Live WWW and its Agents: the CAS-LWWW Project

Yiming Ye John K. Tsotsos
Department of Computer Science
University of Toronto
Toronto Ontario
Canada M5S 1A4

Karen Bennet
IBM Centre for Advanced Studies
Stn. 2G, Dept. 894
1150 Eglinton Avenue East
North York, Ontario
Canada M3C 1H7

Abstract

This paper proposes the idea of vision agents over Internet, outlines the performance models of Live WWW with agents, and describes an object search agent and its communications with other agents. The goal is to reduce the traffic on the Internet for Live WWW by integrating the WWW technology, agent theory and computer vision technology together.

1 Introduction

The most significant recent developments enabling the creation of global information resources and world-wide computer-based interpersonal communications are the incredible growth of the Internet and the success of the World Wide Web (WWW). The Web implements a global hypermedia that ultimately could incorporate much of the world's knowledge that exists in tangible form. The Net implements a global communications system that ultimately could facilitate dialogue and interaction among many of the world's people.

The sheer joy of browsing the world and finding the right nugget of information in the form of hypermedia has attracted researchers from many fields to the problem of how to solve core technical problems posed by the growth of global networks, how to improve the current WWW and Internet technology, and how to predict or influence what the Internet or intranet will look like.

The WWW technology has advanced so rapidly that even graphics, audio, and video can be integrated into the Web. Most previous research has concentrated on handling prerecorded data on the WWW. Recently, efforts have been made to make the WWW live. One spawn of these efforts is the MBONE (Multicast Backbone) [12] technology —— an interconnected set of routers and subnets that provides IP multicast delivery in the Internet. The MBONE is a virtual network. It is layered on top of portions of the physical Internet to support routing of IP multicast packets since that function has not yet been integrated into many production routers. The network is composed of islands that can directly support IP multicast, such as multicast LANs like Ethernet, linked by virtual point-to-point links called "tunnels". The tunnel endpoints are typically workstation-class machines having operating system support for IP multicast and running the "mrouted" multicast routing daemon. In the IP multicast tunnels, IP multicast packets are encapsulated for transmission through tunnels, so that they look like normal unicast datagrams to intervening routers and subnets. A multicast router that wants to send a multicast packet across a tunnel will prepend another IP header, set the destination address in the new header to be the unicast address of the multicast router at the other end of the tunnel, and set the IP protocol field in the new header such that the next protocol is IP. The multicast router at the other end of the tunnel receives the packet, strips off the encapsulating IP header, and forwards the packet as appropriate. Driven by the availability of popular new multicast applications, particularly those providing real-time audio and video teleconferencing capabilities to Internet hosts, the MBONE has grown exponentially. Another effort to make the WWW live is represented by the WWW browser *Vosaic*, or *Video Mosaic*, which extends the architecture of the WWW to encompass the dynamic, real time information space of video and audio [2]. In *Vosaic*, video and audio transfers occur in real time; there is no file retrieval latency. The video and audio result in compelling Web pages. Real time video and audio data can be effectively served over the present day Internet with the proper transmission protocol. A real time protocol VDP is used to handle real time video over the WWW. VDP reduces inter-frame jitter and dynamically adapts to the client CPU load and network congestion. *Vosaic* dynamically changes transfer protocols, adapting to the request stream and the meta-information in requested documents. Bellcore [17] also proposes the real-time data services for the Web, RAVE. RAVE supports live and stored video and audio as well as less typical real-time services such as information services (e.g., news or stock market feeds). The system is extensible so that new real-time services can easily be added. It supports unicast and multicast, as well as both real-time data sources and sinks. RAVE integrates well into the Web so that many multimedia applications may be written using simple HTML extensions. When acting as a live-data source, the RAVE server can connect to hardware input devices (e.g., a video capture device, a sound card, or a serial interface connected to an external text feed). The real time data is packaged in a uniform format and sent to clients over the network. Right now, various Internet sites offer the possibility to view an image captured by a computer with refresh rates varying from one frame every few second to one frame every four hours. In net lingo these images came to be known as "live" in contrast with the rest of the usually static sites. The MCS-WebCam might be one of the most popular site [16]. The WebCam is an easy to install real time video server that allows transmission of live video and audio to a number of remote clients. Rather than go the beaten path of plug-ins, the WebCam server seeks to use established standards such a server-push and JPEG and it works without any change to the configuration of the end-users machine.

The live WWW opens many exciting application areas for WWW. Several problems, however, make the real time transmission of the video data from the server site to the client site unappealing, such as the intense use of bandwidth, poor quality of video image, and high price. When the modem at the client site is too slow, the real time video transmission simply does not work.

The goal of the CAS-LWWW (Centre for Advanced Studies-Live World Wide Web) project is to solve the above problems and reduce the Internet traffic by integrating computer vision technology, agent theory and WWW technology together. Instead of frequently transmitting the image data grabbed from the camera to the client and let the client to analyze the video image, we simply build agents at the server site and ask the agents to help executing the tasks required by the clients. Therefore, the image grabbed from the camera is first **pre-processed** by the agents, only those images that are of interest to the client are sent through the Internet. Thus the bandwidth used is greatly reduced. This strategy can be applied to many areas, such as security surveillance, etc. It is even more interesting when the camera need to be controlled in order to perform a certain task, such as searching for an object in the server's site. The client does not need to interactively control the camera and get the video data over the Internet again and again. The search agent will control the camera and perform the image processing operation at the server's site, and only the most promising images that might contain the target are transmitted to the client over the Internet.

