# Detection of Human Bodies using Computer Analysis of a Sequence of Stereo Images

Jurij Leskovec,
Šentjošt, Slovenia
Jure.Leskovec@ijs.si

**Abstract**

The aim of this research project was to integrate different areas of Artificial Intelligence into a working system for detection and tracking of moving objects and recognition of its position in 3-D space.

*Phantom* is a *real time* system for tracking people and determining their position in 3-D using monochromatic stereo imagery. *Phantom* represents the integration of a real-time stereo system with a real-time object detection and tracking system using Support Vector Machines to increase its reliability. I use STH-V1 stereo camera in combination with SVS, a real-time computer system for computing dense stereo range images which was recently developed by SRI.

*Phantom* has been designed to work with visible monochromatic video sources. Unlike many systems for tracking *Phantom* makes no use of color cues, so that it can operate in outdoor surveillance tasks and for low light level situations. *Phantom* is implemented under Windows NT OS in C++ on a single processor Pentium II PC and can process between 20-25 frames per second depending on the image resolution and the number of objects being tracked.

The procedure of detection of new objects consists of several steps. The range image is calculated first. When a new object is detected, new deformable contour is estimated close enough to the object's silhouette. For each object being tracked a deformable contour is fitted to the object's silhouette. After initialization, for each image from the sequence a deformable contour fitting algorithm is applied.

In machine learning phase of the project I compiled a large set of human silhouettes to account for the wide variability of human body shapes. My first approach to human body recognition by silhouette involved the human body outline database, converting it into polar coordinate system and performing Fast Fourier Transform on the resulting data set. This resulted in a compact representation of the original image data in terms of the FFT coefficients. I took many sets of test FFT coefficients generated from the object's deformable contour as training examples for Machine Learning (ML) algorithms (K Nearest Neighbour and Support Vector Machines). The model generated by the ML algorithm was used to classify a new object's silhouette FFT coefficients into different classes. For instance, one such problem could be to identify an object as *human* or *not human*. Each object was represented by its most influent FFT coefficients.

*Phantom* is a real-time visual surveillance system which tracks humans and tries to answer questions about *What* they are doing and *Where* and

*When* they act. *Phantom* represents the integration of a real-time stereo system for computing range images with a real-time silhouette based person detection and tracking system and machine learning algorithms (Support Vector Machines) for object recognition. *Phantom* is capable of simultaneously tracking multiple objects and identifying them.

# 1 Introduction

Since the advent of digital computer there has been a constant effort to expand the domain of computer applications. Some of the motivation for this effort comes from important practical needs but also some from the challenge of programming a machine to do things that machines have never done before. Both kinds of motivation could be found in the area of artificial intelligence called computer vision.

At present the ability of machines to perceive their environment is very limited. When the environment is carefully controlled and the signals have a simple interpretation perceptual problems become trivial. But as we move beyond having a computer read punched cards to having it read hand-printed characters we move from problems of sensing the data to much more difficult problems of interpreting the data.

## 1.1 Problem Statement

The aim of this project was to integrate different areas of Artificial Intelligence into a working system for detection and tracking of moving objects and recognition of its position in 3-D space. The system should run in *real time* on a PC compatible computer with one processor. The input is a sequence of pairs of stereo and grayscale images.

The requirements can be stated as follows:

1. Use a stereo system to obtain sparse range data of a scene.
2. Determine when a new object enters the system's field of view, and initialize a model for tracking that object.
3. Efficiently separate the object from the background.
4. Employ tracking algorithms to update the position of each object.
5. Use Machine Learning algorithms to interpret the type of objects and the position of its parts in 3-D space.

In this paper I will describe the computational models and algorithms employed by *Phantom* to detect and track people. Sequence of disparity images is the input of *Phantom* and Section 3 will in brief examine the basics of stereopsis. For tracking objects on disparity images *Phantom* uses deformable contours described in Section 4. In next two sections I discuss object tracking methods and data interpretation phase.

## 2 Related Work

Pfinder [?] is a real-time system for tracking a person which uses a multi-class statistical model of color and shape to segment a person from a background scene. It finds and tracks people's head and hands.

[?] is a general purpose system for moving object detection and event recognition where moving objects are detected using change detection and tracked using first-order prediction and nearest neighbour matching. Events are recognized by applying predicates to a graph formed by linking corresponding objects in successive frames.

KidRooms [?] is a tracking system based on mixture models and recursive Kalman and Markov estimation to learn and recognize human dynamics [?].

Real-time stereo systems have recently became available and applied to detection of people. Spfinder [?] is a extension of Pfinder in which a wide baseline stereo camera is used to obtain 3-D models. Spfinder has been used in a small desk-area environment to capture accurate 3-D movements of head and hands. Kanade [?] has implemented a Z-keying method, where a subject is placed in correct alignment with a virtual world. SRI has been developing a person detection system which segments the range image by first learning a background range image and then using statistical image compression methods to distinguish new objects [?], hypothesized to be people.

## 3 Stereopsis

*Stereo vision* refers to the ability to *infer information on the 3-D structure and distance of a scene from two or more images taken from different viewpoints.* One can observe the basic principles of a stereo system through a simple experiment. Hold one thumb at arm's length and close the left and right eye alternatively. One finds that the relative position of the thumb and the background appears to change, depending on which eye is open. It is precisely this difference in retinal location that is used by the brain to reconstruct a 3-D representation of what we see.

### 3.1 The Two Problems of Stereo

From a computational standpoint, a stereo system must solve two problems [?].

1. The first, known as *correspondence*, consists in determining *which item in the left eye corresponds to which item in the right eye.* A rather subtle problem here is that some parts of the scene are visible by one eye only (see Figure 2).

