Simple and Robust Tracking of Hands and Objects for Video-based Multimedia Production

Masatsugu ITOH Motoyuki OZEKI Yuichi NAKAMURA Yuichi OHTA
Institute of Engineering Mechanics and Systems
University of Tsukuba
1-1-1 Ten'noudai, Tsukuba
Ibaraki 305-8573 JAPAN
ozeki@image.esys.tsukuba.ac.jp

Abstract

We propose a simple and robust method for detecting hands and hand-held objects involved in desktop manipulation and its use for indexing the videos. In order to achieve robust tracking with few constraints, we use multiple image sensors, that is, a RGB camera, a stereo camera, and an infrared (IR) camera. By integrating these sensors, our system realized robust tracking without prior knowledge of an object, even if there was movement whether of people or objects in the background. We experimentally verified the object tracking performance and evaluated the effectiveness of integration.

1 Introduction

There is now great demand for video-based multimedia contents and the indexing of those contents. Automating video production will contribute greatly to education, professional training, and lifelong learning programs. For one effective approach to this automation, we have been developing an *intelligent video production system*, as shown in Figure 1[2], which automates video capturing, editing, and indexing. The target of our system is to aid in the production of video-based teaching/operating/instruction manuals.

For those types of videos, the objects appearing in a scene are usually important. For example, in the assembly of a machine, the parts being assembled are the most important factors. Thus, automatic object detection and tracking can add useful indices to the video, in this case by providing links for further information about object. Furthermore, the detection of typical human manipulations of objects can be useful in automatic structuring of video content.

For this purpose, we propose a method for tracking objects that is simple, robust, and relatively free of constraints. Our system uses multiple image sensors, consisting of an RGB camera, a stereo camera, and an IR (infrared) camera, that together track a hand-held object at video rate even if the background contains movement of people or objects, even including skin-color objects.

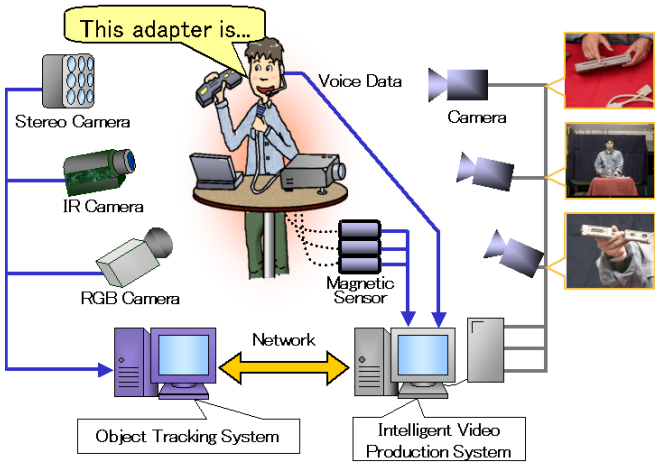


Figure 1. Intelligent video production system

2 Object Tracking and Video Indexing

2.1 Object Tracking Condition

The proposed method utilizes multiple image sensors: an ordinary RGB camera, an IR camera, and a stereo camera. The following explains how we use those sensors to achieve robust tracking.

A typical situation to which we apply our system is as follows:

1. A person holds or points an object in front of his/her body as shown in Figure 1.
2. Referring to the object, the person mentions its name or how to use it; *e.g.*, "This adapter has a connector to attach to a monitor cable".
3. The person manipulates certain parts, *e.g.*, attaches, detaches, or otherwise handles to demonstrate demonstrate how the object works or how, for example, the parts fit together.

In such a case, it is difficult to provide a complete appearance model of an object, since its appearance can be easily altered by rotation, deformation, or assembly. Moreover, it is natural that the background may change as other people or objects move in the scene.

Thus, we consider object tracking under the following conditions:

- The system has no prior knowledge of an object's size, color, texture, etc.
- The background may change at any time during manipulation.

On the other hand, we can naturally make the following assumptions that make the conditions easier to achieve:

- Most of the important objects are moved or manipulated by human hands.
- The space (volume) in which important objects are likely to appear is known on the condition that the work space *e.g.*, a workbench is stationary.