2 The High Bandwidth Requirement for Live WWW

It can be predicated that as more and more people begin to use live WWW, the WWW will burst at the seams due to the lack of bandwidth from servers to clients. To make the video over Internet to appear "live", a high-bandwidth is required. A large percentage of web clients, however, are run over low-speed 28.8 or 14.4 modems, and users are increasingly considering wireless services such as cellular modems at 4800-9600 baud. A recent study by a popular server of shareware, Jumbo, revealed that about 1 in 5 users were connecting with graphics turned off, to eliminate the annoying latency of loading web pages [6], not to say when live video is included in the web pages. To make the live video over Internet popular, a study of how to reduce the bandwidth used by the live WWW is required.

A nature step in reducing the bandwidth used is to reduce the amount of data that has to transmit over the Internet. Thus, image compression, that is to remove the data redundancy, code redundancy, inter-pixel redundancy, and psycho-visual redundancy in an image before it is transmitted, is a necessary step in live WWW. Various compression standards, most of which are based on JPEG [11] and MPEG [7], have been used for live WWW and other multi-media WWW. JPEG offers four modes of operation: lossless encoding, sequential encoding, progressive encoding, and hierarchical encoding. The basic JPEG algorithm first represents the original image in the frequency domain and then achieves data compression by concentrating most of the signal power in the lower spatial frequencies and reducing to zero the high frequencies with small coefficients during the quantization step. The progressive JPEG encodes the image in multiple coarse-to-detailed passes, thus when displayed, a blurry image first appears and it is refined as more image data arrives. The hierarchical JPEG encode the image at multiple resolutions, thus lower-resolution versions may be accessed without first having to decompress the image at its full resolution. The MPEG is based on two basic techniques. One is block-based motion compression for the reduction of temporal redundancies, which is achieved by motion prediction and motion interpolation with the assumption that the blocks of the current picture can be modeled as a translation of blocks of some previous picture or a combination of references to past and future pictures. The other is Discrete Cosine Transform based compression for the reduction of spatial redundancies.

The sizes of the images or video sequences compressed by JPEG and MPEG can be greatly reduced. JPEG achieves a 15:1 average compression ratio and MPEG achieves compression ratios up to 200:1 on average by storing only the difference between successive frames. In spite of the high compression ratio achieved by JPEG and MPEG, the amount of data needed to be transmitted over the Internet is still large for live video. Thus, in order to make the site appear "live", many live WWW sites have to reduce the transmitting rate. For example, Telepresence Systems, Inc. [9] is working on the ProRata system which provides groups of geographically separated people with a sense of shared presence. It takes video snapshots of participating individuals every 5 minutes, merging the pictures into a composite and then sending the composite image back to each person in the group via the Internet.

All of the above approaches for reducing the bandwidth used over the Internet does not involve intelligent agent. There is no consideration of the browser's requirement at the browsing time. All the images are processed in a fixed mode, with no consideration of the perception ability of the browser and the usefulness of the image to be transmitted. We propose an idea of reducing the data by using *vision agents* to intelligently select and transmit only the images that are interesting to the browser and compress the original image according to the perception abilities of the browser.

3 Vision Agents Over Internet

The concept of an agent has become important in user interfaces, artificial intelligence, and mainstream computer science. Research on and discussion about agents has mushroomed in the past few years. Although it is not easy to provide a universally accepted definition, we can say that an *agent* is a hardware or software-based computer system that has some of the following properties:

- *Awareness*: agents have knowledge about itself and the world;
- *Autonomy*: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- *Social ability*: agents can interact with other agents (and possibly humans) via certain communication methods;
- *Reactivity and knowledge adaptation*: agents can perceive their environment (which can be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of them combined) and respond in a timely fashion to changes that occur in it;
- *Pro-activity*: agents do not simply respond to their environment: they can exhibit goal-directed behavior by initiating actions;
- *Trust*: agents usually do what they are supposed to do according to their perception of the task and the world situation;

An Internet agent refers to an agent that acts on the browser's behalf [5] [4] [19]. The most popular agents on the Web are indexing agents such as Lycos, the WebCrawler, and InfoSeek. Indexing agents carry out a massive, autonomous search of the WWW, and store an index of words in document title and document texts. The user can then query the agent by asking for documents containing certain key words. The indexing agents operate by suggesting locations on the Web to the user based on a relatively weak model of what the user wants, and what information is available at the suggested location. The Internet Softbot, however, represents a more ambitious attempt to both determine what the user wants and understand the contents of information services in order to provide better responses to information requests.