   This problem can be solved using correlation based methods. One approach is to match *image windows* of fixed size where the similarity criterion is a measure of the correlation between windows in the two images. The corresponding element is given by the window that maximizes the similarity criterion within search region.

The window size is a compromise, since small windows are more likely to be similar in images with different viewpoints, but larger windows increase the signal-to-noise ratio. Figure (6) shows a sequence of disparity images using window sizes from 7x7 to 13x13. Large windows tend to "smear" foreground objects, so that the image of a close object appears larger in the disparity image than in the original input image but they have a better signal-to-noise ratios, especially for less-textured areas.

2. When the correspondence is solved the coordinates of a 3-D point can be computed from its corresponding image points in both frames using trigonometric rules.

   This second problem is called *reconstruction*. The distance between corresponding items in the left and right frame is called *disparity*.

For solving this problem we have to know both *internal* and *external* parameters of the stereo system. We use triangulation and the disparity information (see Figure 3).

*Internal parameters* describe the distortions introduced in each individual camera by imperfect lenses and lens placement (radial distortion and lens decentering).

*External parameters* define the relative position of both cameras to each other. For stereo matching to work well, the camera image planes must be co-planar, and corresponding scan lines should match.

### 3.2 Disparity

Figure (4) displays stereo geometry. Two images of the same object are taken from different viewpoints. The distance between the viewpoints is called the *baseline* ($b$). The focal length of the lenses is $f$. The horizontal distance from the image center to the object image is $d_l$ for the left image, and $d_r$ for the right image.

Normally, we set up the stereo cameras so that their image planes are embedded within the same plane. Under this condition, the difference between $d_l$ and $d_r$ is called the *disparity*, and is directly related to the distance $r$ of the object normal to the image plane. The relationship is:

$$r = \frac{bf}{d} \qquad \text{where} \qquad d = d_l - d_r \tag{1}$$

The range calculation of Equation (1) assumes that the cameras are perfectly aligned, with parallel image planes.

### 3.3 Horopter

Stereo algorithms typically search only a window of disparities. In this case, the range of objects that they can successfully determine is restricted to some interval. The *horopter* is the 3-D volume that is covered by the search range of the stereo algorithm. The horopter depends on the camera parameters and stereo baseline, the disparity search range, and the X offset. Figure (5) shows a typical horopter. The stereo algorithm searches a 16-pixel range of disparities to find a match. An object that has a valid match must lie in the region between the two planes shown in the figure (figure 8).

### 3.4 Video and Stereo Hardware

*Phantom* computes stereo using area (sum of absolute difference) correlation after a Laplacian of Gaussian transform. The stereo algorithm considers 16, 24 or 32 disparity levels, performs postfiltering with an interest operator and a left-right consistency check, and finally does $4\times$ range interpolation. The stereo computation is done on the host PC. This option provides me the access to much better cameras that those used in STH-V1 [**?**]. SVS is a software implementation of area correlation stereo which was implemented and developed by Kurt Konolige at SRI. The hardware STH-V1 consists of two parallel CMOS 320x240 grayscale imagers and lenses, low power A/D converter and a digital signal processor. A detail description of SVS can be found in [**?**]. SVS performs stereo at different resolutions up to 640x480, but the STH-V1 has the resolution 320x120. The speed is about 40 frames per second. The STH-V1 uses CMOS imagers; these are an order of magnitude noisier and less sensitive than corresponding CCD's. Higher quality cameras can be utilized by *Phantom* to obtain better quality disparity images.

## 4 Snakes

We would like to fit a curve of arbitrary shape to a set of image edge points. We shall deal with *closed contours* only.

A widely used computer vision model to represent and fit general, closed curves is a *snake*, or *active contour*, or again *deformable contour* [**?**]. We can think of a snake as an elastic band of arbitrary shape, sensitive to intensity gradient. Snake is located initially near the image contour of interest, and is attracted towards the target contour by forces depending on intensity gradient (see Figure 7).

The key idea of deformable contours is to associate an *energy functional* to each possible contour shape, in such way that the image contour to be detected corresponds to a minimum of the functional. Typically the energy is a sum of several terms, each corresponding to some force acting on the contours. Each term has also a weight which controls a relative influence of each term [**?**].

Consider a contour, $c = c(s)$, where $s$ a vertex from the contour. A suitable energy functional, $\varepsilon$, consists of the sum of three terms:

$$\varepsilon = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image})ds \tag{2}$$

where each of the terms $E_{cont}, E_{curv}$ and $E_{image}$ is a function of $c$ or of the derivatives of $c$ with respect to $s$. The parameters $\alpha, \beta$ and $\gamma$ control the relative influence of the corresponding energy term.

Each energy term serves a different purpose. The terms $E_{cont}$ and $E_{curv}$ encourage *continuity* and *smoothness* of the deformable contour; they can be regarded as a form of internal energy. $E_{image}$ account for *edge attraction*, dragging the contour toward the closest image edge; it can be regarded as a form of external energy.

1. **Continuity Term.** We can exploit simple analogies with physical systems to devise a rather natural form of the continuity term.

$$E_{cont} = \left\| \frac{dc}{ds} \right\|^2 \tag{3}$$

   In the discrete case, the contour $c$ is replaces by a chain of $N$ image points $p_1, p_2, \ldots, p_N$, so that:

$$E_{cont} = \left\| p_i - p_{i-1} \right\|^2 \tag{4}$$

   A better form for $E_{cont}$, preventing the formation of clusters of snake points, is:

$$E_{cont} = \left( \overline{d} - \left\| p_i - p_{i-1} \right\| \right)^2 \tag{5}$$

   with $\overline{d}$ the average distance between the pairs $(p_i, p_{i-1})$.