Even with these assumptions, the conditions presented above are still severe. Although a number of studies have been investigated hand tracking or object tracking, some of which have reported good results, our situation is much more difficult than the situations assumed in those studies. Object rotation or occlusion caused by grasping can easily alter the object's texture, and people moving in the background add serious noise that cannot be easily eliminated.

2.2 Indexing Desktop Manipulations

Through object tracking and human behavior recognition, we can obtain useful information: when, where, why, and how objects are moved, when and what kind of explanation is given for those objects, etc.

We can effectively use this information for video indexing. For example, if we detect a situation in which a person is explaining an important object, we can add a clickable icon that links the scene to a captured video clip to supplement what are learning about the object.

3 Using Multiple Image Sensors

Figure 2 shows an overview of the system. The system detects the following regions based on information obtained by the three sensors:

RGB camera: the regions showing the color of skin and regions that are moving.

Infrared camera: the *skin-temperature* regions (*i.e.*, regions in which a temperature of around 34°C.

Stereo camera: the *in-volume* regions, which are regions in the workspace area.

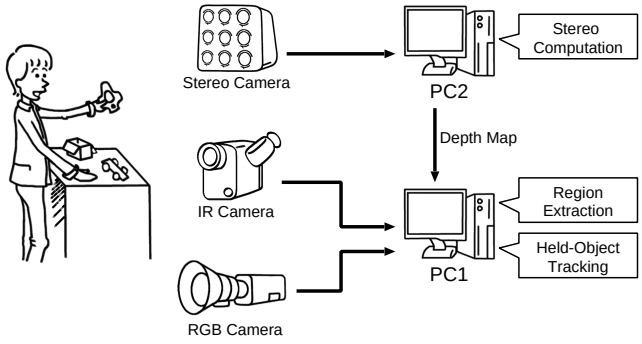


Figure 2. System overview

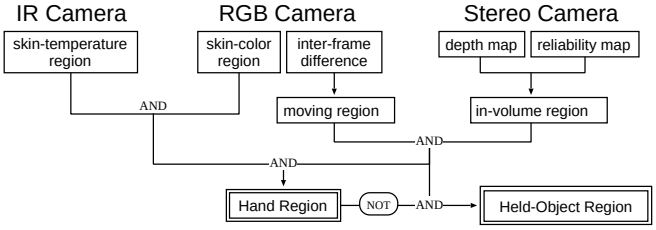


Figure 3. Region detection and integration

By integrating all of these region types, the regions showing the hands and the held objects are detected based on the following principle.

$$\begin{aligned}\text{hand region} &= \text{in-volume region} \wedge \text{moving region} \\ &\wedge \text{skin-temperature region} \wedge \text{skin-color region} \quad (1)\\ \text{held-object region} &= \text{in-volume region} \\ &\wedge \text{moving region} \wedge \neg \text{hand region} \quad (2)\end{aligned}$$

Figure 3 shows an outline of the above process. Once a held-object region is extracted, we can register the object's texture, *i.e.*, its appearance. This texture can be used to detect the object after the person releases it.

4 Process for Each Image Sensor

Before describing the whole system, we discuss problems encountered in model construction, parameter tuning for each sensor, and sensor integration.

4.1 Process for RGB Camera

We created a skin-color model by gathering the statistics regarding pixel values showing skin color, and determined their distribution parameters. This method is based on Kondou's research [1], which determined that the distribution of

