Controller-free exploration of medical image data: experiencing the Kinect

Luigi Gallo, Alessio Pierluigi Placitelli, Mario Ciampi
National Research Council of Italy
Institute for High Performance Computing and Networking
Via Pietro Castellino 111, 80131 Naples, Italy
{luigi.gallo, alessio.placitelli, mario.ciampi}@na.icar.cnr.it

Abstract

In this paper, an open-source system for a controller-free, highly interactive exploration of medical images is presented. By using a Microsoft Xbox Kinect^{TM} as the only input device, the system's user interface allows users to interact at a distance through hand and arm gestures. The paper also details the interaction techniques we have designed specifically for the deviceless exploration of medical imaging data. Since the user interface is touch-free and does not require complex calibration steps, it is suitable for use in operating rooms, where non-sterilizable devices cannot be used.

1. Introduction and background

In recent years, non-invasive imaging techniques such as computed tomography (CT) and magnetic resonance imaging (MRI) have been assuming great importance in clinical practice. Today, clinicians may access large medical datasets, visualize them slice-by-slice or in 3D and explore them interactively by means of different user interfaces. Despite the large availability of fully-featured DICOM viewers that allow clinicians to explore data by omnifarious desktop-based user interfaces, very few systems have been designed to allow a practical and efficient exploration of data in critical medical environments such as operating rooms (OR), in which digital images are commonly used to help surgeons navigate the patient's body.

The main challenge with this kind of interface is that the interface itself has to disappear: surgeons need to browse through scans without having to physically touch any control, since they cannot leave the sterile field around the patient. In fact, scrubbing in and out to access the computer safely can result in a severe increase in the duration of the operation. Moreover, the time-consuming training required to learn how to use the interface has also to be considered as a priority for a specific clinical use [10]. As a consequence, there is a growing interest in controller-free, efficient and easy-to-use interfaces for medical data exploration.

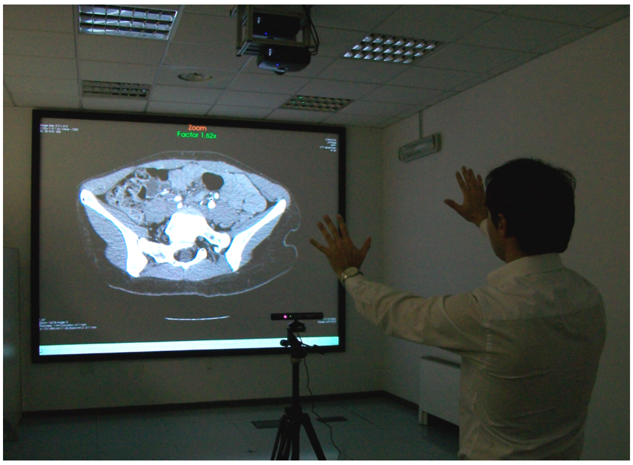

Figure 1. Controller-free interactive exploration of medical images through the Kinect.

Gesture control interfaces, namely interfaces that recognize the gestures made by the user, seem to be suitable for operating rooms. However, at present, there exists no really mature technology for effective gesture control. As reported in [12], there are two main difficulties in the design of this kind of interface: temporal segmentation ambiguity, that is, how to define the starting and ending points of continuous gestures, and; spatial-temporal variability, due to the fact that gestures vary considerably between individuals. Generally, there exist many-to-one mappings from concepts to gestures and vice versa, and hence gestures are ambiguous and incompletely specified [9].

All gesture control interfaces need to sense the human body position, configuration and movement. Sensing devices may be attached to the user, by using magnetic field trackers, instrumented data gloves, body suits or a combination of all of these. These sensing technologies vary in accuracy, resolution, latency, range of motion and user comfort. In [15], a glove-based interface for medical image analysis was presented. The interface uses an instrumented data glove and a magnetic tracker, combining independent hand posture and trajectory recognition to send command messages to the host application. In [2], the tracking system was replaced with a Wiimote to provide a low-cost solution for medical data exploration at a distance. However, these solutions are unsuitable for operating rooms since they require the user to use non-sterilizable devices.

