, w_2 = v_2, \dots, w_n = v_n) = \prod_{i=1}^{n} p(w_i = v_i). \quad (6)$$

Inspired by Shannons's self-information measure [17], $-\log(p(x))$, the information of the joint-likelihood at each local neighborhood is calculated. This information then serves as a measure of calculating salient locations, i.e. the regions that yield the most information (less common within the image) will be recognized as salient within the image.

Using ICA generated features in [15] imposes some limitations including:

- Basis functions generated by ICA do not take into account the color distribution of the object, i.e. training ICA basis functions on two identical objects with different colors will result in similar basis functions.
- Variation in scale, orientation and lighting of the object within an environment makes it challenging for ICA to learn target specific features.
- Computationally, it is neither efficient nor possible to train the system over every individual target feature. For instance, in case of RGB patches of size 21 * 21, there will be 1323 features (treating each individual pixel in each channel as a feature), which means by applying the basis function to the image, assuming a typical image size of 640*480 pixels, we will have a feature space of 1323*630*470.
- Training over a smaller subset of features will result in similar basis functions for different objects (specifically in natural images), therefore it is hard to use to identify a particular object.

Given such characteristics, ICA limits the performance of AIM in generating target specific saliency, but at the same time makes this model extremely efficient in identifying salient points that often correspond to physical structures such as tables, shelves or chairs along the common pattern of floor or wall. Identifying such structures in an image can serve as indirect search clues to guide the attention of the robot to the regions with higher probability of containing the target [18].

### 4.2 Histogram Backprojection

To add object specific information to the AIM generated saliency, we exploit the use of Histogram Backprojection [19], a method to identify similar color distributions of an object within an image.

In order to perform backprojection, we first need to generate a 3D histogram of the target's RGB color distributions. It is important that the object's template used for this purpose does not include any background information to minimize distraction in backprojection. One way of achieving an object template with minimal distracting background colors is to manually crop the object from the background. This method not only is time consuming but also is not suitable for online applications in which we intend to show an instance of the object to the robot that is not previously known. For this reason, we employed a more general approach, widely known as Expectation Maximization (EM) [20], which performs background extraction of the object's template automatically.

In this segmentation technique, a template of the target with uniform background color (preferably distinctive from those of the target) is used. The target and the background colors then are represented in the form of a multivariate Probability Density Function (PDF). The parameters of this PDF are estimated in the form of mixtures of Gaussian distributions, in our case two mixtures, which represents the distribution of the target's colors and the background (see [20] for more details). The colors in the background mixture are replaced with the RGB color black, which will be removed later from the 3D histogram of the target.

The template resulted from the EM algorithm is pixelwise normalized to minimize the effects of illumination changes. For every pixel, color values r, g and b are normalized by,

$$r' = \frac{r}{r+g+b}, g' = \frac{g}{r+g+b}, b' = \frac{b}{r+g+b}. \quad (7)$$

The normalized template is then used to establish a 3D histogram of the RGB color distributions of the object (excluding the color black as it is chosen for the background).

Let $h(C)$ be the histogram function that maps color $C = (R, G, B)$ to a bin of histogram $H(C)$ generated based on the normalized object's template $T'_\theta$. We can perform backprojection of the object over an image as follow:

$$\forall\, x, y: b_{x,y} \coloneqq h(I'_{x,y,c}), \tag{8}$$

where $b$ is the grayscale backprojection image, and $I'$ is the normalized image $I$.

## 5 Applying Saliency to Visual Search

Our proposed method starts by following the search strategy explained in Section 2 of this paper. The stereo camera mounted on the robot captures an image and the system applies the recognition algorithm to detect the target of interest. If it is found, the search is terminated and if not, the image captured is passed to the saliency module to extract interest points. The image is processed by the AIM algorithm and a conspicuity map of the interest regions is generated. This map is refined by applying a percentile threshold (in our work 80%) and then normalized to 40% of their actual values. The reason for lowering the AIM generated values is to avoid overemphasizing indirect clues that might distract the search process. A binary version of the AIM map (before normalization) is also applied to the original image in the form of a mask to extract the RGB values of the interest regions,

$$\hat{I}_{\boldsymbol{\theta}} = I_\theta \times M(\boldsymbol{x}, \boldsymbol{y}),$$

$$\begin{cases} M(\boldsymbol{x}, \boldsymbol{y}) = 1 & info(\boldsymbol{x}, \boldsymbol{y}) > \boldsymbol{p} \\ M(\boldsymbol{x}, \boldsymbol{y}) = 0 & else, \end{cases} \tag{9}$$

where $I_\theta$ is the original image captured through camera configuration $\theta$, $info(\boldsymbol{x}, \boldsymbol{y})$ is the information map resulted from AIM, $M(\boldsymbol{x}, \boldsymbol{y})$ is the binary mask and $\boldsymbol{p}$ denotes the percentile threshold.

