Video Retrieval of Human Interactions Using Model-Based Motion Tracking and Multi-layer Finite State Automata*

Sangho Park[1], Jihun Park[2], and Jake K. Aggarwal[1]

[1] Department of Electrical and Computer Engineering,
The University of Texas at Austin
Austin, TX 78712
{sh.park,aggarwaljk}@mail.utexas.edu
[2] Department of Computer Engineering, Hongik University
Seoul, Korea
jhpark@hongik.ac.kr

Abstract. Recognition of human interactions in a video is useful for video annotation, automated surveillance, and content-based video retrieval. This paper presents a model-based approach to motion tracking and recognition of human interactions using multi-layer finite state automata (FA). The system is used for widely-available, static-background monocular surveillance videos. A three-dimensional human body model is built using a sphere and cylinders and is projected on a two-dimensional image plane to fit the foreground image silhouette. We convert the human motion tracking problem into a parameter optimization problem without the need to compute inverse kinematics. A cost functional is used to estimate the degree of the overlap between the foreground input image silhouette and a projected three-dimensional body model silhouette. Motion data obtained from the tracker is analyzed in terms of feet, torso, and hands by a behavior recognition system. The recognition model represents human behavior as a sequence of states that register the configuration of individual body parts in space and time. In order to overcome the exponential growth of the number of states that usually occurs in single-level FA, we propose a multi-layer FA that abstracts states and events from motion data at multiple levels: low-level FA analyzes body parts only, and high-level FA analyzes the human interaction. Motion tracking results from video sequences are presented. Our recognition framework successfully recognizes various human interactions such as approaching, departing, pushing, pointing, and handshaking.

1 Introduction

Analysis of video data is important due to the rapid increase in the volume of information recorded in the form of video. Most research has focused on

* This work was partially supported by grant No. 2000-2-30400-011-1 from the Korea Science and Engineering Foundation.

E. M. Bakker et al. (Eds.): CIVR 2003, LNCS 2728, pp. 394–403, 2003.

shot detection [1], video indexing [2], and video summarization [3] by analysis of meta-data. Detailed recognition of human interaction in a video is desired for content-based video retrieval. Recognizing human interactions is a challenging task because it involves segmentation and tracking of deformable human body parts at low level and recognition of semantics in behavior at high level. There have been two types of approaches to human motion analysis: model-based approaches and view-based approaches [4] depending on the availability of an explicitly defined *a priori* model.

This paper presents a model-based approach to motion tracking and recognition of human interaction in widely-available, static-background monocular video sequences. We assume the following constraints: the use of a monocular video camera with fixed parameters and a stationary background, that the projection plane is perpendicular to the camera viewing direction, and that people move parallel to the projection plane within tolerance. We represent human interactions as sequences of states that register the configuration of individual body parts in space and time. In order to overcome the exponential growth of the number of states that usually occurs in single-level finite state automata (FA), we propose a multi-layer FA that abstracts states and events from motion data at multiple levels: low-level FA analyzes body parts only, and high-level FA analyzes the human interaction.

The rest of the paper is organized as follows: Section 2 summarizes previous work related to model-based human tracking and behavior recognition. Section 3 describes an overview of our system. Section 4 describes the procedure of image background subtraction. Section 5 presents human body modeling and cost functional for fitting, and Section 6 explains how to generate finite states and events for behavior recognition. Experimental results and conclusions follow in Sections 7 and 8 respectively.

2 Previous Work

Model-based human tracking aims at estimating the kinematic parameters of body configuration. *Kinematics* deals only with the motion of a body, its displacement, velocity, and acceleration. All kinematics-based motion tracking methods may be classified into two groups: *inverse kinematics-based* and *forward kinematics-based* methods. The inverse method [5, 6] computes joint angles given the end-tip parameters (i.e., the parameters of end points of the body parts), whereas the forward method [7] computes the end-tip parameters given the joint angles.

Morris et al.[5] presented an early model-based work for deriving differential inverse kinematics equations for image overlap. In [5], they used a 2D (two-dimensional) scaled prismatic model for figure fitting, and reduced the singularity problem by working in the projection plane (2D). But the appearance of the singularity is inevitable because this method is based on differential inverse kinematics. Huang, et al., [6] extended the inverse kinematics work presented in [5] to solve motion parameters of the articulated body in a statistical framework

using the expectation-maximization (EM) algorithm. Sidenbladh et al.[7] converted the human motion tracking problem into a probabilistic inference problem aimed at estimating posterior probability of body model parameters given the input image.

