Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 1159–1172
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Automatic object extraction and reconstruction in active video

Ye Lu, Ze-Nian Li*

School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6

Received 15 August 2006; received in revised form 11 May 2007; accepted 2 July 2007

Abstract

A new method of video object extraction is proposed to automatically extract the object of interest from actively acquired videos. Traditional video object extraction techniques often operate under the assumption of homogeneous object motion and extract various parts of the video that are motion consistent as objects. In contrast, the proposed active video object extraction (AVOE) approach assumes that the object of interest is being actively tracked by a non-calibrated camera under general motion and classifies the possible movements of the camera that result in the 2D motion patterns as recovered from the image sequence. Consequently, the AVOE method is able to extract the single object of interest from the active video. We formalize the AVOE process using notions from Gestalt psychology. We define a new Gestalt factor called "shift and hold" and present 2D object extraction algorithms. Moreover, since an active video sequence naturally contains multiple views of the object of interest, we demonstrate that these views can be combined to form a single 3D object regardless of whether the object is static or moving in the video.

Keywords: Active video; Camera motions; Shift and hold; Object extraction; 3D object reconstruction

1. Introduction

Fully automatic extraction of semantically meaningful objects from visual data is one of the ultimate aspirations in computer vision and pattern recognition. In addition to the obvious academic interest in this problem, there is a wide array of practical applications that can benefit tremendously from successful object extraction algorithms. One application that can immediately take advantage of object extraction is video compression. A video compression engine can selectively compress objects with higher bit-rates to produce subjectively pleasing results while lowering the bit-rates used to compress less important regions to maintain storage and transmission efficiency. Furthermore, with the proliferation of digital media, rapid searching and retrieval of multimedia data are of paramount importance to industries such as communication, education, and entertainment. It is widely believed that object extraction is the key to more efficient, accurate, and user friendly implementations of such systems. Last but not least, advanced functionalities in surveillance systems such as recognition of suspicious actions and identification of known individuals can be made much simpler with the availability of extracted objects [1].

Digital video carries rich multimedia information and it involves an insurmountable amount of data. For interactive use of the video data, the research community has been focusing on newer standards MPEG-4/H.264 and MPEG-7 where the notion of *video object* (*VO*) is the key, because in most cases VOs and their behavior are the contents! MPEG-4/H.264 has specified many VO-based coding methods. However, one thing was made clear, MPEG-4/H.264 (as other MPEG standards) is a decoding standard. The message is that we do not yet know how to accurately extract VOs.

It is observed that, in general, videos can be classified into two types: *passive video* and *active video*. A video produced by a static surveillance camera is a good example of the former. The camera's function is to (passively) record all objects passing by in front of it. Because of various security concerns, vast amount of this type of video data is generated daily and various software/systems (such as Blue Eye Video) have been developed for automated processing and analysis of these data.

* Corresponding author. Tel.: +1 604 291 3761.
E-mail addresses: yel@cs.sfu.ca (Y. Lu), li@cs.sfu.ca (Z.-N. Li).

0031-3203/$30.00
doi:10.1016/j.patcog.2007.07.015

However, video generated by an active vision system, such as our eyes, will not look like that. In general, digital videos taken by human subjects are more purposive. Typical examples will be filming, professional video cameramen covering sporting events, or an amateur shooting at a tourist scene (e.g., buildings, sculptures/statues, and activities of crowd/people). We call the video thus produced *active video*.

Active videos are very much object-centered, and often exhibit prominent catching and holding behaviors of the human operator. In order to capture the object-of-interest and its movements, it is common for the videographer to initiate various camera movements. Now the rapid pan/tilt movement is analogous to saccades, which is often triggered by object movements or distinct visual features (color, texture, shape, etc.) in the periphery, indicating a shift of attention. When dealing with moving objects, smooth (and usually not so rapid) pan/tilt movements are used for smooth pursuit. When multiple views of the object are desirable, we will witness body movement of the videographer. In professional filming, such movements are often facilitated by sliding rails and moving platforms. It should be apparent that active video is by definition object-based and full of actions.

Object extraction can be considered as a process of identifying an arbitrary collection of image regions that are usually not coherent in low level image features or motion, but somehow form a semantically meaningful entity called an "object". The lack of clear and rigorous definition of what an object is makes this problem exceptionally difficult to solve. Traditional methods of object segmentation follows the configuration laid out by Marr [2] which takes a passive approach by casting the computer as an observer from which useful information is gathered and processed. Although there have been many fruitful results along this line of research, it is very difficult to perform high level vision tasks without active participation from the vision system. It is precisely for this reason that *active vision* is proposed [3,4] for which efforts are made for computer controlled cameras to actively participate in the visual perception process, much similar to the body and eye movements of human vision systems [5]. Of course, when the camera is operated by a human being as in the case of movie making and even home video making, the lines of reasoning advocated in active vision research can essentially be reversed to form a bridge to connect the conceptual gap between the visual world and the underlying semantic meanings. From here on, we shall assume that the input image sequences or videos are acquired by intelligent active vision systems or in most cases by human beings. We thus use the term *object extraction* as opposed to the term *object segmentation* to reflect the active nature of our input data.

In this paper, we introduce a new Gestalt factor called *shift and hold* that describes the motion pattern of the potential object of interest on the image plane. We then develop the required algorithms to extract image regions corresponding to the particular motion pattern that we seek. These image regions form the object of interest and can be tracked throughout the sequence. This is our general strategy for active video object extraction or AVOE for short.

Computing 3D models from 2D views is an important but yet difficult problem in computer vision. An immediate application of visual 3D modeling through 2D views is video indexing and retrieval. If accurate 3D object models can be computed from video sequences, the retrieval system can extract useful 3D shape information from them and use this information to search for similar objects as well as eliminate false matches through shape verification. In viewing this need for 3D object models, we present our AVOE and reconstruction algorithm which extracts objects of interest from active videos and integrates various views of the same object into a single unified 3D surface model. In order to reconstruct the *Euclidean* shape of the object of interest, it is necessary to determine the calibration of the camera. However, since no calibration object was present at the time when the video was taken, traditional calibration method cannot be applied. Instead, we perform a procedure called *self-calibration* to determine the internal parameters of the camera without using any pre-made calibration objects.

2. Shift and hold: a new gestalt factor

The Gestaltist's view of perceptual organization found in 2D images provides much of the underlying principles behind modern image segmentation algorithms. Although these organizational principles can be applied in a similar manner to image sequences (or videos) to perform figure and ground segregation, doing so will likely fail to exploit the richness of information contained within image sequences and may not capture the intentions of the author of the video. In this section, we introduce a new Gestalt factor called *shift and hold* which bridges the gap between static images and video sequences.

2.1. Motivation

Figure and ground segregation is not only an interesting problem in the academic sense but also has a large number of potential practical applications. Gestalt psychology defines a number of factors that can aid in figure and ground segregation on static 2D images [6]. However, because 2D images are perspective projections of the 3D world, much information is lost during the projection process. As a result, it is sometimes extremely ambiguous to separate the figure from the ground even after we apply these Gestalt principles. Some examples of well-known ambiguities are shown in Fig. 1. These ambiguities occur when both the black and white regions have valid semantic interpretations. It is evident that these ambiguities remain even to the human eyes. The fact that our biological vision system rarely produces ambiguous interpretations of the world suggests that most of these artificially designed 2D visual ambiguities can be resolved when we attempt to perceive objects in 3D using various cues such as lighting, shading, shadows, and through the *stereopsis* process.

Fig. 2 shows another illustration of Rubin's vase. In that illustration, there are various visual cues on the vase so that it is immediately perceived as the figure while the black areas are the ground. Comparing to Fig. 1(a) where figure and ground reversals often occur, the shadings, reflections, the deformations

Fig. 1. Ambiguous figures and grounds: (a) Rubin's vase (courtesy of Makio Kashino), (b) a piece of Escher's art work (All M.C. Escher's works © 2004 by The M.C. Escher Company—the Netherlands. All rights reserved.), and (c) the letters "WIN" in white (courtesy of Prof. Dr. Jürg Nänni). All figures used by permission.