To avoid contamination of the patient, the OR and the surgeon, some vision-based interfaces have been proposed, too. In [16] the Gestix system, a video-based hand gesture capture and recognition system used to manipulate magnetic resonance images (MRI), was presented. However, vision-based interfaces need to contend with other problems related to occlusion, lighting, speed of movement, cluttered background, distance of operation and, most of all, the inherent loss in information that happens whenever a 3D image is projected onto a 2D plane.

The use of stereo cameras overcomes this limitation but raises new difficulties such as camera placement and calibration and the correspondence problem [5]. Although algorithms that capture full skeletal motion at near real-time frame rates are available, the special controlled recording conditions required make them unsuitable for use in operating rooms. In [8] the WagO system, a hand gesture control plug-in for the open source DICOM viewer OsiriX [11], was presented. Hand gestures and 3D hand positions are recognized by using two webcams in stereo configuration. However, the system can only recognize a small set of static gestures, the user position is tightly constrained and the recognition rate is not high enough to allow surgeons an effective interaction.

Recently, range sensors have stood out as an option to approach human motion capture with a non-invasive system setup. Time-of-flight (TOF) sensors provide, at high frame rates, dense depth measurements at every point in the scene. TOF cameras capture an ordinary RGB image and in addition create a distance map of the scene using the light detection and ranging (LIDAR) detection schema: modulated light is emitted by LEDs or lasers and the depth is estimated by measuring the delay between emitted and reflected light. This approach makes the TOF cameras insensitive to shadows and changes in lighting, so allowing a disambiguation of poses with a similar appearance. More recently, a less expensive solution to obtain 3D information from video, with respect to the one implemented in the TOF cameras, has emerged: projecting structured IR light patterns on the scene and retrieving depth information from the way structured light interferes with the objects in the scene. This is the mechanism used in the Microsoft Xbox Kinect^{TM} and in the recently announced Asus Xtion PRO^{TM} to derive the distance map of the scene.

In particular the Kinect, since it is inexpensive, off-the-shelf and widely available, is being considered to repeat the success of the Wiimote, which has already been adopted as a user interface for medical data exploration at a distance [4]. As reported in [13], surgeons at Sunnybrook Hospital in Toronto are experimenting with the Kinect as a means of visualizing MRI or CT images in the operating room without leaving the sterile field around the patient. According to their tests, the use of this device can decrease surgery delays by as much as two hours. The aforementioned interface allows surgeons to scroll through the images and lock the one of interest onto the screen. However, this interface does not allow an interactive exploration of the medical images, but only the selection of a single image from a study. Zoom, rotation and modification of the transfer function used have to be applied before entering the images into the system.

In this paper, we describe a whole open-source system for a fully-featured, highly interactive exploration of CT, MRI or PET images, thanks to a gesture control interface that makes use of the Kinect as the only input device. Such an interface allows users, by using both kinetographic and metaphoric hand and arm gestures, to execute basic tasks such as image selection, zooming, translating, rotating and pointing, and complex tasks such as the manual selection and extraction of a region-of-interest (ROI) as well as the interactive modification of the transfer function used to visualize the medical images. The interface has been integrated in MITO (Medical Imaging TOolkit) [6], an open-source, PACS (Picture Archive and Communication System) integrated medical image viewer fully compliant with the DICOM (Digital Imaging and COmmunications in Medicine) standard for image communication and file formats. Since the interface is touch-free and does not require complex calibration steps, it is suitable for use in operating rooms and also every time there is the need to explore interactively medical images at a distance.

2. User interface description

In the system's controller-free interface all the interaction commands are mapped to gestures, which can be executed at a distance from the display without touching it. Moreover, filters have been implemented to reduce the noise in the device signal, to increase the accuracy of the remote pointing and to filter hand tremors during all the interaction tasks. The user's body is represented as a stick figure, which consists of line segments linked by joints. The motion of the joints provides the key to motion estimation and recognition of the whole figure. The skeleton fitting process is performed automatically in a non-intrusive way, thanks to the calibration procedure described in section 2.2.

Recognized gestures have both static elements (the user assumes a certain pose or configuration) and dynamic elements (with pre-stroke, stroke, and post-stroke phases). Static postures, represented by a single image, are used to discriminate between possible actions. Dynamic gestures, characterized by the spatio-temporal motion structures in image sequences, are used to further discriminate between actions.