The motion data obtained from the human tracker is used for human behavior recognition. We may classify human interaction recognition into two groups: gross-level and detailed-level. Gross-level behavior is relevant to wide-view video data where each person is represented as a small moving blob. Examples include interactions between people such as *approaching*, *departing*, and *meeting* [8, 9]. Detailed-level behavior involves movement of individual body parts such as head, torso, hand, and leg, ect. Examples include interactions such as *hand-shaking*, *pointing*, *pushing*, and *kicking*. [10, 11]. Both levels of recognition are desired for surveillance applications, but most research has focused on the gross-level recognition tasks.

We may classify human motion analysis methods according to the recognition algorithms used: the algorithms are either stochastic, such as hidden Markov models (HMM), or deterministic, such as finite state automata (FA). In general, stochastic methods are useful to handle uncertainty due to image noise, imperfect segmentation / tracking, etc., at low level processes in view-based approaches. If the uncertainty can be effectively resolved by model-based methods at low level, then we can use deterministic methods for interaction recognition. In this case, the reliable fitting of the model body to image data is important.

Many approaches have been proposed for behavior recognition using various methods including hidden Markov models, finite state automata, context-free grammar, etc. Oliver et al.[8] presented a coupled hidden Markov model (CHMM) for gross-level human interactions between two persons such as 'approach', 'meet', 'walk together', and 'change direction'. Hongeng et al.[9] proposed probabilistic finite state automata(FA) for gross-level human interactions. Their system utilized user-defined hierarchical multiple scenarios of human interaction. Hong et al.[11] proposed a deterministic FA for detailed-level recognition of human gestures such as 'hand-waving', 'drawing a circle', and 'drawing a figure 8'. Their system was used for computer games based on a human computer interface. Park et al.[10] proposed a string-matching method using a nearest neighbor classifier for detailed-level recognition of two-person interactions such as 'hand-shaking', 'pointing', and 'standing hand-in-hand'. Wada et al. [12] used nondeterministic finite state automata (NFA) using *state product space*. They preferred NFA to HMM because NFA provides transparent information about state transitions whereas HMM's state transition is *hidden* to the user.

3 Overview of Our System

Our system is motivated by model-based human motion tracking and recognition of human interactions in a surveillance video. We use a 3D (three-dimensional) human body model with 11 degrees of freedom (DOF) using a sphere and cylinders (See figure 1(a).) In order to apply the model-based human tracker, we re-

move the background of the input image using a background subtraction method similar to [13]. A 3D human body model projected to the 2D projection plane is used to fit the foreground image silhouette.

We convert the human motion tracking problem into a parameter optimization problem. A cost functional for optimization is used to estimate the degree of overlap between the foreground input image silhouette and the projected 3D body model silhouette. The degree of overlap is computed using computational geometry by converting a set of pixels from the image domain to a polygon in the real projection plane domain.

The kinematic parameters of the fitted model body are concatenated along the sequence and give the motion parameters, and the motion data is analyzed by a recognition system. We propose a *multilevel* deterministic finite state automata (DFA) as the recognition model. The multilevel DFA is composed of basic-level DFA's to abstract numerical motion data and analyze the motion data with respect to feet, torso, and hands. The low-level DFAs independently represent the individual body-part poses as discrete states and the body part motion as a transition between the states. The high-level DFAs concatenate the outputs of the low-level DFAs along the sequence, and analyze the patterns for the recognition of body gestures and interactions between two persons.

4 Background Subtraction

Our video data involves the use of a monocular video camera with fixed parameters and a stationary background. We transform the color video from RGB color space to HSV (hue, saturation, value) color space to make the intensity or brightness explicit and independent of the chromaticity: $Z \in \{H, S, V\}$. We build the background model in terms of a Gaussian distribution with the mean $\mu_Z(x, y)$ and standard deviation $\sigma_Z(x, y)$ of each color channel, Z, at every pixel location (x, y). The Gaussian parameters μ_Z and σ_Z are estimated using k_b background frames ($k_b = 20$) that do not contain humans.