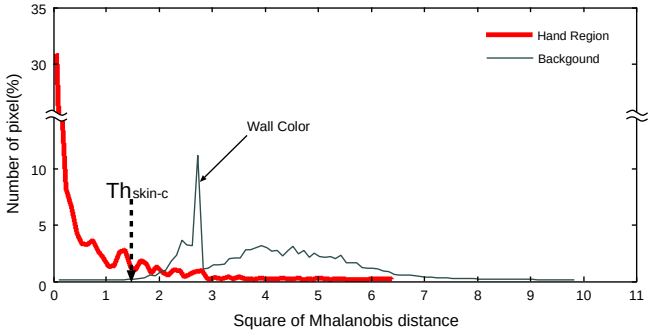


Figure 4. $D^2(r, g)$ **for skin color model**

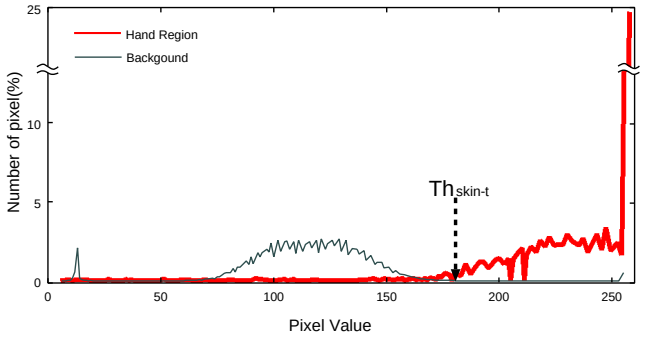


Figure 5. Pixel values from IR camera

Japanese face color taken from TV broadcasts is compactly modeled on the rg-plane[1].

First, the skin color regions are manually extracted, and the mean value and the covariance matrix Σ are calculated. Their actual values are as follows:

$$mean(\bar{r}, \bar{g}) = (0.437773, 0.334845)$$
$$\Sigma = \begin{pmatrix} 0.003915 & -0.000230 \\ -0.000230 & 0.000935 \end{pmatrix}$$

The square of Mahalanobis distance $D^2(r, g)$ from skin color is calculated, and from this skin color regions are extracted.

$$D^2(r, g) = \begin{pmatrix} r - \bar{r} \\ g - \bar{g} \end{pmatrix}^T \Sigma^{-1} \begin{pmatrix} r - \bar{r} \\ g - \bar{g} \end{pmatrix}$$

The graph in Figure 4 shows the statistics obtained from a typical image in our environment. $D^2(r, g)$ values in actual skin regions and those in the background are plotted. Based on those statistics, we determined a threshold value of $Th_{\text{skin-c}} = 1.5$.

Moving regions are delineated by using interframe subtraction of every fourth frame. Through our experiments, threshold Th_{move} of around 30 showed good performance.

4.2 Process for Infrared Camera

Our IR camera captures infrared light with a wavelength between 8 and 12μm, which covers the dominant wavelength that the human body emits. We checked the pixel values in the real hand region and those in a typical background, and determined the threshold for extracting the skin temperature region.

In our experiments, a threshold $Th_{\text{skin-t}}$ of around 180 well separates those regions, as shown by the graph in Figure 5. Since the actual pixel value depends on the iris, focus, and other camera parameters, the threshold must be adjusted if those parameters are changed.

4.3 Process for Stereo Camera

As mentioned above, we assume that the space (volume) where hands and related objects appear is known to the system. In our experiments, we assumed that the width, height, and depth of the volume are 2.5m, 2m, and 0.5m, respectively. These can be changed according to the spatial arrangement of the workspace and the camera position.

Objects in this volume can be detected by using the depth map obtained by the stereo camera. A problem in this step concerns the noise caused by homogeneous regions, periodic textures, and occlusion. In order to overcome this problem, the reliability map provided by the stereo camera is used. The sharpness of the peak in the disparity computation is evaluated; the sharper the peak, the larger the value of the reliability map[4].

For each pixel, the system uses the depth value only if its reliability is higher than the threshold. This simple operation works well for typical indoor scenes. Since the latency of the stereo computation is a few frames, we need the ability to synchronize the image with the other images that can be obtained with much less latency. This is described in the next section.

5 Integration for Multiple Image Sensors

Prior to actual region extraction and tracking, we need geometric compensation and synchronization among the three images.

