

Using interaction signatures to find and label chairs and floors

AUTHOR(S)

P Peursum, Svetha Venkatesh, G West, H Bui

PUBLICATION DATE

01-01-2004

HANDLE

10536/DRO/DU:30044290

Downloaded from Deakin University's Figshare repository

Deakin University CRICOS Provider Code: 00113B

DRO
Deakin University's Research Repository
DEAKIN UNIVERSITY AUSTRALIA Worldly

This is the published version:

Peursum, Patrick, Venkatesh, Svetha, West, Geoff A. W. and Bui, Hung Hai 2004, Using interaction signatures to find and label chairs and floors, IEEE pervasive computing, vol. 3, no. 4, pp. 58-65.

Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30044290

©2004 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright : 2004, IEEE

dro.deakin.edu.au
Deakin University CRICOS Provider Code: 00113B

Using Interaction Signatures to Find and Label Chairs and Floors

Interaction signatures are a proposed method to find and classify objects on the basis of how humans interact with those objects. The method addresses many key problems encountered in smart-home monitoring systems.

Patrick Peursum, Svetha Venkatesh, and Geoff A.W. West
Curtin University of Technology

Hung Hai Bui
SRI International

For smart homes to truly live up to their "smart" moniker, systems must be able to recognize objects in indoor scenes and detect when and how humans interact with them. Without object recognition, smart homes can't make full use of video cameras because vision systems can't provide object-related context to the human activities they monitor. The homes thus underutilize a rich, versatile information source.

Traditional shape-based object recognition tends to fail when presented with a smart home's typical scenes—wide-angle views of indoor scenes containing a variety of objects. Partial occlusions, unconstrained orientations, irregular shapes, people moving the objects, and lack of detail for distant objects all make shape-based recognition difficult. Unfortunately, these same factors are also the defining characteristics of household environments. The result is that little object recognition research has proven robust enough to be deployed in smart-home testbeds such as Microsoft's EasyLiving project (http://research.microsoft.com/easyliving) and the Georgia Institute of Technology's AwareHome (www.cc.gatech.edu/fce/ahri).

To access the benefits of context, many researchers manually label objects or areas of interest (see the "Related Work" sidebar). Such approaches can facilitate intelligent device control and simplify human-behavior monitoring. Knowing a phone's position, for example, simplifies the task of recognizing when someone is making a call.[1] Similarly, identifying the act of page-flipping is more probable when the system recognizes the object of interest as a book.[2]

Our research takes an action-centered approach to automatically learning and classifying functional objects. Our premise is that interpreting human motion is much easier than recognizing arbitrary objects because the human body has constraints on its motion. Moreover, humans tend to interact differently with different objects, so you should be able to identify an object by analyzing how people move when they manipulate it. We call these motions the human-object *interaction signature*.

Systems can use interaction signatures to recognize objects without considering the object's physical structure, thus bypassing many difficulties inherent in shape-based recognition. Although this means that the system can't label objects that humans never interact with (walls and ceilings, for example), such objects are generally less relevant than manipulated objects. Another advantage to interaction signatures is that people frequently and repeatedly interact with household objects, so the system can build up evidence for object locations and labels. Object labels are strengthened or weakened as the system accumulates interaction signature evidence

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/04/$20.00 © 2004 IEEE

Related Work

Most current approaches to object recognition classify objects by comparing shape-based object models against a database of known objects.[1] However, this approach has several serious drawbacks. Typically, large variations in shape and orientation will occur in any particular object class. To address this, Louise Stark and Kevin Bowyer proposed using function-based object recognition, which classifies object models on the basis of their functional components.[2] A chair, for example, might be defined as any object with a flat, stable sitting surface. Unfortunately, the basic problem of trying to extract the object's 3D model from its 2D image remains. Moreover, actually finding and segmenting the object's 2D image out of a wide-angle view is difficult. Recent work by Brandon Sanders and his colleagues addresses this problem using background subtraction and temporal evidence to accurately segment objects that are occasionally moved by humans (dubbed *quasi-static objects*).[3] In contrast to our work, Sanders focuses on object segmentation without concern for what the object is, whereas we wish to observe the human's actions to infer both the object's location and label.

