

Movement-based interaction in camera spaces: a conceptual framework

Eva Eriksson · Thomas Riisgaard Hansen ·
Andreas Lykke-Olesen

Received: 27 November 2005 / Accepted: 1 June 2006
© Springer-Verlag London Limited 2006

Abstract In this paper we present three concepts that address movement-based interaction using camera tracking. Based on our work with several movement-based projects we present four selected applications, and use these applications to leverage our discussion, and to describe our three main concepts *space*, *relations*, and *feedback*. We see these as central for describing and analysing movement-based systems using camera tracking and we show how these three concepts can be used to analyse other camera tracking applications.

Keywords Movement-based interaction ·
Camera spaces · Mixed interaction spaces ·
Camera interaction · Mobile phone interaction

1 Introduction

Every interaction is in some sense movement-based: pressing a key, moving the mouse, or uttering a sound.

E. Eriksson (✉)
IDC—Interaction Design Collegium,
Department of Computer Science and Engineering,
Chalmers University of Technology, Gothenburg, Sweden
e-mail: eva.eriksson@chalmers.se

T. R. Hansen
Center for Pervasive Healthcare, Department of Computer
science, Aarhus University, Aarhus, Denmark
e-mail: thomasr@daimi.au.dk

A. Lykke-Olesen
Center for InteractiveSpaces, Institute for Design,
Aarhus School of Architecture, Aarhus, Denmark
e-mail: alo@interactivespaces.net

By emphasizing the word *movement* in movement-based interaction though, movement is no longer just the source of interaction; it becomes the central element in the interaction. Movement-based interaction seems to be especially suited for interaction that takes place in a public or social context, and it provides interesting alternatives to traditional interaction techniques within social settings, games, public places, for encouraging exercise, and in mobile settings. Using movement-based interfaces, however, can be strenuous and is thus less suited for continuous e.g. desktop work. Location-aware games [1], fitness games [2], and interfaces based on accelerometer input [3, 4] are examples of already available systems based on movement-based interaction.

Cameras are a common ubiquitous sensor in movement-based interfaces and the wide-spread use of camera phones and webcams makes camera-based computer vision a feasible platform for novel interfaces. Hence applications that use real-time camera input are already common within the research areas of tangible user interfaces (TUIs), virtual reality, sensor-based computing, ubiquitous computing, pervasive computing, and augmented reality. Within the computer vision community there is thus a great interest in analysing and extracting information from the video stream and use this information to provide new ways of interaction [5, 6]. However, little research has focused on how to actually use vision to design applications and on how to describe, compare, and characterise different approaches towards camera-based interfaces.

In this paper we focus on movement-based interaction that uses cameras to detect movement. Cameras have a limited field of view and the area within the camera's view can be seen as a bounded space. We call

this space a *camera space*. It is only within this space movements can be detected and registered. By mapping the movement within a camera space to a virtual space in an application a combined space is obtained. We refer to this type of space as a *mixed interaction space* [7] pointing to the space being both physical and virtual. The mixed interaction space is a subset of the mixed reality concept with a major focus on space.

Based on our work with these types of spaces we present a conceptual framework for movement-based interaction based on camera spaces. The framework is grounded in four projects briefly described and discussed. Due to the spatial nature of camera spaces we have drawn on an architectural understanding of space which will be unfolded later on in this paper. The framework is built around the three central concepts of *space*, *relations*, and *feedback*. The concept of *space* describes properties of the mixed interaction space. The mapping between the captured physical movements and the virtual domain is captured by the concept of *relations*. The concept of *feedback* finally describes how the digital events are visualized to the users. The framework is finally used to present and discuss a number of movement-based interfaces, and hereby we demonstrate how the presented framework provides explanatory power beyond the scope of our own projects.

1.1 Related work

Within several different research fields there are frameworks and taxonomies briefly touching upon the capabilities and aspects of camera sensor technologies. Together these frameworks form an important base, but being too general in their nature none of these go into depth with the specifics and potentials of camera-based interaction technologies and the use of them. As the main contribution of this paper is to present a conceptual framework for movement-based interfaces using camera tracking, we here present a short overview over some of the extensive related work, and use this as a springboard to a more in depth analysis.

In [8] Mackay presents the concept of augmented reality as opposite to the, then, increasing focus on virtual reality. Three basic strategies to augmented reality are presented, where video cameras as tracking sensors are used as examples to augmenting the environment surrounding the user and the object, but not discussed further. In [9] Benford et al. analyse sensor-based interfaces in general, including a discussion of camera-tracking. They point out several problems with camera-tracking, such as the number of cameras needed, the frame rate, the field of view limits the

extent of traceable surfaces, and that camera-tracking systems are usually unable to cope with different objects, multiple objects, occlusion, and changes in lightning. The two papers do not further discuss the possibilities with this technology. We take the opposite approach and explore how camera-tracking systems' strengths and weaknesses can be used in the process of developing movement-based interfaces.