To allow a deterministic state transition and to provide an unambiguous way of specifying the start and end points of gestures, both in time and in space, the gesture control interface relies on the concept of an *activation area*. The system continuously checks the user's stick figure and hand postures only if at least one of her/his hands lie inside the activation area, that is, the arm is outstretched more than 55% of its total length. If not, only the neck and left and right hand joints are checked. This approach allows the system also to reduce the computational burden, because the computer vision algorithms used to detect the hand posture run only in the check state. Moreover, it allows the system to minimize unwanted state transitions, because, moving from one state to another, users have to explicitly move their hands out of the activation area.

2.1. The input device

Figure 2 shows a picture of the Kinect. It is equipped with a laser-based IR projector and monochrome CMOS sensor. As reported in section 1, the IR projector is used to send a fixed speckle pattern towards the focused area. The aforementioned pattern is then detected by the CMOS sensor and used to calculate depth data by triangulation against a hardwired pattern. The Kinect also embeds a tilt motor for sensor adjustment, a microphone array and an RGB camera.

The device features a horizontal field of view of 57°, a 43° vertical field of view and an operating distance range between 0.8 m and 3.5 m. The spatial resolution is 3 mm for X/Y and 10 mm for the Z depth within 2 m from the sensor. The produced datastreams have a resolution of 640×480 at 30 Hz. Depth data have an 11 bit resolution with values ranging from 0 to 2047.

In the following, we will call *XY* the plane on which the Kinect camera resides, and *Z* the axis ortogonal to this plane directed towards the user.

2.2. Calibration

The calibration procedure, which takes less than 30 seconds, is required to allow the system to: perform the skeleton fitting process; compute the arm length; identify the dominant hand; map the motion with the display space; compute the parameters needed to tune the filters used in the interaction techniques; and compute the area of the palm facing forward and of the clenched fist.

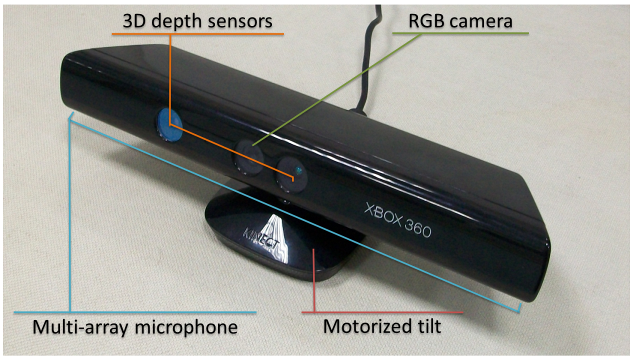

Figure 2. The Microsoft Xbox Kinect$^{\text{TM}}$.

To proceed with the calibration, first of all the user has to assume the calibration pose, or hold her/his arms out from her/his sides bent at 90°, to start the skeleton fitting process. Once completed, the user is represented as a stick figure, which consists of line segments linked by joints. For the interaction techniques we have designed, only the head, neck and left/right shoulder, elbow and hand are used. Then, the user is required to point to the top left corner of the screen and keep her/his hand steady for at least 1 second. The same procedure is repeated for the bottom right corner of the screen. The effect of this procedure is threefold: first of all, it allows the system to identify the dominant hand; secondly, it allows it to retrieve the display size; finally it allows it to measure the hand jitter so as to tailor the filter used to enhance the accuracy of pointing.

Finally the user, while still staying fronto-parallel with respect to the camera, is required to put the palm of her/his dominant hand facing forward and then to close it in a clenched fist. The system computes the area of the user's hands when open and when closed, so as to use this information to discriminate between hand poses. The hand posture detection relies on a classification function which is based on the analysis of the area of the user's hand. First, the system extracts the hand surrounding area from the distance map provided by the Kinect buffer and transforms it into a black and white binary matrix. The resulting image is then processed to find the external contours of the hand, by using a technique derived from the border following the algorithm described in [14] for the topological analysis of digitized binary images. From now on, the system is able to discriminate between the open palm and clenched fist by comparing the current user's hand area with a dynamically computed threshold that considers also the the user distance from the Kinect.

Table 1. Static postures and dynamic gestures currently used in the system.