5.1 Geometric Compensation

A calibration board is placed on the worktable. Markers visible to all cameras are attached to the board. Based on the markers' locations, the projection parameters that map the IR image or the depth map to the RGB image are computed using the following quadratic model.

$$\begin{pmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ 1 & \dots & 1 \end{pmatrix} = M_{3\times 5} \begin{pmatrix} u_1 & \dots & u_n \\ {u_1}^2 & \dots & {u_n}^2 \\ v_1 & \dots & v_n \\ {v_1}^2 & \dots & {v_n}^2 \\ 1 & \dots & 1 \end{pmatrix}$$

where (x_i, y_i) represents the marker position in the RGB image, and (u_i, v_i) is the marker position in the IR image or in the depth map. Although the IR camera has heavy radial distortion, 25 markers are sufficient to calculate the above parameters.

[1] A normalized color space. $r \equiv \frac{R}{R+G+B}$, $g \equiv \frac{G}{R+G+B}$.

5.2 Synchronization

As shown in Figure 2, images from the RGB camera and images from the IR camera are captured in PC1, and images from the stereo camera are captured in PC2. The depth map images stored in PC2 are transmitted through the Ethernet to PC1, which then executes region extraction and integration process. To compensate for the latency of stereo computation and transmission time, the captured time is attached to each image. The depth map image captured at the nearest time is used with the other two images.

5.3 Region Detection and Tracking

As shown in Figure 3, the hand regions are detected by taking logical AND operations of the four regions as shown in equation (1). The extracted hand region candidates are labeled after region expansion-contraction. Then, at most two regions whose areas are larger than the threshold are registered as hand regions.

The held-object regions are detected according to the idea shown in formula (2). In this case, however, the first "$\wedge$" does not simply mean the logical AND operation. Held-object region candidates are detected by calculating the ratio of the pixels in an in-volume region that are included in any moving regions. An in-volume region whose ratio exceeds the threshold is detected as a held-object region candidate. These candidates are labeled after region expansion-contraction, as well the case with the hand regions described above. The final position of the object is estimated through the use of the Kalman filter for position smoothing. By repeating this process at video rate, the estimated position of a held object is obtained in every frame.

The detected regions are shown in Figure 6, and examples of the tracking results are shown in Figure 7. As the figures show, the held object is well detected and tracked even when the intermediate result from each sensor contains much noise. Figure 6 suggests, the skin-color region detection, which is often used for detecting hands, is not satisfactory. By combining the depth and temperature information, we can easily identify for deletion the region with the skin-color box and the region containing the moving person.

6 Detecting Operations for Objects

Important operations in desktop manipulations are detected by integrating information from our object tracking system and intelligent video production system, as mentioned in Section 1. These systems obtain the following data:

our object tracking system: position of each hand-held object, distance between objects, texture of an object, etc.

intelligent video production system: typical behaviors for explaining objects (*e.g.*, holding out, illustration/demonstration, etc.), annotation to an object, video clips

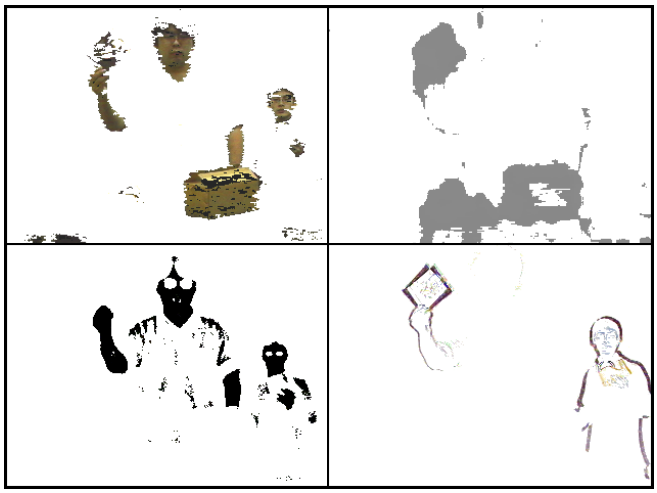


Figure 6. Detected regions (upper left: skin-color region, lower left: skin-temperature region, upper right: in-volume region, lower right: moving region)

Table 1. Operations

operation	number	distance	recognized behavior
present	1	don't care	holding-out
detach	1 →2	come apart	illustration/ demonstration
attach	2 →1	get together	illustration/ demonstration

This information allows the system to detect and identify important operation involving an object, that is, presenting, detaching, and attaching: presenting is a behavior in which an object is explained; detaching or attaching is an operation involving two or more objects. These operations are basically determined by the following clues, as shown in Table 1: the number of objects in a hand or hands and the change in that number, the distance between the objects, and the behavior detected by the intelligent video production system.