In [10], Abowd et al. state that research in ubiquitous computing implicitly requires addressing some notion of scale, whether in the number and type of devices, the physical space of distributed computing, or the number of people using a system. They posit a new area of applications research, everyday computing, focussed on scaling interaction with respect to time. Scale is further discussed by Ullmer and Ishii in the conceptual framework [11], which focuses on the characteristics of TUI. Tangible interfaces are here divided into groups labelled spatial, constructive, relational, and associative. Camera-tracking systems can be found in all of the presented groups. The paper states that several concepts need to be explored further, e.g. physical scale and distance. Our aim is to continue some of these discussions by focusing on e.g. the aspects of space and scale in camera-tracking systems.

In [12] Holmquist et al. strive to create a common vocabulary to systems where a physical object is used to access digital information stored outside the object. In Fishkin's taxonomy for tangible user interfaces [13], categorizations and definitions from previous frameworks are unified, such as the vocabulary of [12] and the classification system from [14]. Fishkin further suggests that tangible user interfaces are leaving the traditional computer-human interfaces into the realm of human interfaces in general, and draws more towards the communities of industrial design, kinesthetics, architecture, and anthropology. We agree in this change in departure for TUIs in general, and especially for systems based on camera-tracking. In our work we have a base in the conventional computer virtual world, but we use inspiration and relations from the physical world, especially from the fields of architecture and kinesthetics.

Surely, a lot have been left out, but the frameworks presented here are examples that cover a wide spectrum and involve different general perspectives on camera-tracking systems. These frameworks create a framing to the context of camera-based systems, and they provide tools of how to analyse, define, and redesign different types of systems in this wide context. Still, we stress the need for a more specific tool developed for camera-tracking, since these related

frameworks present a too general picture and do not pay enough respect to the specific characteristics of camera spaces.

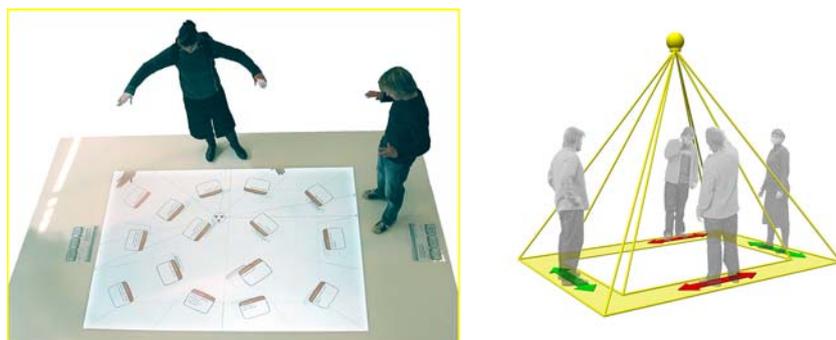
2 Movement-based applications in camera spaces

To frame and inspire the discussion of the movement-based framework we start out with a brief presentation of four selected movement-based projects. The first two applications are developed around an interactive floor with a ceiling mounted camera-tracking the people within the camera space. The last two applications use the mobile phone's camera to track different features, e.g. circles, coloured objects, or a person's face. The movement passed to the application is either the movement of the camera (the mobile device) or the movement of the tracked objects.

2.1 Application one: iFloor

iFloor is an interactive floor facilitating the exchange of information between users of a public library, as well as bringing some of the services that the library offers on the internet into the physical library. A video tracking system tracks the movements and size of the people present along the edges of the display. A single person or a group of people will attract a circular cursor that expands and highlights the different questions and answers displayed on the floor. As soon as a person is recognized by the camera within the legitimate space a string is drawn from the shared cursor to the person indicating a successful established relation and ongoing interaction. The cursor distributes a string to each person around the floor and calculates the resulting vector which determines the overall movement direction. The cursor is shared between all participants, why a collaborative effort and physical movement is necessary in order to navigate the cursor on the floor [15]. The iFloor prototype and the tracked movements are illustrated in Fig. 1.

Fig. 1 The iFloor prototype and a diagram of the tracked movements



2.2 Application two: StorySurfer

StorySurfer is an interactive floor application displaying book covers which provide an alternative way for children to browse the library's collection of books. The book covers are evoked by stepping on buttons on the edge of the floor. Each button is associated with a keyword. Hitting a keyword button will evoke a cloud-like shape on the floor containing book covers associated to the selected keyword; overlapping clouds contain book covers associated with several keywords. A cover can be further examined by moving into the floor. Each person entering the floor and the camera space is provided with a cursor in the shape of a "magnifying lens" oriented and positioned in front of the user turning towards the centre of the floor. Thus the "lens" is controlled by the children's body movements. Keeping the lens icon still over a projected book cover causes it to enlarge for better inspection and maintaining the position even a bit longer will cause the image to move across the floor to an interactive table [16]. Figure 2 shows the StorySurfer prototype and the tracked movements.