Other researchers have begun using human activity to reason about a scene's contents. Applications include finding paths in outdoors scenes, either to detect unusual behavior or to determine the extent of pathways and obstacles in the scene.[4,5] Similarly, Kimberle Koile and her colleagues accumulated evidence of human activity in a scene and used this evidence to map heavily used areas, or *activity zones*.[6] They were limited, however, to manually creating descriptive labels for each zone. In an attempt to use action recognition to assist in object labeling, Darnell Moore and his colleagues tracked human-hand movements as they interacted with an object to refine its initial shape-based object recognition classification.[7] To do this, they worked with top-down, close-up views of office desks monitored by a camera. Although successful (and incorporated into the AwareHome project), the method has limited deployment opportunities in a smart home for three reasons:

- It requires uniplanar scenes (such as a desk's flat surface).
- It relies on an initial shape-based object classification.
- It requires very close-up views.

These factors constrain its potential deployment to household areas that are fixed and that experience significant, cohesive activity within a small area (such as dining tables or kitchen sinks).

REFERENCES

1. R.J. Campbell and P.J. Flynn, "A Survey of Free-Form Object Representation and Recognition Techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, 2001, pp. 166–210.
2. L. Stark and K. Bowyer, "Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, 1991, pp. 1097–1104.
3. B.C. Sanders, R.C. Nelson, and R. Sukthankar, "A Theory of the Quasi-Static World," *Proc. IEEE Int'l Conf. Pattern Recognition* (ICPR 02), IEEE CS Press, 2002, pp. 1–6.
4. W.E.L. Grimson et al., "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 98), IEEE CS Press, 1998, pp. 22–29.
5. D. Makris and T. Ellis, "Finding Paths in Video Sequences," *Proc. British Machine Vision Conf.*, British Machine Vision Assoc., 2001, pp. 263–272.
6. K. Koile et al., "Activity Zones for Context-Aware Computing," *Proc. 5th Int'l Conf. Ubiquitous Computing* (UbiComp 03), Springer-Verlag, 2003, pp. 90–106.
7. D.J. Moore, I.A. Essa, and M.H. Hayes, "Exploiting Human Actions and Object Context for Recognition Tasks," *Proc. IEEE Int'l Conf. Computer Vision* (ICCV 99), vol. 1, IEEE CS Press, 1999, pp. 80–86.

over time, which makes the system adaptable as the scene changes.

To demonstrate our approach's potential, we used a moving person's bounding-box statistics to recognize the signatures for that person's interactions with chairs and floors (that is, sitting and walking). Although our system's features are currently too coarse to detect more subtle interaction motions with smaller objects (such as cups or telephones), it can successfully label chairs and floors using standard action recognition algorithms. The system can also adapt its labeling after a person relocates a chair within the scene.

Interaction signatures: An overview

The interaction signatures approach involves four basic steps: foreground object segmentation and tracking, object relocation detection, action segmentation, and scene object labeling (see Figure 1). Our system can also refine the basic labeling by using higher-level signatures, such as partial occlusions of people and object transference.

Foreground object segmentation and tracking

We use background subtraction to segment target objects from the video stream, using a mixture of Gaussian distributions to model the background.[3] We chose this background model because it can robustly adapt to background definition changes over time, which is essential for our research. The system segments out foreground objects—that is, people—from the background, outlines them with a bounding box, and tracks them using a Kalman filter. Background segmentation

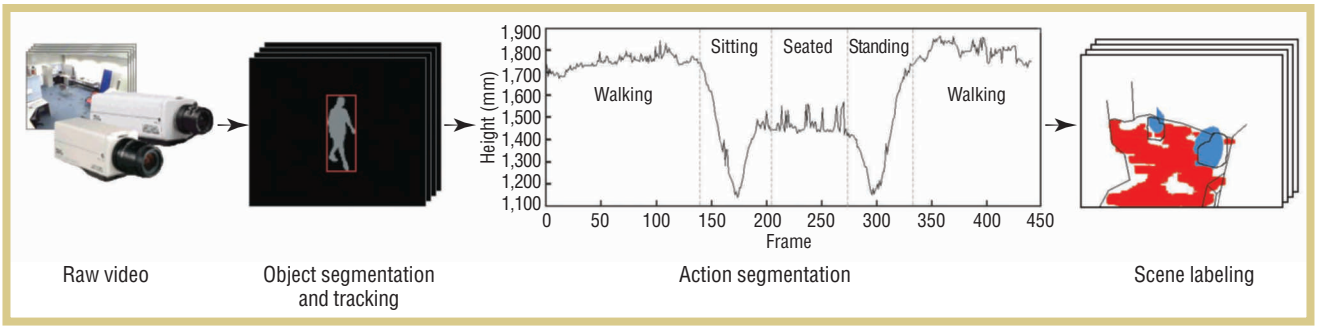


Figure 1. The four major steps in interaction signature scene labeling. The system captures video from the four ceiling-mounted cameras, saves it to disk, and processes the video offline to segment and track objects. Using that raw data, it then segments the action and incrementally labels the scenes, ultimately producing a labeled map of each view (chairs are in blue, floors in red).