Next, the image $\hat{I}_\theta$ is used to generate backprojection saliency, based on the predefined 3D color histogram of the target's template. This map then is normalized to 60% of its actual values.

The two normalized saliency maps are merged to form the final map to be used in the search process. With the aid of a stereo camera, the 3D coordinates of the salient locations are calculated and mapped to the 2D grid of the search environment.

The salient locations that fall within the effective range of the recognition algorithm are ignored, otherwise based on their values, the probability distribution of the target's locations will be increased accordingly. Figure 1 illustrates the entire process of building and applying saliency to the search process.

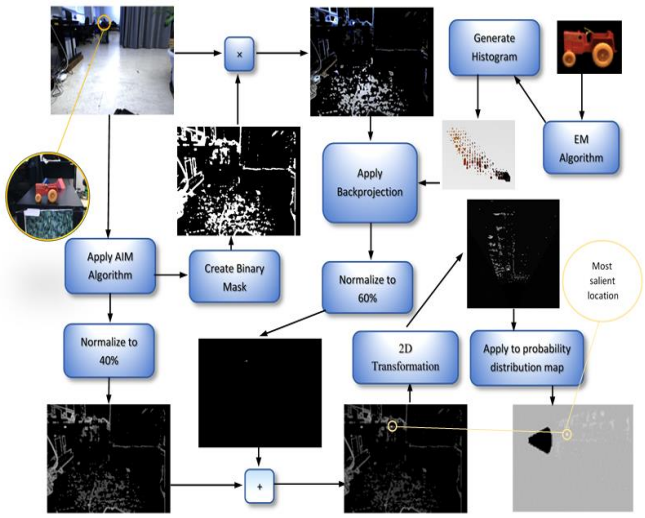


**Figure 1. Applying the saliency map to the robotic visual search. The bottom right image shows the effect of saliency on PDF of the target's location. The black cone shape indicates the effective field of view and lighter spots show the salient regions detected within the stereo camera's field of view. The grayscale color represents the probability of detecting the target, ranges from black, the lowest, to white, the highest.**

## 6 Experiments

The proposed search model was implemented on a Pioneer 3, a four-wheeled differential-drive mobile robot. The robot is equipped with a Point Grey Bumblebee stereo camera mounted on a Directed Perception pan-tilt unit. The search strategy used in our model is similar to the one in [4] with the difference of using saliency results to dynamically modify the probability distribution of the target's locations.

Each search environment was divided into $50^3\ mm^3$ voxels, which hold the target's probability and solidity values. At the time of the search, the robot did not have any prior knowledge of the target's locations, i.e. a uniform probability distribution for the target's locations was considered for the entire environment. The pan and tilt ranges of the camera are $(-158°, 158°)$ and $(-20°, 30°)$ respectively. A total of 142 different combinations of pan and tilt angles were used to select the direction that yields the highest probability. Maximum height of 1 meter was considered for the search and $\Theta_{\text{move}}$ threshold also empirically was set to allow the robot to explore its current location properly before deciding to move to the next location.

The detection method used in our experiments is based on normalized gray-scale correlation [21]. This algorithm is not view-independent, meaning that the target of interest only

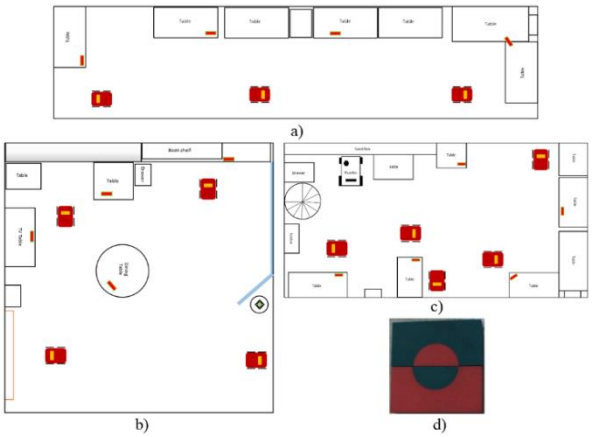


**Figure 2. Three different office environments in which experiments were conducted. Smaller red rectangles indicate the object of interest and bigger ones the robot used in the experiments. Dimensions of the environments are: a) 2.8$m$ × 11.5 , b) 6.23 $m$ × 6.20$m$, c) 4.73$m$ × 9.30$m$. d) The object used in the experiments.**

will be recognized when facing toward the camera with limited degree of transformation.