2.3 Application three: ImageZoomViewer

The ImageZoomViewer that is built on the Mixis tracking technique [17] is an application for mobile devices. It uses movement-based interaction to navigate in a map or a large image. The mobile device tracks either a hand-drawn circle, any coloured object, or the user's own face if the device is equipped with a second camera pointing towards the user. If the mobile device is close to the feature the application zooms in on the map, if the device is far away from the feature the application zooms out, if the device is to the left, right, up, or down in relation to the tracked feature the application pans accordingly. Figure 3 shows the ImageZoomViewer application running on a mobile phone and a diagram of the tracked movements. [17]

Fig. 2 The StorySurfer prototype and a diagram of the tracked movements

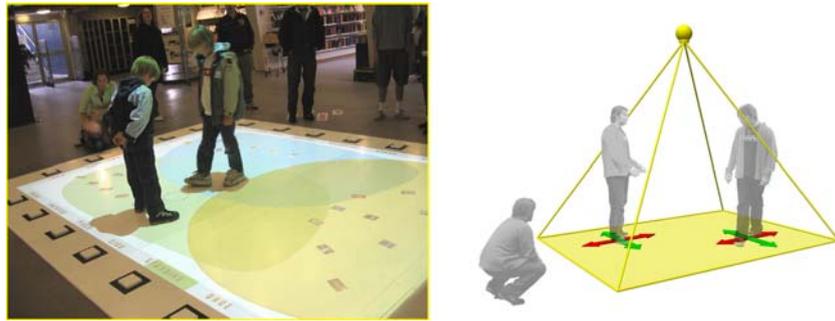


Fig. 3 The ImageZoomViewer prototype navigating in a map with gestures and a diagram of the tracked movements

2.4 Application four: Photo-Swapper

The Photo-Swapper is built on the Mixis tracking technique [17]. Photo-Swapper is an application that allows the mobile phone to operate a cursor on a shared display. Several users can connect to the shared display with their own personal device resulting in several simultaneous cursors. The cursor can be moved on the shared display by moving the mobile device in relation to the tracked feature: moving the device closer to the feature results in a pick-up action, while moving the device away from the feature is mapped to a drop action. It is possible for up to seven users to connect to the same shared display, thus operating seven independent camera spaces simultaneously and using them as input in the same application. Figure 4 shows the Photo-Swapper application and a diagram of three camera spaces connected to the shared display. [18]

2.5 Application summary

Despite the projects different foci, setups, and use situations we present some recurrent themes binding these projects together. The main findings from the individual projects are presented elsewhere [15, 16, 17, 18].

First, *movement and space* play a central role in the four presented applications, but are used in different ways. In the first two applications the tracked features are human bodies moving around in a large static camera space, whereas in the last two applications it is the entire camera space that moves in relation to a set of tracked features and not the tracked features that move within the camera space.

Second, a special relationship exists between the camera and the features being tracked and used for interacting with the system. In application one and two several features (human shapes) are tracked and

Fig. 4 The Photo-Swapper prototype and a diagram of the camera spaces and feedback areas



all user movements affect the system. In applications three and four a single feature (a symbol, a coloured object, or the user’s face) is tracked, where changes in the location of the feature as well as changes in the camera position will affect the interaction. We call the relationship between the camera and a tracked feature a *relation*, because it is the changes in this relationship that trigger the interaction.

Third, these interfaces are not traditional desktop interfaces, why there is a clear need as well as many possibilities for providing user *feedback*. We distinguish between feedback mainly focused on the input system (*input feedback*), and feedback from the application about its state (*application feedback*).

In application one and two the input feedback is provided visually on the floor on top of the application feedback. In application three input feedback is visually overlaid the application feedback but in a very limited screen area. Finally, in application four all feedback is moved from the mobile device onto the shared display, combining input feedback and application feedback on the same display for multiple users. The feedback used in these applications is purely visual but other types of feedback will also be discussed.

Based on our work with the above described applications we hence find *space*, *relation*, and *feedback* to be central concepts useful for describing, explaining, and comparing movement-based interfaces based on camera spaces. *Relations* describe how users manipulate the system and provide input. *Feedback* describes

how the computer system informs the user about its state and provides output, and *space* provides a context for the interaction by constraining and influencing the way in which interaction can take place.

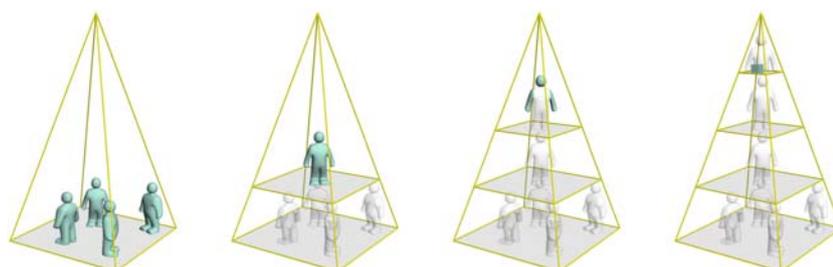
3 Describing movement-based interaction in camera spaces: three central concepts

3.1 Space

A camera space has the shape of a pyramid. Close to a tracked feature the space is small, but expands the further away from the feature the camera is, until it finally blurs out (when a feature is too far away from the camera to be registered). Combined with a digital application the space becomes what we call a mixed interaction space. The mixed interaction space is the combination of a physical camera space and a digital application space, existing within the same setting. The setting can be seen as a physical space containing the mixed interaction space, e.g. a library, a hallway, a street corner, or an office.

