Pose Estimation in Stereo

F.W. Fikkert

f.w.fikkert@student.utwente.nl

ABSTRACT

Ambient intelligence applications often use technologies from the computer vision research area to obtain information on its user(s). An existing computer vision application uses pose estimation to obtain real-time full human body poses. This paper describes how this existing application may be adapted to use stereoscopic vision in order to solve problems that persist in usage of singular camera computer vision applications such as occlusion.

1. Context

This research took place in the ambient intelligence research area. There, many types of sensors are used to provide ambient intelligence systems with information on its users. One type of sensors which are used in this manner are cameras, which are available and useful in all shapes and sizes. Computer vision is the research area in which the usage of cameras is studied. There, the passive camera sensor is used to extract desired information from an environment. This research is focussed on the use of stereo vision, a part of the computer vision research area, in an ambient intelligence environment.

In the Human Media Interaction project 2004 an ambient intelligence system needed to be developed. For this purpose, the Screpe system was conceived, designed and implemented. In this application a user is confronted with a virtual mirror. That mirror contains a virtual representation of the user (which can be anything, varying from a humanoid to an animal) and is augmented with a virtual pet. The user can, via his or her mirror image, interact with the pet using certain predefined gestures. These gestures have been based upon natural human movements to attribute to the systems credibility. The global design of the Screpe ambient intelligence system is depicted in Figure 1.

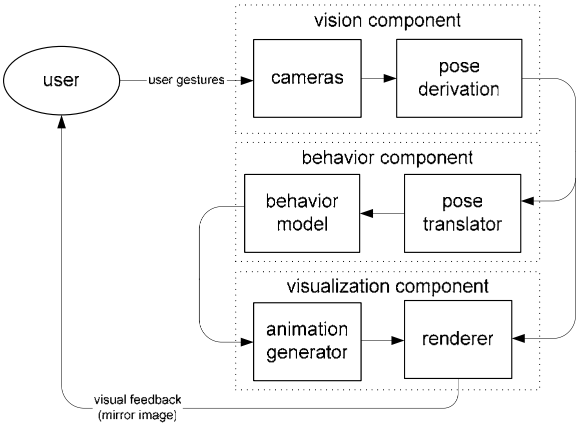

Figure 1. Global design of the Screpe system

In Figure 1 the component of the Screpe system on which the interest of this research is focused can be seen; the vision component. I will describe here how the vision component can be realized in the Screpe system. The goal of this specific component in the whole Screpe system is to detect a limited set of gestures made by the user and to pass that information on to the other two components for processing.

1.1 Problem statement

There are various possibilities for application of computer vision to obtain the described goal. For starters it is possible to use either a single camera or a set of multiple cameras from which the information is combined. It is also possible to use specialized cameras to obtain information outside the visual spectrum, for instance a infrared camera. However, such hardware was not available and will therefore not be used here.

The described goal can be met using a single camera approach because, using various assumptions, an acceptable pose estimation can be obtained [Pop04] [WA97] [Tom02]. The term pose is to be seen as a human pose in a specific time which captures a (part of a) human gesture. An example of such a system is the Posio [Pop04] system in which a human pose is estimated used camera image sequences. However, whilst using a single camera approach various problems will be present:

- Occlusion: objects or parts of the to-be-detected object cover (parts of) that object which prevents extraction of information on that occluded (parts of) object.
- Depth detection can only be estimated or be achieved using workarounds. By using accurate camera positioning depth can be estimated. It is also possible to enhance the environment with markers which can be used to obtain depth information. By example: if object *O* is present at marker *M* than *O* is at depth *D* of *M*.

It is, as has been mentioned, also possible to use more than one camera of extract useful information from the environment. The use of multiple cameras to do so is called stereoscopic vision or, in short, stereo vision. Using stereo vision the above problems may (in part) be solved since information from multiple vantage points of the same environment is obtained. Occlusion may still be present whenever all cameras can not see (part of) the occluded object, but in most cases it is reduced to a bare minimum. Depth detection can be achieved when a object-point is visible in all images, the knowledge of the camera positions and the location of the point representations on the images can be used to achieve it. However, using stereo vision a number of problems are also introduced:

- A lot of extra mathematics is involved in the form of execution various calculations. For the usage of two cameras this results in:
    - Two corresponding image-points, both representing the same object-point, need to be identified in both images. This is called stereo correspondence.
    - An object-point needs to be reconstructed from the two point representations on the images (see Figure 2). This is called stereo reconstruction.
    - Both images must be of the same timeframe which entails good synchronization of the

cameras. This is required in order to make meaningful statements of a dynamic scene.