Fig. 2. A more realistic picture of Rubin's vase (courtesy of Makio Kashino, by permission).

of pattern on its surface, and the specular lights reflecting off the vase cause it to be perceived as the dominant figure that stands out in front of the ground. By the Law of Prägnanz, this simplest and most stable shape makes figure and ground reversal difficult. Although people can still consciously reverse the figure and ground in this case, they will have to do so with a lot of effort. Following this line of reasoning, many vision researchers [2,7–9] argue that depth segregation occurs before other processes.

In our approach, we do not perform full depth segregation but only recover the image pixel motions between consecutive frames. Our approach can be related to a number of hierarchical interactive processing models [9,10] which indicate that depth segregation need only be attempted but not necessarily fully accomplished. In maintaining this view, we propose our new Gestalt factor *shift and hold* to operate on dense pixel motion estimates which are inversely related to depth when the camera motion is a horizontal translation on the principal plane.

Another motivation for defining "shift and hold" as a Gestalt factor is to take advantage of possible reduction in perceptual complexity of moving objects. According to the Law of Prägnanz, the perceptual world is organized into the simplest and best shapes. The notation of simplicity is explored by Restle [11] in the case of motion. He studied the ways how dots moving across a display are perceived. The most complicated approach would be to treat each dot as completely separate from all the others and to calculate its starting position, speed, and direction of movement, and so on. In contrast, it is possible to treat the moving dots as belonging to groups, especially if they move together in the same direction and at the same speed. Restle [11] showed that whatever grouping of moving dots in a display involved the least calculation generally corresponded to what was actually perceived. This observation is summarized into the Gestalt factor "common fate". Similar to "common fate", our Gestalt factor "shift and hold" takes advantage of motion of group elements to reduce complexity. However, "shift and hold" differs from "common fate" in that it groups by using a very specific type of motion which will be defined in Section 2.2. In addition, "common fate" operates on the smallest perceived elements such as image pixels, but "shift and hold" can in principle be applied to much larger sets such as the grouping results from other Gestalt factors. In this sense, "shift and hold" can be viewed as a higher level Gestalt factor than the primitive ones defined in [6].

A third motivation for developing "shift and hold" is to take advantage of the semantic connection between the object of interest and the resulting video sequence. The human visual system does not have uniform resolution; the highest visual acuity occurs at the fovea and gradually decreases into the peripheral area. If our visual system were to have uniform resolution, the head would weigh in the order of 5000 lbs [12]. As a result of this variable resolution nature of the human visual system, a person moves his eye to relocate the object of interest to the middle of the retinal image and lock it there for closer examination. The same behavior can often be observed during the video making process. Thus, this movement and locking of the object of interest to the middle of the each frame in the video provide a strong cue as to what the object of interest might be. This observation demonstrates that the video itself contains clues to what objects are semantically important and the parts

that make up these objects. The Gestalt factor "shift and hold" is developed precisely to exploit these clues in the video.

2.2. Definition

A static image of a scene may contain a number of "objects" in the everyday sense. For example, an image of an office would contain a desk, chairs, bookshelves, some kind of light fixture, or possibly even coffee mugs. These are all valid objects that can be obtained from a combination of the Gestalt grouping factors. However, which object is actually being perceived as the dominant figure in the scene is very much dependent on other external factors such as the mood of the human observer, the context of the conversation, and so on. However, if a video of the scene is shown, it is much easier for the audience to identify the dominant object that appeals to the video maker. This bias towards a particular object as the figure is conveyed through a series of actions during the video making process. Interestingly, these actions all share a similar signature in the visual motion field. Thus, by defining a Gestalt grouping factor on recovered visual motion fields, it is possible to extract the object of interest from a video (or image sequence).

The definition of this new Gestalt grouping factor is inspired by the physical process of foveation. During foveation, the eye first relocates the object of interest into the foveal region and then tries to maintain its position in the fovea by employing various eye and body movements in order to examine the object of interest in high resolution. To emulate this process in an active vision system, the camera first needs to find the object of interest and translate its position to the center of the frame. Then, the system must pursuit the object of interest to stabilize its position at the center of the frame. Finally, the camera zooms in on the object of interest to emulate the high resolution views at the fovea. Essentially, the camera needs to first catch the object of interest by a shift of attention and then hold its position in the foveal region before acquiring more detailed views of it. Thus, the shifting and holding actions provide a strong cue for the object of interest. For this reason, we defined the Gestalt grouping factor "shift and hold" to capture the unique signature of the shifting and holding actions on the visual motion field.

In this section, we develop a precise definition of the Gestalt factor "shift and hold" which we will apply to extract the object of interest. Unlike previously defined Gestalt grouping factors that act on the image domain, "shift and hold" is defined directly on the visual motion field. The definition of "shift and hold" is given below.

Definition 2.1 (*Shift and hold*). Visual elements (pixels, or groups of pixels) that shift towards the center of a video frame and maintain their position for a number of subsequent frames with little or no motion relative to the peripheral regions tend to be perceived as the figure.

The first part of Definition 2.1 encapsulates the tendency for people to move the object of interest from the peripheral regions of the frame into the central portion of the frame while the second part indicates that the object of interest is locked on to the central foveal region of the frame for a duration of time. However, this duration is not explicitly defined in the definition. Different applications may have very different requirements for the duration in which the object of interest has to maintain its position in the foveal region. Thus, a simple threshold can be set to adapt the "shift and hold" factor to specific applications.

At first glance, our definition of "shift and hold" may seem to be a specialized version of the Gestalt grouping factor "common fate". Although they are both defined using motion, these two definitions have a very distinct difference. By definition, the Gestalt factor "common fate" states that when basic visual elements move in the same direction, we tend to group them as a unit. In contrast, our definition of "shift and hold" does not insist that visual elements move in the same direction in order to be grouped. Instead, in the "shift" part of our definition, visual elements that are part of the figure can relocate themselves to the central part of a frame from various different directions. In addition, the "hold" part of the definition only insists that the motion be small, but not all in the same direction. To further clarify the difference between these two grouping factors, we can say that "shift and hold" defines the destination of motion but not the direction of motion while "common fate" defines the direction of motion but not the destination.

The effects of camera zooming have very well-defined signatures on the visual motion field. Even though these special properties of zooming is not explicitly included in the definition of "shift and hold", it is easy to show that the effects of camera zooming can be subsumed under the definition of this new Gestalt grouping factor. When the camera zooms in, applying Definition 2.1 will not detect any objects since there are no visual elements moving towards the center of the frame and there are no elements with small motion around the central region. Interestingly, notice that the only time that the videographer would want to zoom in is when the object of interest has already been located and positioned in the central region. Thus, prior to the zooming operation, there must be frames in the video performing the shifting and holding actions as defined in Definition 2.1. Therefore, the object of interest can be found in these frames using the "shift and hold" factor and then tracked through the frames that are zooming in using the recovered visual motion field. When the camera zooms out, all motion vectors would point towards the center of the frame. Under these circumstances, there are two cases to consider. In the first case, the object of interest has not yet been located by the camera prior to the zooming operation. Thus, the camera is simply zooming out in order to obtain a wider field of view. In this case, a saccadic camera motion would follow immediately after the camera zooms out to search for the object of interest. Therefore, the "hold" part of Definition 2.1 will not be satisfied and so no object will be detected as desired. In the second case, an object of interest has been located prior to the zooming operation. Then, frames performing the zooming out operation would simply be interpreted as holding the object of interest in the foveal region since the motion vectors within the foveal region have smaller magnitudes compared to those in

