LASER POINTERS AS INTERACTION DEVICES FOR COLLABORATIVE PERVASIVE COMPUTING

Andriy Pavlovych[1]
Wolfgang Stuerzlinger[1]

Abstract

We present a system that supports collaborative interactions for arbitrary environments. The system uses laser pointers as interaction devices and employs a video camera to detect where each laser pointer is pointing. Time-division multiplexing together with a wireless link affords discrimination between laser pointer dots belonging to different pointers. Safety issues with laser pointers are addressed with a new technique. We also discuss how new users can be dynamically added and removed from the system.

1. Introduction

The idea of using pointing devices for interaction in a pervasive/ubiquitous environment has been presented before (see e.g. [5] for recent work). However, many systems use pointing devices that by themselves do not produce a visible cursor. Another limitation of most recent presented systems is that they support only a single interaction device, which does not allow for collaboration. Our present contribution is to allow multiple people to interact, while still being able to keep track of who did what. That makes the recording of who did what for a purpose of subsequent analysis possible, enables undo of an operation of a particular user, and is essential for collaborative activities. We also choose to use laser pointers, as the visibility of the generated laser spot has been demonstrated to result in better pointing performance [1]. The system presented in this paper can be used for interaction with posters or photos on the wall, with standard appliances (light switch, blinds, or a VCR) or non-standard appliances (e.g. a statue that can be used to control the room temperature), as in the concept of “Roomware”. Additionally, interactive whiteboards can be integrated seamlessly and used via the interaction devices, too. Another example usage scenario is to control DVD playback *directly* on a large screen, rather than via using a standard infrared remote control unit (DVD menus), where any of the viewers can interact with the system.

2. Intelligent Laser Pointers

The system described in this Section is an enhanced version of the work done by Oh and Stuerzlinger [3]. The system presented there introduced computer controlled laser pointers to afford

[1] Dept. of Computer Science, York University, Toronto, ON M3J 1P3, Canada;
www.cs.yorku.ca/{~andriyp|~wolfgang}

interaction with a large projection screen. The laser dots are identified via a camera, and different dots corresponding to different users are distinguished via a time-multiplexing scheme. Synchronization between the lasers and the camera is provided via cables linking the computer and the laser pointers, which also transmit button press information back to the computer.

The new system presented in this paper differs from the above-mentioned system in that it uses a wireless link to communicate with the laser pointers, it uses a novel blinking scheme that provides better visibility of laser dots, a novel safety feature, the ability to add and remove lasers on the fly, as well as support for a single unmodified laser pointer, and support many planar surfaces for interaction as well as the extension to multiple cameras. We will discuss these features in turn. To simplify the discussion we assume for the next few sections, that only a single, rectangular, and planar surface is used together with a single camera. This restriction is lifted in section 2.4.

First, the hardware for our system utilizes a Pocket PC handheld computer (here an iPAQ 3850) to control a laser pointer in a wireless manner. The handheld is small enough to be clipped to a belt or to be worn in a pocket. We use the serial port of the handheld to control the laser diode. More precisely, whenever we receive a particular signal via the wireless link, we utilize the RTS signal to turn the laser pointer on. This is done via a simple transistor switch hidden in a plastic case of a rechargeable battery of an old cellular phone, which provides the necessary power. The state of the button of the laser pointer is read via the CTS signal and this information is transmitted back to the controlling computer via the wireless link. Three more input signals (CD, DSR, RI) are currently unused, but could support additional interaction buttons or something similar to a scroll wheel.

Figure 1 Pocket PC, battery pack and a pointer

We utilize a monochrome Firewire camera capable of delivering VGA images at 120Hz to identify laser dots. The high frame rate ensures that even fast moving laser dots are captured well. The *shutter/exposure time* of the camera is adjusted so that it is only laser dots that are exposed. In our system the shutter time is usually 1/250th of a second. Note that we leave the *aperture* of the lens constant at a setting corresponding to the lowest amount of optical distortions, usually it is around f/4.0-f/5.6 for small lenses). Then, for each image, we do a simple threshold operation to identify the brightest pixels in the image (corresponding to the tightly focused bright spot caused by a laser pointer) and compute the centroid of the cluster. This identifies a laser pointer dots with minimal computation.

Typically, a laser spot causes the camera sensor to saturate in one very small area (the area in which we are interested) while the rest of the image remains severely underexposed (the area we wanted to ignore). The performance does not seem to be affected by the nature of the surface used (e.g. wood, fabric, plastic), as there are few materials that absorb or transmit almost all of the light arriving at them (e.g. charcoal and glass respectively). Very shiny objects cause problems however, as they reflect light in arbitrary different directions. Similarly, the presence of bright light sources or very high levels of ambient illumination in the field of view of the camera causes problems, too. Currently, we deal with such cases by excluding the corresponding image area(s) from analysis. The performance can be further improved, if necessary, via using narrowband optical filters tuned to the laser pointer bandwidth. The downside of this scheme is that it also darkens the image. This in turn puts more stringent requirements on light sensitivity of the used cameras.