The Live WWW agents are new kind of agents that aims at reducing the bandwidth used by Web pages that contain live video. They act as intelligent assistants between the camera and the server or between other agents to perform different kinds of image processing, vision, or robotics tasks required by the remote browser. The general system architecture for Live WWW with agents (LWWW-A) is shown in Figure 1. The LWWW-A is basically a traditional WWW architecture plus a set of agents that connect the camera and the server.

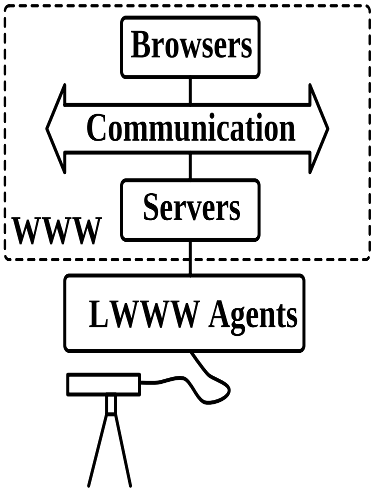

Figure 1: The LWWW system architecture. It is basically a traditional WWW architecture plus a set of agents that connecting the camera and the server to perform tasks required by the browser.

4 Performance Modeling

4.1 LWWW With Passive Camera

LWWW with passive camera means that the zoom, pan, and tilt of the camera at the server's site is fixed. So, when the vision agent is not used, the camera grabs images at a certain rate, and the grabbed images are transmitted through the Internet. In many situations, vision agents can be used to reduce the bandwidth used. For example, a security person is responsible for the security of a warehouse. In the warehouse, there are several cameras fixed in various places such that any part of the warehouse can be examined by at least one of the cameras. The images of the cameras are transmitted through Internet to a LWWW site that is used by the security person to check the situations of the warehouse. Since the number of images displayed at the Web site is large (which equals the number of cameras within the warehouse), the bandwidth used by this LWWW site is also large. The security person has to check every image on the computer screen, thus he might easily get tired. Vision agents can be used in order to reduce the bandwidth used and to make the task of the security person easier. Instead of immediately sending the image grabbed by a camera to the browser, a change detection agent is employed to analyze the image first, only those images that have enough changes are sent through the Internet and displayed at the client site. The number of images transmitted is decreased, thus the bandwidth used is also decreased. The security person's task is also easier, because he has less images to check. Another example is to reduce the size of the image at the server's site by using vision agents that can act according to the browser's requirement and perception ability. For example, a browser visiting a LWWW page is first asked several questions about his understanding of image processing technology. If the browser is an image processing expert and he has a deep understanding of the line detection technology. Then the images grabbed by the camera can be first processed by a line detector and only the resulted images that contain only lines or curves are transmitted through the Internet. Because the compressed line images take very few space, the bandwidth used is greatly reduced. In this case, we can even increase the frame rates such that the browser can see a "true" live video without feeling the interval between the frames.

4.1.1 The Waiting Time for a Single Image

For Live WWW, we define the waiting time for a single image frame as the time between the submission of the viewer's request and the display update resulting from the request.

When agents are not used, the waiting time T_{wait} for a single image frame is given by

$$\begin{aligned} T_{wait} &= T_{setup} + T_{grab} + T_{prop} \\ &+ T_{trans} + T_{comDecompre} \\ &+ T_{display} \end{aligned} \quad (1)$$

where T_{setup} is the network connection setup time (the overhead in setting up the connection between the source and the destination computer); T_{grab} is the image grabbing time for the camera; T_{prop} is the *propagation delay* of the connection (the difference between the time the last bit is transmitted at the head node of the link and the time the last bit is received at the tail node); T_{trans} is the network transmission time for the compressed image (the difference between the time the first bit of the first packet of the communication packets that form the image object and the last bit of the last packet of the communication packets that form the image object are transmitted); $T_{comDecompre}$ is the compression and decompression time (the time needed to compress the image before it is transmitted at the server site and the time needed to decompress the image before it is displayed at the browser's site); $T_{display}$ is the time needed to display the decompressed image at the browser's site.

When agents are used, the waiting time is given by

$$\begin{aligned} T_{wait_A} &= T_{setup_A} + T_{grab_A} + T_{prop_A} \\ &+ T_{trans_A} + T_{agent} \\ &+ T_{comDecompre_A} + T_{display_A} \end{aligned} (2)$$

where T_{setup_A} is the network connection setup time; T_{grab_A} is the image grabbing time; T_{prop_A} is the *propagation delay* of the connection; T_{agent} is the agents operation time (the time for the agent to process the grabbed image); T_{trans_A} is the network transmission time for the image after being processed by agents and being compressed; $T_{comDecompre_A}$ is the compression and decompression time; $T_{display_A}$ is the time needed to display the decompressed image at the browser's site.

For the above two equations, we can assume that $T_{setup} = T_{setup_A}$, $T_{grab} = T_{grab_A}$, $T_{comDecompre} = T_{comDecompre_A}$, and $T_{prop} = T_{prop_A}$. From the aspect of waiting time for a single image, the network setup time and transmission time can occupy a significant portion of the total waiting time. The services provided by the Internet architecture are *best effort* services —– they perform their functions as well as possible, using as much of a shared resource as available, but do not guarantee their performance or resource availability. When the number of users, especially the LWWW users of the Internet increases, several simultaneous high-bandwidth sessions might easily saturate network links and routers. As a result, the bandwidth available decreases and the transmission increases. To maintain an acceptable waiting time and send a useful stream, it is very important to reduce the data to be transmitted over the Internet. The Internet agents are used for this purpose.