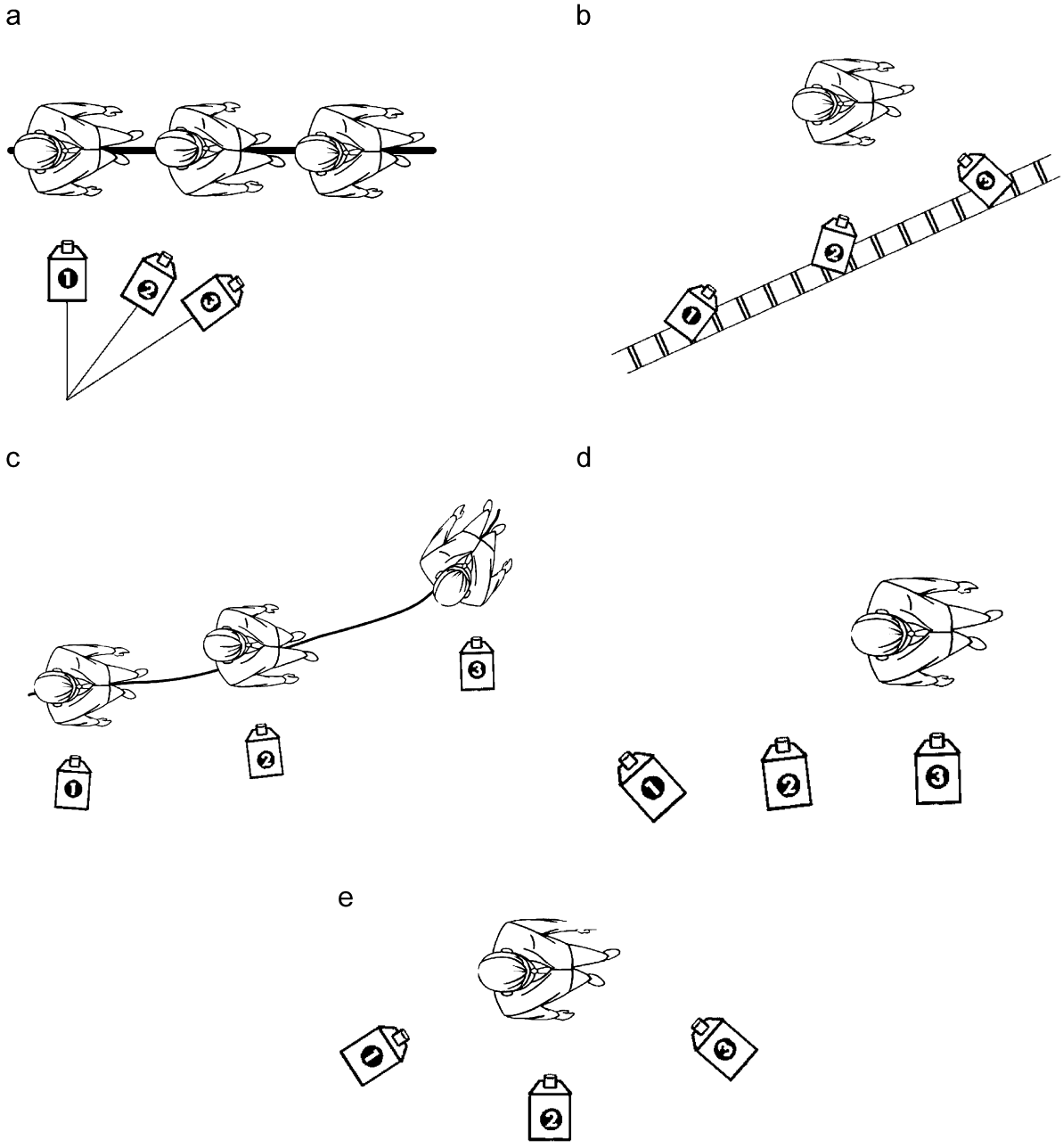

Fig. 3. Different types of camera motion: (a) panning, (b) dollying, (c) tracking, (d) saccadic movement, and (e) revolving movement.

the peripheral region. Therefore, the object of interest will still be detected through these frames.

2.3. Application to VO extraction

The newly defined Gestalt factor "shift and hold" can be readily applied to VO extraction. Recall that the motions of visual elements need to be recovered first. Motion estimation from multi-view stereo images is a well-studied area [13]. The recent article by Seitz et al. [14] presented a comparative study of the state-of-the-art. In [15,16], we presented an improved cooperative motion estimation algorithm that recovers the motion of pixels. Using that, we are able to obtain relatively accurate visual motion fields as inputs to our VO extraction algorithm.

A video can be decomposed into hierarchical levels: video, scenes, shots, and key frames, with each level increasing in granularity [17]. The highest level in this hierarchy that preserves continuity in video content is at the shot level. This level is thus appropriate for VO extraction algorithms. Various techniques [18–20] have been proposed for retrieving the different shots from videos.

Each shot may contain one or several different types of camera movements. Different camera motions are utilized in

cinematography to convey or emphasize certain things and of course focus on different objects of interest (or areas of action) in the process of doing so. Here, we list three different camera motions frequently used in film making [21].

- Panning—The camera rotates around a vertical axis to follow its subject or action.
- Dollying—The camera is mounted on a platform and moves into or away from the subject or action.
- Tracking—The camera moves along with a moving subject while keeping approximately the same distance to the subject.

We also list two additional types of camera motions commonly used by amateur video makers.

- Saccadic camera movement—The camera moves rapidly to search for the object of interest. This includes rapid panning (or tilting).
- Revolving camera movement—The camera revolves around the object of interest to capture views of it from different angles.

These different types of camera motion are illustrated in Fig. 3. In film making, the differentiation between these camera movements is mainly caused by the setup requirements or the movement of the camera operators. However, from a computational perspective, it is possible to group some of them together and reduce from five types of camera motion into only three categories. The three categories of camera motions are:

- Saccadic motion—Same as saccadic camera movement.
- Smooth pursuit—Smooth panning (or tilting) shots in which the camera rotates about the same point.
- Multi-baseline pursuit—The camera may freely translate and rotate to follow the object of interest. This category includes dolly shots, tracking shots, and revolving camera movements.

Instead of turning off the camera during context switch, amateur video makers often use saccadic camera motion to switch the object of interest. Therefore, one shot could contain more than one object of interest, but separated by a number of frames of rapid camera saccade. Locating frames of saccadic camera movement is relatively simple. Since frames of saccadic motion generally consist of rapid movement of visual elements and since the camera does not fixate on any object, applying the "shift and hold" factor on the visual motion recovered from these frames would not return any objects. Therefore, frames in which no objects can be found using "shift and hold" can be labeled as being performing saccadic camera motion.

The difference between smooth pursuit and multi-baseline pursuit is the way in which the camera moves. For smooth pursuit, the camera rotates about the same location as shown in Fig. 3(a). Multi-baseline pursuit, on the other hand, does not place any constraint on the camera motion. The camera can freely translate and rotate to follow the object of interest. The distance between the camera and the object of interest is relatively constant in this case. As a result, the image of the object of interest in the video stays approximately the same size. The "shift and hold" factor can be applied to frames of both smooth pursuit and multi-baseline pursuit for object extraction. In both cases, the object of interest will be fixated on the foveal region of each frame. Since the camera is mainly compensating for the object's (or the figure's) motion in order to fixate its location on the image, the visual elements in the peripheral region constituting the background will most likely possess large visual motion. Therefore, to extract the object of interest, we only need to examine the magnitude of the recovered motion vectors. The central region with small motion is extracted as the object while the remaining part of the image is the background.

Visual motion fields recovered from smooth pursuits and multi-baseline pursuits are generally very similar. The only way to differentiate between the two is to examine the change in size of the extracted object from successive frames. With smooth pursuit, the image of the object gradually changes as the object moves closer to or further away from the camera. In contrast, multi-baseline pursuit keeps the size of the object relatively constant on each frame since the distance between the camera and the object in 3D is more or less the same from frame to frame. However, for the purpose of object extraction, very similar methods can be applied to both kinds of camera movements.

3. Active VO extraction

In this paper, we assume that the actions taken by the videographer are purposive. Thus, the resulting video is not a series of random shots, but a combination of well-intended camera movements capturing a set of objects of interest. Our goal is to extract only the objects of interest from active videos and leave behind other objects that just happen to be in the scene. We model three types of actions performed by the active observer: saccadic movement, smooth pursuit, and multi-baseline pursuit. Each of these three types of actions result in different characteristics in the visual motion of pixels computed from consecutive frames of the video as well as give important hints as to what objects the videographers are interested in.

