Evaluation of a Vision Based 2-Button Remote Control for Interactive Television

Faisal Shafait, Marco Grimm, Rolf-Rainer Grigat
Vision Systems Department
Hamburg University of Technology
Germany
{faisal.shafait,grimm,grigat}@tu-harburg.de

Abstract—This contribution presents a new vision based remote control for interactive television (ViBReCIT) and DVD players. This device is easily and intuitively used due to its vision based pointing ability in combination with only two push buttons. The purpose of this remote control is to use a pointer based scheme for menu selections on the television screen. For this purpose a low-resolution, low-cost camera is integrated on the front side of the remote control beside the IR transmitter. Using the images obtained from this camera, an estimate of the remote (i.e. camera) movement is made. This estimate of camera motion is used to control the pointer position on the television screen. We have implemented both absolute tracking and relative tracking algorithms for motion estimation and have evaluated their performance with respect to this application. An experiment is presented which evaluates this interaction style by comparing it to the status-quo arrow-keys based remote controls. In the experiment participants strongly preferred ViBReCIT over the arrow-keys based remote control, as ViBReCIT not only increased browsing speed through the menu structure but also was more user-friendly.

I. INTRODUCTION

Menu selection in a television set is done today by arrow-keys based remote controls. These remote controls suffer from the inherent problem of sequential or step-by-step movement through the menu items. So in order to move from one menu item to the other, one has to pass through all the menu items in the path. This results in a slower response particularly for complex menu structures, where navigation in all four directions (up, down, left, right) is to be done. The better alternative for menu selections is a pointer based approach. In this approach a pointer is used to navigate through the menu structure, just like a mouse pointer on a computer.

In this paper we present a new pointer based approach for menu selections in a television set with the help of a camera. In order to control a pointer on the screen with the camera, two different approaches were considered.

1) Estimate the position of television screen in the camera image. Then depending on this position relative to image center point, pointer position can be controlled. Details of this approach along with the problems encountered are discussed in section II-A.

2) Use camera ego-motion estimation to move the pointer on the screen. This approach eliminates the need to detect the television screen in the image.

There are many direct approaches to calculate camera ego-motion from video sequences. Acceleration sensors in combination with image analysis have proved to be useful for camera ego-motion estimation [1]. The indirect methods first compute a corresponding feature set or a displacement vector field. The camera parameters are then estimated in a second step based on this information. Broszio et al. [2] have presented a three-step approach for camera parameters estimation. In the first step corners are detected for each image. Then correspondences between features of an image to the next in the sequence are established. The final parameter estimation employs the random sampling procedure RANSAC [3]. Another approach presented by Jones [4] involves camera motion estimation using MPEG motion vectors.

All these approaches involve calculation of camera parameters (pan, tilt, and zoom), which is redundant information for our case. Secondly, only [1] claims to work in real-time (as the others were developed for batch processing on video signals). Many approaches exist which estimate camera ego-motion in real-time. Choon et al. [5] have proposed an optical flow based algorithm to compute camera motion in real time. Spindler [6] has used an approach for robust multi-resolution estimation of parametric motion models to estimate dominant motion for underwater video images. Again, these approaches involve a lot of computations for precise and robust estimation which is not required in our case. We have used a variation of the three step approach by Broszio [2], extracting only the necessary
information of 2-D camera parameter estimation. We use some approximations to reduce the computational complexity. Our approach is discussed in section II-B.

II. METHODS

A. Absolute Tracking using Television Screen Position Estimation

Television screen position estimation is done to use it as a reference for absolute tracking. As a first approach for robust and reliable estimation of television screen position from camera images in real-time, we have embedded particular colors in the menu structure, and detect those colors by color segmentation. After color segmentation a $3 \times 3$ smoothing filter is applied to remove noise and smoothen the boundaries of segmented regions. Then segments corresponding to the region of interest (television screen) are separated from the surrounding regions by using information about shape, size, and relative position of the regions. We use the facts that regions corresponding to color markers in the menu items are equal in size, have same shape, and their centers form four corners of a parallelogram. These steps are illustrated in Fig. 1. In the last stage the pointer position is calculated with reference to the center point of the detected television screen.

Hence we are able to calculate position and orientation of the television screen for moving the pointer on the screen in this constrained menu environment.