From Equation 1 and Equation 2, we get

$$\begin{aligned} T_{wait} - T_{wait_A} &= T_{trans} + T_{display} \\ &\quad - T_{agent} - T_{trans_A} \\ &\quad - T_{display_A} \qquad (3) \end{aligned}$$

Given a fixed network bandwidth, the transmission times T_{trans} grow proportionally with the size of the file being transmitted. It can be obtained by the size of the image file to be transmitted S_{image} [1] and the average network bandwidth B,

$$T_{trans} = \frac{S_{image}}{B}$$

The value of $T_{display}$ can also be approximated by the size of the image to be displayed

$$T_{display} = cS^{dis}_{image}$$

where c is a constant.

Similarly, $T_{trans_A} = \frac{S_{image_A}}{B}$ and $T_{display_A} = cS^{dis}_{image_A}$. S_{image_A} is the size of the image file to be transmitted after the transformation by vision agents. $S^{dis}_{image_A}$ is the size of the image file to be displayed at the browser's side. Since S_{image_A} (e.g. the image only contain lines) and $S^{dis}_{image_A}$ can be much smaller than S_{image} and S^{dis}_{image} respectively, the waiting time by using agents T_{wait_A} can be much smaller than the waiting time without using agents T_{wait}, although the agents have to spend some time to process the image.

4.1.2 Transmitted Data for a Video Sequence

We study how the vision agents can reduce the Internet traffic in this section.

Suppose the video sequence contains N image frames. When agents are not used, the amount of data that is transmitted over the Internet is:

$$M_{data} = NS_{image} \qquad (4)$$

When agents are used, suppose they filtered out N^* images that are not interesting, then the amount of data that is transmitted is:

$$M_{data_A} = (N - N^*)S_{image_A} \qquad (5)$$

Thus, the amount of data that is reduced by using agents is

$$\begin{aligned} M_{data} - M_{data_A} &= NS_{image} \\ &\quad -(N - N^*)S_{image_A} (6) \end{aligned}$$

Since S_{image_A} can be much smaller than S_{image} (e.g., in the line image case) and N^* can be pretty large (e.g., in the security person case), the bandwidth used when the agents are used can be much smaller than the bandwidth used when the agents are not used.

[1] which is proportional to the dimensions of the $2D$ image, its resolution (including color quantization), and its quality.

4.2 LWWW with Active Camera

The spontaneous growth of the WWW over the past several years has resulted in a plethora of remote controlled mechanical devices which can be accessed via the WWW. LWWW with active camera means that the zoom, tilt, pan, and even the position of the camera at the server's site can be interactively controlled by the remote browser. The use of an on-line controlled camera can make the browser experience the state at a particular moment in time as if he is in the actual remote space where the camera is situated.

Right now, there are more than a hundred interesting mechanical devices are connected to the WWW [18]. For example, Paul Cooper et. al developed a interactive, telerobotic system InterCam [1]. Ken Goldberg et al. [8] developed a three axis telerobotic system where users were able to explore a remote world with buried objects and alter it by blowing bursts of compressed air into its sand filled world. Eric Paulos and John Canny [18] developed a WWW browser Mechanical Gaze, which allows multiple remote WWW users to actively control up to six degrees of freedom of a robot arm with an attached camera to explore a real remote environment. The initial environment is a collection of physical museum exhibits which WWW users can view at various positions, orientations, and levels of resolution. Cooperstock et. al [3] proposed the idea of World-Wide Media Space (WMS) for video-conferencing, which supports navigation with an active floor plan and exploits sensors to provide additional information for remote activity.

LWWW with active camera requires a high bandwidth to operate because the browser need to check the images constantly while interactively controlling the camera. Vision agents can be used to greatly reduce the bandwidth used.

4.2.1 Waiting Time

For LWWW with active camera, we define the waiting time as the time between the submission of the viewer's request and the display of a image at the client's site that satisfies the viewer's request.

When agents are not used, the browser controls the camera remotely and check the images transmitted over the Internet until a satisfactory image arrives. Suppose the satisfactory image is the Nth image that is transmitted over the Internet, then the waiting time T_{wait} is given by

$$\begin{aligned} T_{wait} &= T_{setup} \\ &\quad +N\Big(T_{grab} + T_{prop} \\ &\quad T_{trans} + T_{comDecompre} \\ &\quad +T_{display}\Big) \end{aligned} \tag{7}$$