2. **Smoothness Term.** The aim of the smoothness term is to *avoid oscillations* of the deformable contour by penalizing high contour curvatures. Since $E_{cont}$ encourages equally spaces points on the contour, the curvature is well approximated by the second derivative of the contour;

   hence we can define $E_{curv}$ as

$$E_{curv} = \left\| p_{i-1} - 2p_i + p_{i+1} \right\|^2 \tag{6}$$

3. **Edge Attraction Term.** The third term corresponds to the energy associated to external force attracting the deformable contour towards the desired image contour. This can be achieved by a simple function [**?**]:

$$E_{image} = -\left\| \nabla I \right\| \tag{7}$$

   where $\nabla I$ is the spatial gradient of the intensity image $I$.

Clearly $E_image$ becomes very small (negative) wherever the norm of the spatial gradient is large (near image edges), making $\varepsilon$ small and attracting the snake towards image contours. The contour fitting method is based on the minimization of the energy functions (2) [**?**] which is described in next section.

## 4.1 Greedy Algorithm

Let $I$ be an image and $\overline{p_1}, \ldots, \overline{p_N}$ the chain of image locations representing the initial position of the deformable contour.

Starting from $\overline{p_1}, \ldots, \overline{p_N}$ find the deformable contour $p_1, \ldots, p_N$, which fit the target image contour best, by minimizing the energy functional:

$$\sum_{i=1}^{N} (\alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{image}) \tag{8}$$

with $\alpha_i, \beta_i, \gamma_i > 0$ and $E_{cont}, E_{curv}$ and $E_{image}$ as in (5), (6) in (7);

Of the many algorithms proposed to fit a deformable contour, I have selected a *greedy algorithm.* A greedy algorithm makes *locally optimal choices*, in the hope that they lead to a *globally* optimal solution. Among the reasons for selecting the greedy algorithm I emphasize its low computational complexity and usabitily systems working in real time.

The core of a greedy algorithm for the computation of a deformable contour consists of two basic steps. First, at each iteration, each point of the contour is moved within a small neighbourhood to the point which minimizes the energy functional. Second before starting a new iteration, the algorithm removes or inserts new points into the chain so that the average distance between a pair of points is around a user defined constant value.

**Step 1: Greedy Minimization.** The area over which the energy functional is locally minimized is typically small (for instance, a $5 \times 5$ or $9 \times 9$ window centered at each contour point). Keeping the size of the small lowers the computational load of the method (complexity being linear in the size of the ). The local minimization is done by direct comparison of the normalized energy functional values at each location.

**Step 2: Insertion and removal of points.** During the second step, the algorithm examines distance between a pair of consecutive points and if they are too far away it inserts $n$ points so that they are distanced for $d_u \pm 50\%$ (a user defined value). If the distance is too small (smaller than $d_u$) it removes a point from the contour.

For a correct implementation of the method, it is important to normalize the contribution of each energy term. For the terms $E_{cont}$ and $E_{curv}$, it is sufficient to divide by the largest value in the in which the point can move. For $E_{image}$ instead it is useful to normalize the norm of spatial gradient $\|\nabla I\|$ as:

$$\frac{\|\nabla I\| - m}{M - m} \tag{9}$$

with $M$ in $m$ maximum and minimum of $\|\nabla I\|$ over the .

The iterations stop when a predefined fraction of all points reaches a local minimum; however the algorithm's greed does not guarantee convergence to the global minimum. It usually works very well as far as the initialization is not too far from the desired solution.

**Algorithm SNAKE:**

The input is formed by an intensity image, $I$, which contains a closed contour of interest, and by a chain of image locations, $p_1, \ldots, p_N$, defining the initial position and shape of the snake. $d_u$ is a minimal distance between a pair of consecutive snake points.

Let $f$ be the minimum fraction of snake points that must move in each iteration before convergence, and $U(p)$ a small of point $p$. In the beginning, $p_i = \overline{p_i}$ and $d = \overline{d}$ (used in $E_{cont}$).

**1.** for each $i = 1, \ldots, N$ find location of $U(p)$ for which the functional $\varepsilon$ defined in (3) is minimum, and move the snake point $p_i$ to that location

**2.** for each $i = 1, \ldots, N$ calculate distance $d_p$ between $p_i$ and $p_{i+1}$. If $d_p < d_u$ then remove one of two snake points. If $d_p > 2d_u$ then insert a new snake point to position $p_{i+1}$.

**3.** Update the value of average distance $\overline{d}$.

On output this algorithm returns a chain of points$p_i$ that represent a deformable contour.

# 5 Tracking of Objects

The goals of the object tracking are:

- determine when a new object enters the system's field of view and initialize data structures for tracking that object.
- employ tracking algorithms to estimate the position of each object and update data structures used for tracking.

I assumed that at the initialization stage of the system there are no objects in the scene. At the initialization stage the camera calibration is done, some other tracking parameters are calculated and basic models are initialized.

1. The procedure of detection of new objects consists of several steps. The range image is calculated first. The system then searches for any new objects in the scene so that already tracked objects are not detected as new in the scene. $N$ random pixels in range image are checked to identify chunks of pixels closer to the camera.

2. When a new object is detected, new deformable contour is estimated close enough to the object's silhouette (see Figure 10).

3. For each object deformable contour is fitted to the object's silhouette. After initialization, for each image from the sequence the deformable contour fitting algorithm is applied. For the estimation of new contour of interest a previous fitted deformable contour is taken. This drastically decreases the computational cost of the execution of the algorithm, because in the time between two consecutive images I assume that the object can not

move far away from its original position. Due to *real time* constraints and the fact that object can not move to much from one frame to the next algorithm SNAKE is run in only one iteration for each image. For each point from the chain representing a deformable contour its distance from the camera is also calculated on the basis of the range image (Figure 9).