The foreground image region is segmented by background subtraction performed in each frame [13]. Foreground segregation is performed for every pixel $v = [v_H, v_S, v_V]^T$ as follows: at each image pixel (x, y) of a given input frame, the change in pixel intensity is evaluated by computing the Mahalanobis distance $\delta_Z(x, y)$ from the Gaussian background model for each color channel Z.

$$\delta_Z(x, y) = \frac{|v_Z(x, y) - \mu_Z(x, y)|}{\sigma_Z(x, y)} \tag{1}$$

The foreground image $F(x, y)$ is obtained by choosing the maximum of the three distance measures, δ_H, δ_S, and δ_V for the H, S, V channels;

$$F(x, y) = max[\delta_H(x, y), \delta_S(x, y), \delta_V(x, y)] \tag{2}$$

A binary foreground mask image is obtained by thresholding F. At this stage, morphological operations are performed as a post-processing step to remove small regions of noise pixels.

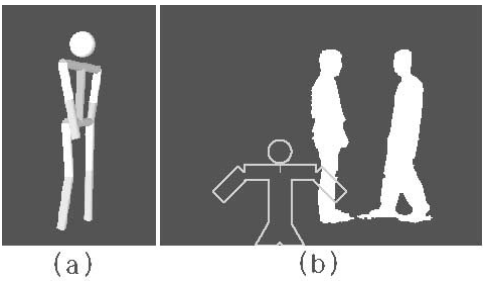

Fig. 1. 3D body model (a), and the initial stage for fitting the model to image (b)

5 Human Body Modeling and Cost Functional

5.1 Human Body Modeling

As shown in fig. 1(a), the body is modeled as a configuration of nine cylinders and one sphere according to anthropometric data. The body model is projected onto a 2D real projection plane. The sphere represents the head, while the rest of model body is modeled using cylinders of various radii and lengths. Our body model is similar to that used in [7]. Currently, we use only nine 1-DOF (degree-of-freedom) joints plus body displacement, and a vertical rotation to compensate for the camera view. These are our control variables for the cost functional. Body parts are linked together by kinematic constraints in a hierarchical manner. This may be considered to be a tree structure with the base at the pelvis (i.e., the bottom of the cylinder representing the torso) of the model body. We have overcome the body occlusion problem using computational geometry, by computing the union of the projected model body parts and then computing the intersection with overlapping input image silhouettes. Fig. 1(b) shows the initial step of optimization.

5.2 Forward Kinematics-Based Cost Functional

Given a set of joint angles and body displacement values, the forward kinematics function, $h(\cdot)$, where $\cdot$ is a generic variable(s), computes the boundary points of individual body segments. The P matrix projects the computed boundary points on a 2D projection plane, which will be compared to a foreground image silhouette. $g(\cdot)$ converts projected points to a polygon(s). The input image is preprocessed using the background subtraction function, $f(\cdot)$. The projection plane is represented in real numbers. $r(\cdot)$ converts the input image silhouette to a polygon(s) in the real number domain. The representation in the real number domain makes the derivative-based parameter optimization possible.

$$\begin{aligned} c(I,\bar{\theta}) = &-w_1 \times [a(r(f(I)) \cap (\cup_l g(P \cdot h_l(\bar{\theta},t))))] \\ &+ w_2 \times \sum_{xy} (w_d(x,y) \times a(d(x,y) \cap (\cup_l g(P \cdot h_l(\bar{\theta},t))))) \end{aligned} \tag{3}$$

Let us explain the notation used in equation (3) in more detail.

$c(I, \bar{\theta})$ is a cost functional for parameter optimization, which depends on a raw input image I and model DOF variables $\bar{\theta}$.
$\bar{\theta}$ is a joint DOF vector, represented in a column matrix. $\bar{\theta}$ is a function of time when a complete sequence is considered.
P is an orthographic camera projection matrix, projecting a 3D body model to the 2D plane.
$h_l(\cdot)$ is a nonlinear forward kinematics function of an l-th body part in terms of joint DOF.
$g(\cdot)$ is a function with the argument 2D input points and converts them to a polygon.
$r(\cdot)$ is a real function with the argument an input imag and converts its foreground (non-zero value) part to a set of polygons, possibly with holes.
$f(\cdot)$ represents a preprocessed foreground image, given a raw image.
I is a raw input image.
$I(x, y)$ denotes a grey level pixel value at image location (x, y).
$d(x, y)$ is a square polygon of area size 1, representing a pixel located at the (x, y)-th position in the distance map [14].
t represents time related with frames.
$w_d(x, y)$ is a distance map value at position (x, y), and has a scalar value.
$\cap$ is an operation that takes two polygons and returns their intersection.
$\cup$ is an operation that takes two polygons and returns their union.
$a(\cdot)$ is a function that gives the area of a polygon.
w_i , $i = 1, 2$ are weighting factors.