## 6.1 Indoor Environment

We primarily conducted our experiments in three office environments of various sizes and furniture configurations to evaluate the performance of the proposed model in comparison to the method in [4]. Figure 2 shows the layout of each environment and placement of the robot and the target in the experiments.

### 6.1.1 Quantitative Results

All possible combinations of the robot and the target locations were considered, comprising total of 106 experiments. For each configuration, the search methods with and without saliency were conducted and their performances were measured in terms of the number of actions performed, the time of search and distance travelled by the robot. The average performance of each method is reflected in Table 1. The results are divided into two groups of "No Move" in which the object was found before the robot moves to a new location, and "Move" in which the robot at least moved once to find the object.

**Table 1. The average results of 106 experiments conducted in 3 environments**

| Method | Factor | Search Process | | |
|---|---|---|---|---|
| | | *No Move* | *Move* | *Overall* |
| **Search with no Saliency** | *No. Actions* | 2.03 | 10.50 | 8.30 |
| | *Time(min)* | 1.48 | 12.93 | 10.03 |
| | *Distance Travelled(m)* | 0 | 11.852 | 8.915 |
| **Search with Saliency** | *No. Actions* | 2.03 | 8.49 | 6.79 |
| | *Time (min)* | 1.56 | 10.07 | 7.85 |
| | *Distance Travelled (m)* | 0 | 9.811 | 7.277 |

As Table 1 demonstrates, both search methods performed similarly in cases were the target was found from the first location of the robot. These results are anticipated as in both methods the robot first searches its surrounding environment before deciding to relocate. On the other hand, the search with saliency performed better in cases where the robot at least moved once to detect the target.

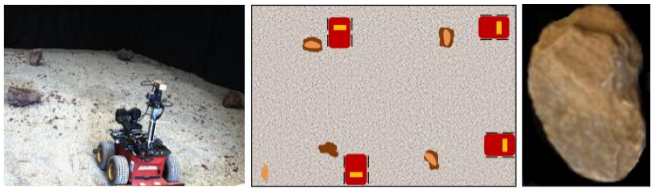


**Figure 3. The simulated environment for the planetary surface experiments. Dimensions of the environment is 5.5 x 3.8 m. a) An image of the Mars simulated ground with the pioneer 3 robot used in the experiments. b) The environment configurations out of which 9 was chosen for each search method. The orange shapes and the red rectangles are the target and the robot accordingly. c) The rock used in the experiments.**

Table 2 presents the percentage each method performed better in terms of the number of actions performed to find the object in each environment. Given the similar performance of the search methods in cases of "No Move", only situations in which the robot at least moved once are considered.

**Table 2. Performance comparison of the search methods for each environment**

| Method Performed Better | Env. (2a) | Env. (2b) | Env. (2c) | Total |
|---|---|---|---|---|
| **Proposed Method** | 77.77 % | 76.92 % | 68.75 % | 74.48 % |
| **Search with No Saliency** | 11.11 % | 7.69 % | 18.75 % | 12.51 % |
| **Similar Performance** | 11.11 % | 15.38% | 12.5 % | 12.99 % |

The proposed model of visual search using saliency clues performed significantly better in each of the three environments. However, the search with no saliency outperformed our algorithm in a number of cases, which shows that saliency information not only can be beneficial but also can distract the attention of the search agent to locations away from the target.

Another implication of the above results is variation in performance of the proposed algorithm in different environments. The saliency search performed at its worst in environment 2c, where was populated with a large amount of furniture, which created the highest amount of distraction for the robot.

## 6.2 Mars Environment Simulation

We also performed 18 experiments in a simulated Mars environment (Figure 3a) to evaluate performance of the proposed model on planetary surfaces. Similar to the above approach, the primary objective of these experiments was to detect the object of interest (a rock), using the methods of