B. Relative Tracking Using Camera Ego-Motion Estimation

The motion observed in an image sequence can be caused by camera motion (ego-motion) and by motion of objects moving in the scene. In this paper the case of a camera moving in a static scene is addressed. We have used the simplest (translational) model for 2-D motion. It accounts for a rigid translational 3-D motion under orthographic projection resulting in a spatially-constant 2-D motion. The precision of the motion field also depends on the region of support for the model, i.e., the set of image points to which the model applies. The region of support for the global motion model consists of the whole image. Furthermore, we make the assumption that the image intensity remains constant along the motion trajectory.

The block diagram of the algorithm we have used is as shown in Fig. 2.

1) Reference Template Selection: In the first stage corners are detected from the camera image based on SUSAN corner detector [7]. For a well-structured image, the number of detected corners is quite high. So it is not possible to track all of them in real-time. In order to ensure getting features from every image area, the image can be divided into several tiles [8]. We perform corner selection based on the nearest neighborhood criterion to a reference grid. Corners closest to particular points on the grid in terms of Euclidean distance are selected. The reference grid we have used is as shown in Fig. 3.

2) Template Matching: In order to track the selected corners in the next image, a full search $8 \times 8$ block-matching is applied. The search region for each corner is limited to a $50 \times 50$ search window centered around the position of corner in the reference image. This puts a theoretical limit on the maximum allowed speed which can be estimated. In practice the maximum allowed moving speed is limited by motion blur and not by search window size. Mean absolute difference (MAD) is used as the error criterion for selecting the best match.

3) Dominant Motion Estimation: The template matching module gives us one motion vector for each selected corner. The next task is to estimate the 2-D camera motion, i.e. translation in x and y directions, from these motion vectors. Choon et al. [5] have proposed repeated median filtering for dominant motion estimation. This requires at least more than half of the vectors to have the same value for correct estimation. We use the concept of simple majority for dominant motion estimation. This introduces more flexibility by having freedom to choose the threshold and tolerance for selection among candidate vectors.

III. EXPERIMENT: NAVIGATION THROUGH A MENU STRUCTURE

In order to evaluate the performance of ViBReCIT, and compare it with the existing arrow-keys based remote control we have designed an experiment. The prototype we used for
ViBReCIT is as shown in Fig. 4. We have implemented two types of menu structures for evaluation as shown in Fig. 5.

Fig. 4. Prototype used for ViBReCIT. The red button is used for menu on/off, while the green button is used for menu selection.

Fig. 5. Two types of menu structures used in the experiment. Left menu structure is denoted by M1 while the right menu structure is denoted by M2.

First menu type, called M1, closely corresponds to a menu structure found in today's television sets, in which usually navigation in only two directions (up-down or right-left) is done at one time. This type of menu structure is optimal for arrow-keys based remote control as the list of menu items is arranged either vertically or horizontally so that the users only need two out of four arrow keys for navigation at one time. However, for a pointer-based menu selection the possibility of moving in all four directions is more appropriate as it corresponds to natural real-world expectation of users as they move the remote control. So we have implemented menu type M2 suitable for use with ViBReCIT.

We have tested both devices on each type of menu structure. The actual comparison is between the optimum achieved with arrow-keys based remote control and ViBReCIT, as the structure of menu can always be adjusted for optimum performance. For ViBReCIT, both absolute tracking and relative tracking algorithms are tested to find the most suitable algorithm for it. Therefore we have three competing methods on two types of menu structures as summarized in table I.

A. Task and Stimuli

The task is to perform particular menu selections through both menu structures with each method. We measure the time taken by the user in performing that task. The presentation order of the three methods was counterbalanced across the participants. For each technique, participants performed 10 blocks of trials. Each block consisted of selecting each of the 9 menu items turn by turn, while the next menu item to be selected was presented in a constrained pseudo-random order within the menu structure. The constraint imposed was that the target always appeared on a different item of the menu structure from the menu items previously used as targets. This ensured that participants had to manipulate the camera or arrow-keys in order to select each target, and that each menu item was used as a target only once.

Participants were given three practice trials with each device to familiarize themselves with the task. They were allowed breaks after each block of 9 trials. 10 volunteers participated in this experiment, so the experiment consisted of 5400 total trials. The camera frame rate used for experiments with ViBReCIT was 30 frames/sec. For each subject, the experiment was conducted in one sitting and lasted under 45 minutes. Subjects were alternately assigned to one of three experimental orders. A questionnaire designed to elicit participants' subjective preferences for the three techniques was completed by the participants at the end of the experiment.