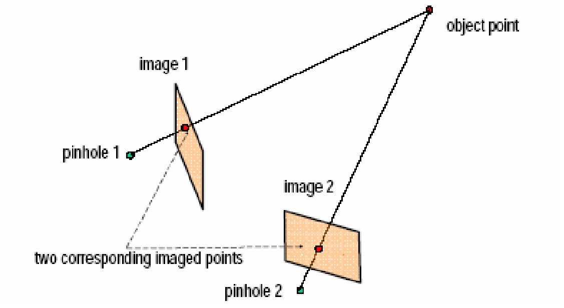

Figure 2. The stereo reconstruction process

- It is necessary (in order to execute the above mathematical calculations) to have available the camera parameters, describing the used cameras:
  - Intrinsic parameters which describe the internal camera properties. These values keep constant no matter the camera setup.
  - Extrinsic parameters which describe the rotations and translations of all cameras in use relative one point, which can also be, of course, the axis system of one of those cameras. These values change per camera setup and will have to be obtained anew for each new setup. See Figure 3 for an illustration of these camera parameters.

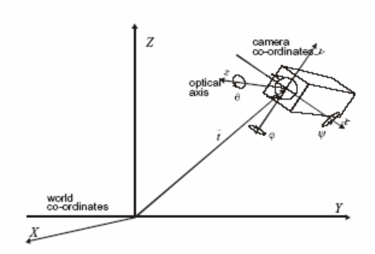

Figure 3. Extrinsic camera parameters

2. Goal definition

I am interested in, in part thanks to the described problems with stereo vision, how to use stereo vision in combination with the existing Posio pose estimation system to detect user gestures and how to apply that information in the Screpe ambient intelligence system. I therefore ask

Can the Posio pose estimation application be adapted to use stereo vision and if so, in what way?

Some sub goals can de defined here:

- Which methods can be applied to realize the use of stereo vision in this context?
- How will the shortcomings of Posio be improved upon by the usage of stereo vision?
- To what extend will the stereo vision enhanced version of Posio improve upon the usage of the original Posio system?

2.1 Boundaries

Since this research is related to the Screpe Human Media Interaction project the boundaries stated for that project will also be used here. I will therefore not go into detail on those boundaries. Stereo vision will be limited to two cameras which will be used to detect user gestures in the form of pose estimation. At runtime one Caucasian person will be present to prevent ambiguity when more limbs (for example) are present than is expected. This user may be of either sex and is of average height, meaning 1.70 to 1.90 meters. He or she will be present in both camera images during run time. Also, no occlusion caused by environment objects will be present. There will be continuous lighting conditions. Also, there is a limited set of user poses which are to be detected. Below, in Figure 4 a few of these poses are depicted. The poses are limited in such a way that only arm movements are to be detected.

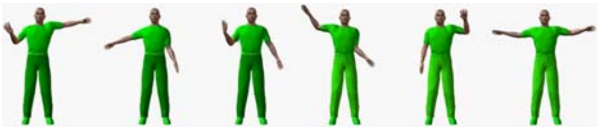

Figure 4. Some to-be-detected poses in the Screpe system

Also, the two cameras which are used here will be setup in a static camera setup. No changes in this setup are assumed during the usage of the system. To conclude the boundaries which will be used I state that only a proof of concept is to be delivered. I do so since the Screpe system also resulted in a proof of concept.

2.2 Requirements

Two Logitech Quickcam cameras are used for this research. These are USB 1.1 cameras with a default resolution of 320x240 pixels and frame rate of 30 fps. A personal computer with a 2 GHz processor is assumed with 512 MB Ram memory, Windows XP and the Direct X 9c SDK. Also, the software package MS Visual C++ is used in combination with the open source imaging library OpenCV.

3. Related work

There has been much research on applications of stereo vision and pose estimation. The Alive project [MBP95], related to the Screpe project, uses for example a single camera for user gesture detection via pose estimation. One of the requirements in that work was a clear contrast between user and background which is realized by dressing the user in dark colored clothes whilst keeping the background white.

The Pfinder system [WA97] uses blob tracking combined with stereo vision. However, there were no means to make the system robust in its usage. The setup of its stereo configuration is not easily realized.