The problems appear what to do when two or more objects occlude. In this case tracking must still work and after the objects are not occluded, each tracking model has to track appropriate object (the one who was tracked before occlusion). Another critical moment is when an object splits into pieces (possibly due to a person depositing an object in the scene, or a person being partially occluded by a small object). Finally separately tracked objects might merge into one because of interactions between people. Under these conditions deformable contours would fail, and the system instead relies on stereo to locate the objects. Stereo is very helpful in analyzing occlusion and intersection.

Object tracking can be employed either on range or intensity image. Each of them has its faults and benefits.

The main advantage of the intensity-based analysis are that range data may not be available in background areas which do not have sufficient texture to measure disparity with high confidence. Changes in those areas will not be detected in the disparity image. Foreground regions detected by the intensity-based method have more accurate shape (silhouette) then the range-based method.

However there are more important situations where the stereo-based tracking algorithm has advantages over the intensity image. When there is a sudden change in illumination the intensity-based method could fail but the stereo-based tracking is not effected by the illumination changes over short periods of time. Shadows which makes intensity detection and tracking harder do not cause a problem in the disparity images as the disparity image does not change from the background model when a shadow is cast on the background.

Pixels inside a close contour are that object's picture while other pixels the background. This approach to the object tracking problem has advantages in comparison with other technics.

## 6 Data Interpretation

For detecting the object position two basic approaches are suggested. One hand we have a model based techniques where we try to estimate and calculate model's parameters on the basis of captured data. And on the other hand we can use machine learning. I used machine learning because of the properties and complexity of human body. Human body is far too complex for a good model description and there are numerous different positions of our limbs which the model should cover. Human body is also not rigid, so that the model design should also consider this fact. The problem with machine learning algorithms is that they have to be efficiently trained on both positive and negative cases and they are sensitive to noise, which is highly present in computer vision.

In the beginning of the data interpretation process the position of points from the chain presenting a deformable contour is normalized and transformed into polar coordinate system. We do this transformation that we calculate the center of the silhouette and then draw a circle so that the silhouette intersects our circle as many times as possible. After this we have two functions for each point we know its distance from the circle (if it is positive then it lies outside and if it is negative then it lies inside the circle) and another describes the change of the angle for each point from the contour.

We do this transformation in order to get a periodic function which intersects zero axis many times. Then on this data a discrete Fast Fourier Transform (FFT) is applied. As an output we get 2 sets of coefficients, one for angle and the other for distance.

*Learning phase:* I compiled a large set of human silhouettes to account for the wide variability of human body shapes in the real world. My first approach to human body recognition by silhouette involved the human body outline database and performing Fast Fourie Transform on the resulting data set (Figures 11 and 13). This resulted in a compact representation of the original image data in terms of the FFT coefficients. The first coefficients represent the characteristic features of the human body distribution; the last coefficients represent noise. For example, preliminary findings have shown that I need fewer that 64 FFT coefficients from each set to account for most of variability in human body data.

I took many sets of test FFT coefficients generated from the object's deformable contour as training examples for Machine Learning algorithms. Two machine learning methods were used: Support Vector Machines and k-Nearest Neighbour.

The model generated by the ML algorithm was used to classify a new object's silhouette FFT coefficients into classes. For instance, one such problem could be to identify an object as *human* or *not human* (see Figures 11 and 13). Each object was represented by its most influent FFT coefficients.

In one of many experiments I compiled a set of 100 positive (humans) and 100 negative (no humans: hogs, cars, human body parts, ...) examples and then I made a cross validation text, which means that I took 1 of 200 examples out of the training set, train the sistem on 199 examples and then try to classify that one example. I did this for all 200 sets and the sistem was able do make a correct prediction in 85.6% of cases. I used Support Vector Machines for classification.

# 7 Conclusion and Results

The result of the project is a working application *Phantom*. *Phantom* is a real time system for tracking people and determining their position in 3-D space using monochromatic stereo imagery. *Phantom* represents the integration of a real-time stereo (SVS) system with a real-time object detection and tracking system to increase its reliability. STH-V1 [**?**] is a compact, inexpensive stereo camera which in combination with SVS [**?**], a real-time computer system for computing dense stereo range images which was recently developed by SRI.

*Phantom* has been designed to work with visible monochromatic video sources. While most previous work on detection and tracking of people has relied heavily on color cues. *Phantom* is also suitable for operating in outdoor surveillance tasks and for low light level situations. In such cases color will not be available and people need to be detected and tracked based on weaker appearance and disparity cues. *Phantom* is implemented under Windows NT OS in C++ on a single processor Pentium II PC and can process between 15-25 frames per second depending on the image resolution and the number of objects being tracked.

The incorporation of stereo allowed me to overcome the difficulties due to sudden illumination changes, shadows and occlusions. Even low resolution range maps allow the system to continue to track objects successfully, since stereo analysis is not significantly effected by sudden illumination changes and shadows, which make tracking much harder in intensity images. *Phantom* currently operates on video taken from a stationary camera but its image analysis algorithms can be easily generalized to images taken from a moving camera.

## 7.1 Future Work

There are several directions that I am pursuing to improve the performance of *Phantom* and to extend its capabilities. First some optimization on active contour tracking could be done. Second, I would like to be able to recognize and track people in other generic poses, such as crawling, climbing, etc. I believe this might be accomplished based on analysis of silhouettes of people which I am currently using (the result of tracking is a 3-D silhouette of an object).