Interaction, which results in a division of space between what is interaction sensitive and what is not, can only occur within the camera space. Before the Bauhaus period [19] space was understood and defined as a container that could contain other containers (spaces). During the Bauhaus period space was seen as a continuum where spaces dynamically would intertwine

Fig. 5 Scales of camera spaces; diagram showing how the same camera space can adapt to different scales of space and feature



and flow among each other. This continuous space was changed by the observer moving in space. In our work we expand this understanding of space further through ubiquitous computing and virtual augmentations. With the dynamic nature of digital systems and interfaces the perception of space is not only changed by the observers moving point of view, but the space itself is dynamic, both regarding appearance and functionality. We hence see space as being highly defined by the potential functionalities afforded by areas or spaces within a continuous space, and not only as a container defined by a three-dimensional set of physical and virtual boundaries.

We see space and the physical environment as a design resource open to virtual and interactive augmentation. Using camera-based interaction we can design spaces that correspond perfectly with traditional physical spaces, where different connected but distributed spaces afford different functions and norms for social and working behaviours. An example of this is the kitchen where you cook, compared with the living room where you can crash on the couch and watch TV. As camera spaces are physically constrained they mime the pure physical spaces loaded with a certain functionality, however the augmented digital properties make the nature of these spaces different from traditional spaces for a number of reasons. Camera spaces can afford numerous functionalities depending on the specific user/users, time of day, kind of activity, and so on. This opens up for temporary ownerships of space or situations where different users of the camera space perceive it the space differently in a use perspective, or don't see it at all. Furthermore, functions are usually associated with specific parts of our built environment as e.g. the kitchen or bathroom, but camera spaces can adapt to any space because of its multi-scaled nature, understood in the sense that the kitchen has a scale that is adjusted to the human body, whereas camera spaces can take on any scale. As the camera space is not a physical container but just an area with extra or advanced properties, it can be established, moved, or wiped out instantly, changing the way user and space can engage and interact with the environment. Figure 5 shows how a camera space can be scaled to cover from small objects to several users depending on the distance from the camera to the tracked objects.

As computer systems migrate into our physical environments space becomes an important player in the design of future interactive environments. Therefore we have to accept and play with the properties of physical space and their influences on the types of interactions. We characterize the camera space by a

number of properties—type, scale, and orientation. The four above described applications show how the camera space can either be *static* or *dynamic*. In static camera spaces movement occurs when tracked features move in the camera space. In dynamic camera spaces it is the camera space itself that moves in relation to the tracked feature, see Fig. 6.

In the large scale applications iFloor and StorySurfer the camera space is static and the ceiling-mounted camera tracks the people who, at the same time, are the users of the system. In the small scale applications (ImageZoomViewer and Photo-Swapper) the camera space is dynamic and used to track primarily small static features. The user is in charge of moving and orienting the camera space.

Another property we have identified regarding *space* is the orientation of the camera space. As described earlier the camera space exists within a larger but continuous space. The importance of *orientation* is highly related to *scale*, and the relation between the user and the *space*. If we look at basic architectural elements such as walls, floors, and ceilings taking part in the definition and framing of physical space, we see that the orientation of the camera space influences the way in which a feature, being static or dynamic, can interact with the system. The floor is due to gravity our most shared architectural surface [20], why we as humans are used to act on the horizontal plane, see Fig. 7. As gravity forces objects to the ground, tracked features in a horizontal camera-space will most often exist on the two dimensional ground plan. Trackingwise the horizontal ground plan serves as a two-dimensional coordinate system for measuring positions and movements of tracked features. Orienting the space in the vertical direction, e.g. towards a wall, affords a new set of potential interactions where the feature has to overcome gravity. Most features will not continue to hang in free space, thus this type of space is similar to many situations where gestures and acting are used. Further acting within a vertical camera space the notion of a solid plane is replaced by a more free space in which the z-axis roughly seen as the distance between feature and camera can play a more dominant role in the feature moving away or towards the camera. This difference is most prominent with the larger scale camera spaces, where the users are the features themselves. With dynamic camera spaces orientation becomes less important because of the user's changed role from being a tracked acting feature to controlling the entire camera space. In these setups the focus on physical space diminishes because gravity in some sense has less influence—we are able to move the world.

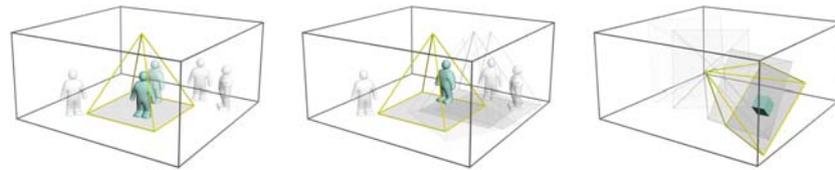


Fig. 6 *Static and dynamic camera spaces; a static camera space and dynamic features, b 2D dynamic camera space and static feature, c 3D dynamic camera space and static feature*