3.1. Core extraction algorithm

In this section, we present a simple object extraction algorithm from active videos. This algorithm proceeds by examining the magnitudes of the recovered visual motion in the foveal region $\mathscr{F}$. It first tries to decipher which one of the three kinds of camera movements has occurred from two consecutive frames of the video. If the camera is performing a smooth pursuit or multi-baseline pursuit, then it implies that an object of interest exists in these frames. Otherwise, the object of interest is not present in these frames and the algorithm continues to process the next two frames. We summarize the core object extraction algorithm as follows:

Algorithm 3.1. Active object extraction core

1. Recover the dense 2D motion field for two consecutive frames in a shot.

2. Examine the magnitude of the motion vectors in the foveal region $\mathscr{F}$ of the recovered motion field.
3. If the average motion vector magnitude in the foveal region is less than a threshold τ, then the current action must be either smooth pursuit or multi-baseline pursuit.
   - Grow a region from the fovea containing only pixels having motion vector magnitudes less than τ.

   Otherwise
   - These two frames must be part of a saccadic movement, and so no object is detected. Go back to Step 1.
4. Fill holes in the region computed in the previous step and output the resulting object.

This approach to object extraction is very different from the traditional feature based extraction techniques. In the above algorithm, it is evident that no low level image features are employed during the object extraction process. We will show, in our experimental results, that this surprisingly simple algorithm is very effective in detecting and extracting the object of interest from active videos. However, the performance of the core algorithm can be improved dramatically by taking into account only the edge information in each frame. We demonstrate this using a linear programming based boundary adjustment algorithm.

3.2. Linear programming based object boundary adjustment

The outline of the extracted object of interest from the core algorithm can be rather jagged and there is no guarantee that it will coincide with meaningful object boundaries. An easy way to remedy this problem is to adjust the object boundaries produced by the core algorithm in such a way that it is close to the original boundary, coincides with a correctly oriented edge, and minimizes the change from scan line to scan line.

The boundary adjustment problem can be effectively solved by a linear programming based approach. Essentially, the boundary adjustment problem becomes a minimization problem

$$\min \sum_{y_j} \sum_{x_i \in \Phi_j} \xi_{x_i,y_j} c(x_i, y_j) + \lambda(d_j^+ + d_j^-) \tag{1}$$

subject to

$$\sum_{x_i \in \Phi_j} \xi_{x_i,y_j} = 1, \tag{2}$$

$$\sum_{x_i \in \Phi_j} \xi_{x_i,y_j} x_i = x_j^*, \tag{3}$$

$$x_j^* - x_{j+1}^* = d_j^+ - d_j^- \tag{4}$$

with bounds

$$\xi_{x_i} \geqslant 0, \tag{5}$$

$$0 \leqslant d_j^+, d_j^- \leqslant R, \tag{6}$$

where λ is an adjustable weight. ξ_{x_i,y_j} are the real valued weights for the boundary candidate (x_i, y_j). x_j^* is the final location of the boundary at scan line y_j. The cost $c(x, y)$ of putting an object boundary at (x_i, y_j) is defined as

$$c(x, y) = E_\theta(x, y) + kE_M(x, y) \tag{7}$$

with $E_\theta(x, y)$ and $E_M(x, y)$ representing the normalized orientation and magnitude of the edge at (x, y), and k a constant weighting factor. The quantities d_j^+ and d_j^- are the positive and negative displacements of the object boundary from scan line j to $j + 1$. Separate variables are needed to represent the positive and negative displacements because all variables in the objective function need to be non-negative. The variables ξ_{x_i,y_j} act as indicator variables that select boundary pixels x_i at scan line y_j. The set Φ_j contains all the candidate boundary pixel locations at y_j. The candidates are pixels in the search range of $[-R, R]$ centered at the original boundary pixel returned by the core algorithm.

The value of ξ_{x_i,y_j} determines whether there is a boundary pixel at location (x_i, y_j). Ideally, ξ_{x_i,y_j} should strictly be binary valued variables, and together with Eq. (2), this implies that only one of ξ_{x_i,y_j} will receive a value of 1 while the others all have values of 0 on any scan line j. Strictly enforcing this constraint would turn the minimization problem in Eq. (1) into a mixed integer programming problem which is much harder to solve. Thus, we relax the variables ξ_{x_i,y_j} into a range [0, 1]. However, when the linear programming algorithm terminates, the values of ξ_{x_i,y_j} only take on binary values in most cases. When there are more than one non-zero ξ_{x_i,y_j} on a scan line j, the location having the largest value will be chosen to be the boundary pixel.

Two choices of parameters need to be selected in order to solve the linear programming problem given previously. The first parameter λ controls the vertical continuity in successive scan lines in the extracted object. A larger value of λ would cause the second term in Eq. (1) to have a higher weight, so it would result in a smoother object boundary in which the position of boundary pixels change only slightly from one scan line to the next. By the same token, a smaller value of λ would result in a more jagged object boundary. The second parameter to choose is k which weights the influence of $E_\theta(x, y)$ against $E_M(x, y)$. Intuitively, k decides how well the object boundary should agree which strong edges. For a large value of k, the resulting object boundary will be very likely to coincide with a strong edge regardless of the edge's orientation. However, a smaller value of k allows the algorithm to weigh the orientation more in the resulting object boundary.

We first divide the object boundary vertically into the left and right halves. We then independently adjust the two halves using our linear programming algorithm. The results of the improved object boundaries are shown in the section of experimental results.

4. 3D object reconstruction

Since an active video sequence naturally contains multiple views of the object of interest, a logical step forward from

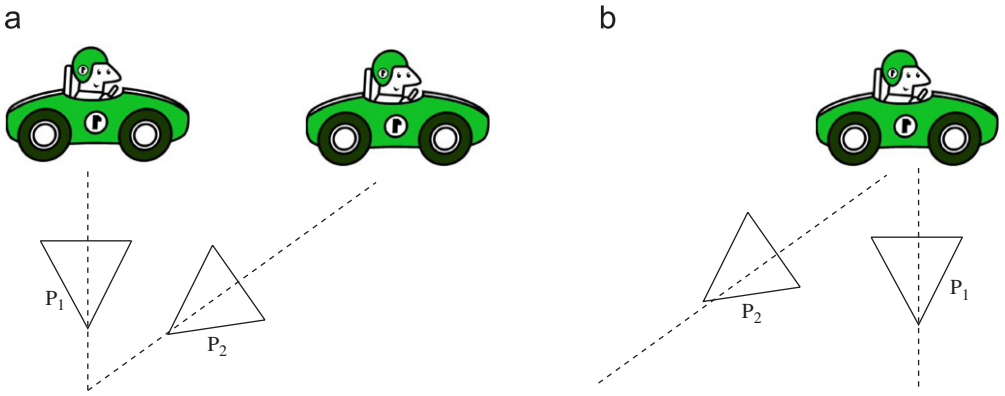

Fig. 4. Reconstruction of moving objects of interest.

2D object extraction is the reconstruction of the 3D object of interest from multiple views. Recovering the 3D structure of the object of interest allows indexing and retrieval of active videos using 3D object structure instead of the common 2D descriptors such as color, texture, and motion. Below, we will outline the main steps needed to reconstruct the object of interest in 3D. For details, readers are invited to read Lu's dissertation [16].