The Posio system [Pop04] which is used in this research describes a pose estimation method for image sequences of a single camera. Also, various techniques for image processing are explained and weighed against each other, also to be found in [Moe01].

One way to solve the stereo correspondence problem is to use a neural network. In [MP00], for example, a Hopfield algorithm was developed which favored the unicity of matches of points of interest in the left and right camera images.

A very different approach to the stereo correspondence problem was proposed in [ZABD98]. Here, a continues calibrating system was described which enables a visually impaired person to navigate a simple hallway setup. Depth measurements were computed using a pixel-to-pixel correspondence method using an epipolar geometry constraint.

A related study, [SB98], also applied a dynamic recalibration approach. Such an extensive approach to the calibration problem is not needed for this work since I have used a static camera setup. However, the techniques used may be applied for use here.

In [FP03] computer vision theory in general is discussed to great extend. Topics of interest are stereo vision and image processing techniques.

4. Approach

In this chapter I will first describe which methods may be applied when using stereoscopic vision in this context. Then I will continue by describing the global steps that the to-be-designed system will include. This will then be followed by a short discussion on the alternatives at each step and a selection from those alternatives.

4.1 Stereo vision possibilities

There are many possible methods in which stereovision can be applied. First is the well-known model of two, more or less parallel cameras. This is analogous to the way in which humans and some animal species can see. Below, in Figure 5, an illustration of this model is included. For each eye, an image of the environment is constructed. Those images differ slightly from eye to eye, which is called disparity. In the brain, those two images are combined with each other to yield a depth reconstruction of the environment.

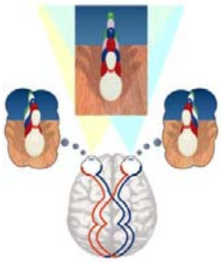

Figure 5. Stereo vision in humans

This approach is often used in the computer vision research area when implementing stereo vision as well. As described, it requires the position and rotation of the cameras that are used to be known. After recognizing two corresponding image points in both camera images and using the extrinsic camera matrices, it is possible to compute the original world point which is represented by those two image points.

Another approach is to obtain information on the environment using motion. Doing so will provide in information from multiple perspectives of the environment. Distinctive features of the images will be extracted after which 3D reconstruction is possible. However, taking into account the context in which the stereo vision is to be used it is not feasible to use this approach. First of all, the necessary equipment is not available and more importantly, the method is not applicable in a dynamic environment which is present in this context. The user will be moving continuously and it will therefore not be possible to obtain stereo images of one and the same environment from those different vantage points.

A third option when applying two cameras to obtain stereo vision in an application is to position those cameras opposite each other to view the object from two vantage points. However, this will not solve the occlusion problem, in fact it will add to it since now there is occlusion in two images, and also it is impractical to accurately detect depth positions since stereo correlation is difficult to obtain in such a setup. I have therefore chosen to use two cameras in a more or less parallel setup for creating a stereoscopic vision setup.

Further, it is possible to choose from two approaches regarding the setup of two more or less parallel cameras to estimate poses. Firstly, it is possible to fully process both camera images which will result in two poses, one for each camera. Those poses are then correlated with one another after which, using the calibration information of the camera setup, a 3D estimation is completed of the user's pose. Secondly, the other approach is to determine the stereo correlation of the images beforehand and to apply the pose estimation process on the resulting disparity map. The disparity map describes the correlation between two images. Using this disparity map a single 3D estimation of the user's pose may be completed.

The mayor benefit of first calculating the disparity map in the latter approach is that less image processing needs to be done. This will benefit the necessary required calculation time. Also, occlusion problems will be handled more precisely in this approach [ZABD98]. There will be no faulty conclusions when there is occlusion of a limb, that information is already included in the disparity map. The main benefit of processing both camera images first and correlating the results afterwards is that determining the stereo correlation is much easier since the processing results in both cases yields the same pose model. Stereo correlation can therefore be calculated very fast and simple. In Posio the assumption is made that occlusion of limbs is handled adequately [Pop04]. It is therefore possible use the second approach.

Although determining the disparity map beforehand will likely be more robust I have chosen to determine the user's pose per image first and correlating the results afterwards. The faster calculations of stereo correlation combined with the low expected faulty modeling of occlusion generating limbs are the main reasons for this choice. From this point on I will therefore work from the situation in which both images are processed fully to result in a pose estimation and that the results are correlated with one another afterwards.