can often fail to perfectly segment the person, but because we measure only coarse features such as height and width, only major segmentation failures significantly affect the measurements. (Our previous research offers more details on our tracker and test environment.[4])

The system precalibrates each view to the world coordinate system using a set of landmark points. It then finds correspondences between different views of a person by his or her proximity in the world coordinate system (assuming that the person is standing on the ground plane). Additionally, we detect partial occlusions of people by comparing the world heights and the person's positions in all views. If the person's lower portion is occluded in one view, the system reports a smaller height than the other views, indicating occlusion by an object.

Object relocation detection

To detect scene objects' addition or removal, we rely on the fact that household objects generally don't move without human intervention—making them what Sanders calls *quasi-static* objects.[5]

Adding a new object leads to a new foreground blob suddenly appearing in the scene. Because a quasi-static object doesn't move on its own, the system infers that the new blob relates to a new, unknown object at that location. The system eliminates any existing labels in the area to reflect that a new object is there, then adds the blob to the statistical background so that it no longer appears in the foreground.

Removal of objects presents an additional complexity: when an object is removed, it leaves a "ghost" in its previous location. This occurs because when an object is removed, the scene behind it no longer matches the background (which the system learned with the object in place). This causes a foreground blob to appear, even though no physical object is there (hence the term "ghost"). Fortunately, the ghost's color generally matches its surroundings, because now no object is in the way. We use this to distinguish *removed* objects (which produce matching blob colors) from *introduced* objects (in which the blob doesn't match its surroundings). However, the system doesn't make the connection that an object's removal and addition to another area indicates a transfer—currently, the system assumes that two different objects exist.

Action segmentation

Given hidden Markov models' proven aptitude in modeling human motion,[6,7] we trained one HMM for each interaction signature using features extracted from the video. We modeled four actions:

- Walking
- Sitting down into a chair
- Being seated on a chair
- Standing up from a chair

Training data consisted of six examples with four views per example (24 sequences total) of a person walking into a room, sitting down, standing back up, and leaving the room. For each sequence, we positioned the chair at different orientations and positions in the room. We then manually segmented each sequence into the four constituent actions and used them to train the HMMs. We used four training features:

- Real-world height (in millimeters)
- The height change between consecutive frames (expressed as a proportion of the total height) to minimize dependency on object height
- Change in width, expressed as a proportion of the total width
- The object's ground speed (absolute velocity, in mm/frame)

On the basis of the HMMs, the system automatically segments test video sequences into individual action blocks that relate to a particular interaction signature. (We opted for automated rather than manual segmentation to demonstrate that our proposed evidence-based labeling is robust to noisy action segmentation.) We use a simple sliding-window approach[6] in which the system segments sequences by considering only the frames that fall within a fixed-sized moving window. For our actions, a single window size of 30 frames provided the best results. The system uses the features from the frames within this window to calculate each HMM's log likelihood, selecting the most likely HMM as

the frame block's action label. The system then estimates that the selected action began halfway in the window (because an HMM dominates the previous action's HMM when at least one-half of the window frames relate to the new action). The system then moves the window one frame forward and repeats the entire process.

However, sliding windows tend to produce short bursts of incorrect action labeling because an incorrectly classifying HMM can become temporarily more probable than the correctly classifying HMM. This might be due to background subtraction failures, occlusions, or other random factors. To solve this, we introduced a heuristic confidence test on the HMM log likelihoods; it mandates that the most likely HMM must significantly outperform the next most likely model. To decide this, the system calculates the ratio between the highest and second-highest HMM log likelihoods, using an arbitrary threshold (currently 0.75) to define the "significant" difference. If it finds no significant HMM, it reinstates the last significant action.

Each view performs action segmentation independently of other views. Then, to further improve segmentation, each view casts an equally weighted vote as to the action being performed (in the case of a deadlock, the system reinstates the current model). All views use the elected action to label scenes (again, independently). Although we could fuse each camera's features into a single, corresponded set, the features we measure are not fine-grained enough to benefit from such a fusion. In fact, the voting mechanism will generally obtain better results because it's essentially a form of *bagging* (where combining several classifiers together results in a more reliable classification).