In the long run *Phantom* will be extended to recognize actions of the people it tracks. Specifically, I am interested in interactions between people and objects, e.g. people exchanging objects, leaving objects in the scene, taking objects from the scene.

## List of Figures


1 STH-V1 stereo camera. . . . . . . . . . . . . . . . . . . . . . . . . 13
2 An illustration of the correspondence problem. A matching between corresponding points of an image pair is established. . . . 13
3 A simple stereo system. 3-D reconstruction is depends on the solution of the correspondence problem (a); depth is estimated from the disparity of corresponding points (b). . . . . . . . . . . 13
4 Definition of disparity: offset of the image location of an object. . 14
5 Horopter planes for a 16-pixels disparity search. . . . . . . . . . . 14
6 Effects of the area correlation window size. The images show windows of 7x7, 9x9, 11x11 and 13x13 pixels. . . . . . . . . . . . 15
7 From left to right, images show initial position of the snake. Intermediate position and final position. . . . . . . . . . . . . . . 15
8 Planes of constant disparity for verged stereo cameras. A search range of 5 pixels can cover different horopters, depending on how the search is offset between the cameras. . . . . . . . . . . . . . . 16
9 On the left is intensity image with corresponding Disparity image. Right image shows the tracked object separated from the background. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
10 Two situations when a new object is detected. A grayscale image represents the scene and black-white image represents the new object found on the basis of the stereo image. . . . . . . . . . . . 16
11 A Human Object with it's sillhouete, range image and a silhouette with a cicle for transformation of the contour in polar coordinate system. . . . . . . . . . . . . . . . . . . . . . . . . . . 17
12 Typical graph for a contour of a human, radius and angle. . . . . 17
13 Typical graph for a contour of a non-human, radius and angle. . 18
14 Human objects traing set. . . . . . . . . . . . . . . . . . . . . . . 18
15 Non human objects traing set. . . . . . . . . . . . . . . . . . . . 18
16 Screen shot of my Application in action. . . . . . . . . . . . . . . 19
17 A shape of a dog as an example of a negative training example from the training set of image deformable contours (classified as *not human*). . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

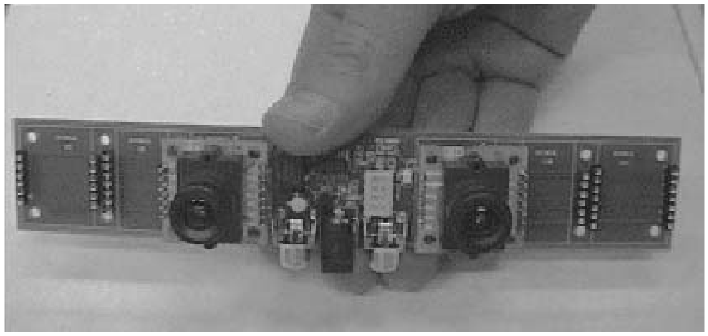

Figure 1: STH-V1 stereo camera.

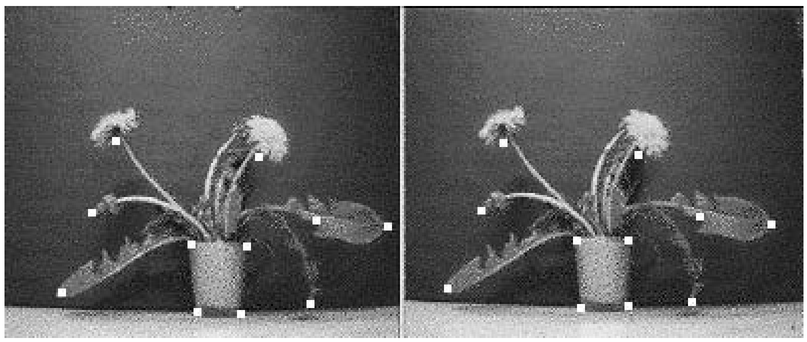

Figure 2: An illustration of the correspondence problem. A matching between corresponding points of an image pair is established.

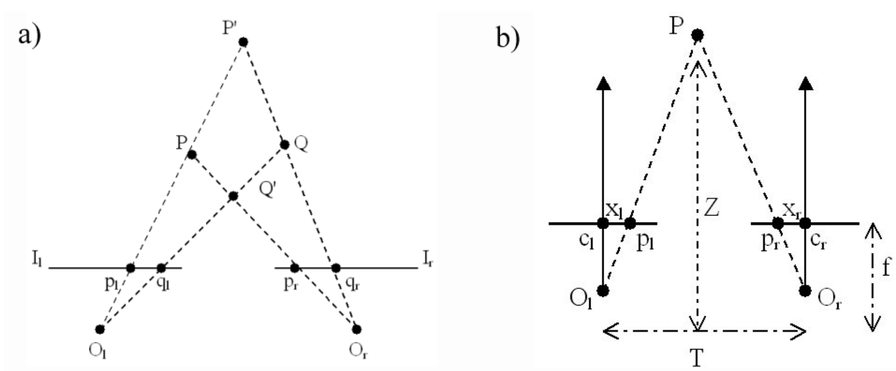


Figure 3: A simple stereo system. 3-D reconstruction is depends on the solution of the correspondence problem (a); depth is estimated from the disparity of corresponding points (b).