4.2 Steps to be undertaken

The goal is to obtain a full human body pose from two simultaneously obtained images of an environment. That pose should be as accurate as possible in three dimensions. I will use a two-camera setup in which the cameras are positioned more or less parallel to each other and facing the same direction. In practice it will not be the case that those two cameras are precisely in the same plane so that will not be my assumption here. First of all the information of the used cameras in the form of intrinsic and extrinsic camera parameters needs to be obtained. Determining these camera parameters is done via a process known as camera calibration.

After camera calibration it is possible to remove lens distortion from the camera images. This means that so called radial lens distortion is removed and that straight lines in the environment are also depicted as such. It is not always possible to remove lens distortion but it should then be assumed that the used camera lenses are identical, which is not the case in practical situations. I will therefore remove the radial lens distortion before trying to estimate the user pose.

As was mentioned earlier, I will use the Posio pose estimation application. This system makes the assumption that the shoulders and hips are in a plane perpendicular to the image plane. As was mentioned above, this is impossible since the cameras used will, in practice, never be positioned in exactly the same plane. A solution must be found to circumvent or solve this assumption.

Using the above adaptation of the Posio pose estimation process processing of the camera images may commence. It must be ensured that the camera images are from the exact same time frame since a temporal variation between the images of the two cameras will result in inconsistencies in the overall results. This will not yet be the case for the pose estimation processes which are performed on the stereo image pair. However, the temporal variation will be obvious in the stereo reconstruction process in which the information of the two poses is combined.

The execution steps which are undertaken in the Posio begin by extracting, per camera image, the foreground representing the user. The resulting silhouette still contains shadows which is false information and which is therefore removed. Also, other image noise, e.g. pixels which have been falsely classified as foreground, are removed. Using a combination of color spaces to detect skin colorings, the positions of the face and both hands are derived from the silhouette. The final step is to model a predefined human body model, e.g. a simplified version of an international standard [HAW], onto the silhouette and thus to estimate the user's pose.

This process is, as has been mentioned, performed on both images in the stereo image pair. The information contained in the resulting two poses still needs to be combined. The stereo correspondence needs to be determined, a disparity map describing the differences between the two poses has to be created and both poses need to be used to reconstruct a more accurate 3D user pose. I will now go into detail on the execution steps I just described.

4.2.1 Camera calibration

How to extract the intrinsic and extrinsic camera parameters. Intrinsic camera parameters describe the lens properties whilst the extrinsic parameters describe the position and orientation of the camera relative a to-be-defined world reference frame. This relation has to be made for both cameras of course in order to be able to combine the information from these cameras. Although the cameras I used are the same this does not necessarily entail that the lenses in both cameras are exactly identical (production deviations for example). I will therefore not assume identical intrinsic camera parameters for the cameras I have used.

The intrinsic camera parameters are thus extracted per camera only once since I will assume that they remain constant. The extrinsic parameters however have to be extracted simultaneously for both cameras and for each stereo setup.

Intrinsic camera parameters

I will use the extended pinhole camera model to represent my two cameras. This is in fact just the standard pinhole mode expanded to cope with radial distortion. The intrinsic camera parameters are often described via a set of variables. Below I will summarize these variables where the last are the ones which deal with radial distortion.

f Distance from the focal point of the camera image plane to the lens (focal length)

(u_0, v_0) Coordinates for the principal point on the camera image plane

(s_x, s_y) Skew factors (often assumed to be orthogonal

(d_x, d_y) Radial distortion variables (describe the centre of the radial lens distortion in the image)

The radial distortion parameters are necessary as was described above. Using the above intrinsic parameters, the matrix κ can be created which contains all parameters to represent the used camera model:

$$\kappa = \begin{bmatrix} \alpha & -\alpha\cot(\theta) & u_0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where $\alpha = s_x f$ and $\beta = s_y f$. Removing the radial distortion is described in section 4.2.2 and is normally not included in the above matrix κ .

Extrinsic camera parameters

These values are often represented in the matrix *M*:

$$M = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix}$$

Where *R* is the rotation matrix (3x3) and where *t* is the translation vector (3x1). *R* contains three combined rotations, one for every axis x, y and z of the camera reference frame.

Calibration

How does the actual extraction of these camera parameters performed then? This is called the camera calibration and for this process often an object is used of which the shape and sizes are predefined. This object is seen in the camera images.

