Visual Saliency Improves Autonomous Visual Search

Amir Rasouli, John K. Tsotsos
Dept. of Electrical Engineering and Computer Science and Center for Vision Research
York University
Toronto, Canada
{aras, tsotsos}@cse.yorku.ca

***Abstract*— Visual search for a specific object in an unknown environment by autonomous robots is a complex task. The key challenge is to locate the object of interest while minimizing the cost of search in terms of time or energy consumption. Given the impracticality of examining all possible views of the search environment, recent studies suggest the use of attentive processes to optimize visual search. In this paper, we describe a method of visual search that exploits the use of attention in the form of a saliency map. This map is used to update the probability distribution of which areas to examine next, increasing the utility of spatial volumes where objects consistent with the target's visual saliency are observed. We present experimental results on a mobile robot and conclude that our method improves the process of visual search in terms of reducing the time and number of actions to be performed to complete the process.**

Keywords- Visual Attention; Visual Search; Autonomy; Robotic; Motion Planning

I. INTRODUCTION

Autonomous mobile robots are increasingly gaining popularity in various applications including space exploration, rescue missions and assistive companionship for the elderly or infirm. Autonomy in such applications can greatly enhance the efficiency and even, in some cases, makes such systems useable for end users. For example, in space applications such as Mars exploration, removing human involvement from each step of a mission can speed up the overall process by one sol (Martian solar day) [1]. More importantly, autonomy can play a vital role in assistive applications where a large portion of their users, such as elderly or people with significant disabilities, lack the ability of operating manual systems [2].

A major component of an autonomous robot is the capability of searching for a particular object for the purpose of environment manipulation, item detection and pick up. However, the process of search and detection for an object in a given image without any attentive processes or knowledge is proven to be NP-hard. It has exponential time complexity and the result is independent of its implementation [3,4]. To simplify this process, the most common approach in computer vision research 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 [5]. Although such approaches simplify the search process significantly, they are not realistic for robotic applications.

In an early version of visual search, Garvey [6] 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. This idea appears in more recent work by Aydemir et al. [7]. They 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. For instance, "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. [8] 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 instruction, and then continues the search process, if a location of interest is identified. Kunze and Hawes [9] used more detailed descriptions of object relationships to minimize the search region such as keyboard "*in front of*" monitor and "*left of*" laptop. In this work, only simulation results are presented and possible locations of the target are known in advance.

The major drawback of indirect search algorithms is the fact 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.

An alternative approach to improve visual search was introduced by Butko et al. [10] who used saliency to guide the attention of a companion robot toward humans for social interactions. The saliency information was extracted by analyzing temporal data and detecting motion within the environment. Once developed, this information was used to move the robot to locations with a higher chance of detecting the human subjects. In this work, no evidence of detecting stationary objects is presented. Cantrell et al. [11] proposed 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. Their experiments do not

demonstrate performance of the proposed model in a cluttered background that contains similar color distributions to the target.

Shubina and Tsotsos [5] proposed 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), 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. [12] 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, we propose an extension to [5] that employs a general framework of saliency to guide 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.

II. Searching an Unknown Environment

Assume we want to search an unknown 3D environment with known exterior boundaries for a particular object. The direct approach would consider every possible configuration of camera geometry to capture images of previously unseen locations. Such a brute-force approach would suffice for a solution but for the reasons mentioned earlier it is not computationally or mechanically feasible.

To address this issue, Ye and Tsotsos [13,14] formulated the visual object search as a problem of maximizing the probability of detecting the target within a predefined cost constraint. In this model, the search region is characterized by the probability distribution function (PDF) of the target's presence. The control of the sensing parameters depends on the current search region and the recognition algorithm's ability to detect the target. The massive search space is reduced to a finite number of actions to be considered, and each in turn, updates the status of the search space.

A. Problem Statement

A search region Ω 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, $i = 1 \dots n$. The search agent's action is defined by an operation $\boldsymbol{f}$ on Ω, 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 τ. 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 τ 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 Ω at time τ.