2.1. Distinguishing Different Users

Other systems that use laser pointer-based interaction usually rely on a single-user interaction. Our system, on a contrary, from the very beginning supports simultaneous active users. To achieve that, we use a technique similar to time-division multiplexing. That is, at any given time, we only turn one pointer on and observe its location. Then, we turn the next visible pointer on and observe its position. Unlike in [3], we activate only *one* laser per captured image, to simplify the processing and avoid any potential problems when multiple pointers converge in one location. As discussed in [3], this leads to reduced brightness of laser pointers, as they are on for only $1/n^{th}$ of the time.

The solution to the brightness problem is to realize that the exposure time (e.g. $1/250^{th}$s) is shorter than the duration of a frame ($1/120^{th}$s). We utilize the fact by turning *all* laser pointers on while the camera is not acquiring an image (i.e. during the remaining approx. $1/250^{th}$s of the frame). This increases the average brightness of all laser pointers to more than 50%, while still allowing for robust detection of different pointers. If we use 60Hz acquisition, the average brightness increases to more than 75%. Clearly, this idea can also be combined with other temporal encoding schemes (see e.g. [3]).

2.2. Eyes-safety Feature

High power laser sources such as the ones used in industry, physics research or medicine can instantly cause permanent eye damage. The laser diodes used in our system are Class IIIa, which is considered eyes-safe and is not known to be able cause any permanent damage.

Still, many people have lingering safety concerns and prefer to avoid even temporary blindness, similar to the one caused by incoming traffic during night driving on a highway. Hence, we decided that eliminating this problem is a desirable goal. Our approach to the problem is the following. As soon as the camera loses track of a laser spot, it assumes that the device is pointing in a potentially harmful direction and switches the corresponding laser into a low duty-cycle mode. In this mode, the laser pointer is switched on for only one shutter time during each second (i.e. for $1/250^{th}$s). This reduces the power level to practically negligible levels and as far as we can determine it from the laser literature will not produce temporary blindness. The benefit over turning the laser pointer off completely is that there is still a visual indication that the pointing device is active. Whenever the camera detects a spot from a laser pointer in low duty-cycle mode (i.e. the laser pointer points at the screen), it returns the laser pointer to normal operation.

2.3. Dynamic Allocation of Pointers in the System and Support for Unmodified Laser Pointer

Normally, only a limited number of people collaborate in a normal-sized room. However, we expect that, occasionally, additional persons would want to participate in the interaction. Supporting the maximum possible number of the users at all times would dramatically slow down our system due to the use of time-division multiplexing. There, each pointer's update rate is simply the camera's frame rate divided by the number of pointers. To put this into a perspective, with a 120 Hz frame rate, the position update rate for three pointers is 40 Hz, which is equal to a non-USB mouse's query rate under Windows 98. We observed that even with rates as low as 10 Hz (slow camera) the system was still usable. Yet, to reduce the aforementioned limitation on the number of users and to maximize each pointer's position update rate, we *dynamically* add and remove pointers from the interaction.

For adding a new pointer to the system it is sufficient for the user to point the laser pointer to the screen and to press the button on the laser pointer, which is transmitted via the wireless link to the system. The system then activates only the new laser pointer in the next frame and checks if a laser spot is visible. If it is, it adjusts the global blinking pattern to include the new device. If the laser spot is not visible, the new pointer is put into low duty-cycle mode. Removal of laser pointers works via a timeout on the low duty-cycle mode. Once a pointer has been in that mode for more than a small number of minutes (3 in our implementation), it is removed from the list of active pointers and turned off. The global blinking pattern is adjusted accordingly. If the user wishes to continue to use the laser pointer, he/she only needs to point the laser pointer at the screen and press the button.

Finally, the system can also support a single unmodified laser pointer. As described above, the system expects to see only a single laser pointer in each image (as all others are turned off). If two laser dots are detected, we assume that there is an unmodified laser pointer present (i.e. one that is not under our control). In this case, we track this pointer by reserving a special timeslot in the blinking pattern, where all computer-controlled laser pointers are turned off. In frames where two laser pointers are on, we use a Kalman Filter to predict the path and hence identify the unmodified laser pointer. However, the button press functionality is currently not supported with an unmodified pointer.