Often, a chessboard is used as a calibration object. I also used one such calibration object. The properties of the chessboard I used are:

$$h_{ch} = w_{ch} = 33mm$$
$$n_w = n_h = 10$$
$$x_{ch} = y_{ch} = n_w * w_{ch} = n_h * h_{ch}$$

Representing the height and width of a chessboard square, the number of squares in the width and height of the chessboard and the resulting sizes in *x* and *y* directions of the chessboard respectively. A calibration using the chessboard results in pictures such as:

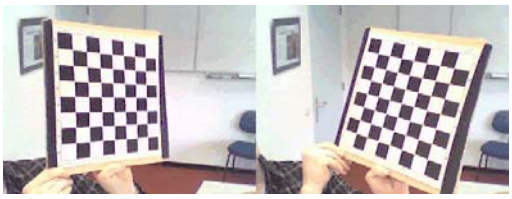

Figure 6. Stereo image pair of the calibration object

Detecting the chessboard is done by first extracting the corners of the board. Then the squares are detected correctly. The distances in pixels can then be used to determine the variables described above [HS97] [FP03]. For this process, a reasonable number of image pairs needs to be used to reduce measurement errors. A good number is 20 image pairs [HS97]. Also, the image resolution also contributes to measurement errors (less pixels results in more errors). As should be clear now, the κ and *M* matrices can be used throughout the usage of the stereo setup.

4.2.2 Lens distortion

The basic pinhole camera model is too simple to be used in most cases [FP03]. Almost every lens has radial distortion to some degree. This might be pincushion distortion, barrel distortion or a combination of these two [Oja99]. In this case I will assume the case in which only radial distortion is present. More complex models do exist in which radial distortion correction is only a basis.

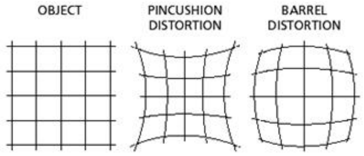

Figure 7. Radial lens distortion

The correction of the lens distortion is performed by applying a polynomial transformation on the image. In most cases it suffices to use a low degree polynomial (2nd or 3rd degree). The pixels in the camera images are then transformed using the equations:

$$x = x_0 + (x_0 - c_x)\left(K_{f1}\frac{r^2}{f^2} + K_{f2}\frac{r^4}{f^4} + \ldots + K_{fn}\frac{r^{2n}}{f^{2n}}\right)$$
$$y = y_0 + (y_0 - c_y)\left(K_{f1}\frac{r^2}{f^2} + K_{f2}\frac{r^4}{f^4} + \ldots + K_{fn}\frac{r^{2n}}{f^{2n}}\right)$$

Where K varies with the focal length.

4.2.3 Simultaneous image extraction

The problem of extracting images in the stereo image pair at the same time is of great importance. Regarding the software side, it may be solved in a variety of ways. The camera images of the used cameras are 320x240 pixels and will therefore not yield problems with the used hardware (enough bandwidth is available). The obvious approach is to use two threads or independent processes. This, combined with the use of mutexes ensures best the simultaneous image pair extraction.

After the images have been extracted, the described lens distortion correction is applied before starting the pose estimation process.

4.2.4 Posio pose estimation

I have described the global steps of the Posio algorithm. Here I will describe this process in more detail. For more detailed information see [Pop04]. There are a total number of five steps to be made.

In the images, the background is separated from the foreground in which we are interested. The foreground is not static, the background is. Using chroma-keying, the background (which is assumed to have a uniform color) is filtered out of the image by applying a threshold on the image. More complex methods may be applied but for a proof of concept this method suffices. The pixel intensity for thresholding is calculated using:

$$I = 0.212671R + 0.715160G + 0.072169B$$

Shadows will, since they have a different intensity compared with the background, not be filtered out of the image. Also, user-shadows (including those generated by the user) are possible. Pixels of an object with shadow are equal in color tone compared with non-shaded pixels of that same object [Moe01]. The intensity differs however and thus, shadows may be removed. In the HSV color space a separate channel is available to describe intensity values in. Using an upper and lower limit the background pixels, including shadows, are filtered:

$$L_{\lim} \prec (\frac{I_B}{I_C}) \prec U_{\lim}$$

Where I_B is the intensity of the background pixel and I_C is the intensity of the current pixel.

Noise in the camera image may result in pixels being falsely classified as either foreground or background. Noise must be removed. This is done using pixel erosion in which solo-pixels are filtered out of the foreground silhouette.