Scene object labeling

We label objects in the scene by taking each of an action block's frames and updating the view on the basis of the action and the person's position in the scene view. We do this by maintaining a weight for each label (chair or floor) for every pixel in a view's background image. The weights lie within the range 0 to 1 and are initialized to 0. When the system updates a pixel, it updates all weights using the exponential-forgetting function

$$w^{L}_{t+1(x,y)} = w^{L}_{t(x,y)}(1-\eta) + (\eta \cdot \kappa),$$
$$\kappa \begin{cases} 1 & \text{if } L = \text{detected object}, \\ 0 & \text{otherwise} \end{cases}$$

in which

- w is the weight of the Lth label (chair or floor) at time t and pixel (x, y).
- η is the learning rate for learning labels and is generally very small (less than 0.05) to avoid building up weights too quickly.
- κ is the update value that controls which label the system will strengthen.

This function ensures that as the system observes new evidence, it views older evidence as increasingly less important. Also, the system quickly resolves evidence conflicts (such as a single pixel having similar evidence for both chair and floor labels) because rival labels are decayed when new observations occur that support one label over the others.

In keeping with the interaction signature concept, the system should label a particular object whenever it observes that object's associated action. So, the system labels chairs whenever it detects the sitting, seated, or standing-up actions. The system uses the seated person's fitted ellipse, rather than the bounding box, as the labeling area because it more closely matches the person's silhouette (and, by implication, is closer to the chair's area). The system labels floor space when the walking action occurs, with the heuristic that it labels only the fitted ellipse's lowest five percent of floor space, which generally corresponds to the person's feet.

High-level labeling constraints

In addition to the basic action-based labeling, we can affect labels by detecting higher-level interaction signatures that aren't specific to any particular object type, including partial occlusions of the person and object transference within the scene. We can use partial occlusions of a person who walks on a chair's far side, for example, to refine chair labels and affect the system's future chair-label learning in the area. When the chair occludes the person's legs, we can infer that the chair doesn't extend into the unoccluded area and thus remove all chair labels and slow the future labeling rate in the unoccluded area. If the occluding object isn't labeled (such as a table or an unlabeled chair), there are no labels to refine. However, the system now has evidence that *some* object is there and will slow future labeling in the unoccluded area. Slowing the labeling rate ensures that the system doesn't quickly reinstate incorrect chair labels. We heuristically defined the retardation rate as linear with respect to the number of times the system observes occlusions in the region—more instances mean a slower relearning rate. We don't reduce the learning rate to zero because we still want the system to recover from mistakes in defining the partial-occlusion area. Although we use the chair example here, these responses to partial occlusion are applicable to almost any object. The response isn't applicable to floors, however, because the floor can't occlude the person.

Another type of high-level interaction signature is detecting when a person relocates an object in the scene. When this occurs, we must destroy the labels that are no longer valid. If the system

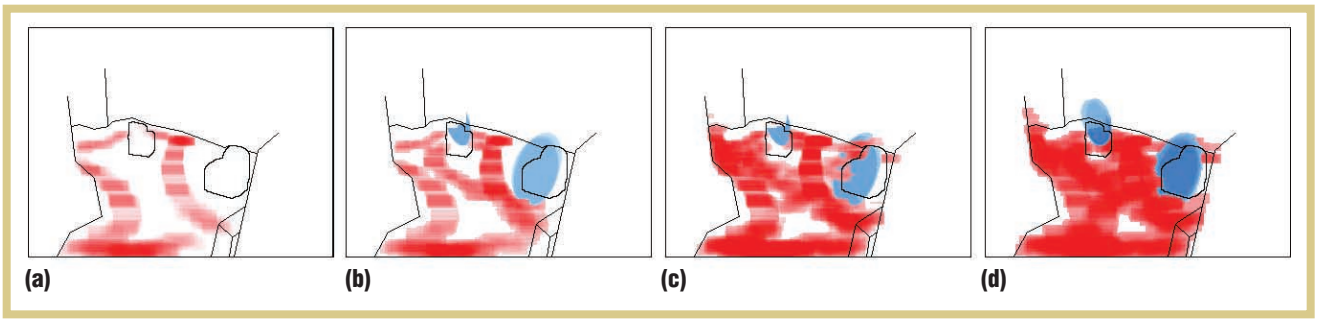


Figure 2. A sequence of images showing the progression of floor (red) and chair (blue) labeling for one run: (a) the person's initial walk, (b) the person has sat once in each chair, (c) more walking and sitting, and (d) final scene labeling. Outlines indicate chair, floor, and obstacle boundaries. Intensity indicates the label's weight, which strengthens as more evidence accumulates.

detects that a chair has been moved from an area, for example, we remove all the area's chair labels to reflect that the chair no longer exists. Similarly, if the system detects that a chair has been added to an area, we remove all the area's nonchair (that is, floor) labels because the chair now occludes the floor and chair labels must take priority.