4.1. Steps for 3D object reconstruction

Recovering the structure of a static object of interest from its images taken by a camera under general motion can be done using the following steps:

- *Motion estimation*: The first step towards finding a 3D reconstruction from multiple 2D images is to find corresponding pixels in successive views. Using these correspondences, the fundamental matrix can be estimated between each pairs of views. This can be done robustly using algorithms such as Random Sample Consensus (RANSAC) or Least Median of Squares (LMedS).
- *Projective reconstruction*: Projective factorization is then performed on the matching points to obtain a projective reconstruction of the 3D structure and the corresponding projection matrices. Since we do not assume the affine camera model, the projective factorization method requires estimates of projective depths for each point. These projective depths can be estimated by stringing them together using the fundamental matrices with the first depth value set to one.
- *Camera calibration*: In order to compute a Euclidean reconstruction from the projective reconstruction, the camera needs to be calibrated. In other words, the internal and external parameters of the cameras must be computed. The internal camera parameters can be estimated using a process called camera autocalibration which translates the unknown internal parameters into constraints on the dual image of the absolute conic.
- *Euclidean reconstruction*: The revised projective reconstruction can then be upgraded into a Euclidean reconstruction using the rectifying homography computed during autocalibration. In order to display the resulting reconstruction, the recovered Euclidean structure needs to be triangulated. The triangulation process produces a set of polygons with the reconstructed points as their vertices. These polygons approximate the 3D surface in question.
- *Presentation details*: For increased realism, the triangulated surface can be texture-mapped. This can be done by computing the texture coordinates on one of the 2D input images and map it onto the polygons.

4.2. 3D reconstruction of moving objects

In general, 3D reconstruction of a moving object from images captured by a moving camera is a very difficult problem. Structure from motion, in the traditional sense, often makes the "static scene" assumption. Previous work on the independent camera and object motion problem has concentrated on the motion segmentation problem [6,22]. However, such reconstruction can usually be readily computed for the class of active videos. The shifting and holding properties of active videos facilitates easy and accurate extraction of the object of interest. Consequently, points belonging to the object of interest exhibit only a single motion resulting from the slightly different views of the object captured by the moving camera. Thus, this reconstruction problem can be (approximately) reduced to that of the reconstruction of static objects captured by different camera movements.

The problem of reconstructing the moving object of interest captured by an actively moving camera is illustrated in Fig. 4. In Fig. 4(a), a moving car is under the smooth pursuit of a panning camera. Since the object of interest is being held in the foveal region of each frame, assuming the object is sufficiently far from the camera, the motion of pixels belonging to the object of interest is equivalent to that of a static car being imaged by a camera under a different motion, as illustrated in Fig. 4(b). Similar analysis can be applied to the case of multi-baseline pursuit. However, in the special case in which the pursuing camera always captures the same view of the object, the reconstruction of this object is not possible. In Section 5, we will present the 3D reconstruction results of moving objects under the pursuit of active cameras.

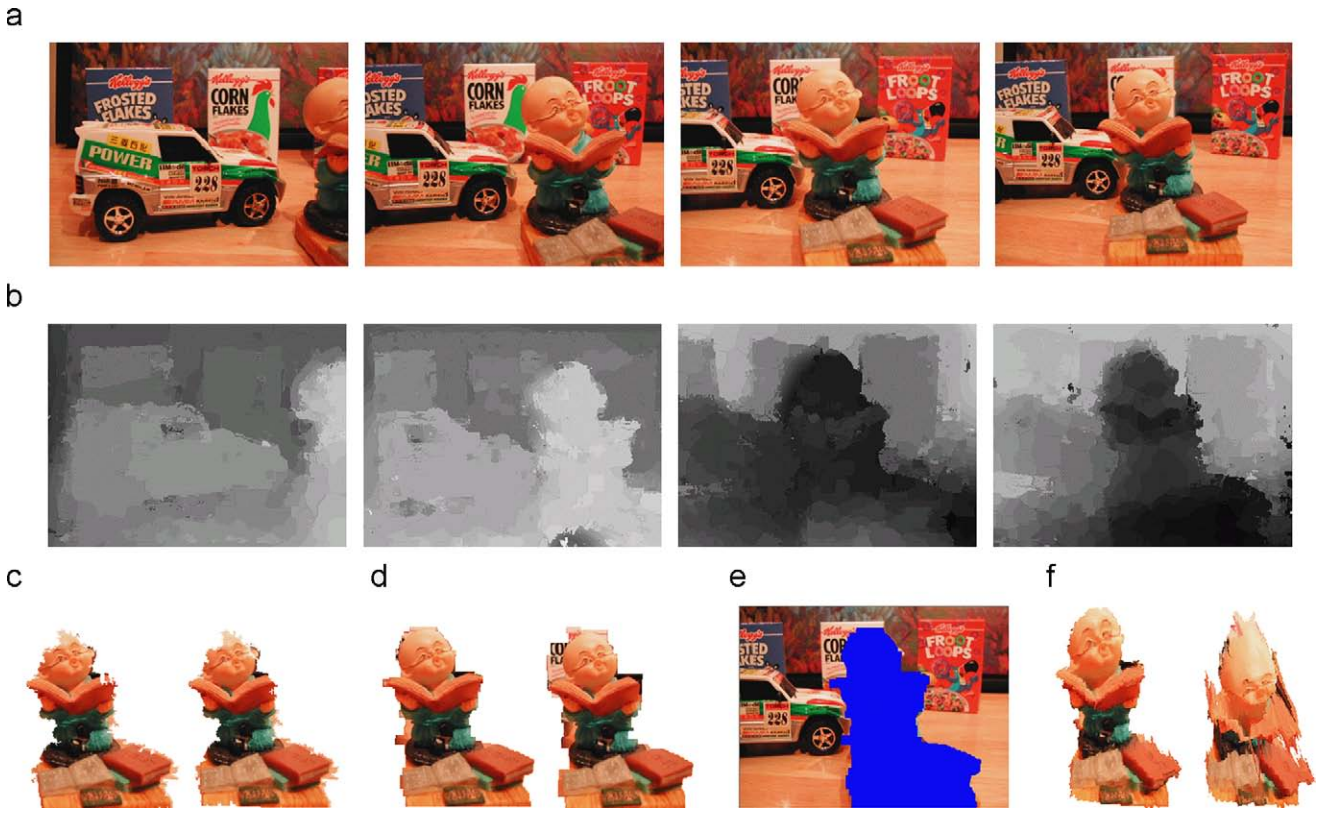

Fig. 5. Object extraction and reconstruction results for the Monk sequence: (a) frames from input sequence, (b) the magnitudes of motion vectors for the corresponding frames, (c) object extracted using the basic algorithm, (d) improved object boundaries using the linear programming based algorithm, (e) extracted object overlaid on top of the original frame, and (f) front and top views of the reconstructed monk figurine.

5. Experimental results

We have implemented the proposed active object extraction and reconstruction algorithm using C++ on a PC platform. We have tested our algorithm on a number of video sequences taken by a human user. These video sequences were taken by non-calibrated cameras. For example, the Monk sequence has 11 frames in total taken by a camera held in the hands of a human videographer. Since a digital camera (in continuous shooting mode) is used to capture these images, the frame rate is only about 4 frames per second. Therefore, the 11 frames would correspond to roughly 80 frames if taken by a video camera. The camera moves rapidly to the right to search for the object of interest in the first eight frames and revolves around the monk figurine in the last three frames. For this reason, this sequence can be seen as a composition of two different camera events: saccadic movement followed by multi-baseline pursuit. In the Bear sequence, the camera simply revolves around the toy bear. The Teapot sequence is taken by a camera mounted on a tripod. The purpose of the sequence is to emulate the situation where a moving object is tracked by a moving camera. As the camera pans to the right, the location of the teapot is manually moved in each frame, as can be seen from the tape measure on the table. We implemented the dense motion estimation algorithm proposed in Ref. [15]. Since the threshold τ depends on the speed of the camera motion, we set it to be 20% of the largest magnitude of the motion vectors in each frame. The search range for the linear programming based boundary adjustment is set to $[-10, 10]$.