The silhouette thus far contains few useful information. Using skin detection we can detect the hands and face. A lookup table including all kinds of possible skin colorings might be used but is not achievable here. Therefore, I have used, analogous to the approach in [Pop04], a majority vote of a combination of color spaces (RGB, HSV and YCrCb) to detect the skin colors.

Using the positions of the hands and the face on the detected silhouette, the silhouette may be mapped on a human body model.

A standard model is available [HAW] having a great many feature points. This model is however, too detailed. In [Pop04] a simplified model, see Figure 8, is proposed which is also used here. This model contains 15 feature points, having in total 16 degrees of freedom, 10 joints and 14 segments. The head is assumed to be in a straight line with the backbone, the same holds for the hands and feet which are assumed to be in a straight line with their connecting limbs. The joints are, as is the case in actual humans, limited in their ranges of motion. This contributes to the elimination of false mapped silhouettes in an early stadium. The basis of this human body model is the human root, the node in the centre of the hip where the centre of mass is also to be found. From this point on the other nodes are defined.

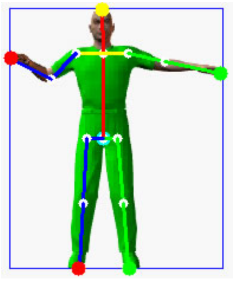

Figure 8. The simplified H-Anim human body model

Estimation of the user's pose is done analogous to [Pop04]. This is achieved by:

1. determining the distance to the camera
2. determining the length of the person
3. locating the human root node of the model
4. feet, hand and head positions extraction
5. shoulder and hip positions extraction
6. elbow and knee positions extraction
7. most probable pose mapping

The distance to the camera is determined in Posio by manually adjusting the camera parameters to approach the position of the camera relative the floor (which is considered the *xy*-plane). Since now a stereoscopic setup is used, the depth is estimated much more precise. However, the human root is needed in order to be able to detect the exact depth. I therefore chose to apply the entire Posio process and thus estimating the depth at first. By using stereo correlation, the depth is detected more accurately which results in repositioning the resulting user pose to a more precise depth-position. A more detailed description of the above steps may be found in [Pop04].

A small adaptation to this process is necessary however. As was mentioned above, in Posio the assumption is made that the shoulders and hips are in a plane perpendicular to the image plane. This is not possible in a stereoscopic system. Some solutions to this problem:

- By using a fixed with/height ratio of a human being, the silhouette may be skewed. A problem here is how to detect the width of the silhouette since it might have arms sticking out. Those arms may be detected using vertical scan lines originating from the human root node.
- Redefinition of the Posio algorithm to cope with this assumption. Although it is expected to be more precise, it will leave the scope of this research project and will therefore not be used.

Although I will not describe the Posio mapping process I will summarize how it goes about its business. Using the found silhouette for each image the most likely positioned human body model is mapped. See Figure 9 for a illustration of this process.

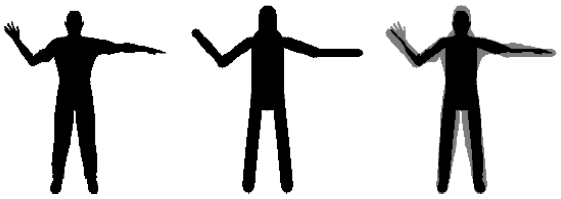

Figure 9. Pose mapping

4.2.5 Stereo correspondence

Having two user poses, one for each camera, this information needs to be combined with one another. Goal is to achieve a single and more precise pose estimation. Since both poses are based upon the same human body model it is straightforward to correlate the feature points of both models. The first point of notice is to determine the correct depth.

Human root extraction

The position and orientation of the cameras is know via the extrinsic camera parameters. Two human root nodes are known at this point in the pose estimation process. Repositioning this human root node to the correct depth location will also transform the rest of the human body model.

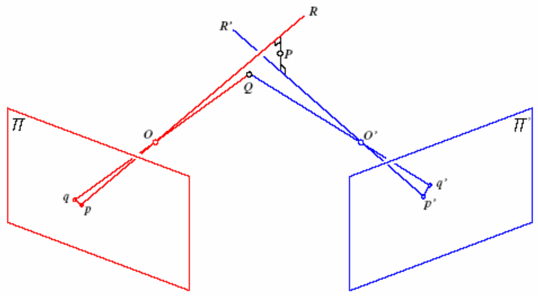

Figure 10. Stereo correspondence solution

In Figure 10, the image points *p* and *p'* are the representations of the point *P*, or the human root. Draw a line from *p* and *p'* through the focal points of the lenses *O* and *O'*. The point of the lines where the lines are closest is the location of *P*. These lines will almost never cross each other (but that is possible, see point *Q*).