Results and analysis

To produce a labeled scene, our system performs its major operations sequentially (see Figure 1). First, the system captures video at 25 frames per second from four ceiling-mounted cameras monitoring the scene (a laboratory). The system then saves the captured video to disk in MPEG-4 format, which it processes offline to segment and track objects. Then, in a separate process, it uses this raw data for action segmentation and scene labeling, producing a labeled scene image from all four camera views (see Figure 2).

We tested labeling accuracy using three video sequences, each about one to two minutes long and comprising four views of a person alternately moving about and sitting down in the target chairs. The chairs remained in fixed positions throughout the experiments. The system performed action segmentation and scene labeling on each sequence to produce three sets of four labeled images (one image for each scene view—see Figure 3).

Finally, we conducted a second set of experiments to evaluate the system's effectiveness in dealing with a person moving a chair around the scene. For this, we took three video sequences in which a person moved around the scene, repeatedly sitting in and relocating a chair.

Action segmentation

We estimated a ground truth for each action's starting frame by manually determining each action's start and end times. The ground truth's uncertainty is roughly ±5 frames, although this judgment is subjective. Table 1 shows the difference (in frames) between the ground truth and the automatic action segmentation, indicating how noisy the segmentation was and whether the subsequent labeling process had a reasonable chance of success. The system segmented the walk and sit actions quite accurately (mean error of –2.94 and 5.15, respectively), especially given the ground truth uncertainty of ±5 frames. Also, it generally segmented the sit action slightly later than the actual sit

Figure 3. Floor (red) and chair (blue) labels for all views of the same test run, thresholded to remove weak labels. The four views are (a) northwest, (b) northeast, (c) southwest, and (d) southeast. Edges show manually defined ground truth for chairs, floors, and occluding objects in the scene. Floor labeling is reasonably adept at detecting edges of occluding objects, such as walls and chairs. The northwest view (a) and southeast view (d) show how occlusion assists the system's attempt to find chair boundaries.

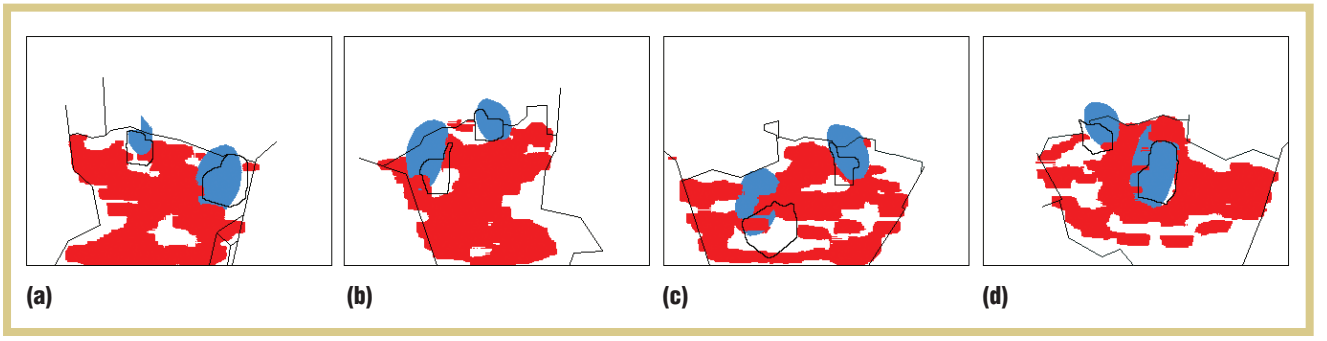


TABLE 1

Error means and variances for action segmentation.

Model	Instances	Found instances	Mean error (frames)	Error variance (frames)
Walk	19	19	–2.94	24.56
Sit	19	19	5.15	48.31
Seated	19	2	0	8.00
Stand	19	19	–50.24	726.32

TABLE 2

Confusion matrices for chair labeling.