When agents are used, instead of directly transmitting each image that is grabbed by the camera, the agents analyze the image and decide whether it is the one the browser required. If the agents believe that the image is not a wanted one, they will control the camera intelligently and grab the next image to analyze. If the agents believe that the image is a wanted one, they will send the image through the Internet to the browser. When the browser checks the transmitted image and is satisfied, the task is finished. If the browser is not satisfied with the result, he/she will inform the agents to try again. Suppose the camera grabbed totally N^* images when the browser is satisfied during the above process, and suppose among them there are $N^{'}$ images are transmitted through the Internet, then the waiting time is:

$$\begin{aligned} T_{wait_A} &= T_{setup} + N^*T_{grab} + N^*T_{agent} \\ &\quad +N^{'}T_{prop} + N^{'}T_{trans_A} \\ &\quad +N^{'}T_{comDecompre} \\ &\quad +N^{'}T_{display} \end{aligned} \tag{8}$$

Generally speaking, N is smaller than N^* (because agents are not as smart as the browser), but much bigger than $N^{'}$ (because many unwanted images are filtered out by agents). So, when the value of T_{agent} is not significantly large, the waiting time by using agents can be much smaller than the waiting time without using agents.

4.2.2 Transmitted Data Reduced

We study in this section the Internet traffic reduction by using agents that intelligently control the camera and analyze the images.

It is clear that

$$M_{data} = NS_{image} \tag{9}$$

$$M_{data_A} = N^{'} S_{image_A} \tag{10}$$

The amount of transmitted data that is reduced by using agents is given by

$$M_{data} - M_{data_A} = NS_{image} - N^{'} S_{image_A} \tag{11}$$

The above data reduction can be very significant when the agents are "smart" enough, that is, when $N^{'}$ is kept very small compared to N.

5 An Example: the LWWW Object Search Agent

In the following section, we formulate the LWWW Object Search Agent (LWWW-OSA), outline its working mechanism, and present experimental results to illustrate how the agent performs its task.

The LWWW-OSA is a useful feature of LWWW equipped with an active camera. For example, a researcher of CAS wants to play baseball after work. The researcher is supposed to bring the baseball to the game, but cannot find it in the office. The researcher wants to know whether the baseball is in someone else's office. The researcher can open the home page of the the possible person and pass the request to the server site of LWWW. At this time, the WWW agent can be activated to automatically control the camera to search for the baseball.

5.1 The LWWW Object Search Agent Formulation

The LWWW Object Search Agent (LWWW-OSA) is a software package that assists the server in its searching for an object in the server site's space environment. It has the properties of autonomy, social ability, reactivity, proactivity, trust, and knowledge about itself and the environment.

5.1.1 Awareness

The LWWW-OSA has the following knowledge about the sensor and the environment.

The sensor unit is a camera with zoom, pan, and tilt capabilities. The state of the sensor unit is uniquely determined by 7 parameters $(x_c, y_c, z_c, p, t, w, h)$, here (x_c, y_c, z_c) is the position of the unit center which cannot be changed, (p, t) is the direction of the camera viewing axis (p is the amount of pan $0 \leq p < 2\pi$, t is the amount of tilt $0 \leq t < \pi$), and w, h are the width and height of the solid viewing angle of the camera.

The LWWW-OSA has a set of recognition algorithms that can be used to analyze the image grabbed by the camera control agent to detect the target.

The LWWW-OSA knows the geometric configuration of the search region Ω. It tessellates the region Ω into a series of elements c_i, $\Omega = \bigcup_{i=1}^{n} c_i$ and $c_i \bigcap c_j = 0$ for $i \neq j$. c_o is the region outside Ω.

According to the information the browser supplies about the possible position of the target, the LWWW-OSA forms a target probability distribution function $\mathbf{p}$. The $\mathbf{p}(c_i)$ gives the probability that the center of the target is within cube c_i. The LWWW-OSA uses $\mathbf{p}(c_o)$ to represent the probability that the target is *not* in the environment.

5.1.2 Reactivity and Knowledge Adaptation

The LWWW-OSA perceives the physical world by executing actions and adjusting its knowledge.

An operation $\mathbf{f} = \mathbf{f}(p, t, w, h, a)$ is an action of the searcher within the region Ω, here a is the recognition algorithm used to detect the target. An operation $\mathbf{f}$ entails two tasks: (1) the LWWW-OSA instructs the LWWW-CCA (Live World Wide Web —– Camera Control Agent) to take a **perspective** projection image according to the camera configuration of $\mathbf{f}$ and to pass the image to LWWW-OSA; (2) the

LWWW-OSA analyzes the result image using the recognition algorithm a.

The cost $\mathbf{t_o}(\mathbf{f})$ gives the total time LWWW-CCA needed to manipulate the hardware to the status specified by $\mathbf{f}$, take a picture and pass the image data to LWWW-OSA, and run the recognition algorithm and update the environment if the target is not detected.

Each operation is associated with a detection function. The detection function on Ω is a function $\mathbf{b}$, such that $\mathbf{b}(c_i, \mathbf{f})$ gives the conditional probability of detecting the target, given that the center of the target is located within c_i and that the operation is $\mathbf{f}$. It is obvious that the probability of detecting the target by applying action $\mathbf{f}$ is given by $P(\mathbf{f}) = \sum_{i=1}^{n} \mathbf{p}(c_i)\mathbf{b}(c_i, \mathbf{f})$.