	Static posture recognition	Dynamic gesture recognition
Pointing - point	ONLY the DOMINANT hand is *active*	none
Pointing - click	BOTH hands are *active* (while already in pointing state)	palm of the NON-DOMINANT hand *closed - open - closed* (sequence)
ROI extraction	*folded arms*	none
ROI erasing	ONLY the DOMINANT hand is *active*	unstable movements for 500 ms
Animating	ONLY the NON-DOMINANT hand is *active*	none
Zoom	BOTH hands are *active* AND with *palms facing forward*	discordant movements in the XY plane
Translation	BOTH hands are *active* AND with *palms facing forward*	concordant movements in the XY plane
Windowing	BOTH hands are *active* AND one with *palm facing forward*, the other with a *clenched fist*	none
Rotation	BOTH hands are *active* AND with *clenched fists*	none

2.3. Interaction techniques

Choosing the appropriate gesture is a key activity in the design of a controller-free interface. The choice has to keep in consideration both the hardware characteristics of the input device and the applicative domain in which the interaction tasks take place. The finite state machine reported in figure 3 depicts the state transitions of the interface. It consists in the following states:

Idle this state consists of three substates: idle, stabilize and check. While in the *idle* substate, the system checks the distance between the user's hands and torso. If at least one of the user's hands enters the activation area, or is outstreched more than 55% of the user's arm length, then the actual joint positions are stored in a circular queue of 30 elements, and the system switches to the stabilize substate.

In the *stabilize* substate, the system checks hand stability on the Z axis, that is, the longitudinal axis starting from the Kinect and going towards the user; if the hand position is stable, that is, there are no notable movements on this axis, then the circular queue is cleaned so that only the current event is stored, and the FSM moves to the check state.

While in the *check* state, the system checks if the hand positions have moved, compared to the ones stored in the circular queue, by at least of 25 mm. If this is the case, it activates the palm/fist detector and checks the static and dynamic conditions to enter the "active" states. Once the system enters an active state, it exits only if the user moves her/his hands back out of the activation area;

Pointing to enter this state, the user has to move her/his dominant hand inside the activation area while keeping the non-dominant hand outside. As the pointing modality, we have implemented the *image-plane* [7], in which a ray is determined through two points in space: the user's eye location, and the position of the tip of the user's index finger; then, the cursor is placed at the intersection of this ray with the display. The effect is that the user can see the cursor aligned with her/his index finger in her/his field of view, even if they are actually at different depths. Such a technique requires a tracking of the dominant eye position and the index finger. In our implementation, we approximate the dominant eye with the head position and the index finger with the dominant hand position. The control-to-display ratio, namely the coefficient that maps the physical movements of the hand to the on-screen cursor movements, is changed dynamically by using the Smoothed Pointing technique [3], which is aimed at enhancing the accuracy of the pointing and at filtering the user's hand jitter.

The click is performed by moving the non-dominant hand in the activation area and executing the sequence fist-palm-fist. By using the point & click features, a user can select a ROI or compute the distance between two points in an image. Exiting by the pointing modality also has the effect of connecting the first selected point with the last one, so as to close the ROI selected;

ROI extraction once a ROI has been selected, to extract it, the user has to put her/his hands in the activation area and assume the static posture of folded arms. This will raise the segmentation command. Then, the system will return to the idle state;

ROI erasing the metaphoric gesture of cleaning, that is, moving quickly the dominant hand like cleaning the display, has the effect of erasing the previously selected ROI and, if a segmentation has been executed, of restoring the original image;

Animating to enter this state, the user has to move her/his non-dominant hand inside the activation area. Then, hand movements going from left to right describing a horizontal line make the system move forward in the image series, vice versa with movements from right to left. To allow a precise selection of an image inside a large medical study, the PRISM technique [1] has been used to map hand movements and velocity of sliding: the faster the user moves, the quicker the images slide. In this way, by moving slowly, the user is able to select each single image even in a large medical dataset;

Zoom the zoom gesture is modeled after the widespread gesture commonly used in touch-screen devices. In fact, it involves keeping both hands with open palms inside the activation area with the open palms moving in opposite directions, regardless of their position in space. The system estimates the zoom factor by comparing the actual distance between the hands with that of the first recognized frame. This technique also takes advantage of the PRISM filter to better stabilize the zooming factor;