	Pixels classified as			
	Chair	Floor	Other*	Recall
Chair	18,127	3,068	5,083	69%
Floor	11,503	168,320	72,435	67%
Other	7,313	7,686	628,065	98%
Precision	49%	94%	89%	

*"Other" relates to unlabeled pixels where no significant action occurred. (Table 3 shows the confusion matrix when "Other" isn't taken into account.)

action but segmented the walk action slightly earlier. This means that the system conservatively estimated the chair interactions' beginning and end.

The data's most worrisome aspects are that the system detected only two of 19 seated actions and generally detected the start of stand actions far too early. In fact, the two failures are related—the last third of the sit action's motion profile looks strikingly similar to the first third of the stand action (see, for example, Figure 1's "Action segmentation"). This resulted in the stand action being prematurely dominant when the seated action actually began. The system didn't correct this mistake over the next few frames because our log-likelihood confidence threshold prevented the seated model from replacing the stand model. Fortunately, given our evidence-accumulation framework, losing the seated action label merely gives less evidence for the chair labeling. This is easily offset when the system observes more instances of a person sitting in the chair.

The coarseness of the system's human-actor measurements (bounding-box statistics and speed) severely limited segmentation accuracy. The system's failure to find seated action was a direct consequence of these simple features. Selecting better features would improve segmentation accuracy and let us classify more interaction signatures, such as drinking from a cup. We could use pose estimation, where techniques include skeletonization[8] or model-based methods.[9] To further improve the results, we could use more sophisticated segmentation techniques, such as referring to the HMM's Viterbi state sequence.[7] That said, the segmentation results are adequate for our purposes.

Scene-labeling accuracy

We evaluated chair labeling by comparing the area labeled "chair" against the chair's true extent in each view (a chair's extent includes the space between its legs). Table 2 shows these results.

For chair labeling, the system achieved a 69 percent recall rate (that is, it correctly labeled 69 percent of the total chair area across all views). So, it's evident that chair labeling manages to locate chairs fairly successfully; the system found nearly seven of 10 chair pixels. The inaccuracies are mostly because the system uses the seated person to produce the labels, and the person is almost always offset slightly from the chair itself because people sit *on* chairs rather than *within* them (see Figure 3).

To measure how closely our chair labeling fit within chair boundaries, we must refer to precision. Even though the 49 percent precision value seems quite low (about one-half of the chair labels were outside the chairs), it isn't unexpected because the seated person's extent is nearly always larger than the chair itself. For example, the person's head and shoulders are almost always higher than the chair's back. Also, the person's offset from the chair further degrades labeling precision.

Precision benefits significantly from using occlusion to localize the chair's extent. Occlusion is effective because it's particularly useful in detecting and reducing one of overlabeling's primary causes—that a person's head and shoulders rise above the chair's back. Unfortunately, we couldn't fully exploit this fact because our experiments contained limited occlusions. In chair views that experienced occlusions, the system had fairly high precision (70 percent) compared to its 49 percent overall precision.

Although Table 2 shows that floor labeling was extremely precise (94 percent), this is misleading; the open-floor space extends over a large proportion of the view. Equally misleading is the floor's recall figure, which seems quite low (67 percent). The system failed here because the person didn't walk over some portions of open floor during the experiments, so gaps exist in the coverage and adversely affected the recall. Without the "Other" labels, floor recall improved markedly—from 67 percent to 94 percent (see Table 3). Given these issues, we didn't analyze floor labeling numerically. Instead, we limited our floor label evaluation to visually inspecting the labeled

TABLE 3
Chair labeling results without the "Other" category.

	Pixels classified as		
	Chair	Floor	Recall
Chair	18,127	3,068	85%
Floor	11,503	168,320	94%
Precision	61%	98%	

images for floor labels that incorrectly spilled into chair areas or occluded walls and partitions (see Figure 3). Overall, the floor labeling detects occluding edges reasonably well, with only minimal overflow. Overlabeling into chair spaces was also minimal, owing to both the floor labeling's success and the fact that chair labels tend to overpower the floor labels.