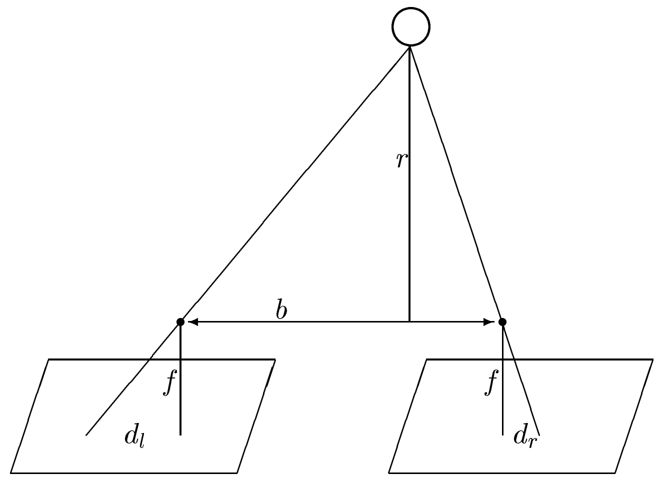


Figure 4: Definition of disparity: offset of the image location of an object.

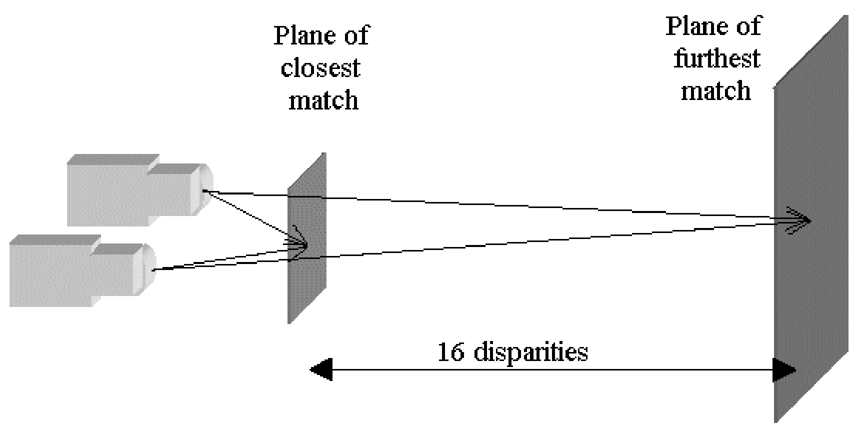


Figure 5: Horopter planes for a 16-pixels disparity search.

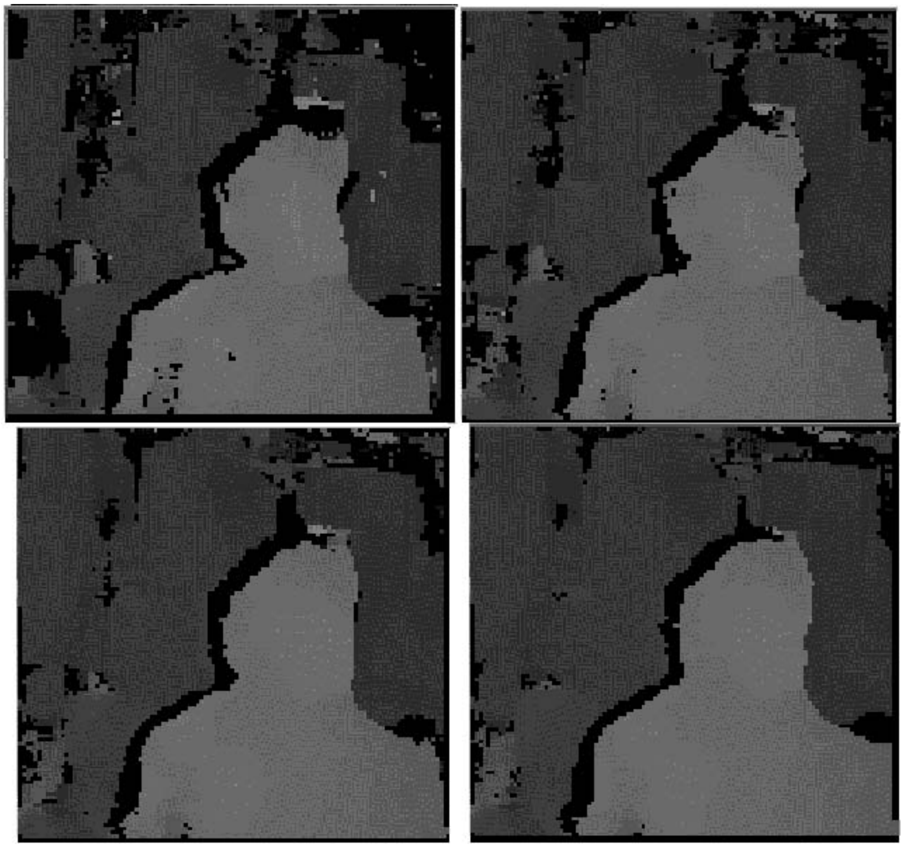

Figure 6: Effects of the area correlation window size. The images show windows of 7x7, 9x9, 11x11 and 13x13 pixels.

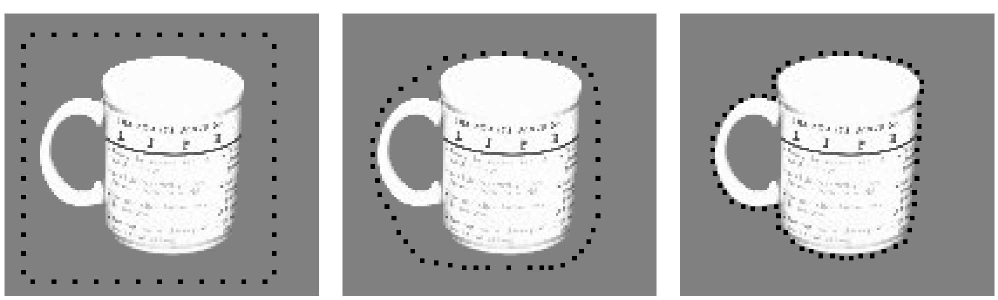

Figure 7: From left to right, images show initial position of the snake. Intermediate position and final position.

