

# Using Camera-Phones to Enhance Human-Computer Interaction

Anil Madhavapeddy,  
David Scott

Computer Laboratory  
University of Cambridge  
Cambridge, CB3 0FD, UK  
{avsm2,djs55}@cl.cam.ac.uk

Richard Sharp,  
Eben Upton

Intel Research Cambridge  
15 JJ Thomson Avenue  
Cambridge, CB3 0FD, UK  
first.last@intel.com

## ABSTRACT

The proliferation of camera-phone technology provides a global opportunity for novel ubiquitous computing applications that is, as yet, largely unexploited. Our work demonstrates some of the untapped potential of this existing infrastructure, focusing on the use of deployed camera-phones as devices to enhance human-computer interaction.

We show that, without requiring additional hardware, an existing camera phone can be used as both a sophisticated pointing-device, facilitating interaction with active displays; and a user interface for devices without displays or input capability of their own. Our implementation relies on visual tags, which can be detected and decoded by camera-phones, and short-range wireless communication between camera-phones and nearby computers (e.g. Bluetooth).

## HIGH-LEVEL SYSTEM OVERVIEW

Our system relies on two core technologies:

1. Visual tags (known as *SpotCodes*), that can be detected by camera-phones, are used to encode data. Note that the tags may be either active (e.g. generated dynamically by a PC and written to a display) or passive (e.g. printed on a poster).
2. Bluetooth is used for short-range wireless communication between a camera-phone and a nearby computer.

Each visual tag encodes two pieces of information: a *service-identifier* and a *data-block*. The service-identifier tells the phone which nearby Bluetooth service the tag relates to; the data-block encodes a few bits of application-specific information. We have implemented robust, high-performance tag-recognition software for a number of

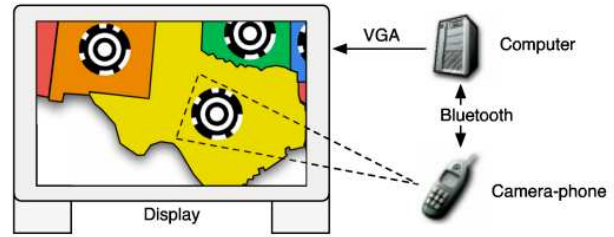


Figure 1. Data flow through our system

existing phones.

Figure 1 shows a diagrammatic view of our system in operation. A typical environment contains several users equipped with camera-phones, several computers exposing Bluetooth services and several visual tags: some passive, some active. A user aims their camera-phone at a visual tag. As soon as a tag appears in the frame, the phone decodes it and transparently initiates a Bluetooth session, connecting to the Bluetooth service specified by the tag's *service-identifier*. Whenever a Bluetooth session is active, the phone constantly transmits the data-blocks of visible tags along with the current (X,Y) positions, sizes and orientations at which the tags are detected (relative to the position of the camera). Key presses on the phone's keypad are also transmitted over Bluetooth.

This data provides a computer-based application with information about what a user is doing with their phone. More specifically an application knows: (i) which tags the phone is being pointed at (determined by the values of the data-blocks); (ii) where the tags are in the camera's field-of-view; (iii) how far the phone is from the tags (inferred from the size of the tags); (iv) the current velocity of the phone (inferred from the rate of change of the tags' positions and sizes); (v) at what orientation the phone is being held (inferred from the tags' orientation); (vi) the phone's angular velocity (inferred from the rate of change of the tags' orientation). Also, at Bluetooth session-initiation time, the phone announces its identity. This facilitates applications that react differently according to which phone is being used

to control them.

As well as using the phone as an input/pointing-device for nearby displays, we also enable nearby computers to throw UIs to the phone's own display. This is useful for a number of reasons. Firstly, consider computers which, due to constraints on form-factor or cost, have no displays or input devices of their own. By using Bluetooth, these devices can expose powerful UIs via nearby phones, using the phone's own display and keypad as the point of human-computer interaction. Secondly, it facilitates UIs that simultaneously use *both* the phone's own display *and* active displays in the environment.

### Motivation

As the price of display technology continues to fall, ubiquitous computing researchers are predicting an increased use of active displays in the environment. With this in mind, we believe that the use of a camera-phone as a *universal pointing and data-entry device* is an important contribution since it provides a convenient way for users to interact with such displays using their existing hardware.

Unlike touchscreens, which require users to stand at most arm's length from a display, a camera-phone can be used to remote control a display from a distance. This allows interactive displays to be situated in places where users cannot reach them, making optimal use of space (e.g. flat panel displays above supermarket aisles). It also enables users to control multiple displays in their field-of-view without having to constantly walk between them.

Since camera-phones announce their identity to Bluetooth services on connection, applications know which phone is being used to control them. Under the (realistic) assumption that each user has their own camera-phone, this allows us to develop *personal applications* which react according to *who* is controlling them (e.g. "click here to email the contents of this display to *your* address"). Furthermore, the camera-phone's persistent memory can be used to record *session state* (cf. HTTP cookies). This enables camera-phone-controlled applications to react differently according to how users have interacted with them previously.

By using a *combination* of visual tags and Bluetooth, we shield the user from tedious and unintuitive device naming and connection protocols. Instead of forcing the user to selecting a Bluetooth device to connect to manually we encode the name of a Bluetooth service in the tag itself. In an environment where tens or hundreds of wireless devices are present this kind of naming abstraction is of great benefit.

Tens of millions of Bluetooth-enabled camera-phones have already been shipped globally, with market trends showing that the rapid adoption of such devices is set to continue for the foreseeable future. Our work demon-



Figure 2. The World Map Application

strates some of the untapped potential of this existing infrastructure, showing how existing, deployed camera-phones can facilitate human-computer interaction.

### Demo Application: World Map

Designed for deployment in a travel agent's shop, the "World Map" application enables users to view and book flight information. The World Map application is designed to demonstrate how camera-phones can be used to control large displays in the environment.

A screenshot of a user interacting with the World Map application is shown in Figure 2. The plasma screen shows a map augmented with Spot Codes. By using their camera-phone to click on a Spot Code, a user can select an area of the map. The application zooms into the selected area, augmenting it with more Spot Codes. In this way, a user can drill-down and select a specific country in only a few clicks. Once a country is selected, a list of its airports are thrown to the camera-phone; the user scrolls through them and uses the phone's keypad to select one.

As soon as an airport is selected a side-bar appears on the plasma screen displaying relevant information (airport name, IATA code, distance from nearby cities etc.) and the user is prompted to select another country and airport (as before). When the second airport is selected, the application prompts a user for times and dates (by throwing suitable UI controls to their phone), queries a web-service and displays information about relevant flights on the plasma screen. At all times, Spot Code-buttons are displayed which enable a user to (i) start again, cancelling their current selections; and (ii) zoom out so the full map is displayed.

Once the required flight information has been obtained, the World Map application sends a WAP URL to the camera-phone over Bluetooth. The URL enables users to review flight information and possibly book on-line. It can either be bookmarked for future reference, or viewed there-and-then on the phone.