The detection function $\boldsymbol{b}((x, y, z), f)$ on Ω 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 Ω). 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

$$\boldsymbol{P}_{\Psi_f}(\boldsymbol{f}) = \sum_{c_i \in \Psi_f} \boldsymbol{p}(c_i, \tau_f) \boldsymbol{b}(c_i, \boldsymbol{f}) , \tag{1}$$

where τ_f denotes the time just before $\boldsymbol{f}$ is applied and Ψ_f is the influence range of the action $\boldsymbol{f}$, i.e. those parts of Ω 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 Ω, 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,

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

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

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

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 $\boldsymbol{P}[\boldsymbol{F}]$.

B. Conducting the Search

Ye and Tsotsos proved that the problem of object search is NP-hard and showed that a "greedy" algorithm suffices as a good approximation to the solution [14]. To that extent, they consider one action at a time, which is selected along all possible actions given the cost and effect of each one. They further simplified the process of search into two stages of "where to look next" and "where to go 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 $\boldsymbol{f} = \boldsymbol{f}(p, t, w, h, a)$ that yields the highest utility given by

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

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

After application of the candidate operation, the target's location probabilities are updated as follow

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

$$i = 1, \dots, n, out\ , \tag{4}$$

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 Ω 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.

Once the "covering probability" of all remaining operations, $Prob_{\Psi_{\mathrm{f}}} = \sum_{c_i \in \Psi_f} \boldsymbol{p}(c_i)$, goes below some threshold Θ_{move}, the robot moves to a different location where the probability of detecting the target is higher.

At this stage, "where to move next", the robot has two criteria to choose the next best location; it must be reachable and have a high probability of detecting the target. This probability at each location j is calculated by $Prob_{\Psi_{\mathrm{j}}} = \sum_{c_i \in \Psi_j} \boldsymbol{p}(c_i)$, where Ψ_j is the region within the union of all effective fields of view at position j. Then the robot chooses the location with the largest $Prob_{\Psi_j}$ and moves there.

III. Saliency as Dynamic prior knowledge for search

Shubina and Tsotsos [5] presented an implementation of the above search algorithm on a mobile robot. They assumed a uniform probability distribution of the target's locations at the beginning of the search. After each operation, the probability of the locations are updated according to the influence range of the recognition algorithm and successes of recognition. Any regions beyond the range of recognition, but within the camera field of view, are simply 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. The stereo camera used, however, is capable of detecting disparity within at least twice as long a range as the recognition algorithm (this can even be more for other types of stereo cameras with larger base line and higher resolutions).

This difference implies 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. Such information could also be analyzed to guide the search more efficiently.

We thus propose an algorithm to dynamically extract visual clues from regions beyond the effective range of each recognition action in the form of a saliency map. This map then can be 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 more potential.

IV. Development of the Saliency Map

It is a common practice in the computer vision literature to identify salient locations within an image based on characteristics of interest such as orientation, color, shape, motion, etc. (see [15, 16]). Despite their promising performance, these methods would not suffice for our application. We require a more general framework that not only pinpoints the possible locations of the required object within an environment but also leads the search agent to those locations with a higher chance of containing the object we are looking for.

A. Attention based on Information Maximization (AIM)

As a first step to develop our saliency map, we employed the work of Bruce and Tsotsos [17], commonly known as AIM. This saliency map algorithm begins by decomposing an image into independent features by applying a basis function previously trained by an Independence Component Analysis (ICA) model [18] 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

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

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 Ψ is the context on which the probability estimate of the coefficients of ω is based. Given the assumption that the ICA generated features are independent, the overall probability density function of features is given by

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

Inspired by Shannons's self-information measure [19], $-\log(p(x))$, the information of joint-likelihood at each local

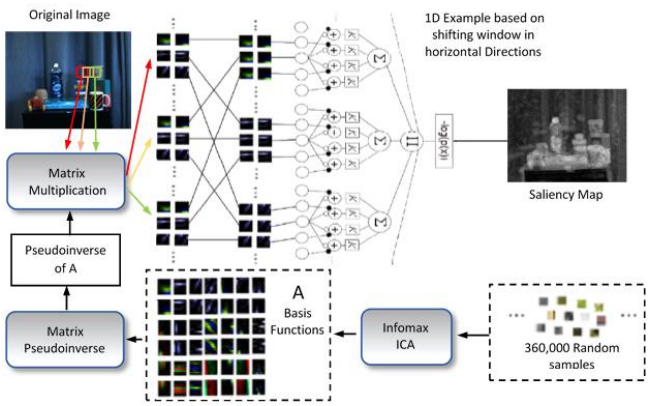

Figure 1. The framework of achieving information measures by application of AIM to a sample image.