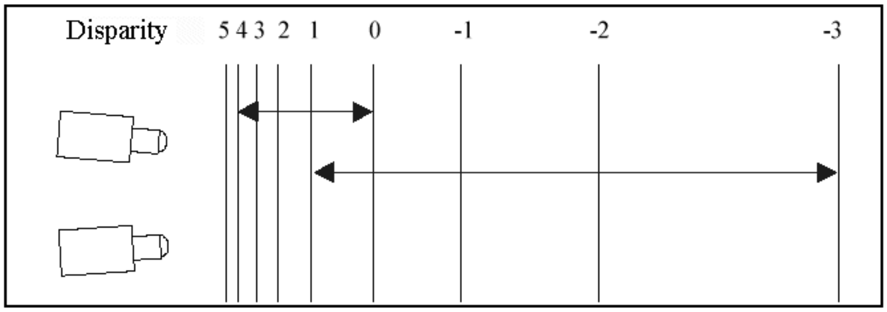


Figure 8: Planes of constant disparity for verged stereo cameras. A search range of 5 pixels can cover different horopters, depending on how the search is offset between the cameras.

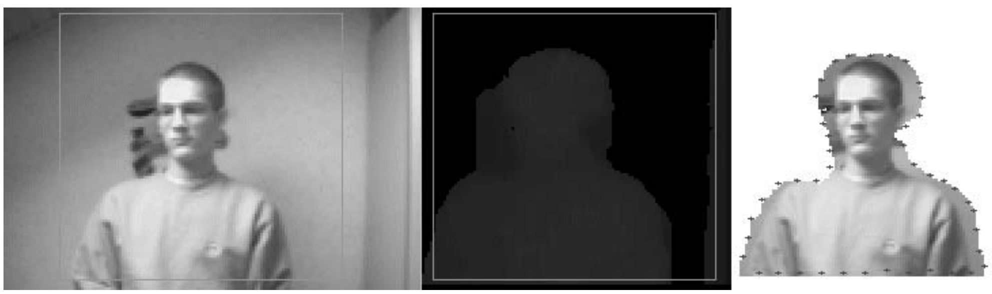

Figure 9: On the left is intensity image with corresponding Disparity image. Right image shows the tracked object separated from the background.

Figure 10: Two situations when a new object is detected. A grayscale image represents the scene and black-white image represents the new object found on the basis of the stereo image.

Figure 11: A Human Object with it's sillhouete, range image and a silhouette with a cicle for transformation of the contour in polar coordinate system.

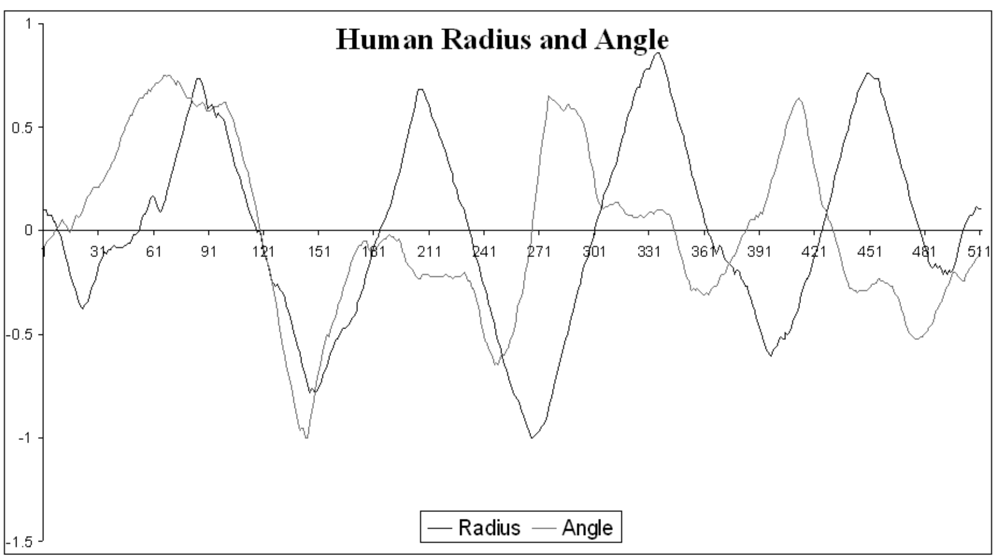


Figure 12: Typical graph for a contour of a human, radius and angle.