5.1. Object extraction and reconstruction for the Monk sequence

The Monk sequence demonstrates several interesting points. Since it is composed of two types of camera motions, it tests the algorithm's ability to distinguish between them. From Fig. 5(b), we can see that the magnitudes of the motion vectors are large in the foveal region during saccadic movement and the first eight frames all share this common property. Therefore, they are easily detected as part of the saccadic movement. As the camera revolves around the monk figurine, we see that the magnitudes of the motion vectors are small on the object of interest and large elsewhere. Thus, these frames are subsequently detected as part of a pursuit movement. This motion sequence can also be used to show the difference between the newly defined "shift and hold" and the Gestalt factor "common fate". When this sequence is processed using "common fate", everything will be classified as a single object in the first eight frames

Fig. 6. Object extraction results for various objects. Column 1 shows a frame from each sequence, Column 2 is the magnitudes of the recovered visual motion, Columns 3 and 4 are the edge magnitudes and orientation costs, Column 5 shows object extraction without using color and texture, and the last column is the improved object extraction results. (a) Toy bear sequence, (b) box sequence, (c) dog sequence, (d) phone sequence, and (e) pineapple sequence.

since all pixels in these frames move in the same direction. However, in subsequent frames, the camera rotation causes the motion vectors on the monk figurine to point in different directions which causes "common fate" to group distinct parts of it as different objects. By contrast, the "shift and hold" factor expects the relocation of the object to the center of the image in the shift phase and so no object will be detected at that time. In the subsequent camera rotation frames, the object of interest is at the center of the frame and even though motion vectors within this object points in different directions, the magnitude of the motion vectors is small on the object compared to the background. Therefore, the desired object will be identified as a whole using "shift and hold". Moreover, this sequence clearly accentuates the monk figurine as the object of interest by the videographer even though there are many other objects in the scene. The result shows that our algorithm is able to correctly detect only the object of interest and successfully extract it. Fig. 5(c) shows monk figurine extracted using the basic algorithm. We can see that the object outline is poor around the head region because of poor motion estimation caused by occlusion. The result of applying the linear programming based boundary adjustment shown in Fig. 5(d) significantly improves the object outline at that region as well as cleaning up some excessive pixels at the base of the figure. We overlaid the extracted object on top of the original frame in Fig. 5(e). This shows that the object of interest is completely and cleanly extracted. The 3D reconstruction result of the monk figurine is shown in Fig. 5(f).

5.2. Improving the accuracy of object extraction

The accuracy of object boundaries can be further improved by combining low level image features such as color and texture with motion vector magnitudes.

Fig. 6 shows object extraction results for various objects. Since the objects of interest are placed on horizontal surfaces, and the camera is tilting downward at an angle when taking the sequences, the pixel motion at the part of the surfaces close to the objects also has small magnitudes. Hence, the magnitudes of the recovered visual motion do not discriminate well between

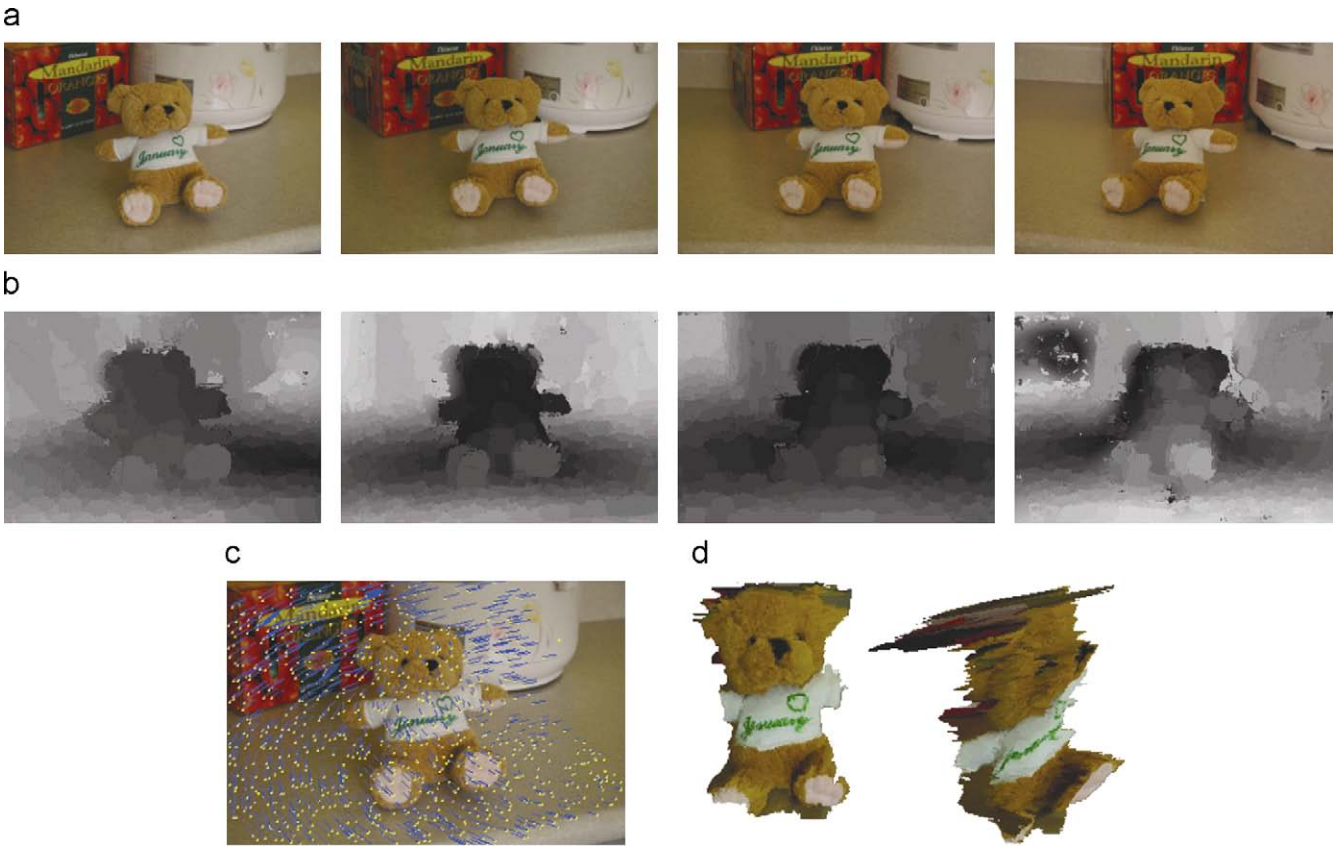

Fig. 7. Object extraction and reconstruction results for the bear sequence: (a) frame from the input sequence, (b) the magnitudes of motion vectors for the corresponding frames, (c) motion vector needlegram of 1000 randomly selected pixels, and (d) front and side views of the reconstructed toy bear.

the objects and the surfaces on which they rest. As shown in Column 5 in Fig. 6, a small portion of the surfaces is extracted as part of the objects. This is a fundamental problem with using only motion data.

Because the quality of object extraction after the linear programming is relatively high, i.e., only some extraneous background pixels near the bottom of the objects need to be removed, we implemented a simple algorithm to incorporate color and texture features as part of the object extraction process.

We call the images in Column 1 "source images" and the intermediate results in Column 5 "object images". First, each object image is used to generate a binary image, i.e., "object mask". Second, a horizontal scanning process is invoked in the object image: at each scan line the boundary points on the object mask are identified; if the color of the boundary pixel is similar to the color of the neighboring background pixel in the source image (a simple L1 distance of the RGB values is used in this implementation), it is changed to "background". Third, use this pixel as the seed to get a connected component in the object image, and then remove this component from the object image. Last, a connected component analysis is conducted to remove small, isolated regions. The threshold is rather big (over 1000). Since an active object is considered as an integral one, we can safely remove those smaller components provided that the object has not been segmented into pieces after the active object extraction.

Only color differences are used for this implementation. The improved results are shown in the last column in Fig. 6. Apparently, the same method will work for texture features if they are also needed.

Fig. 7 shows the extraction and reconstruction results for the bear sequence with improved boundaries.

5.3. More results on moving objects and zooming

The teapot sequence (Fig. 8) demonstrates the extraction and reconstruction of a moving object. The teapot was manually moved on the table and rotated slightly about the vertical axis. The movement of the teapot is evident from the tape measure on the table. The camera performs a multi-baseline pursuit to keep the teapot on its foveal region. We can see from Fig. 8(b) that the magnitude of motion vectors within the teapot is very small at each frame. This clearly demonstrates a pursuit movement of the camera and the teapot object can be easily extracted. Also, even though both the camera and the object of interest are moving, we can still obtain reasonably good reconstruction results shown in Fig. 8(d).