The LWWW-OSA not only senses the environment, it also updates its knowledge according to the sensing result. It uses Bayes' formula to update the probability distribution whenever an action fails (the target is not detected). Let α_i be the event that the center of the target is in cube c_i; let α_o be the event that the center of the target is outside the search region; and let β be the event that after applying a recognition action, the recognizer successfully detects the target. Then $P(\neg\beta \mid \alpha_i) = 1 - \mathbf{b}(c_i, \mathbf{f})$ and $P(\alpha_i \mid \neg\beta) = \mathbf{p}(c_i, t_{\mathbf{f}+})$, where $t_{\mathbf{f}+}$ is the time after $\mathbf{f}$ is applied. Because the events $\alpha_1, \ldots, \alpha_n, \alpha_o$ are mutually complementary and exclusive, the following updating rule holds

$$\mathbf{p}(c_i, t_{\mathbf{f}+}) \leftarrow \frac{\mathbf{p}(c_i, t_{\mathbf{f}})(1 - \mathbf{b}(c_i, \mathbf{f}))}{\mathbf{p}(c_o, t_{\mathbf{f}}) + \sum_{j=1}^{n} \mathbf{p}(c_j, t_{\mathbf{f}})(1 - \mathbf{b}(c_j, \mathbf{f}))}$$

where $i = 1, \ldots, n, o$.

5.1.3 Trust and Pro-activeness

According to the search task assigned by the browser, LWWW-OSA can explicitly represent the goal of the task and analyze its difficulties and propose a practical solution.

Let $\mathbf{O_\Omega}$ be the set of all the possible operations that can be applied. The effort allocation $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$ gives the ordered set of operations applied in the search, where $\mathbf{f}_i \in \mathbf{O_\Omega}$. The probability of detecting the target by this allocation is as follows:

$$\begin{aligned} P[\mathbf{F}] \quad = \quad & P(\mathbf{f}_1) + [1 - P(\mathbf{f}_1)]P(\mathbf{f}_2) \\ & + \ldots + \\ & \{\prod_{i=1}^{k-1}[1 - P(\mathbf{f}_i)]\}P(\mathbf{f}_k) \end{aligned}$$

The total cost for applying this allocation is

$$T[\mathbf{F}] = \sum_{i=1}^{k} \mathbf{t_o}(\mathbf{f}_i)$$

If K is the total time that is allowed for the search, then the goal of object search can be defined as finding an allocation $\mathbf{F} \subset \mathbf{O_\Omega}$, which satisfies $T(\mathbf{F}) \leq K$ and maximizes $P[\mathbf{F}]$.

Because the above task is NP-Complete (for details of our proof, refer to [28] and [24]) and the number of the available candidate actions is usually large, it is necessary to simplify the original problem. Instead of looking for an algorithm that always generates an optimal solution, LWWW-OSA simply uses heuristics that will generate a feasible solution for the original problem. Here, the greedy strategy is used, which suggests that one can devise an algorithm that works in stages, considering one input at a time. At each stage, based on some optimization measure, the next candidate is selected and is included into the partial solution developed so far. The objective function for the selection of the next action is $E(\mathbf{f}) = \frac{P(\mathbf{f})}{\mathbf{t_o}(\mathbf{f})}$. It is interesting to note that the above greedy strategy can generate the optimal answers in many situations and that it can generate a sequence of actions so that the expected time used to detect the target is minimized (for detail of our proof, refer to [28][24]).

5.1.4 Autonomy

The LWWW-OSA selects the camera's state parameters (p, t, w, h) without the intervention of the browser or the server. It determines the next action according to its perception of the world and passes its decision to LWWW-CCA.

For a given recognition algorithm, there are many possible camera viewing angle sizes to select from. However, the whole search region can be examined with high probability of detection using only a small number of them. For a specified angle, the probability of successfully recognizing the target is high only when the target is within a certain range of distance from the camera. This range is called the effective range for the given angle size. The LWWW-OSA only considers angles whose effective ranges cover the entire depth D of the search region and have no overlap of their effective ranges. If the largest viewing angle for the camera is $w_0 \times h_0$ and its effective range is $[N_0, F_0]$, then the necessary angle sizes $< w_i, h_i >$ and the corresponding effective ranges $[N_i, F_i]$ are as follows:

$$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})]$$

$$h_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{h_0}{2})$$

$$N_i = F_0(\frac{F_0}{N_0})^{i-1}$$

$$F_i = F_0(\frac{F_0}{N_0})^i$$

where $1 \leq i \leq \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor$.

For each angle size derived, an infinite number of camera viewing directions can be considered. We have designed an algorithm that can generate only the directions that can cover the whole viewing sphere without overlap. The LWWW-OSA only uses as candidate actions the viewing angle sizes and the corresponding directions obtained by the above method. An infinity of possible sensing actions is reduced to a finite set of actions that must be tried. The LWWW-OSA uses $E(\mathbf{f})$ to select the best viewing angle size and direction from the candidate actions. For each recognition algorithm, LWWW-OSA can find a best action. The next action to be executed is then selected from these best actions.

After the next action is selected, LWWW-OSA passes the result to LWWW-CCA. If the selected action does not find the target, LWWW-OSA updates its world knowledge according to Section 5.1.2 and selects another action to execute.

