LOW-COST SPATIAL INDEXING USING MACHINE VISION

Tom Craig,[1] Joseph Newman,[1] and Andrew Rice[1]

Abstract
Location data is vital for many pervasive computing systems. Printed visual tags are commonly used to enable a machine vision system to determine the position and orientation of an observer. We have extended these tags to represent a general spatial zone which can be used to trigger an action upon ingress or egress. Our system is decentralised and computationally efficient, and it is trivial to create and modify zones. We use these tags in a "follow-me desktop" application.

1. Introduction

One useful concept that has arisen in pervasive computing is "programming with space" [3], which associates particular behaviour with certain regions of space, or *spatial zones*. For example, the Forget-me-not [4] and PlaceIt [7] applications associate reminders with spatial zones, which are triggered when the user enters the zone. Programming with office space has also proven useful: applications include follow-me desktops and video conferencing [2].

Meanwhile, cameras are becoming increasingly ubiquitous: fixed cameras are deployed on many sites (e.g. for security purposes), and mobile cameras are built in to most modern phones. A camera may serve as a location sensor by extracting features from images and calculating its position and orientation relative to these features using machine vision. To increase robustness, it is common to augment the environment with visually distinctive patterns known as *fiducial markers* [5], or *tags*; these are easily recognisable and furthermore may encode useful data.

In this paper we investigate the representation and use of spatial zones in a system that senses location using machine vision and fiducial markers. We describe how to define zones, keep them up to date, and efficiently determine when a significant entity has entered or left a zone. Using these *spatial tags*, we provide a follow-me desktop service.

2. System Infrastructure

The system consists of up to three kinds of entity:

- **Spatial tags** are fiducial markers, each of which: *(i)* represents a spatial zone, *(ii)* encodes context about the action associated with the zone, and *(iii)* allows a user to locate themselves relative to the tag.
- **Clients** are devices equipped with an outward-facing camera, either mobile (carried by a user) or fixed. They capture images, locate and decode spatial tags within those images, and perform

[1]University of Cambridge Computer Laboratory, Cambridge CB3 0FD, UK, e-mail: firstname.lastname@cl.cam.ac.uk

actions based on whether they are inside or outside the located tags' zones.

- **Service providers** are machines from which clients, as part of performing their actions, may request services. For some applications, the clients may be their own service providers.

A spatial tag is composed of two components: spatial and symbolic, illustrated in grey in Figure 1. The **spatial component** represents a zone using a 2-D shape surrounded by a square border. We derive the corresponding 3-D zone – an example of which is illustrated in cross-section in green in Figure 1 – by extruding the 2-D shape vertically relative to the coordinate system of the tag. Because of the simplicity of this mapping between 2-D representation and 3-D zone, this component of the tag is easily user-editable. Furthermore, as described below, it allows us simply to sample a point in the image in order to determine if the user is in the zone, rather than performing a more computationally expensive point-in-polygon test to achieve the spatial indexing. The **symbolic component** of the tag comprises a grid of bits encoding context about the action clients should perform when entering or leaving the zone.

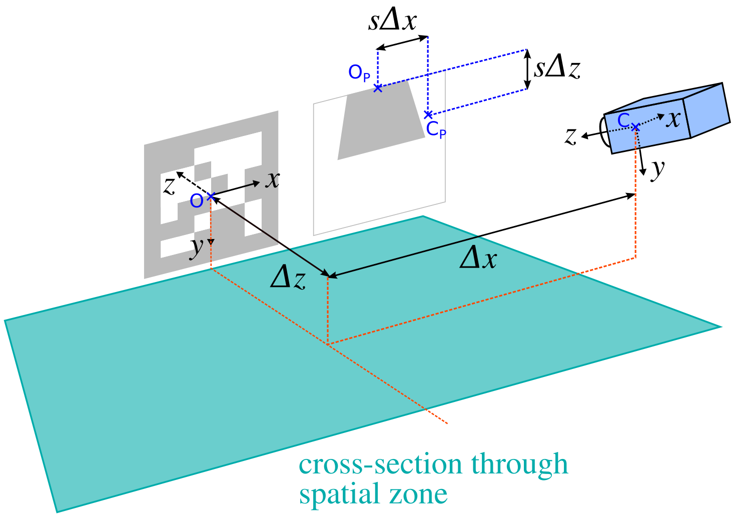

Figure 1. A spatial tag is mounted vertically in the environment. Its symbolic component (left) is a grid of bits, and its spatial component (right) is a 2-D shape which maps to a 3-D zone of the same shape protruding from the tag and extruded vertically.

Each client acts individually to detect when it enters or leaves a zone. We run a dæmon process on clients that performs the following actions in a continuous loop: (1) An image is captured by the camera C (see Figure 1) and thresholded to yield a monochrome image. (2) The symbolic component of each spatial tag in the image is detected and decoded using a fiducial marker tracking library (we use Cantag [6]). (3) The position and orientation of the camera is determined relative to the tag centred at O using standard machine vision techniques [5]. (4) A point in the image is then sampled to determine whether the user is in the zone. This sampling point C_P is determined by first projecting the camera's location into the plane of the tag relative to the spatial component at O_P, and then scaling by the pre-defined factor s in order to relate real-world units to units of tag width. If the sampling point is found to lie within the spatial component and be a black pixel, then the user is within the zone; otherwise, he is outside it.