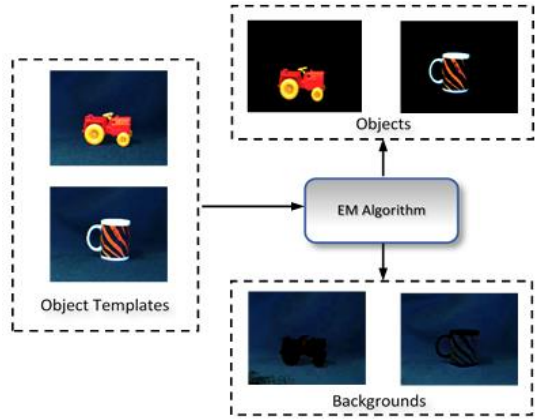

Figure 2. The EM algorithm process to extract two sample objects from their templates' background.

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 [17] 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.

These characteristics of ICA limit the performance of AIM in generating target specific saliency; however, this also 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 [20].

B. Histogram BackProjection

The saliency map generated by AIM is further refined by applying Histogram Backprojection [21], a method that is commonly used to identify similar color distributions of an object within an image.

In this model, the first step is to generate a 3D histogram of an object's RGB color distributions. It is important that the object's template used for this purpose does not contain 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 employ a more general approach that performs background extraction of the object's template automatically.

A clustering method with Gaussian Mixtures [22] is used to perform template segmentation (see Figure 2). In this algorithm, the object and the background colors are represented in the form of a multivariate probability density function. The objective is to estimate parameters of this PDF in the form of mixtures of Gaussian distributions. To gain better performance, it is important to provide the model with templates that have uniform background colors, preferably distinguishable from the object. The Expectation Maximization (EM) algorithm is applied for clustering template color distributions. Assume the following probability distribution,

$$p(x|\Theta) = \sum_{i=1}^{m} \alpha_i p_i(x|\theta_i), \tag{7}$$

where m is the number of mixtures (in our case 2) and parameters are $\Theta = (\alpha_1, \dots, \alpha_m, \mu_1, \dots, \mu_m, \Sigma_1, \dots, \Sigma_m)$ in which $\alpha_i \geq 0$ denotes mixing coefficient (weight) of each mixture such that $\sum_{i=1}^{m} \alpha_i = 1$, μ_i and Σ_i refer to mean and covariance of normal distribution p_i respectively that is given by,

$$p_i(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} , \tag{8}$$

where d denotes dimensionality of Gaussian distribution. Let x_i be the image samples, we intend to find maximum likelihood estimate (MLE) of all mixture parameters of Θ,

$$\log\big(L(\Theta|x)\big) = \log \prod_{i=1}^{N} p(x_i|\Theta)$$
$$= \sum_{i=1}^{N} \log\left(\sum_{j=1}^{m} \alpha_j p_j\big(x_i|\theta_j\big)\right). \quad (9)$$

The EM algorithm estimates the parameters in two steps. First, the Expectation or E-step in which the probability $p_{i,j}$ of sample i belongs to mixture j using currently available parameters is determined,

$$p_{i,j} = \frac{\alpha_j p_j(x|\mu_j, \Sigma_j)}{\sum_{k=1}^{m} \alpha_k \, p_k(x|\mu_k, \Sigma_k)}. \quad (10)$$

At the second stage, M-step, mixture parameters are refined using computed probabilities,

$$\alpha_j = \frac{1}{N}\sum_{i=1}^{N} p_{i,j}, \qquad \mu_j = \frac{\sum_{i=1}^{N} p_{i,j} x_i}{\sum_{i=1}^{N} p_{i,j}},$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} p_{i,j}(x_i-\mu_j)(x_i-\mu_j)^T}{\sum_{i=1}^{N} p_{i,j}}. \quad (11)$$

Alternatively, the E-step and M-step can be applied in reverse order depending on availability of data at the time of calculation (see [22] for more details). Once the background color distribution is calculated, its values are replaced with RGB value zero (black). Then the resulting template is pixelwise normalized to minimize the effects of illumination changes. For every pixel, color values r, g and b are normalized by,

$$r' = \frac{r}{r+g+b}, g' = \frac{g}{r+g+b}, b' = \frac{b}{r+g+b}. \quad (12)$$

The normalized template is then used to establish a 3D histogram of RGB color distributions of the object. Note that the color black is ignored in the histogram as it is defined for the template 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'_θ. We can perform backprojection of the object over an image as follow:

$$\forall\, x, y: b_{x,y} := h(I'_{x,y,c}), \quad (13)$$

where b is the grayscale backprojection image, and I' is the normalized image I(see Figure 3).