7 Experimental Results

7.1 Tracking

Table 2 and Table 3 show the specifications of the PCs and three image sensors, respectively. As shown in Figure 7, we evaluated our system's performance in two situations. Scene A simply contains one person holding and moving an object. Scene B is a more complicated, as it contains multiple objects on the worktable and includes another person walking in the background.

The correct regions were detected in 97% and 93% of the frames in scene A and scene B, respectively. The tracking in Scene B had a slightly higher failure rate because the skin-color box and the walking person in the background created

Table 2. PCs

PC	CPU	RAM	OS
PC1	Xeon 2.2GHz	RDRAM 1GB	Linux kernel2.4
PC2	P3 933MHz	RDRAM 256MB	Linux kernel2.2

Table 3. Image sensors

Sensor	Name	Output image	Vendor
RGB	DFW-VL500	320×240 30Hz	Sony
IR	IR-30	320×240 30Hz	Nihon Avionics
Stereo	FZ-930	280×200 30Hz	Komatsu

Table 4. Detection and tracking performance

	Scene A	Scene B
Total	1350 frames	1350 frames
Detection failure	30 frames (2.2%)	11 frames (0.8%)
Tracking failure	4 frames (0.3%)	80 frames (5.9%)

misleading regions. Nonetheless, this failure rate, which was less than 6%, would be difficult to achieve using a single image sensor.

7.2 Detecting Operations

As shown in Figure 8, we evaluated our system by using toy parts. Figure 8 shows an example of an "attach" operation. In this figure, (a) shows the situation just before the assembly. Two objects are detected and enclosed in red rectangles. Then, (b) shows the moment of assembly. The color of the rectangles was changed to show that the operation was detected. Finally, (c) shows the moment right after assembly.

In our preliminary experiment in this situation, operations were correctly detected in 88%, 76%, and 85% of the total operations for presenting, attaching, and detaching, respectively.

7.3 Application Example

We demonstrated one promising application of our system. By combining object tracking with the intelligent video production system, we can create a clickable icon to link a detected object to supplemental information such as a movie clip about the object. Our prototype system does this simultaneously while video is being recorded.The system directs the cameras to capture a held object; when the person gives any explanation of the object, the system registers its appearance and links it to the movie clip being captured.

An example is shown in Figure 9. In this scene, a host is explaining a dish for the guest. First, when he lifts the dish, the system detects it, and the rectangle with the dotted lines shows the location. When he explains the object ("This dish is ...."), the system recognizes the situation[2], and registers his

[2]Please refer to [3] for the detection.

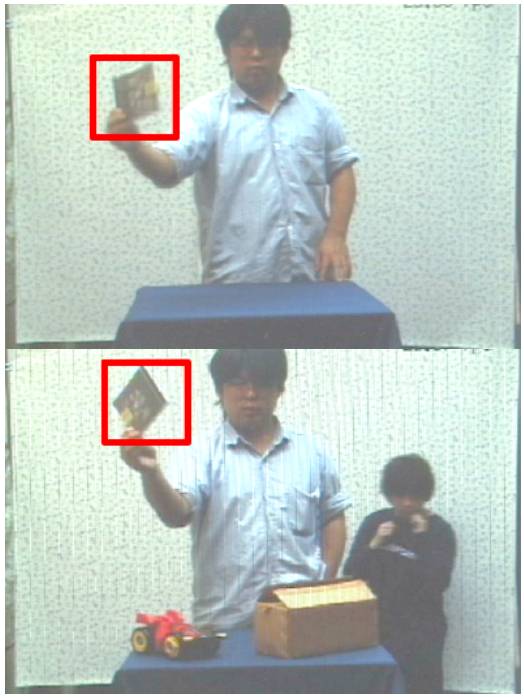


Figure 7. Scene A (above) and scene B (below)