In the zooming sequence (Fig. 9), the camera first revolves around the monk figurine in the center of the image. Then, it zooms in on this object to obtain increasing resolution views of it. In this sequence the camera zooming operation is used

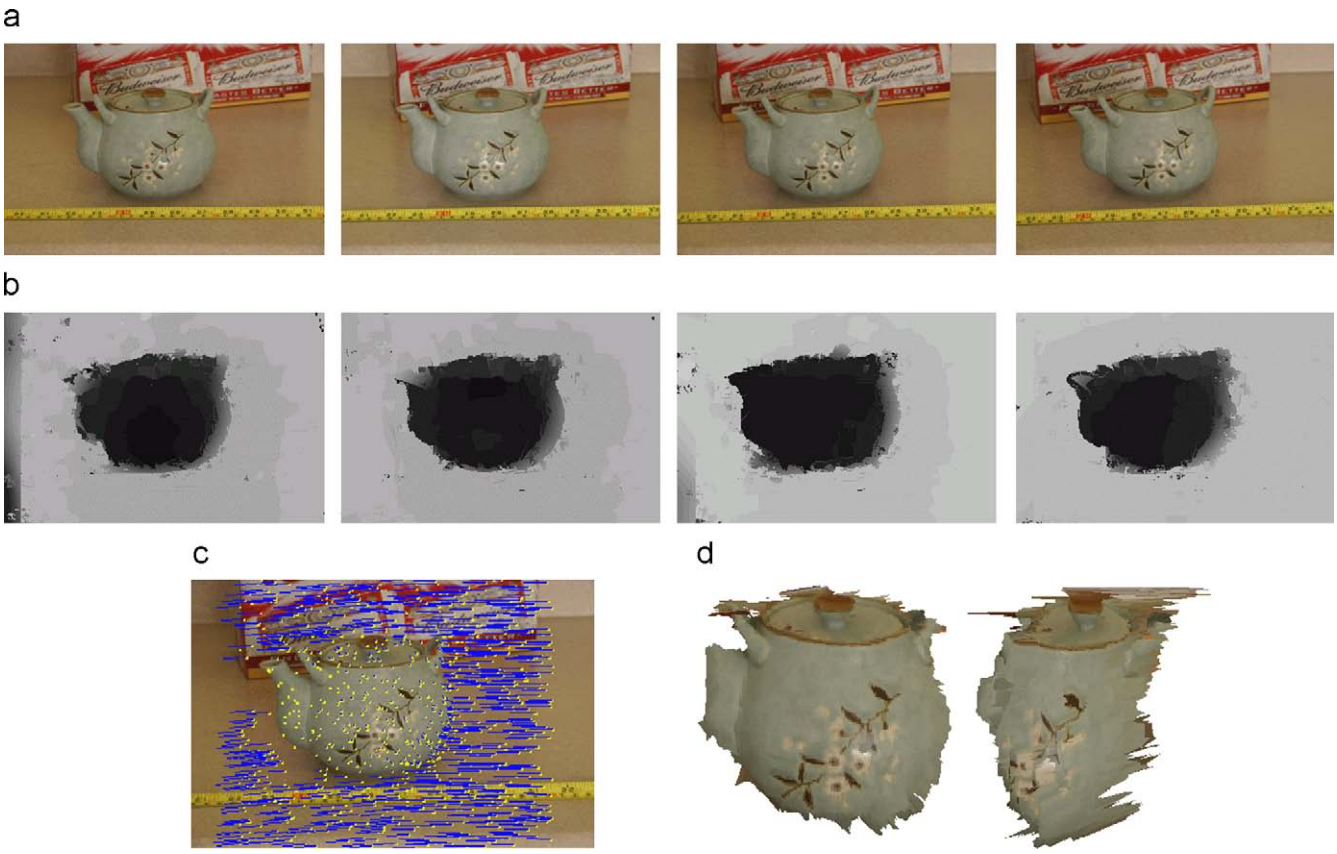

Fig. 8. Object extraction and reconstruction results for the teapot sequence: (a) frame from the input sequence, (b) the magnitudes of motion vectors for the corresponding frames, (c) motion vector needlegram of 1000 randomly selected pixels, and (d) front and side views of the reconstructed teapot.

to emulate the human fovea in which the object of interest is perceived with high resolution. It is evident from Fig. 9(b) that only the revolving movement of the camera gives indication of what the object of interest is while camera zooming does not. Fig. 9(d) clearly shows that the visual motions of the pixels are pointing outwards from the center of the zooming frames. As discussed in Section 2.2, the object of interest is first found using "shift and hold" during the revolving movement of the camera and then tracked through the zooming frames using the recovered visual motion of pixels. The result of this object extraction and tracking combination is shown in Fig. 9(e).

5.4. Some discussions on results

The reconstruction results shown in this section are not completely enclosed 3D object models (they are only reconstructed from some front and side views). Such fully enclosed 3D models cannot be obtained at this point partly because our reconstruction algorithm employs the projective factorization algorithm to determine an initial projective reconstruction of structure and camera motion. This algorithm requires that all point correspondences be visible in every view. If the camera were to circum-navigate to the back of the object of interest, then every point that is visible in the front view will be occluded in the back view. For this reason, the current implementation of our reconstruction algorithm is not able to reconstruct 3D points at the back of the object model. However, because of the modular design of our system, the "Structure and Motion Recovery" module can easily implement other more complex sequential view updating algorithms such as those given in [23,24] to obtain an initial projective reconstruction in a future release.

In terms of execution time, our program currently spends more than 1 min to process each pair of consecutive frames. Exact execution time depends on the contents of the videos. The basic object extraction algorithm together with the linear programming based boundary adjustment usually finishes within about 2 s, running on an Intel Pentium IV at 3 GHz. The majority of computational time is spent on estimating the pixel motion between frames where a search range of 40×40 is used to find the correspondence for each pixel. After the object of interest is located in all frames, the dense 3D reconstruction of the object of interest can usually be done within about 10 s. The algorithms presented in this paper are part of a major effort in the exploration of active videos. We are currently still putting together a multitude of various algorithms into our system. We are confident that improvements and optimization of the system performance can be expected in the near future.