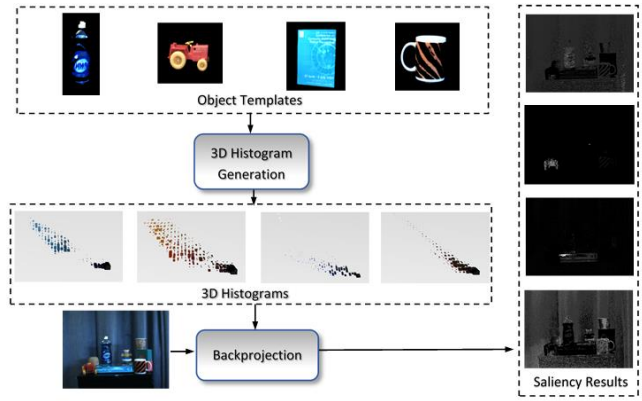

Figure 3. Histogram Backprojection results of four sample objects. Saliency results from top to bottom refer to the object templates from left to right.

V. Applying Saliency to Visual Search

We start by following the search strategy explained in Section II. 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 (as discussed earlier) and a conspicuity map of the interest regions is generated. The AIM results are 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 on 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 RGB values of the interest regions,

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

$$\begin{cases} M(\boldsymbol{x}, \boldsymbol{y}) = 1 & info(\boldsymbol{x}, \boldsymbol{y}) > \boldsymbol{p} \\ M(\boldsymbol{x}, \boldsymbol{y}) = 0 & else, \end{cases} \quad (14)$$

where I_θ is the original image captured through camera configuration θ , $info(\boldsymbol{x}, \boldsymbol{y})$ 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 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 create the final map to be used in the search process. With the aid of the stereo camera, the 3D coordinates of the salient locations are calculated and mapped to the 2D grid of the search environment.

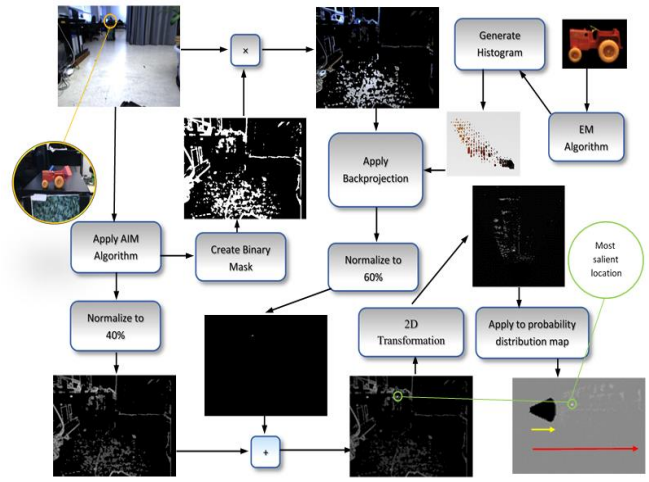

Figure 4. The overall process of generating the saliency map and applying to the probability distribution map of the target. The image on the right bottom is a 2D slice of the probability distribution map demonstrating the range and effect of saliency observations. The yellow and red arrows indicate the effective range and the stereo camera's range respectively.