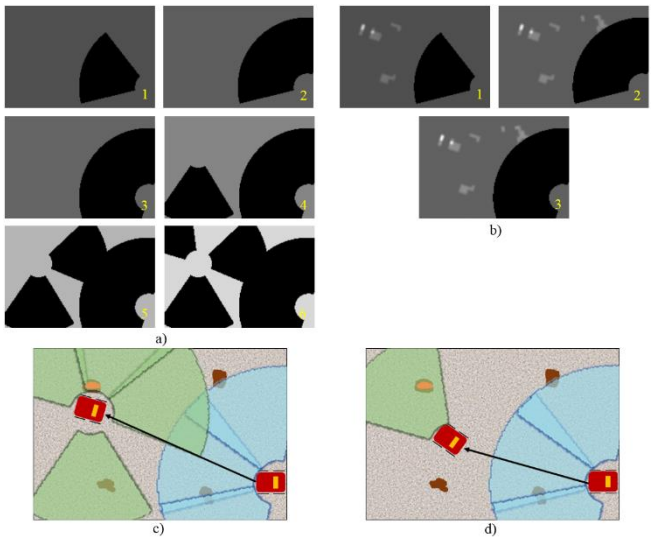


**Figure 4. A complete process of the search using methods in [4] and the proposed model. a, b) a 2D representation of the 3D probability distribution of the target's locations. The background color refers to a uniform distribution and the lighter spots in (b) represent the salient points. c, d) The complete search process for the object shown as an orange shape on the top left of each image.**

search with and without saliency, within a cluttered environment covered with sand and pebbles.

Figure 4 demonstrates a complete search process using methods in [4] and the saliency search on the Mars simulation surface. Figures 4a and 4b illustrate the sequence of a search with and without saliency respectively. The background gray color in each sequence, shows the uniform probability distribution of the target's locations generated by summing the total 3D probability environment to a 2D representation and the black regions show the locations whose probabilities are lowered to zero. Salient locations in 4b can be identified by lighter gray spots.

The entire search process for each method can be seen in Figures 4c and 4d. In these images, colored regions around the robot show the locations searched and the orange shape on the top left is the target of interest (Figure 3c).

As it is shown in Figure 4c, the robot starts searching its surrounding environment by selecting the direction with the highest probability of detecting the target. This process continues until the remaining probabilities fall below some threshold $\Theta_{move}$ at which stage the robot moves to a new location, where it detects the target after looking toward the forth direction.

Similarly, the proposed method (Figure 4d) searches its first location and moves to the next one. The robot, however, positions itself differently due to the presence of saliency responses. It then chooses the direction pointing to the top left of the environment, which not only contains a salient structure but also the target's color distribution. As a result, the target is found by only selecting one direction at the new location, improving the overall search process by 3 actions.

### 6.2.1 Quantitative Results

A total of 18 experiments were conducted by placing the robot and the target in different positions. Along all the possible combinations (Figure 3b), only those in which the robot at least had to move once to detect the target was selected (given the same reason discussed in 6.1.1).

Table 3 reflects the results of the experiments on the simulated planetary surface using both methods of search described earlier.

**Table 3. Performance comparison of the search methods for Mars simulated environment**

| Method | Factor | Average Results | Percentage the method found the target faster |
|---|---|---|---|
| **Search with no Saliency** | *No. Actions* | 7.22 | 11.11% |
| | *Time (min)* | 6.70 | |
| | *Distance Travelled (m)* | 3.47 | |
| **Search with Saliency** | *No. Actions* | 5.88 | 55.55% |
| | *Time (min)* | 5.54 | |
| | *Distance Travelled (m)* | 3.28 | |

In the planetary experiments, the proposed method once again outperformed the search with no saliency. However, a lower percentage of improvement was achieved (comparing to the office experiments) due to the limited search space available. Given the large effective range of the search methods, there is a higher chance that the robot chooses a similar location and direction in either methods of search.

## 7 Conclusion

We proposed an autonomous method of visual search using saliency clues. In this model, the internal configuration of the search environment and the target's locations are not known except those of the exterior boundaries. The saliency search method does not consider the time constraint that was part of the original theory in the selection of search operations because the primary objective of this work was to establish the benefits of a saliency map and the time required for search. If such advantages are confirmed, this method can be added to a search toolkit that is able to satisfy a time constraint.

As the search progresses, the saliency method generates information regarding the target's presence in the form of a saliency map. This map includes clues regarding the physical structure of the environment that may contain the target as well as specific characteristics of the object. By using such information, the proposed method significantly reduces the search space by directing the attention of the search agent to those interest points. Consequently, it improves the overall search process in terms of the number of actions taken, the search time and the energy consumption of the robot.

Extensive empirical evaluations showed that the nature of the environment, where the search is taking place, can greatly influence the overall performance of the search with saliency. In particular, large number of salient regions within

an environment can distract search agent from the target's location.

Our visual search method was studied in small test environments, where the dimensions of each location did not significantly exceed the effective range of the robot's recognition algorithm. It is anticipated that the proposed search method to perform better in comparison to the search without saliency in larger environments, something to be studied in the future.