5.2 The Behaviors of LWWW-OSA

The behavior of LWWW-OSA can be summarized as follows:

1. The LWWW-OSA receives an object search request from the server. The request contains the target for which to search, the time allowed, and the possible locations of the target that are supplied by the remote browser.

2. The LWWW-OSA initializes the target probability distribution according to the information provided by the browser, and collects the available object recognition algorithms needed to detect the target.

3. For each available recognition algorithm, the LWWW-OSA selects a best action.

   (a) The LWWW-OSA selects possible camera angle sizes $< w, h >$ needed to examine the search region.

   (b) The LWWW-OSA selects the best direction $< p_k, t_k >$ for each camera angle size $< w_k, h_k >$.

   (c) The LWWW-OSA compares the best directions for each camera angle size to find the best action for the given recognition algorithm.

4. The LWWW-OSA compares the best actions for each available recognition algorithm to find the parameters $< w, h, p, t, a >$ for the next action.

5. The LWWW-OSA sends the best $< w, h, p, t, a >$ to LWWW-CCA.

6. The LWWW-CCA manipulates the hardware, grabs an image, and sends the image data to LWWW-OSA.

7. The LWWW-OSA analyzes the image by using recognition algorithm a. If the target is detected, it sends the FOUND signal to the server and performs EXIT.

8. If the allocated time is used up, LWWW-OSA sends the FAILURE signal to the server and performs EXIT.

9. The LWWW-OSA updates the probability distribution and goes back to Step 3.

5.3 Experiments

Experiments are performed to test the LWWW-OSA. The Laser Eye sensor shown in Figure 2(b) is used in the experiment. It is a sensing unit with pan and tilt capabilities, and consists of a camera with controlled focal length (zoom), a laser range-finder, and two mirrors. The mirrors ensure collinearity of effective optical axes of the camera lens and the range finder. The laser emits from the center of the camera (see [10] for details) to measure the distance from the center of the camera to objects in the environment in a specified direction. The direction of the camera's viewing axis and the size of the camera's viewing angle can be adjusted according to the visual task. The search task assigned by the browser is to search for a baseball in the environment as shown in Figure 2(a). The browser also specifies to LWWW-OSA that the baseball is probably on tables. After LWWW-OSA initializes the probability distribution of the search environment, the search begins. Figure 2(c) shows the first action selected by LWWW-OSA. The camera viewing angle size is $41^o \times 39^o$. Although the baseball is in the image, the given recognition algorithm does not detect the target because the target is outside the effective range of the view angle size $41^o \times 39^o$. Figure 2(e) shows the third action selected by LWWW-OSA. The camera viewing angle size is $20.7^o \times 19.7^o$. This action detects the target. Figures 2(f) and 2(g) show the image analyze result, where the baseball is detected (refer to [27][24] for a more detailed discussion of the experiments).

6 Conclusion

This paper proposes the concept of vision agents over Live World Wide Web. It outlines the general architecture of the LWWW agent technology and gives the performance mse

The Web now contains over 100 million documents, and is said to be doubling in size every 52 days. The incredible interest in the Net and the Web, and in the more general concept of information highways, is evident in the media today and is shared by peoples of various background. We believe that by offering the vision agents over Internet can make the Live WWW more attractive and open more applications of WWW technology. Vision agents over Internet will become one of the features of the future Internet technology.

The results presented in this paper can be taken as a starting point for further study of the LWWW technology. It opens new avenues of research, such as, how to extend the current communication protocol so that it is suitable for LWWW technology, how to integrate real time channels to make LWWW work effectively, and how to maintain the consistency, integrity and security of LWWW.

Acknowledgments

This work[2] was supported by IBM Centre for Advanced Studies and ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., Industry Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro Technologies, and Atomic Energy of Canada Limited. The authors are grateful to Karel Vredenburg and Don Hameluck for stimulating discussions and to CASCON96 reviewers for valuable comments.

About the Author

Yiming Ye is a Ph.D candidate at Department of Computer Science, University of Toronto and a Research Fellow student at IBM Canada Centre for Advanced Studies. He has papers in the areas of Internet agent and WWW, computer vision and image processing, robotics, computational geometry, machine translation, computational complexity, and knowledge based systems been published in journals and conference proceedings such as: *Proceedings of CASCON96*, *Proceedings of Second International Conference on Multiagent Systems (ICMAS96)*, *Proceedings of the International Symposium on Intelligent Robotic Systems (SIRS96)*, *Proceedings of 1995 IEEE International Symposium for Computer Vision (ISCV95)*, *Proceedings of the 4th International Symposium on Artificial Intelligence and Mathematics (AIM95)*, *Proceedings of the Second Asian Symposium on Computer Mathematics (ASCM96)*, *Proceedings of the IJCAI Workshop for handicapped Children*, *Proceedings of 1990 ACM Eighteenth Annual Computer Science Conference*, and *Science in China* etc. His current research interests are multimedia, image processing, computer vision, robotics, Internet agent, and multiagent system, etc. He can be reached at yiming@vis.toronto.edu.