Translation the translation gesture is similar to the zoom one, except for the dynamic gesture: in this case, the hands movements in the XY plane have to be concordant. Once the system enters this state, the medical image follows the user's hand movements. While the direction and sense of the image movement vector are determined by considering only the current position of the user's hands, the magnitude of the vector is computed by also considering the user's hand velocity, so as to scale down the translation when the user moves her/his hands slowly;

Windowing to enter this state, both the user's hands have to be active, one with the palm facing forward, the other with a clenched fist. Then, the techniques used are the same as in the zoom and translation states: moving both hands on the Y axis will modify the window level (WL), whereas moving the hands in opposite directions on the X axis will modify the window width (WW). Since the PRISM technique adopts axis independent scaling, that is, it operates on each axis independently, we have used it to help users to control separately WL and WW without the need to move into the idle state;

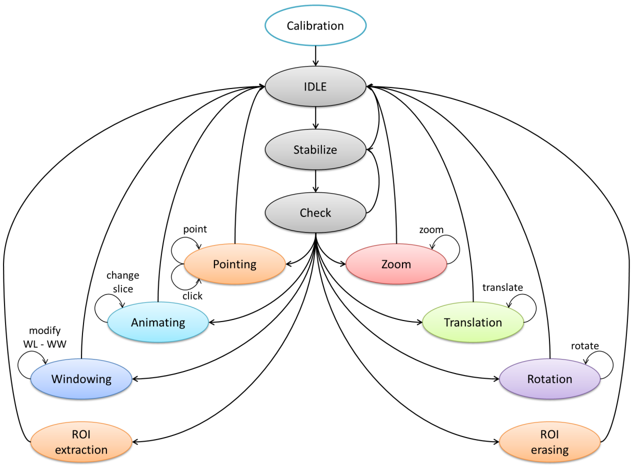

Figure 3. The finite state machine describing the state transitions of the interface.

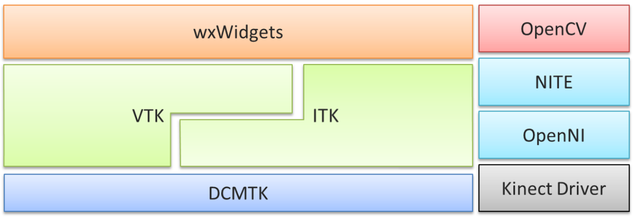

Figure 4. The stack of software libraries used in the system.

Rotation to enter this state, both hands need to be active and with clenched fists. This dynamic gesture involves the movement of both hands in a circular motion, describing an arc. The vector passing through user's hands, at the instant the state becomes active, is identified as the rotation reference axis. Then, the rotation is performed by calculating the signed angle between the rotation reference axis and the vector which passes through the current hand positions. Also in this case, a velocity-based filter has been used to help users to set the rotation angle and to filter hand tremors.

All the above techniques share a stabilizing mechanism to avoid any alteration of the final effect of the action performed on the image when returning to the idle state. In fact, since exiting from a state requires the user to move her/his hand or hands out of the activation area, during this movement the interface keeps detecting the hand movements on the XY plane and so produces unwanted image modifications. To avoid this, the system continuously keeps track of the hand movements on the Z axis and, if they are predominant with respect to the movements on the XY plane, it does not execute any action.

3. Implementation details

The interface has been written entirely in C++ language and built on open-source and cross-platform libraries, which have been combined and extended to support medical imaging processing and interaction functionalities. OpenNI, an open-source framework for natural interfaces, and PrimeSense's NITE Middleware, which provides a closed-source but free to use implementation for OpenNI functionalities, have been used to communicate with the

Kinect. To recognize the static hand postures, the system also makes use of the open-source OpenCV image processing library. The stack of software libraries used in the system is depicted in figure 4. A video showing the system in action is avaiable at `http://www.youtube.com/watch?v=CsIK8D4RLtY`

The gesture control interface described in this paper has been integrated in MITO [6], an open-source, PACS-integrated medical image viewer. MITO was developed from scratch by using only open-source libraries and is fully compliant with the DICOM standard for image communication and file formats. With the exception of the Kinect and Wiimote drivers, presently available only for MS Windows OSs, MITO is a platform-independent application, ready for use in every medical facility.