Because the vector of joint DOF variables, $\bar{\theta}$, is the input to the optimization process, this computation is purely forward kinematics-based and thus presents no singularity problems. We may limit the range of joint variation for individual model-body parts. If there is an occlusion between two persons, our method relies on a distance map [14] for each person to be tracked correctly.

5.3 Computational Geometry for the Cost Functional

An image silhouette is one that is converted from the 2D integer pixel domain to a real domain such that the resulting image silhouette becomes a jagged-edge polygon with only horizontal and vertical edges. The resulting polygons may have holes in them. We compute polygon intersection between the input image silhouette and the model silhouette. Pixel-based computational geometry is needed to compute the distance map [14] that is used to fit the model to foreground image regions. Our computation is a modified version of Weiler-Atherton's polygon union/intersection computation[15]. We found the best fitting configuration using the GRG2[16] optimization package with the cost functional in equation (3). Figure 2 shows possible cases where either a model head or polygon-shaped body overlaps with a pixel. In figure 2, the circle represents a projected head outline and an oblique polygon represents a body part, while a square represents a pixel. The union of these irregular-shaped objects results in a projected model body silhouette. After obtaining the union, a triangulation process is used to compute the area of the union of polygon-shaped object. It may be noted that

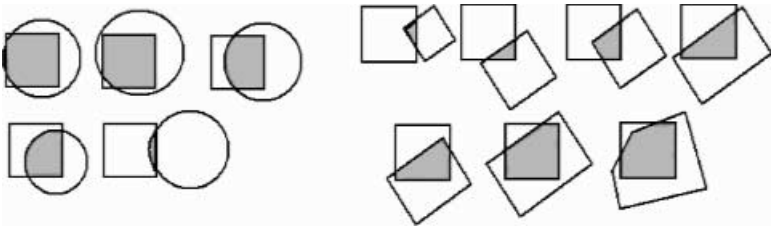

Fig. 2. Five possible cases of a pixel(square) partially occluded by a model head, and seven possible cases of a pixel(square) partially occluded by a polygon body

there are arcs involved in the computation. Our cost functional does not allow ridges, an area of a function with zero gradient. As a consequence, we cannot work on a purely pixel-based integer cost functional. Therefore, we compute the area of overlap between pixels and the projected body model in the sub-pixel accuracy in order to eliminate ridges in our cost functional. Even a small change in the 3D body model is reflected in the cost functional.

6 Multi-layer Deterministic Finite State Automata (DFA) for Behavior Recognition

We employ a *sequence analyzer* and an *event detector* that are similar to those of [12]. The sequence analyzer is a DFA, while the event detector allows state transition. Our finite state automata consists of a finite set of states(Q), an initial state(q^0), a finite set of events($\sum$), a state transition function(δ), and a finite set of final states(F). The sequence analyzer is represented by $(Q, q^0, \sum, \delta, F)$. Each intermediate state q^i in the sequence $(q^0, q^1, \cdots, q^n)$ corresponds to a frame. The event detector analyzes motion data obtained from a parameter optimization sequence and detects events. We designed separate sequence analyzers (DFAs) for each body part: hands, body center (torso), and feet. We consider all possible states for feet, hands and torso, independent of the rest of the body. These DFAs abstract motion data from each body part. We employ a higher-level DFAs to handle motion recognition given low-level motion abstraction. For a single person, we employ three low-level sequence analyzers. Since there can be more than one person present in a scene, our low level sequence analyzer is of the form $({}^p_m Q, {}^p_m q^0, {}^p_m \sum, {}^p_m \delta, {}^p_m F)$, where $p \in \{1, 2, 3\}$ is an index for body parts, and m is an index for each person in the scene. ${}^1_2 q^i \in {}^1_2 Q$ means ${}^1_2 q^i$ is a state of sequence index number i, of a *second* person in the scene, of body part index *one*.