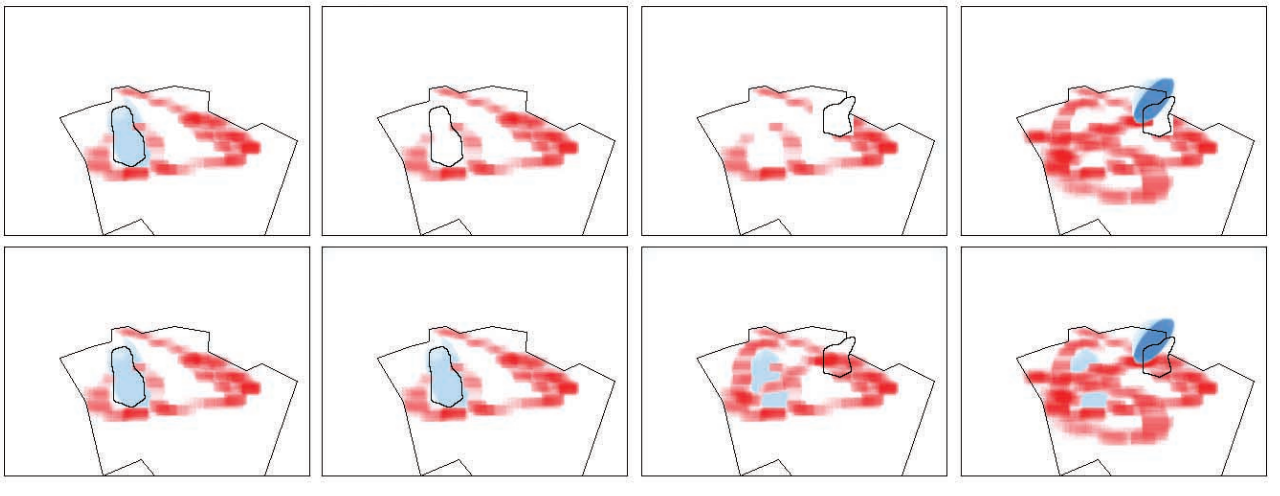
Handling chair relocation

To demonstrate interaction signatures' possibilities for object-relocation handling, we performed an additional experiment that examined how moving chairs around the scene affected labeling. Because we assume that only chairs are transferable, we erased only chair labels (not floors) at the chair's former location. If we dealt with more object types, we'd erase the strongest-weighted label type.

Figure 4 shows a progressive labeling example for one view, comparing the effect on labeling with and without object-relocation detection. Figure 4a shows the initial position. As Figure 4b shows, as soon as the person picked up the chair, the system immediately recognized the action and removed all chair-related pixel labels. Conversely, in the lower image, in which we'd disabled pick-up detection, the labels were left unchanged. Similarly, when the person put the chair down (see Figure 4c), the system removed the floor labels from the chair's new location. Figure 4d indicates the benefits of object-relocation detection—in the lower image, a chair still appears to be in the original position, and even the new position has a large component labeled "floor space." No such problems affect the upper image.

Object-relocation detection isn't foolproof, however. If we consider all four views independently, the system detected object-relocation events only 70 percent of the time—finding only 34 of the 48 events (four views of 12 physical events). Fortunately, because the system nearly always detects a relocation event in at least one view, we can correctly update views that fail to detect that event. Still, in some cases this cross-view fusion won't work, often because the person blocks one view of the relocation event for too long, making the view out of sync with the other views. So, when we used fusion, system accuracy improved to 42 of 48 (87 percent), but it still missed six events. In the missed cases, the system can recover somewhat because accumulated evidence will tend to erase incorrect chair labels over time (see Figure 4d).

Figure 4. A chair relocation sequence. The upper row shows how detecting chair relocations affects labeling; the lower shows labeling with relocation detection disabled. (a) the initial position; (b) the chair being picked up; (c) the chair being put down; (d) the final labeling.



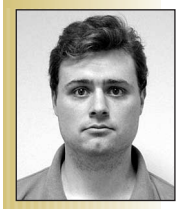
While this is a fairly early investigation into using interaction signatures for action-centered object labeling, it's encouraging that we've obtained reasonable results with crude measurements and that the method can adapt to object relocation within the scene. Given that we've dealt only with chairs and floors, however, it would be premature to judge our approach's general effectiveness. Moreover, the labeling process uses too many heuristics and thresholds, which we must eliminate to make the system more robust and portable to new environments.

To address this, we intend to improve our research along multiple paths. First, we plan to refine human foreground blob measurements to reveal more interesting information, such as where the person's limbs are. This will offer several benefits, including the ability to handle more complex interactions and extend our limited object range.