Interaction in MITO follows the *event - state - action* paradigm, so that the control flow is completely determined by the user's inputs. To integrate a new interaction modality, a designer has to develop an event handler, which is the procedure in charge of associating events to corresponding actions, and a driver for the input device, which is in charge of generating properly formatted events. The state of the interface (the finite state machine is reported in figure 3) defines the set of possible events that can be received. All the Kinect updates are propagated through the system by using the observer software design pattern.

The interface was developed specifically for fast operation at video frame rates, since natural communication requires a low latency action-reaction cycle. An analysis of the average lags between a change in the Kinect depth sensor data and the generation of an interaction event shows that, on commodity hardware, the execution of the whole pipeline, which comprises recognition and filtering, requires less than 1 millisecond. Only when the computer vision algorithms are also executed, for example in the pointing state, when the non-dominant hand enters the activation area in order to click, and in the check state,may the total lag reach 10 milliseconds.

4. Conclusions and future work

In this work, a user interface that allows a controller-free interaction with medical images through an inexpensive, off-the-shelf range sensor has been proposed. The interface has been implemented and integrated into a medical image viewer and issued as open source software, which may promote its use and evolution.

Future work will be focused on extending the gesture control interface so as to enable a controller-free 3D interaction in collaborative semi-immersive medical imaging environments.

References

[1] S. Frees, G. D. Kessler, and E. Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction*, 14(1), 2007.

[2] L. Gallo. A Glove-Based Interface for 3D Medical Image Visualization. In R. J. Howlett et al., editors, *Intelligent Interactive Multimedia Systems and Services*, pages 221–230, Berlin Heidelberg, 2010. Springer-Verlag.

[3] L. Gallo, M. Ciampi, and A. Minutolo. Smoothed Pointing: a User-Friendly Technique for Precision Enhanced Remote Pointing. In *CISIS '10*, pages 712–717, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[4] L. Gallo, A. Minutolo, and G. De Pietro. A user interface for VR-ready 3D medical imaging by off-the-shelf input devices. *Computers in Biology and Medicine*, 40(3):350–358, 2010.

[5] M. B. Holte, T. B. Moeslund, and P. Fihl. View-invariant gesture recognition using 3D optical flow and harmonic motion context. *Computer Vision and Image Understanding*, 114(12):1353–1361, 2010.

[6] ICAR-CNR and IBB-CNR. Medical Imaging TOolkit (MITO). `http://amico.icar.cnr.it/mito.php`.

[7] R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale, and S. Greenberg. A comparison of ray pointing techniques for very large displays. In *GI '10*, pages 269–276, Toronto, Ont., Canada, 2010. CIPS.

[8] T. Kipshagen, M. Graw, V. Tronnier, M. Bonsanto, and U. G. Hofmann. Touch- and marker-free interaction with medical software. In R. Magjarevic et al., editors, *WC '09*, pages 75–78. Springer Berlin Heidelberg, 2009.

[9] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

[10] K. Montgomery, M. Stephanides, S. Schendel, and M. Ross. User interface paradigms for patient-specific surgical planning: lessons learned over a decade of research. *Computerized Medical Imaging and Graphics*, 29(5):203–222, 2005.

[11] A. Rosset, L. Spadola, and O. Ratib. Osirix: An open-source software for navigating in multidimensional dicom images. *Journal of Digital Imaging*, 17(3):205–216, 2004.

[12] C. Shan. Gesture control for consumer electronics. In L. Shao et al., editors, *Multimedia Interaction and Intelligent User Interfaces*, Advances in Pattern Recognition, pages 107–128. Springer London, 2010.

[13] L. Steakley. Canadian hospital tests Kinect in the operating room, March 2011. `http://scopeblog.stanford.edu/archives/2011/03/kinect-in-the-operating-room.html`.

[14] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[15] B. S. Tani, R. S. Maia, and A. v. Wangenheim. A gesture interface for radiological workstations. In *IEEE CBMS '07*, pages 27–32, Washington, DC, USA, 2007. IEEE Computer Society.

[16] J. P. Wachs, H. I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith. A gesture-based tool for sterile browsing of radiology images. *Journal of the American Medical Informatics Association*, 15(3):321–323, 2008.