B. Results and Discussion

1) Trial Completion Time: Figure 6 compares participants' mean trial completion time for each technique over the 10 blocks of trials. Trial completion time was measured beginning when the first target appeared on the menu structure and ending when the last target was selected.

The results of the experiment show that for linear menu structure M1, arrow-keys based remote control was better than ViBReCIT. However for two-dimensional menu M2, ViBReCIT using absolute tracking was remarkably better than status-quo arrow keys based remote control. Since ViBReCIT was new to participants, the effect of learning is also there. The average completion time of last five blocks of trials is lower than first five blocks of trials for all cases of ViBReCIT. The high variations in trial completion time with ViBReCIT using relative tracking are due to the dependence of relative tracking algorithm on structure in the image. While moving the remote control, users sometimes pointed away from the screen to ill-structured areas like wall. This resulted in poor performance. This event was more frequent in the case of two-dimensional menu structure M2, as large camera motions were necessary to move to different areas of screen.

The results show that after some learning (i.e. after first five blocks of trials) trial completion time with ViBReCIT using absolute tracking with menu structure M2 was 20% faster than status-quo arrow-keys based remote control with menu

<table>
<thead>
<tr>
<th>Method</th>
<th>Menu Type</th>
<th>No. of Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViBReCIT with Absolute Tracking (A.T.)</td>
<td>M1</td>
<td>900</td>
</tr>
<tr>
<td>ViBReCIT with Relative Tracking (R.T.)</td>
<td>M2</td>
<td>900</td>
</tr>
<tr>
<td>Arrow-Keys</td>
<td>M1</td>
<td>900</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>900</td>
</tr>
</tbody>
</table>
structure M1, and 17% faster than arrow-keys based remote control with menu structure M2.

2) Subjective Evaluation: At the end of the experiment, participants were asked to rate their preference for each technique on a scale of -2 (very low) to 2 (very high). For rating the actual comparison is between ViBReCIT on menu structure M2 with arrow-keys based system on menu structure M1. The reason for this is that a menu structure optimal for a particular device can easily be implemented to work with that device. As discussed before, menu structure M1 is optimal for arrow-keys based system, whereas menu structure M2 is optimal for ViBReCIT. The results, summarized in Table II, clearly show the user-friendliness of ViBReCIT using absolute tracking and are consistent with the quantitative trial completion time data. The dependence of relative tracking algorithms with respect to use with ViBReCIT. For this purpose we performed to measure the time taken by the users to navigate through the menu structure implemented in the demonstrator. The experiments showed that navigating through the menu structure with ViBReCIT using absolute tracking was 20% faster than the status-quo arrow keys based remote control. Secondly absolute tracking proved more suitable than relative tracking for use with ViBReCIT. The experiments clearly showed the feasibility of ViBReCIT using absolute tracking as a user-friendly remote control for television or DVD menu selections.

IV. CONCLUSION

We presented a new vision based remote control for interactive television (ViBReCIT). We showed that a camera integrated with the television remote control can be used to control a pointer on the television screen. For this purpose we compared the performance of absolute tracking and relative tracking algorithms with respect to use with ViBReCIT. For absolute tracking, television screen detection from camera images in a constrained environment was used. The constraints were menu items of particular color and fixed lighting condition. The other alternative of relative tracking using camera ego-motion estimation was implemented. An existing algorithm for camera ego-motion estimation was modified such that the new algorithm was suitable for this particular application.

A demonstrator was built to evaluate the performance of relative tracking and absolute tracking algorithms with respect to use with ViBReCIT, and to compare ViBReCIT with the existing arrow-keys based remote controls. Experiments were performed to measure the time taken by the users to navigate through the menu structure implemented in the demonstrator. The experiments showed that navigating through the menu structure with ViBReCIT using absolute tracking was 20% faster than the status-quo arrow keys based remote control. Secondly absolute tracking proved more suitable than relative tracking for use with ViBReCIT. The experiments clearly showed the feasibility of ViBReCIT using absolute tracking as a user-friendly remote control for television or DVD menu selections.

REFERENCES