Fig. 7 *Orientation of camera spaces; a none, b horizontal, c vertical*

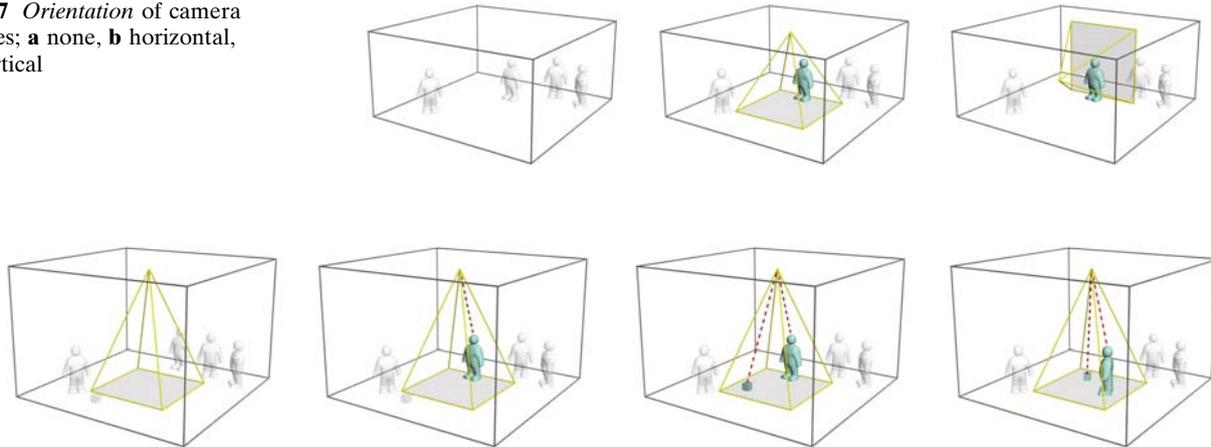


Fig. 8 *Relations between feature and camera; a none, b one relation, c two relations, d movement*

3.2 Relations

Where space defines the context for movement-based interaction, relations describe the connection between a camera and the tracked features within the camera space.

3.2.1 Entities and properties

A relation is an edge between a camera node and a tracked feature node. The edge can have a number of properties, and since vision algorithms are able to track multiple features a single camera can have multiple attached edges connected to the different features. However, a feature can also be tracked by different cameras, implying that also a feature can have multiple edges attached. Figure 8 shows how an interaction relation is created as a feature enters a camera space and how several relations can exist simultaneously.

The Maxis ImageZoomViewer application only utilizes a single relation between the mobile device’s camera and a tracked feature (circle, object or user’s face). In the iFloor application multiple features in form of human shapes are tracked within one camera space. Every time a new person enters the camera space a new relation is created, but the relation is not associated with a specific identity and no distinction is

made between the different users. The Photo-Swapper application is the opposite case, where multiple camera spaces are facilitated, still with only a single relation associated to each space. The different relations are combined by the shared application where each user is able to manipulate the interface and receive feedback, see Fig. 4.

A relation can be described by a set of properties that defines the potential interaction inputs. The number of properties depends on the algorithm used to analyse the input from the camera. The presence of a feature (on/off), the position of the feature in 1D, 2D or 3D space, rotation of the feature, the feature’s size, its state, identity, or information about uncertainty, are examples of properties associated with a relation. Interaction is triggered by mapping a different action to changes in a relation’s property.

The number of relations and the number of properties associated with each relation greatly determine the complexity of the interaction. With a complex setup there is a great need to visualize the way in which the user is actually able to influence the application through feedback. The iFloor application directly visualizes the relations present by drawing a line between the cursor and each user (tracked feature). Furthermore, each relation contains a 2D location and a size property based on the volume of the tracked

object. Changes in the size property control the force associated with each user's pull in the cursor.

The Mixis applications use only a single relation, but this relation has a 3D location property, and can have a 1D rotation property as well. This design space opens up for a 3D spatial interface, and is hence richer compared to both StorySurfer and iFloor. The Image-ZoomViewer application maps the movement in the physical space directly to pan and zoom in the application and it is therefore possible to pan and zoom simultaneously.

We found *direct input* and *gesture input* to be two different approaches on how to map changes in a relation's property to actions within an application. Direct input describes a mapping strategy where changes in a relation's properties directly influence the application. E.g. when a feature is positioned to the left in a camera space an application starts scrolling left. Gesture input describes a strategy where changes in a relation's property is monitored over time and matched to predefined patterns.

3.2.2 Multi-user interaction

With multi-user systems the relation concept opens up for a discussion about how to map the different relations to the multiple users. In the StorySurfer application each user is given a separate relation associated with an independent cursor. While one user browses the floor content by moving on the floor and hereby invoking a change in the position property, other users can use the magnifying lens to examine a book by standing still, hereby starting a selection timer. The Photo-Swapper also gives each user a separate relation, but in this application each relation is associated with its own camera. In the Photo-Swapper the relations have an extra property where the colour of the tracked feature is transferred to the corresponding cursor on the shared display as a sort of an identity.

3.3 Feedback

Movement-based interaction in camera spaces is problematic in the sense that the interaction tool is invisible to the user. The user cannot see what the camera registers or what the algorithms applied calculate. Feedback is hence important in order to visualize the relations that govern the interaction. Feedback from movement-based systems can be divided into *input feedback* and *application feedback*.

Input feedback focuses on telling the user that the input system is actually working; that a relation exists,

and that the user is able to control its properties. Bellotti et al. call it attention and use it to describe the problem of knowing when the system is ready and attending to actions [20].

Application feedback provides feedback about the application and its state. Bellotti et al. call this type of feedback alignment and address how to tell the users that the system does the right thing [21].

In the iFloor application input feedback about the relations is provided by a special cursor with a number of strings to each user. The application feedback is simultaneously provided on the floor in form of pictures, questions, and videos, which are highlighted and expanded as the cursor moves over them.