2.4. Extension to Multiple Active Surfaces and Multiple Cameras

So far, the discussion focused on a single active area with a single camera pointed at it. It is interesting to note, that if this active area is rectangular and roughly planar, calibration is very easy. We point one of the laser pointers to the 4 corners of the area, which yields a perspectively distorted rectangle in image space. A simple bi-linear interpolation between the corners in the image then yields the corresponding coordinates on the world surface. This allows the user to point at particular areas of a planar surface, or to allow for gesture recognition, or to use the laser pointer as a "mouse" for an interactive whiteboard. Note that if we are only interested in detecting if a laser pointer is pointed at an object (such as e.g. a lamp) we can simply designate an area in the image to correspond to a give object and do not need to perform bi-linear interpolation. By repeating the above procedure for each surface or object that is in view of a camera, we can support multiple active areas.

Naturally, it is hard to cover everything in a room with one camera due to the issues of non-optimal viewing angles, as well as due to occlusion by objects in the environment and by the users themselves (e.g. user stands with back to the camera and points at an object in front of him/her). In all such cases, using additional cameras helps. In particular, if cameras are positioned so that their field of view does not overlap, it is very simple to deal with this – any given laser spot can appear in only a single image. If the field of view of two cameras overlaps, we have to deal with the fact that a single laser spot can show up in multiple images. In this case, we use a weighted average of the results of multiple cameras, where the weight is being determined by the distance from the centre of the field of view and the angle at which the cameras are pointing at the surface (determined via the perspective distortion of the camera image).

2.5. Network Latency Issues

Our system is very demanding with respect to the precise timing of control signals. For example, at 120 Hz camera frame rate, we need to control the laser pointer on/off states with better than 10 ms precision.

It is well known that transmission delays over wireless digital links tend to be longer than for wired links. This is especially so if any of the media access protocols (MAC) are employed in the communication, as it is done in IEEE 802.11*x*, a version of which is used in our system. However, we observed that the wireless link delay in our case was roughly 1 ms and was very stable during work (usually, the delay is larger only if the device comes back from sleep – not an issue in our case). Of course, if there are many other devices actively working in the same electromagnetic band, the performance can go down severely (by *many* here we mean dozens). To increase the robustness of the system, one could dynamically control the frequency of laser pointer pattern rotation, so that as soon as the network delays (or any other delays) cause problems, the interactivity will be degrade gracefully for error-free operation.

2.6. Additional Uses of Pocket PC's Functionality

In our current implementation, we do not use the handheld computer for anything other than controlling a laser pointer. However, as the device has a decent colour display, as well as a moderate amount of processing power, it is possible to employ these capabilities in particular applications. For example, a system could display GUI elements on a handheld to enable, for example, control of surrounding appliances [2], [5], text input, selection from large lists, etc.

However, care should be taken designing interfaces that significantly extend onto secondary displays, as this may disrupt collaboration.

3. Discussion and Future Work

One of the major problems with using laser pointers is jitter due to hand tremor. One option is to apply filtering techniques to the captured video sequence, an approach tried by Wilson et al. [5]. However, we are not aware of solutions that eliminate the problem while still keeping the operation fluid and transparent. One may argue that using a Pocket PC as a remote controller and sensor is a bit of technical overkill. A better solution would be to use a dedicated microcontroller chip and a miniature transceiver (a technology currently employed in wireless keyboards and mice). We are currently implementing this, but do not expect any functional differences to the current system. Finally, we are also investigating blinking schemes that depend only on the number of laser pointers visible in the field of view of a *single* camera, as this will allow for even more laser pointers to be used globally.

In this paper, we do not introduce any applications that could use our system. Clearly, all applications that use wired pointers with collaboration [3], or regular pointers without explicit collaboration [4] should still work and some other could be developed in the future.

4. References

[1] CAVENS, D., VOGT, F., FELS, S., MEITNER, M., Laser Pointers as Collaborative Pointing Devices, Poster at CHI 2002, pp. 678-679.

[2] MYERS, Brad A. "Mobile Devices for Control," The Fourth Symposium on Human-Computer Interaction for Mobile Devices, Mobile HCI'02. (Keynote speech), September 18-20, 2002, Pisa, Italy. pp. 1-8.

[3] OH, J.-Y., STUERZLINGER, W., Laser Pointers as Collaborative Pointing Devices, Graphics Interface 2002, pp. 141-149.

[4] REKIMOTO, J., SAITOH, M., Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments, Proceedings of CHI'99, 1999.

[5] WILSON, A., PHAM, H., Pointing in Intelligent Environments with the WorldCursor, INTERACT International Conference on Human-Computer Interaction 2003.