The use of multi-layer DFAs reduces the large number of states to be handled in interaction recognition. Assuming that there are two persons in a scene to generate motion states of the model body, each person has 27 (3^3) possible states (three states for each of three body parts.) If two persons are involved in an interaction, the DFA will classify 729 (27^2) states. Generally, we need

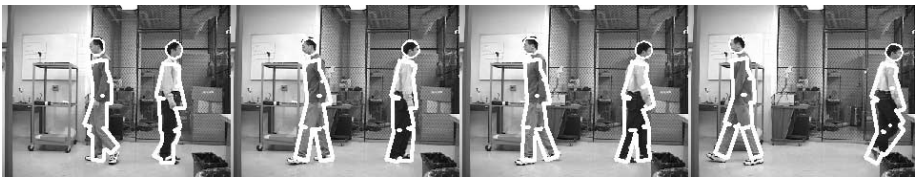

Fig. 3. The subject with the model figure superimposed, shown over a walking motion

$|{}^1_1Q| \times |{}^2_1Q| \times |{}^3_1Q| \times |{}^1_2Q| \times |{}^2_2Q| \times |{}^3_2Q|$ states for an interaction of two persons, where $|Q|$ is the number of states in Q. This exponential growth quickly becomes intractable. Rather than generating 729 states and designing state transitions, we design three states to recognize each motion of a body part, totaling nine states for a single person. Then we consider a tuple of states, $({}^1_m q^i, {}^2_m q^i, {}^3_m q^i)$, a token made of low-level state transitions, to recognize the motion of a person with an index number m. To recognize an interactive motion between two persons, we use a tuple of states, $({}^1_1 q^i, {}^2_1 q^i, {}^3_1 q^i, {}^1_2 q^i, {}^2_2 q^i, {}^3_2 q^i)$. Therefore, we design a higher-level DFA to recognize behavior based on lower-level sequence analyzers, plus nine lower-level states to abstract motion data rather than 729 state and state transition designs. The high-level DFA also refers to the data related to the low-level state transition. Figure 3 shows an example. For the departing/approaching motion, abstract motion data allows us to recognize that two persons are walking, but we cannot tell whether they are departing/approaching without referring to the distance between the two persons. (Refer figure 3.)

The high-level DFAs analyze the interaction in a cause and effect framework. Figure 4 shows an example of 'pushing' interaction, in which the right person approaches and pushes the left person (cause), and the left person moves backward as a result of being pushed (effect). Our observation shows that meaningful interactions are characterized by a sequence of specific states belonging to relatively small number of states. Therefore, our high-level DFAs focus on a subset of all possible states instead of analyzing all 729 states. In the 'pushing' interaction, we define a minimum of four states to recognize the pushing motion: two states representing the contact states by the left/right pushing person, respectively, and the other two states representing the moving backwards of the right/left pushed person, respectively.

7 Experimental Results

We have analyzed several different 2D-based human motions; walking (i.e., approaching, departing), pushing, kicking, pointing, and hand-shaking. The example shown in figure 5 involves two persons in hand-shaking interaction. The body model properly fitted to image sequence shows the effectiveness of our geometry union process. The model body cannot distinguish between the left/right side of the body to be tracked. This limitation is the result of using monocular video

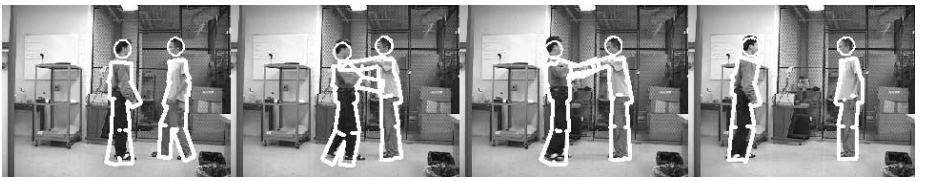

Fig. 4. The subject with the model figure superimposed, shown over a pushing motion