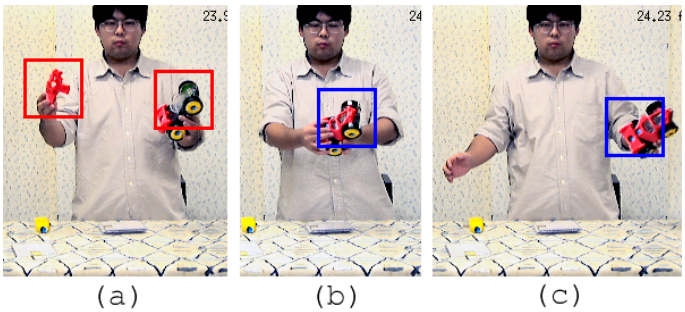


Figure 8. Operation example

annotation as information regarding dishes. This step is noted by the thick red lines overlaid on the dotted lines. When the host places the object on the table, the texture and the position of the object are registered, and the captured annotation is linked to the object region.

8 Conclusion

We proposed a novel method for detecting hands and handheld objects in desktop manipulation situations. By using multiple image sensors, our system realized robust tracking without prior knowledge of an object, even if other skin-color objects are in the scene or the background contains movement. We experimentally verified the system's object tracking performance using three sensors and evaluated the effectiveness of the sensors' integration.

Our future research will focus on tracking smaller objects,

Table 5. Operation detection performance

	showing	separation	assemble
total	80	80	80
correct	70 (87.50%)	61 (76.25%)	68 (85.00%)
false negative	10 (12.50%)	19 (23.75%)	11 (13.75%)
false positive	0 (0.00%)	0 (0.00%)	1 (1.25%)

efficient tracking by motion prediction, and utilization of our system in video-based multimedia production.

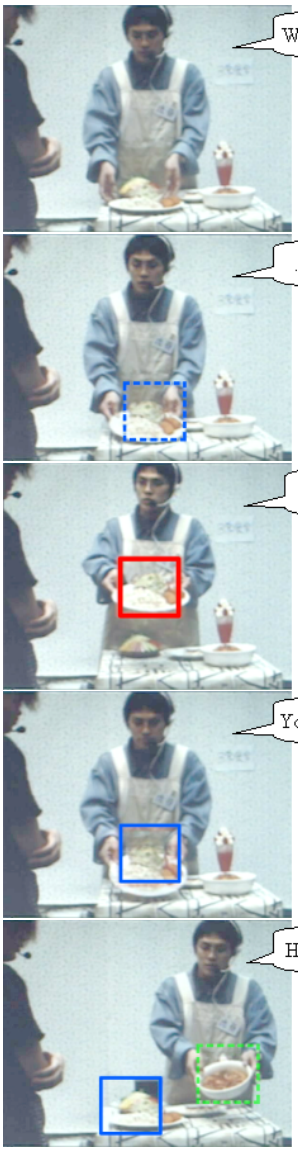
References

[1] H. Koundou, H. Mou, S. Satou, and M. Sakauchi. Indexing persons in news video by telop recognision and face matching. *Proc. IEICE Annual Conference*, D-12-190, 1999.

[2] M. Ozeki, Y. Nakamura, and Y. Ohta. Camerawork for intelligent video production – capturing desktop manipulations. *Proc. ICME*, pages 41–44, aug 2001.

[3] M. Ozeki, Y. Nakamura, and Y. Ohta. Human behavior recognition for an intelligent video production system. *Proc. PCM*, pages 1153–1160, dec 2002.

[4] O. Yoshimi and H. Yamaguchi. Sharpening of object contours disparity image using coefficient of swelling. *Proc. SSII*, pages 227–230, 2000.



(a) Left image shows the result of held-object tracking. The person on the right is explaining a dish for a guest.

(b) When he held the dish, the system detected it, and the rectangle with the dotted lines indicates the location.

(c) When he gave the explanation of the dish, the system recognized the situation and registered his annotation as the dish information.

(d) When he put the object on the table, the texture and the position of the object were registered, and the captured annotation was linked to the object region.

(e) Then, the system detected a new held-object.

Figure 9. Example of combining our object tracking system with an intelligent video production system