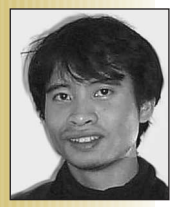
Also, we have not considered the scene image itself at all—only the human's motion. Even simple image segmentation techniques that divide the image into similarly colored areas would offer a wealth of information and let us move from pixel-level labeling to region-level labeling. This should significantly improve labeling accuracy by using the image segmentation as secondary evidence in determining an object's boundary.

Finally, we could use complementary sensors such as microphones to provide further evidence for interaction signature recognition. Our approach has definite limits, however. As we mentioned before, our system can never detect objects that people don't interact with, such as walls and ceilings. In addition, the system would have difficulty detecting objects such as tables, because humans don't normally interact with a table directly but rather with the objects on top of it.

the AUTHORS



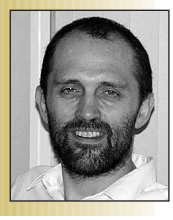
Patrick Peursum is a PhD student in the Curtin University of Technology's Department of Computing . His research interests include computer vision, human-motion analysis, and machine learning. He received his BSc (with honors) in computer science from the Curtin University of Technology. Contact him at the Dept. of Computing, Curtin Univ. of Technology, GPO Box U1987 Perth, Western Australia, 6845, Australia; peursump@cs.curtin.edu.au.



Hung Hai Bui is a researcher at SRI International's Artificial Intelligence Center. Previously, he was a lecturer at the Curtin University of Technology's Department of Computer Science. His research interests are activity recognition, probabilistic reasoning, and machine learning. He received his PhD in computer science from the Curtin University of Technology. Contact him at the Artificial Intelligence Center, SRI Int'l, 333 Ravenswood Ave., Menlo Park, CA 94025; bui@ai.sri.com.



Svetha Venkatesh is a professor at the Curtin University of Technology's Department of Computing. Her research interests are large-scale pattern recognition, image understanding, and applications of computer vision to image and video indexing and retrieval. She codirects the university's Centre of Excellence in Intelligent Operations Management and its Institute of Multi-Sensor Processing and Content Analysis. Contact her at the Dept. of Computing, Curtin Univ. of Technology, GPO Box U1987 Perth, Western Australia, 6845 Australia; svetha@cs.curtin.edu.au.



Geoff A.W. West is a professor of computer science at the Curtin University of Technology. His research interests include 3D object recognition, feature extraction, surveillance, smart homes, pervasive computing, small-scale systems, automatic visual inspection, data mining, telemedicine, and related applications. He received a PhD in systems engineering from City University, London. He's a senior member of the IEEE, a member of the IEE (UK), a fellow of the Institution of Engineers, Australia, and a Chartered Engineer. Contact him at the Dept. of Computing, Curtin Univ. of Technology, GPO Box U1987 Perth, Western Australia, 6845, Australia; geoff@cs.curtin.edu.au.

REFERENCES

1. D. Ayers and M. Shah, "Monitoring Human Behavior from Video Taken in an Office Environment," *Image and Vision Computing*, vol. 19, no. 12, 2001, pp. 833–846.
2. D.J. Moore, I.A. Essa, and M.H. Hayes, "Exploiting Human Actions and Object Context for Recognition Tasks," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, IEEE CS Press, 1999, pp. 80–86.
3. C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 747–757.
4. P. Peursum et al., "Object Labeling from Human Action Recognition," *Proc. IEEE Int'l Conf. Pervasive Computing and Communications*, IEEE CS Press, 2003, pp. 399–406.
5. B.C. Sanders, R.C. Nelson, and R. Sukthankar, "A Theory of the Quasi-Static World," *Proc. 16th Int'l Conf. Pattern Recognition*, IEEE CS Press, 2002, pp. 1–6.
6. A.F. Bobick and Y.A. Ivanov, "Action Recognition Using Probabilistic Parsing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, 1998, pp. 196–202.
7. M. Brand and V. Kettnaker, "Discovery and Segmentation of Activities in Video," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000, pp. 844–851.
8. H. Fujiyoshi and A. Lipton, "Real-Time Human Motion Analysis by Image Skeletonization," *Proc. IEEE Workshop Application of Computer Vision*, IEEE CS Press, 1999, pp. 15–21.
9. M.W. Lee, I. Cohen, and S.K. Jung, "Particle Filter with Analytical Inference for Human Body Tracking," *Proc. IEEE Workshop Motion and Video Computing*, IEEE CS Press, 2002, pp. 159–165.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.