In the above figure, the assumption is made that the image points are in the same reference frame. However, at this stage this is not yet the case, world points are still defined in the individual camera reference frames. Using

$$O'p' \cdot (O'O \wedge ROp) = 0$$

We can transform the points of the 1st camera reference frame to the 2nd one. Here, *R* is the matrix describing the rotation between the two camera reference frames. Therefore, *R* can be calculated using the already available extrinsic camera parameters. Inclusion of translation is also required such that

$$p'^T \cdot ([t] \times Rp) = p'^T TRp = 0$$

Where p'^T is used to redefine *O'p'*, *O'O* and *Op* from the previous equation. Here, *t* is the translation vector (along the x, y and z axis) which can be found in the extrinsic parameters. T thus becomes

$$T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

Which enables us to redefine the last equation to

$$p'^T Ep = 0$$

Where *E* is the so called essential matrix. *E* describes the Euclidian rotation and translation between the camera reference frames. By now translating the image points to one reference frame (for example to the second camera's one) en by subsequently calculating the point *P*, we can complete the more accurate estimation of the human root.

Model correspondence

Having the human root available it remains straightforward to combine the information of the two human body model poses with one another. Translating the models to the newly defined human root node location is followed by a comparison of the distances between the other remaining feature points. This may be done in two related ways:

- Mediation of the two feature points
- Weighted comparison by valuing the information of camera X more on X's side of the model. Thus, for calculation of the right-hand feature point the information of the right camera is regarded more important. The reason for this approach is that the right camera is more close to that point and thus should be able to have more and more accurate information on the exact location of that point.

We now have completed the stereo correspondence process for the stereo image pair we obtained. It is not necessary to test the found pose on validity of limb-positions since a mediation between two possible poses should always result in a valid pose as well. However, measurement errors may result in an invalid or false estimated user pose. From this point of view it might be necessary to include a final check on pose validity.

5. Results

In this chapter I will describe how I implemented the above approach to the problem. For the implementation I used the OpenCV imaging library [OCV] which contains an array of functions for image processing purposes. I also made use of an existing Matlab toolbox for the camera calibration [Bou]. In Figure 11 the result is depicted after analyzing the orientations and locations of the chessboard in 20 stereo image pairs. The individual extrinsic camera parameters yielded the positions of the two cameras here.

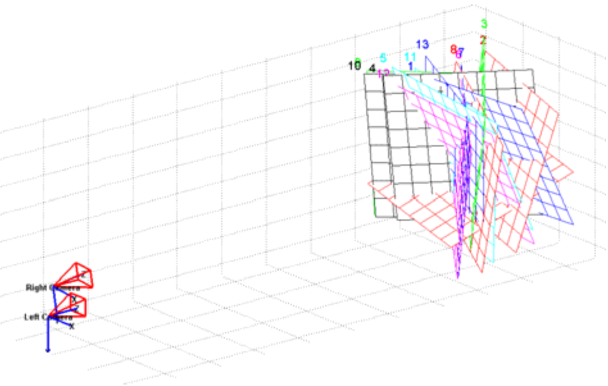

Figure 11. Camera calibration results in [Bou]

Using the intrinsic and extrinsic camera parameters which are obtained using the Matlab calibration toolbox we can move onwards to the processing of the camera image pairs. First, the images need to be obtained simultaneously. By modifying an existing function in the OpenCV library it is possible to select two cameras and to subsequently obtain their images in two parallel processes. This adaptation was the use of mutexes to ensure the processes extract the camera images at the same time.

Next, the radial image distortion is removed. Below, in Figure 12, it is shown that the distortion which is seen in the left image is corrected in the right image. The chessboard edges in the right image are straight lines after correction.

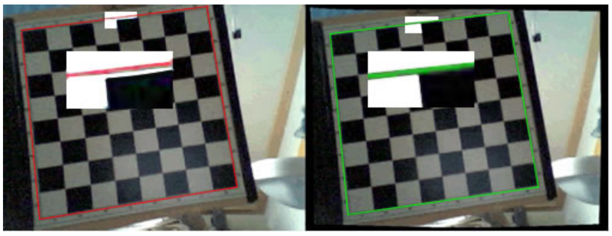

Figure 12. Radial distortion correction

In the OpenCV library the below equation is used to correct radial lens distortion. This has been based upon [HS97].