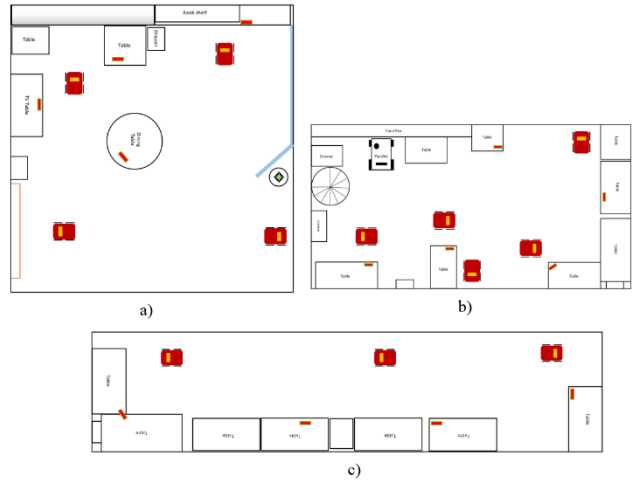

Figure 5. Three different environments in which experiments were conducted. Smaller red rectangles indicate the object of interest and bigger ones the robot used in the experiments. Dimensions of the environments are: a) $6.23\ m\ \times 6.20m$, b) $4.73m \times 9.30m$, c) $11.5m \times 2.8m$.

If salient locations fall within the effective range of the recognition algorithm, they are ignored, otherwise based on their values, the probability distribution of the target's locations will be increased accordingly.

As it is illustrated in Figure 4, the general interest points, such as those corresponding to the physical structures within the environment, are extracted by the AIM algorithm. The AIM map is further refined by using histogram backprojection to distinguish between the structures that more likely contain the target. This step can be helpful in cases where similar saliency responses are generated by AIM for different structures.

VI. Experiments

We implemented our saliency model on a Pioneer 3, a four-wheeled differential-drive mobile robot. The robot is equipped with a Point Gray Bumblebee stereo camera mounted on a Directed Perception pan-tilt unit. The search strategy used in our model is similar to the one in [5] with the difference of using saliency results to dynamically modify the probability distribution of the target's locations.

In our experiments, three office environments of various sizes and furniture configurations were used (see Figure 5). Each location was searched up to height of 1 meter from the ground and divided into $50^3 mm^3$ voxels that 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. Θ_{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 [23]. This algorithm is not view-independent, meaning that the target of interest only will be recognized when facing toward the camera with limited degree of transformation.

Figure 6 demonstrates a complete search process inside office environment 5b using method in [5] and our proposed model. Figure 6a illustrates sequence of a search without using saliency. The background color shows the uniform probability distribution of the target's locations generated by summing the total 3D probability environment to a 2D representation and black regions show the locations whose probabilities are lowered to zero. The obstacles within the environment are also presented with the color green. Similarly in Figure 6b, the search process with the use of saliency is illustrated. The lighter spots on the uniform gray background refer to salient locations that are detected in earlier stages of the search.

The overall process of the search for each method can be seen in Figures 6c and 6d. In these images, the cone shape colored regions show the robot's effective field of view. The red rectangle on the bottom left is the target of interest (Figure 6f) and the red rectangular shape with black and yellow spots represents the robot.

As illustrated in Figure 6c, the robot first starts searching its surrounding environment by choosing the directions with the highest probability of detecting the target. This process continues until the remaining probability of the current location falls below threshold Θ_{move}. At this point, the robot moves to the next location that yields a higher probability of finding the target at which stage it repeats a similar routine until it detects the target after looking toward the third direction.

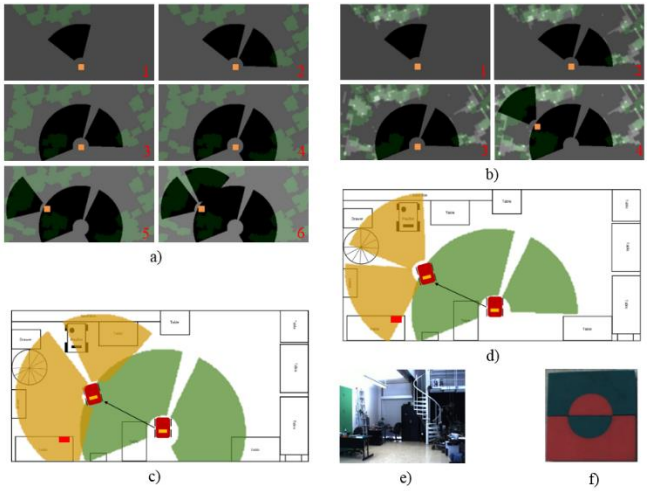

Figure 6. A complete process of the search using methods in [5] and the proposed model. a,b) 2D representation of the 3D probability distribution of the target's locations. The background color refers to a uniform distribution and the lighter color spots in (b) represent the salient points. The green regions represent the obstacles within the environment. c,d) The complete search process for the object shown as a red rectangle on the bottom left of each image. e) An image from the office environment 5b, where the experiments were conducted. f) The object used in the experiments.