This approach to spatial indexing is more computationally efficient than alternatives, such as point-in-polygon tests (which take logarithmic time in the number of vertices) and quadtree analysis [1] (which depends on the complexity of the shape of the zone and the level of granularity). Once we have localised the camera in step (3) above, then step (4) – determining the sampling point in the image – takes constant time, irrespective of the shape of the zone. Furthermore, each client runs its own pipeline, rather than requiring a centralised spatial indexer.

3. Deployment

We are using spatial tags to provide a "follow-me desktop" service, wherein a user's desktop is automatically teleported to the computer nearest him. A spatial tag is attached to every participating computer's monitor: the tag's spatial component represents a zone in front of the monitor, and the symbolic component encodes the lowest 16 bits of the IP address of the PC to which the monitor is attached (the highest 16 bits are not required because all machines are on the same Class B subnet). When notified over the network, a process on each PC (the service provider) will log in remotely to any other machine using the VNC desktop teleportation protocol (using the client's security credentials or prompting for authentication if required).

4. Evaluation

Ideally a spatial zone system should be widely deployable, scalable and unintrusive. It should support a variety of applications using low-cost, low-maintenance equipment. The system we have presented requires only spatial tags – patterns printed on ordinary paper – and software as its infrastructure. These tags are easily printed, mounted and modified, and they require no power source. With these in place, any sufficiently powerful commodity device may use the system; we currently use a tablet PC and webcam, but in fact many recent mobile phones satisfy these requirements [8]. In comparison, RFID or card-swipe systems require hardware that is not standard on all PCs, and they only give rough proximity information rather than a precise zone.

Using spatial tags, we suffer from the same limitations as any machine vision system. In particular, tags may only be recognised within a certain volume of space (the *viewable volume*), because the symbolic component may only be read if each of its bits occupies one pixel or greater in both width and height in the image. Viewable volume is further limited by adverse environmental conditions, i.e. noise in the imaging process and occlusions in the environment. The range within which sensibly sized tags may be read is sufficient for the follow-me desktop application; future work will evaluate where the viewable volume proves practically restrictive.

Incremental deployment of the spatial tag system is feasible because its architecture is decentralised. Clients may interact with zones independently of each other and of any other infrastructure. There is no limit to the number of clients, and the only limits to the number of zones are physical space and the number of bits of information encoded by tags. Being decentralised also benefits privacy: a user may deploy a private spatial zone application without interaction with a central entity.

Maintenance of the system involves:

Changing spatial zones. When the user wants to change the position of a spatial zone, he or she may simply edit the tag by hand using a marker pen and/or eraser fluid. Many other systems require specialist software or database knowledge to edit zones.

Maintaining consistency. When the object with which a spatial zone is associated changes position or identity, the zone must be updated. A spatial tag will typically be attached to the object to which it refers (such as a monitor for follow-me desktop), and so the coordinate systems of tag and object remain correctly aligned when the object moves.

Dealing with unexpected behaviour. The system has no central point of failure and the infrastructure required is passive. Furthermore, because the medium is visual, unexpected behaviour may in some cases be diagnosed by visual analysis of the images.

As future work, we intend to quantitatively evaluate the viewable volume of spatial tags and the relationship between error in localising the camera and error in choosing the sampling point in the image. Thus we can decide which applications can feasibly be implemented using spatial tags.

5. Conclusions

We have presented a decentralised, incrementally deployable system that allows a camera-equipped entity to recognise when it has entered or left a particular zone of 3-D space and to trigger an event based on some context associated with this zone. The zone and its context are represented using fiducial markers, which are easily user-editable; we call our particular design of marker a *spatial tag*. Determining containment within a zone is more computationally efficient than established alternatives such as centralised spatial indexing. We have demonstrated the system by implementing a follow-me desktop application, in which zones are associated with computer monitors.

6. Acknowledgements

The authors would like to thank David Cottingham and Jonathan Davies for their insightful comments and suggestions. Grateful thanks are due to Andy Hopper for his supervision and support.

References

[1] R. Harle. Spatial indexing for location-aware systems. In *Proceedings of MobiQuitous*, pages 1–8, August 2007.

[2] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2–3):187–197, March 2002.

[3] A. Hopper. The Clifford Paterson lecture, 1999. Sentient computing. *Phil. Trans. R. Soc. Lond.*, pages 2349–2358, August 2000.

[4] M. Lamming and M. Flynn. Forget-me-not: Intimate computing in support of human memory. In *Proceedings of FRIEND21 Symposium on Next Generation Human Interfaces*, February 1994.

[5] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Proceedings of Asia Pacific Computer Human Interaction*, pages 63–68, July 1998.

[6] A. Rice, A. Beresford, and R. Harle. Cantag: an open source software toolkit for designing and deploying marker-based vision systems. In *Proceedings of PerCom*, pages 12–21, March 2006.

[7] T. Sohn, K. Li, G. Lee, I. Smith, J. Scott, and W. Griswold. Place-its: A study of location-based reminders on mobile phones. In *Proceedings of Ubicomp*, pages 232–250, September 2005.

[8] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proceedings of ISMAR*, pages 121–124, September 2008.