In order to reduce the effect of distractors within the search environment, characteristics of the target such as size, orientation or shape also can be used to further refine the saliency map generated throughout the search process.

We conducted several experiments on a simulated Mars surface to search for a particular object. This method also can be extended to find unknown objects for the purpose of applications such as sample acquisition and exploration. The saliency algorithm used in the proposed method can be repurposed to identify anomalies or those regions that have a higher chance of containing them within an environment.

## 8 Acknowledgment

We acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the NSERC Canadian Field Robotic Network (NCFRN), and the Canada Research Chairs Program through grants to JKT.

## References

[1] M.W. Maimone, I.A. Nesnas and H. Das, "Autonomous Rock Tracking and Acquisition from a Mars Rover", *in Proc. 5th Int. Symposium on AI, Robotics and Automation in Space*, 1999, pp. 329-334.

[2] M.W. Maimone, P. C. Leger and J. J. Biesiadecki, "Overview of the Mars Exploration Rovers Autonomous Mobility and Vision Capabilities", *in Proc. IEEE ICRA*, 2007.

[3] J.K. Tsotsos, "The complexity of perceptual search tasks", in *Proc. IJCA.*, 1989, pp. 1571-1577.

[4] K. Shubina and J.K. Tsotsos, "Visual search for an object in a 3D environment using a mobile robot", *Computer Vision and Image Understanding*, vol. 114, pp. 535-547, May 2010.

[5] T.D. Garvey, "Perceptual strategies for purposive vision", Technical report, SRI International, note117, Sep. 1976.

[6] A. Aydemir, K. Sjoo, J. Folkesson, A. Pronobis, "Search in the real world: Active visual object search based on spatial relations", in *Proc. IEEE ICRA*, 2011, pp. 2818-2824.

[7] M. Göbelbecker, A. Aydemir, A. Pronobis, K. Sjöö, and P. Jensfelt, "A planning approach to active visual search in large environments", in *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[8] L. Kunze and N. Hawes, "Indirect Object Search based on Qualitative Spatial Relations", in *Proc. IROS, Workshop on AI-based Robotics*, 2013.

[9] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan, "Visual Saliency Model for Robot Cameras", in *Proc. IEEE ICRA*, 2008, pp. 2398-2403.

[10] F. Orabona, G. Metta and G. Sandini, "Object-based visual attention: a model for a behaving robot", in *Proc. IEEE CVPR workshop: Attention and Performance in Computational Vision*, 2005, pp. 89.

[11] F. Saidi, O. Stasse, and K. Yokoi, "Active Visual Search by a Humanoid Robot", *Recent Progress in Robotics: Viable Robotic Service for Human*, vol. 370, pp. 171-184, 2008.

[12] Y. Ye, J.K. Tsotsos, "Sensor planning for 3d object search", *Computer Vision and Image Understanding*, vol. 73, no. 2, pp 145–168, 1996.

[13] H. Jiang, J. Wang, Z. Yuan, T. Liu and N. Zheng, "Automatic salient object segmentation based on context and shape prior", in *Proc. British Machine Vision Conference*, 2011.

[14] E. Rahtu, J. Kannala, M. Salo and J. Heikkil, "Segmenting salient objects from images and videos", in *Proc. of ECCV*, 2010, pp. 366-379.

[15] N.D.B, Bruce and J.K.Tsotsos, "Attention Based on Information Maximization", *Journal of Vision*, vol. 7, Jun. 2007.

[16] A. Hyvarinen and E. Oja, "Independent Component Analysis: Algorithms and Applications", *Journal of Neural Networks*, vol. 13, no. 4-5, pp. 411-430, Jun. 2000.

[17] C.E. Shannon, "A Mathematical Theory of Communication", *The Bell Systems Technical Journal*, vol. 27, pp. 379-423, Jul. 1948.

[18] T. Garvey, "Perceptual strategies for purposive vision", Tech. Rep., Technical Note 117, SRI Int'l., 1976.

[19] M. J. Swain and D. H. Ballard, "Color Indexing", International Journal of Computer Vision, vol. 7, pp11-32, Nov. 1991.

[20] J. A. Bilmes. "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models". Technical Report TR-97-021, Computer Science Division, University of California at Berkeley, Berkeley, CA, Apr. 1998.

[21] W. MacLean and J.K. Tsotsos, "Fast pattern recognition using gradient-decent search in an image pyramid", in *Proc. Int. conf. on Pattern Recognition*, 2000, pp.877-881.