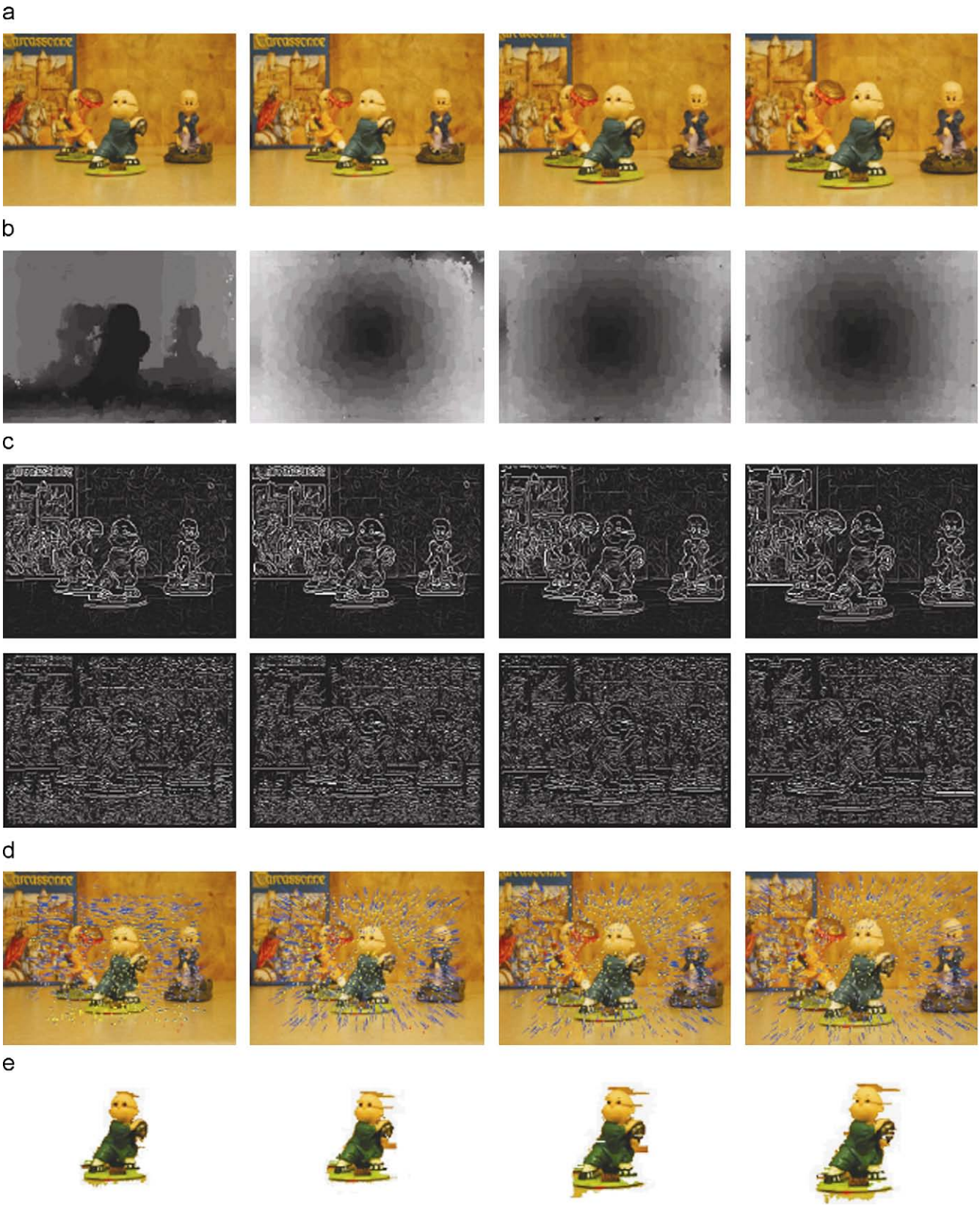

Fig. 9. The zooming sequence: (a) selected original frames, (b) the magnitudes of the recovered visual motion, (c) edge magnitudes and orientation costs for the corresponding frames, (d) the needlegram of 1000 randomly selected pixels, and (e) the object extraction results.

6. Conclusion

Object extraction from active videos is by and large an unexplored area in computer vision and pattern recognition. In this paper, we have presented a technique to extract only the object of interest from active videos taken using non-calibrated cameras under active and unrestricted (hand-held) motions. We justified our approach using recent results from cognitive psychology and proposed a new Gestalt factor called "shift and hold" to describe the catching and holding behavior of active video and subsequently used it to extract the object of interest. We improved the quality of the basic active object extraction algorithm by using a linear programming based object boundary adjustment. The adjustment algorithm is shown to be effective in improving the object outline when inaccurate motion vector estimates caused by occlusion exist around object boundaries. Additional improvement in object extraction is achieved by incorporating color and texture features. Furthermore, we also demonstrated that multiple views of the object of interest can be combined into a single 3D shape regardless of whether the object is static or moving in the video. The results presented in this paper can have immediate applications in indexing and retrieval as well as compression of actively acquired videos. This paper not only makes it possible for search engines to selectively index just the semantically relevant VOs, it can also potentially add an extra searching modality by allowing the object of interest to be queried using its 3D shape.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under the Grant RGP36726.

References

[1] H. Jiang, Z.N. Li, M.S. Drew, Recognizing posture in pictures with successive convexification and linear programming, IEEE Multimedia 14 (2) (2007) 26–37.
[2] D. Marr, Vision, W.H. Freeman and Co., New York, 1982.
[3] Y. Aloimonos, Active Perception, Lawrence Erlbaum Associates Publishers, London, 1993.
[4] D.H. Ballard, Animate vision, Artif. Intell. (48) (1991) 57–86.
[5] J. Wei, Z.N. Li, On active camera control and camera motion recovery with foveated wavelet transform, IEEE Trans. Pattern Anal. Mach. Intell. 23 (8) (2001) 896–903.
[6] D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice-Hall, Upper Saddle River, NJ, 2003.
[7] K. Gottschaldt, Gestalt factors and repetition, in: A Source Book of Gestalt Psychology, Kegan Paul, Trench, Trubner and Co, Ltd, 1938.
[8] I. Rock, An Introduction to Perception, Macmillan Publishing, Co. Inc., New York, 1975.
[9] S.P. Vecera, R.C. O'Reill, Figure-ground organization and object recognition processes: an interactive account, J. Exp. Psychol.: Hum. Percept. Perform. 24 (1998) 441–462.
[10] J.L. McClelland, Putting knowledge in its place: a scheme for programming parallel processing structures on the fly, Cognitive Sci. 9 (1985) 113–146.
[11] F. Restle, Coding theory of the perception of motion configuration, Psychol. Rev. (1979) 1–24.
[12] M. Bolduc, M.D. Levine, A review of biologically motivated space-variant data reduction models for robotic vision, Comput. Vision Image Understanding 69 (2) (1998) 170–184.
[13] Y. Lu, J.Z. Zhang, Q.M.J. Wu, Z.N. Li, A survey of motion-parallax-based 3D reconstruction algorithms, IEEE Trans. Syst. Man Cybern. 34 (4) (2004) 532–548.
[14] S.M. Seitz, et al., A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '06), 2006.
[15] Y. Lu, Z.N. Li, Active video object extraction, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2004.
[16] Y. Lu, Automatic object extraction and reconstruction in active video, Ph.D. Thesis, Simon Fraser University, 2005.
[17] H.J. Zhang, A. Kankanhlli, S.W. Smoliar, Automatic partitioning of full-motion video, ACM Multimedia Syst. 1 (1) (1993).
[18] S.C. Pei, Z. Yu, Efficient MPEG compressed video analysis using macro block type information, IEEE Trans. Multimedia 1 (4) (1999) 321–333.
[19] Y. Rui, T.S. Huang, S. Mehrotra, Constructing table of content for videos, ACM Multimedia Syst. J. (Special Issue Multimedia Systems on Video Libraries) 7 (5) (1999) 359–368.
[20] U. Gargi, R. Kasturi, S.H. Strayer, Performance characterization of video shot change detection methods, IEEE Trans. Circuits Syst. Video Technol. 10 (1) (2000) 1–13.
[21] J.V. Mascelli, The Five C's of Cinematography, Radstone Publications, North Hollywood, CA, 1966.
[22] P.H.S. Torr, D.W. Murray, Statistical detection of independent movement from a moving camera, Image Vision Comput. 1 (4) (1993) 180–187.
[23] P. Beardsley, A. Zisserman, D. Murray, Sequential updating of projective and affine structure from motion, Int. J. Comput. Vision 23 (3) (1997) 235–259.
[24] S. Avidan, A. Shashua, Threading fundamental matrices, IEEE Trans. Pattern Anal. Mach. Intell. 23 (1) (2001) 73–77.

About the Author—YE LU received his Ph.D. degree in computer science from Simon Fraser University, Burnaby, BC, Canada in 2005. His research interest includes computer vision, multimedia, and data compression. He is currently working at Business Objects Inc. where he is developing large scale server applications for enterprise data processing.

About the Author—ZE-NIAN LI is a Professor in the School of Computing Science at Simon Fraser University in Vancouver, Canada. Dr. Li received his undergraduate education in Electrical Engineering from the University of Science and Technology of China, and M.Sc. and Ph.D. degrees in Computer Sciences from the University of Wisconsin-Madison. Earlier in his career he was an electronic engineer in charge of design of digital and analogical systems. His current research interests include computer vision, pattern recognition, multimedia, image processing, and artificial intelligence. Dr. Li is the Director of Vision and Media Lab at Simon Fraser University. He has published over 100 refereed papers in journals and conference proceedings. He is the co-author of the book "Fundamentals of Multimedia" published by Prentice Hall in 2004. Dr. Li was the Director of the School of Computing Science from 2001 to 2004.