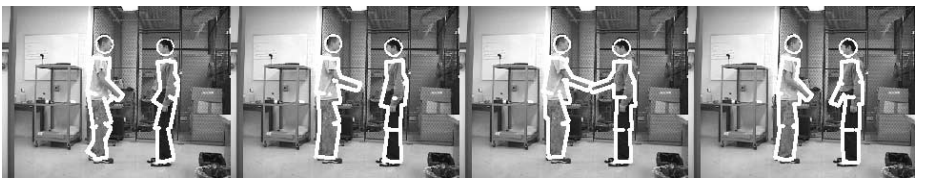

Fig. 5. The subject with the model figure superimposed, shown over a hand-shaking motion

input, which provides very limited input information. Our method finds one locally optimum solution from a search space. The motion tracking is excellent, as shown in figures 3, 4, and 5. After motion tracking, we get two sets of motion data in terms of frames, one for each person appearing in the scene. The multilevel DFA processes the motion data to recognize human interactions; the low-level DFA analyzes the motion of individual body parts to generate states, and the high-level DFA analyzes the sequence of the state changes to generate the recognition results of the interaction. Recognition of interaction is achieved when the multilevel DFA stops in a final accepting state. Computation time depends on the degree of accuracy that is sought and the size of the input image.

8 Conclusion

In this paper, we have presented a model-based approach to human motion tracking and a multilevel DFA-based approach to behavior recognition. The model based-method uses a 3D human body model and parameter optimization techniques to track the moving humans. Our forward kinematics-based system overcomes the problem of singularity in inverse kinematics-based systems. We have presented a solution to the model body part occlusion problem using computational geometry. The motion data of the body model enables us to apply our DFAs to the problem of recognizing human interaction. We used a multilevel DFA to overcome the exponential growth of the number of states in usual single-level DFA. The results of motion tracking from video sequences are excellent for a number of activities, and our recognition framework successfully recognizes various human interactions between two persons.

References

[1] Kim, K., Choi, J., Kim, N., Kim, P.: Extracting semantic information from basketball video based on audio-visual features. Lecture Notes in Computer Science **2383** (2002) 268–277 395

[2] Chang, Y., Zeng, W., Camel, I., Aonso, R.: Integrated image and speech analysis for content-based video indexing. In: IEEE proc. Int'l Conference on Multimedia Computing and Systems. (1996) 306–313 395

[3] Denman, H., Rea, N., Kokaram, A.: Content based analysis for video from snooker broadcasts. In: Int'l Conference on Image and Video Retrieval, Lecture Notes in Computer Science. Volume 2383., Springer (2002) 186–193 395

[4] Aggarwal, J., Cai, Q.: Human motion analysis: a review. Computer Vision and Image Understanding **73(3)** (1999) 295–304 395

[5] Morris, D., Rehg, J.: Singularity analysis for articulated object tracking. In: Computer Vision and Pattern Recognition, Santa Barbara, California (1998) 289–296 395

[6] Huang, Y., Huang, T. S.: Model-based human body tracking. In: International Conference on Pattern Recognition. Volume 1., Quebec city, Canada (2002) 552–555 395

[7] Sidenbladh, H., Black, M. J., Fleet, D. J.: Stochastic tracking of 3d human figures using 2d image motion. In: ECCV (2). (2000) 702–718 395, 396, 398

[8] Oliver, N. M., Rosario, B., Pentland, A. P.: A Bayesian Computer Vision System for Modeling Human Interactions. IEEE Trans. Pattern Analysis and Machine Intelligence **22** (2000) 831–843 396

[9] Hongeng, S., Bremond, F., Nevatia, R.: Representation and optimal recognition of human activities. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2000) 818–825 396

[10] Park, S., Aggarwal, J.: Recognition of human interaction using multiple features in grayscale images. In: Int'l Conference on Pattern Recognition. Volume 1., Barcelona, Spain (2000) 51–54 396

[11] Hong, P., Turk, M., Huang, T. S.: Gesture modeling and recognition using finite state machines. In: IEEE Conf. on Face and Gesture Recognition. (2000) 396

[12] Wada, T., Matsuyama, T.: Multiobject behavior recognition by event driven selective attention method. IEEE transaction on Pattern Analysis and Machine Intelligence **22** (2000) 873–887 396, 400

[13] Park, S., Aggarwal, J.: Segmentation and tracking of interacting human body parts under occlusion and shadowing. In: IEEE Workshop on Motion and Video Computing, Orlando, FL (2002) 105–111 397

[14] Gavrila, D. M., Philomin, V.: Real-time object detection using distance transforms. In: Proc. IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany (1998) 274–279 399

[15] Hill, F.: Computer Graphics. Macmillan (1990) 399

[16] Lasdon, L., Waren, A.: GRG2 User's Guide. (1989) 399