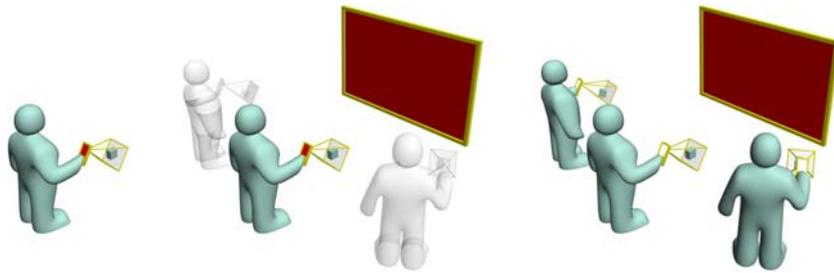
The four presented applications mainly use visual feedback and during our research it became evident that visual feedback needed to be provided close to the user's focus area. To further analyse feedback we found it useful to study focus shifts inspired by Bødker's work [23]. Bødker differentiates between focus shifts that are deliberate and focus shifts resulting from breakdowns and unsuccessful interaction design.

In the first iteration of Photo-Swapper we had two sources of feedback. The shared display showed information about the cursor and provided application feedback, whereas the display on the mobile device showed information about the position of the feature in the camera space (input feedback). This resulted in a below-average performance and user experience because of constant focus shifts between the mobile device's display and the shared display. We addressed this issue by designing a special cursor on the shared display with information about the input feedback previously provided on the phone. By eliminating a large number of focus shifts we were able to greatly improve the performance of the system. Figure 9 shows the different options for visual feedback in the Photo-Swapper setup.

4 Analysis of movement-based interaction systems

In this section we use the presented conceptual framework to analyse other movement-based interaction systems in order to demonstrate the framework's explanatory abilities. Furthermore, we use these systems to clarify our conceptual framework and fill in the gaps of the design space not covered by our own work, for instance non-visual feedback. The systems are selected as prototypes representing a number of different approaches to movement-based camera space interaction.

Fig. 9 Options of *input-feedback* and *application-feedback*; **a** local feedback, **b** local and remote feedback, **c** remote feedback



4.1 A number of movement-based systems

Sony Eyetoy is a motion recognition camera that plugs into a Playstation2 game console. The camera detects movements in the vertical plane from a user, and delimited areas of the screen are able to register input motion during a limited time period. In Beat Freak players are required to move their hands over a speaker in one of the four corners of the screen simultaneously as a CD flies across the speaker [24].

Urp is a tangible interface for urban planning, based on a workbench for simulating the interactions among buildings in an urban environment. The interface combines series of physical building models and interactive tools with an integrated projector/camera/computer node, the “I/O Bulb.” [14].

Mouthesizer consists of a miniature head-mounted camera which acquires video input from the region of the mouth. It extracts the shape of the mouth with a computer vision algorithm, and converts shape parameters to MIDI commands, so that the users’ facial gestures control a synthesizer or musical effects device. [25]

Kick Ass Kung Fu is a large display martial arts game installation where the player fights virtual enemies with kicks, punches, and acrobatic moves such as cartwheels. With the use of real-time image processing and computer vision the user’s video image is embedded inside 3D graphics. By shouting the player can go into a special power mode for a limited time. [26]

The ARTennis is a face-to-face collaborative application for mobile phones that use a set of three ARToolkit markers arranged in a line. When the players point the connected camera phones at the markers they can see and play on a virtual tennis court model superimposed over the real world. [27]

The above described projects are analysed based on the framework and presented in Table 1.

5 Discussion

By using the above described framework to analyse different movement-based interaction applications a picture is starting to form of how these applications

relate and differ. By looking at the space property in Table 1 we see that the applications generally fall into two groups; they either use static spaces or dynamic spaces. The applications that use dynamic spaces are generally used to support mobile interaction, while the static spaces are augmentations of specific physical spaces. The Mouthesizer application provides an interesting combination: while the space is static the whole setup is mobile. In the selected projects only applications based on dynamic spaces use multiple spaces, e.g. ARtoolkit tennis and Photo-Swapper; however, applications that combine e.g. two static spaces appear as an uninhabited space open for exploration by new applications.

Looking at the orientation we have three different types represented: horizontal surface, vertical surface, or a dynamic space. Looking at Table 1 scale and orientation seem to be related. To use a horizontal orientation setup tracking human movements requires a large surface to move on, e.g. a floor, whereas if the setup scale is smaller and the tracking objects are e.g. limbs or objects a table top is more suitable (Urp). The applications that use dynamic spaces can potentially be used with large scale spaces, however, most of the applications we have looked at use relatively small spaces (ARTennis, Photo-Swapper and ImageZoom-Viewer). The applications we have chosen cover the scale from multiple human shapes to small objects, thus since camera spaces can be resized freely (only depending on the optics in the camera) this conceptual framework will also be able to describe and analyse smaller or larger camera spaces, e.g. spaces under a microscope or tracking cars in a parking lot.