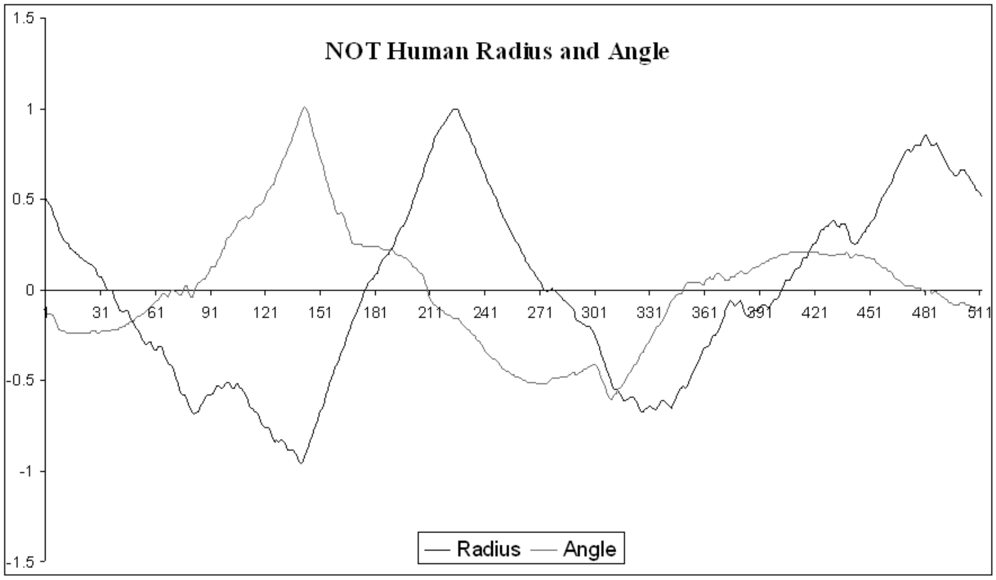


Figure 13: Typical graph for a contour of a non-human, radius and angle.

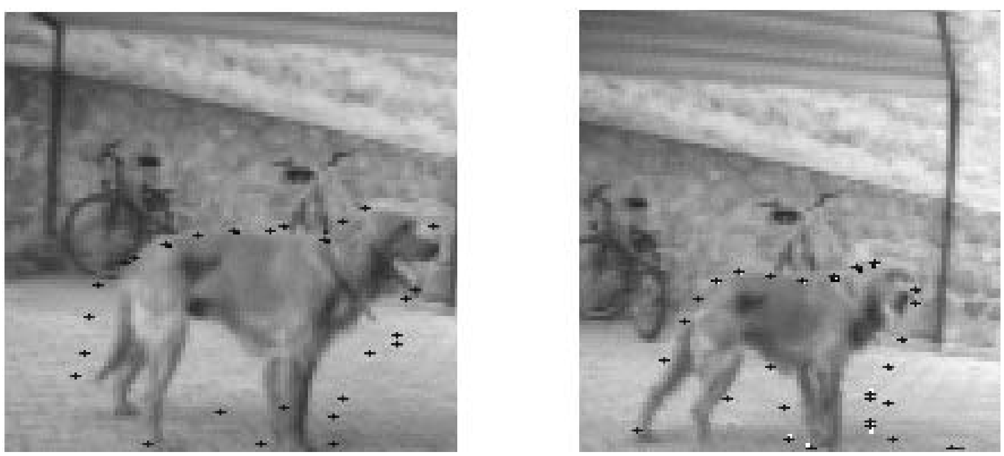

Figure 14: A shape of a dog as an example of a negative training example from the training set of image deformable contours (classified as *not human*).

Figure 15: Human objects traing set.

Figure 16: Non human objects traing set.

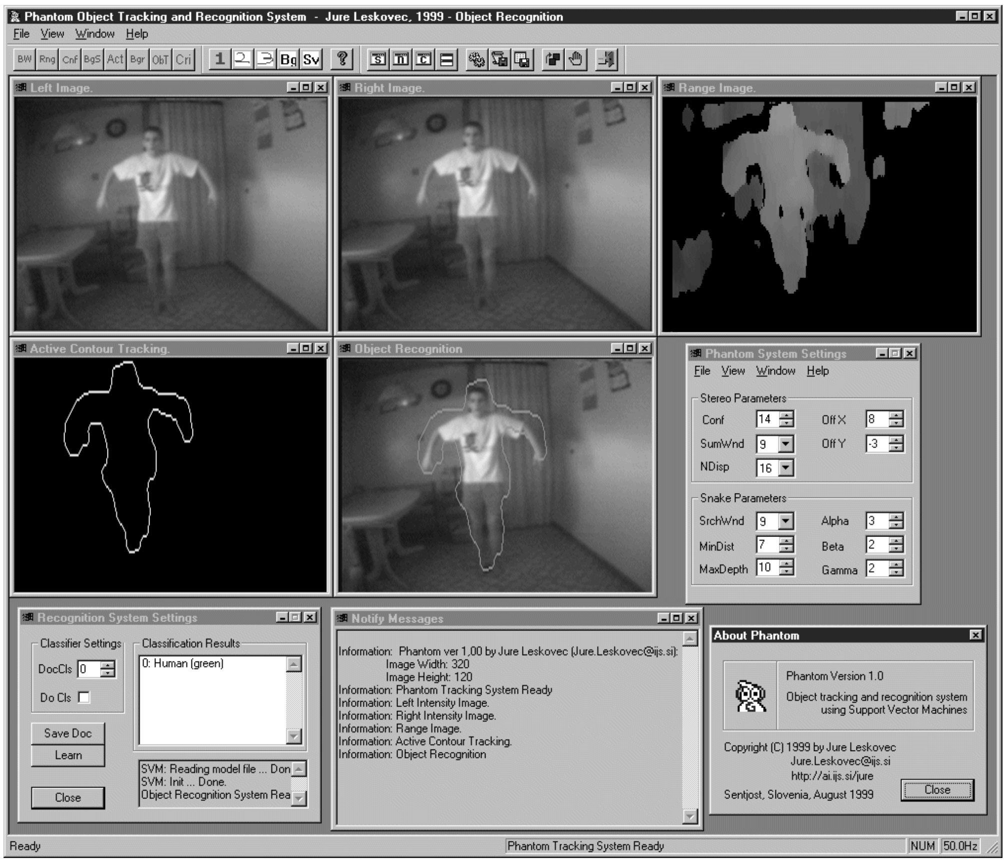


Figure 17: Screen shot of my Application in action.

Jure.Leskovec@ijs.si,
Šentjošt, 27th May 1999.