When the robot uses the saliency algorithm (Figure 6d), it follows the same strategy as 6c to conduct the search from its initial position. However, after observing each direction, saliency information is generated that corresponds to the potential locations of the target (Figure 6b). Based on this information, the robot only performs three actions at its initial location, and then moves to the new position, where it detects the target by looking toward the second direction. As a result, the overall process of the search is improved by two actions in comparison to the search without using saliency.

A. *Quantitative results*

We conducted total of 106 experiments by placing the target and the robot at different unique locations (see Figure 5) and ran the object search with and without the use of saliency. The average performances of both methods are reflected in Table 1, where 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 I. 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	8.54	6.83
	Time (min)	1.56	10.07	7.85
	Distance Travelled (m)	0	9.811	7.277

As shown in Table 1, both search methods performed similarly in cases were the target was found from the first location of the robot. These results are expected as, in the proposed model, saliency clues are only effective after the first movement of the robot to a new location. On the other hand, the search with saliency performed superior in cases where the robot at least moved once to detect the target. A performance comparison of both methods for each test environment is also presented in Table 2 in terms of the number of actions performed to find the object. 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 II. PERFORMANCE COMPARISION OF THE SEARCH METHODS FOR EACH ENVIRONMENT

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

Our proposed model of visual search using saliency clues performed significantly better in each of the three environments. However, there were cases in which the search with no saliency outperformed our algorithm due to the fact 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 proposed model performed at its worst in environment 5b, where was populated with a large amount of furniture, which created the highest amount of distraction for the robot. In contrast, the saliency search performed better in environments, where less furniture was present.

VII. CONCLUSION

We showed the benefits of attentive mechanisms in the visual search for an object in a 3D environment. In the proposed model, no assumption is made about the target's locations or the environment's configuration except those of the exterior boundaries. The time constraint that was part of the original theory also is not applied in the selection of search operations because this work was concerned with establishing the benefits of a saliency map and time required for search, which if confirmed, allow this method to be added to a search toolkit that is able to satisfy a time constraint.

In the saliency model, as the search progresses, information regarding the target's presence is generated in the form of a saliency map, which contains clues regarding the physical structure of the environment that may contain the target as well as specific characteristics of the object. By relying on such knowledge, the search with saliency 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.

Through extensive empirical evaluations, it also was shown that the nature of the environment, where the search is taking place, can greatly influence the overall performance of the proposed model. As the number of salient regions within an environment increases, there is a higher chance that the robot will be distracted from the target's location.

We have evaluated our method in small office 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 performance gap between our proposed model and the search with no saliency grows as the size of the search environment increases, something to be studied in the future.

Furthermore, our saliency algorithm was only used within a search policy in which the robot always completed searching its current location before it considers moving to a new position. It would be beneficial to evaluate the performance of the saliency search using different strategies. For instance, the robot can evaluate all actions within the entire environment instead of searching its current location before moving to the next one. In this approach, the saliency information can be used to guide the search not after the first movement of the search agent but instead, after the first action is performed.

We also only used one characteristic of the object of interest to refine the saliency map. Using additional target features such as size, orientation or shape can further enhance the performance of the saliency search by reducing the effects of distractors in the search environment.

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

[2] J. K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M. Black, D. Metaxas, S. Culhane, Y. Ye, and R. Mann, "PLAYBOT: A visually-guided robot to assist physically disabled children in play", *Image & Vision Computing Journal, Special Issue on Vision for the Disabled*, vol. 16, pp. 275-292, Apr. 1998.

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

[4] J.K. Tsotsos, "A 'complexity level' analysis of vision", in *Proc. International Conference on Computer Vision: Human and Machine Vision Workshop*, 1987, pp. 346–355.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] Y. Ye, "Sensor planning for object search", Ph.D. thesis, Dept. Computer Science, University of Toronto, Toronto, ON,1997.

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

[15] 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.

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

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

[18] 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.

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

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

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

[22] 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.

[23] 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.