With relations, the tracked feature is often closely connected to the scale of the space in use. Many projects use several relations, and in most of the projects a single user is given control of only one relation, and the relation controls a single cursor or object. However, two applications are interesting to point out. In Kick-AssKungFu one user controls several relations as each limb of the body is used to control a relation. In iFloor the approach is exactly the opposite since several users are given their own relation, and these relations are coupled to a single cursor.

Table 1 The nine applications arranged after the concept of *space, relation and feedback*

	StorySurfer	iFloor	Kick Ass Kung Fu	EyeToy beat freak	Mouthesizer	URP	ImageZoomViewer	ARTennis	Photo-Swapper
Space Type	Static single	Static single	Static single	Static single	Static single	Static single	Dynamic single	Dynamic two	Dynamic multiple
Orientation	Static horizontal	Static horizontal	Static vertical	Static vertical	Static vertical	Static horizontal	Dynamic	Dynamic	Dynamic
Scale	Humans	Humans	Human	Human/limbs	Limbs	Object	Object	Object	Object
Relation Feature	Human shape	Human shape	Human shape	Movement Shape	Movement Shape	Color dots	Color/circle object	Tags	Color/circle object
Number Presence	Multiple Yes	Multiple Yes	Single Yes	Multiple Yes	Single Yes	Multiple Yes	Single Yes	Multiple Yes	Single Yes
Position	2D	2D	2D	2D	2D	2D	3D	3D	3D
Id	No	No	No	No	No	Yes	No	Yes	No
Other Feedback	Size	Size	Size	Shape	Shape	Rotation	Rotation	Tilt	Rotation
Type Input	Visual Looking glass (floor projection)	Visual Strings (floor projection)	Visual Player rep. (wall projection)	Visual Player rep. (TV)	Audio Changes in sound	Visual Overlay e.g. shadows (table projection)	Visual Dynamic cursor (phone display)	Visual Court/racket (phone display)	Visual Dynamic cursor (phone and wall display)
Application	Visual (floor projection)	Visual (floor projection)	Visual and sound (wall projection)	Visual and sound (TV)	Audio (speakers)	Visual (table projection)	Visual (phone display)	Visual, sound and vibration (phone)	Visual (wall display)

Concerning feedback, the chosen applications mainly rely on visual feedback, only the Mouthesizer rely purely on auditory feedback. Even though several applications using more ambient feedback can be found and designed, the use of visual feedback seems to be the most common feedback mechanism for movement-based camera-space systems. To minimize focus shifts almost all the discussed applications use overlays or special cursors to present input feedback close to the application feedback.

Since some of these applications rely on complex interaction with multiple relations with many properties a standard cursor provides too little input feedback, hence specially designed cursors or overlays that visualize the properties of the relations need to be considered and designed. In the ImageZoomViewer application the cursor uses colours and changes in its size to visualize the distance to the feature and the presence/non-presence of a feature. In the KickAssKungFu input feedback is addressed by letting the user be the cursor, and all movements are thus mirrored in real time. However, in EyeToy Beat Freak the application feedback is weaker. Tracking the limbs being or not being in the right position of the camera space to intersect with an object in space-time only confirm the user in being right or wrong, the application do not provide the user with any type of information in orientation, for instance if leaning too much to the left. In Larssen et al. [22] an evaluation of Sony EyeToy was performed on how movement as input would hold as communication in the interaction. The evaluation highlights how challenging it would be to facilitate the interaction, without the use of a conventional GUI for feedback, even though the interaction is not based on detailed knowledge of orientation.

A further issue to discuss in relation to designing camera spaces is frame rate. All the projects described here are exploiting the maximum possible frame rate for the camera to give instant feedback. The frame rate can be seen as a property that connects the relation between space and feature dealing with the match or mismatch between physical space time and interaction space time, which could inspire to new ways of designing camera space interfaces.

6 Conclusion

Building camera tracking systems is not only about developing technically sound algorithms. Being able to describe and understand the design possibilities and limitations is an equally important factor in the development of a successful system.

With this conceptual framework we have covered some basic concepts relating to movement-based interaction using camera tracking, but there are other important concepts we have left for future work. Mapping, privacy, tracking inaccuracy, ambient feedback, and affordance seem equally relevant, but to focus our discussion we have chosen the space, relation and feedback concepts.

With the space concept, properties of the setting that the system is deployed in are taken into account. Relation describes different approaches for mapping tracked features to interaction, and feedback address how the users are informed about the events taking place within the digital application. These concepts have proven not only to be useful for analysing our own four applications, but also to point out interesting aspects of a number of other very different movement-based camera tracking applications.

We believe the framework and table presented in this article can be used to describe and analyse a wide variety of movement-based applications in camera spaces. The aim has been to present both a general conceptual framework for comparison as well as provide concrete suggestions for the analysis of individual applications. We also hope that this framework will be useful for exploring novel variants and approaches to the design of movement-based applications with camera spaces.