John K. Tsotsos is a professor of Department of Computer Science, University of Toronto and a Fellow of Canadian Institute of Advanced Studies. He can be reached at tsotsos@vis.toronto.edu.

Karen Bennet is a Principal Investigator and Operational Manager for Center for Advanced Studies, IBM Canada. She can be reached at bennet@vnet.ibm.com.

References

[1] David Abrams. Toronto web society. In *http://www.cs.utoronto.ca/ abrams/tws.html*, Toronto, 1996.

[2] Z. Chen, S.E Tan, R.H Campbell, and Y. Li. Real-time video and audio in the world wide web. In *Proceedings WWW95*, 1995.

[3] Jeremy Cooperstock, Kelvin Ho, Kimiya Yamaashi, and Bill Buxton. Opening the

[2] The CAS-LWWW team now consists Karen Bennet and Abhay Shah from IBM side, and John K. Tsotsos and Yiming Ye from University of Toronto side.

doors of communication: The world-wide media space. In *Submitted to UIST'96: Ninth Annual ACM Symposium on User Interface Software and Technology*, 1996.

[4] L.L. Daigle and P. Deutsch. Agents for internet information clients. In *CIKM'95 Intelligent Information Agents Workshop*, Baltimore, MD, December 1995.

[5] Oren Etzioni and Daniel S. Weld. Intelligent agents on the internet: Fact, fiction, and forecast. *IEEE Expert*, August,1995.

[6] Armando Fox and Eric A. Brewer. Reducing www latency and bandwidth requirements by real-time distillation. In *Fifth International World Wide Web Conference*, Paris, France, May 1996.

[7] Didier La Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):47–58, 1991.

[8] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutte, and Jeff Wiegley. Robot teleoperation via www. In *IEEE International Conference on Robotics and Automation*, Toronto, May 1995.

[9] Telepresence Systems Inc. Prorata: Group awareness and collaboration software. In *A Deme at: The Internet Beyond the Year 2000*, Toronto, 1996.

[10] P. Jasiobedzki, M. Jenkin, E. Milios, B. Down, and J. Tsotsos. Laser eye - a new 3d sensor for active vision. In *Intelligent Robotics and Computer Vision: Sensor Fusion VI. Proceedings of SPIE. vol. 2059*, pages 316–321, Boston, Sept. 1993.

[11] Tom Lane. Jpeg image compression: Frequently asked questions. In *Independent JPEG Group: http://www.smartpages.com/faqs/jpeg-faq/faq.html*, November 1994.

[12] Case T. Larsen. Introduction to videoconferencing and the mbone. In *http://www.lbl.gov/ctl/vconf-faq.html*, 1996.

[13] Y. Lashkari, M. Metral, and P. Maes. Collaborative interface agents. In *Proceedings of the National Conference on Artificial Intelligence*, USA, 1994.

[14] Chris Lilly. Not just decoration: Quality graphics for the web. In *Fourth International World Wide Web Conference*, Boston, 1995.

[15] P. Maes, T. Darrell, B. Blumberg, and S. Pentland. Interacting with animated autonomous agents. In *Working Notes AAAI Spring Symposium on 'Believeable Agents'*, USA, 1994.

[16] Jacques A Mattheij. Web camera info. In *http://www.mattheij.nl/webcam*, 1996.

[17] England P., Allen R., and Underwood R. Rave: Real-time services for the web. In *Fifth International World Wide Web Conference*, Paris, France, May 1996.

[18] Eric Paulos and John Canny. A world wide web telerobotic remote environment browser. In *Fourth International World Wide Web Conference*, Massachusetts, USA, 1995.

[19] D. Riecken. Intelligent agents: Introduction to special issue. *Communications of the ACM*, 37(7):107–116, July 1994.

[20] J.K. Tsotsos. Analyzing vision at the complexity level. *The behavioral and brain science*, 13:423–469, 1990.

[21] Jacco van Ossenbruggen and Anton Elikns. Bringing music to the web. In *Fourth International World Wide Web Conference*, Boston, 1995.

[22] Xiaohuan Corina Wang. An adaptive rendering and display model for networked applications, 1996.

[23] Mike Wooldridge and Nick Jennings. Intelligent agents: Theory and practice. USA, 1994.

[24] Yiming Ye. *Sensor planning in 3D object search.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4, 1996.

[25] Yiming Ye and John K. Tsotsos. Sensor planning for object search. Technical Report RBCV-TR-94-47, Computer Science Department, University of Toronto, 1994.

[26] Yiming Ye and John K. Tsotsos. The detection function in object search. In *Proceedings of the fourth international conference for young computer scientist*, pages 868–873, Beijing, 1995.

[27] Yiming Ye and John K. Tsotsos. Where to look next in 3d object search. In *1995 IEEE International Symposium for Computer Vision*, Florida, U.S.A, November 19-21 1995.

[28] Yiming Ye and John K. Tsotsos. Sensor planning in 3d object search: its formulation and complexity. In *The 4th International Symposium on Artificial Intelligence and Mathematics*, Florida, U.S.A, January 3-5 1996.

[29] Frank Yellin. Low level security in java. In *Fourth International World Wide Web Conference*, Boston, 1995.