References

1. Barkhuus M, Chalmers M, Tennent, P, Bell M, Hall M, Sherwood S, Brown B (2005) Picking pockets on the lawn: the development of tactics and strategies in a mobile game. In: Proceedings of ubicomp, Springer 3-540-28760-4, pp 358–374
2. Dance Dance Revolution UltraMix 3. Available at http://www.konami-usa.com/production/ddr_ultramix3/index.html
3. Samsung SCH-S310 Available at <http://www.samsung.com/>
4. Patridge K, Chatterjee S, Sazawal V, Borriello G, Want R (2002) TiltType: accelerometer-supported text entry for very small devices. In: Proceedings of the 15th annual ACM symposium on user interface software and technology UIST'02, Paris, 27–30 October 2002, pp 201–204
5. Pavlovic V, Sharma R, Huang T (1997) Visual Interpretation of Hand Gestures for Human-Computer Interaction. IEEE Trans Pattern Anal Machine Intell 19(7):677–695
6. Bradski G, (1998) Computer vision face tracking for use in a perceptual user interface, intel technology. J Q2, pp 1–15
7. Hansen TR, Eriksson E, Lykke-Olesen A (2005) Mixed interaction space – designing for camera based interaction with mobile devices. In: Conference on human factors in computing systems in CHI '05 extended abstracts on human factors in computing systems, Portland, ACM Press
8. Mackay W (1998) Augmented reality: linking real and virtual worlds: a new paradigm for interacting with computers. In:

- Proceedings of the working conference on advanced visual interfaces, L'Aquila, 24–27 May 1998
9. Benford S, Gaver B, Boucher A, Walker B, Pennington S, Schmidt A, Gellersen H, Steed A., Schnädelbach H, Koleva B, Anastasi R, Greenhalgh C, Rodden T, Green J, Ghali A, Pridmore T (2005) Expected, sensed, and desired: a framework for designing sensing-based interaction. *ACM transactions on computer-human interaction (TOCHI)* vol. 12 , Issue 1. ACM Press, pp 3–30
 10. Abowd G, Mynatt E (2000) Charting past, present, and future research in ubiquitous computing. In: *ACM transactions on computer-human interaction (TOCHI)*, vol 7 , Issue 1. Special issue on human-computer interaction in the new millennium, part 1, ACM Press, pp 29–58
 11. Ullmer B, Ishii H (2000) Emerging frameworks for tangible user interfaces. *IBM Syst J* 39(3–4):915–931
 12. Holmquist LE, Redström J, Ljungstrand P (1999) Token-based access to digital information. In: *Proceedings of the 1st international symposium on handheld and ubiquitous computing (HUC'99)*, Karlsruhe, pp 234–245
 13. Fishkin K (2004) A taxonomy for and analysis of tangible interfaces. *Pers Ubiquit Comput* 8(5):347–358
 14. Underkoffler J, Ishii H (1999) Urp: a luminous-tangible workbench for urban planning and design. In: *Conference on human factors in computing systems, proceedings of the SIGCHI conference on human factors in computing systems*. Pittsburgh, Pennsylvania, pp 386–393
 15. Krogh PG, Ludvigsen M, Lykke-Olesen A (2004) “Help Me Pull That Cursor”—a collaborative interactive floor enhancing community interaction. In: *Proceedings of OZCHI 2004*, Wollongong, CD-ROM, 22–24 November 2004. ISBN:1 74128 079
 16. StorySurfer. Available at <http://www.interactivespaces.net>
 17. Hansen TR, Eriksson E, Lykke-Olesen A (2005) Mixed interaction spaces—expanding the interaction space with mobile devices. In: *Proceedings of the 19th British HCI Group annual conference*, Edingburgh, 5–9 September 2005
 18. Hansen TR, Eriksson E, Lykke-Olesen A (2005) Mixed interaction spaces—a new interaction technique for mobile devices. In: *Proceedings of the 7th international conference on ubiquitous computing*, Tokyo
 19. Forty A (2000) *Words and buildings—a vocabulary of modern architecture*. Thames and Hudson, New York
 20. Gehl J, (1987) *Life between buildings—using public space*, Van Nostrand Reinhold, New York
 21. Bellotti V, Back M, Edwards WK, Henderson A, Lopes C. (2002) Making sense of sensing systems: five questions for designers and researchers. *Conference on human factors in computing systems*. In: *Proceedings of CHI 2002*, ACM Press, Minneapolis, pp 471–509
 22. Larssen AT, Loke L, Robertson T, Edwards J (2004) Movement as input for interaction—a study and evaluation of two eyetoy (TM) games. In: *Proceedings of OZCHI 2004*, Wollongong, CD-ROM, 22–24 November 2004. ISBN:1 74128 079
 23. Bødker S (1996) Applying activity theory to video analysis: how to make sense of video data in human-computer interaction. In: Nardi B (ed) *Context and consciousness. Activity theory and human computer interaction*, MIT press, pp 147–174
 24. Sony Eye-Toy. Available at <http://www.eyetoy.com>
 25. Lyons M, Haehnel M, Tetsutani N (2001) The mouthesizer: a facial gesture musical interface. *conference abstracts, Siggraph 2001*, Los Angeles, p 230
 26. Hämäläinen P, Ilmonen T, Höysniemi J, Lindholm M, Nykänen A (2005) Martial arts in artificial reality. In: *Proceedings of CHI 2005 conference on human factors in computing systems*, Portland, pp 781–790
 27. Henrysson A, Billinghurst M, Ollila M (2005) Face to face collaborative AR on mobile phones. In: *Proceedings of the international symposium on mixed and augmented reality (ISMAR 2005)*, Vienna