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u (\tilde{u}_i + \delta u_i^{(r)} + \delta u_i^{(t)}) \\ D_v (\tilde{v}_i + \delta v_i^{(r)} + \delta v_i^{(t)}) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

This is, as can be seen, the same equation which I described in section 4.2.2.

The pose estimation is executed analogous to the approach in Posio. The adjustment here should be that the shoulders and hips are not necessarily in a plane perpendicular to the image plane. However, the solution described above has not been implemented and verified which entails the lack of using a real world camera setup. Using an application Poser, as was done in [Pop04], the possibility exists to precisely position two cameras so that their image planes are contained in the same plane. The necessity to improve the [Pop04] algorithm has hereby been put off for now. Using Poser, a proof of concept is still achievable.

Mapping the simplified [HAW] human body models is implemented in a straightforward manner. It was largely possible to used components from Posio, resulting in:

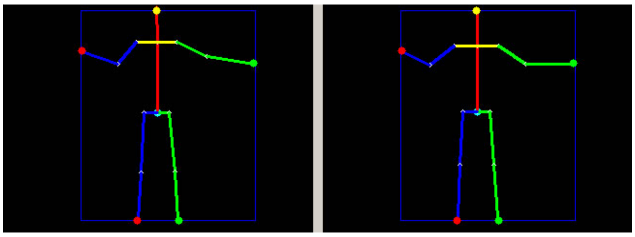

Figure 13. Two resulting user poses for the left and right cameras

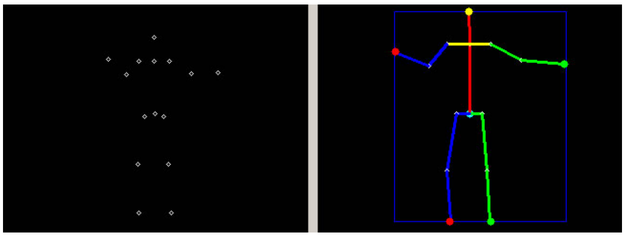

Figure 14. Stereo correlation results, left the features points in 3D, right the feature points in the estimated user pose

The results in Figure 14 are based upon the correlation of the two user poses depicted in Figure 13. Significant differences between the two methods described in section 4.2.4 to correlate the stereo user pose pair have not been found. However, it must be remarked here that I have only been able to research this to a limited degree.

I also investigated how the correlation between the stereo user pose pair, as described above, may be used when not using the extrinsic camera parameters. In such a scenario the pose pair is only mediated. Although this approach is no different than applying two processes of Posio on the scene, the result is likely to be better than using a single Posio process. The number of errors is then mediated.

6. Evaluation

To evaluate the results of the new stereoscopic vision based approach which has been described thus far it needs to be compared with another approach. The logical course it to compare it with the original Posio approach. Also, it is conceivable to implement alternative methods, as were described above, and to compare it with those methods.

For the evaluation it is required to have validation data. That data describes the correct positions of the feature points of the human body model which I have used. Since it is not realistic to real world data extracted validation data, an alternative should be sought.

As was also described, I have use Poser to generate test image sequences for the pose estimation process. It is possible to maneuver the camera vantage point in Poser with great accuracy so that the calibration steps are not needed. The camera parameters can be obtained directly from Poser. The feature points then need to be extracted from Poser. However, that is a limitation to the Poser usage since Poser calculates real-time the positions of the limbs using only a few key frames which are fully described. It is possible to obtain that data but that would require a lot of manual labor to extract each feature point of each camera of each sequence frame. This is beyond the scope of the research. An alternative might be to use another specialized 3D rendering application such as 3D Studio Max and its toolboxes. However, the accuracy which is obtainable in Poser is not met there. Such an alternative is also not realistic in use.

Assuming validation data is available of stereo movie pairs of a specific scene, it is straightforward to compare multiple pose estimation techniques. The pose estimation techniques are applied to the image sequence pairs. The results are a pose for each frame pair. That pose is not necessarily calculated real-time in order to be able to accurately compare the techniques themselves. After all frames are processed the errors may simply calculated using the validation data.

The average calculated error in the pose estimation process is, combined with a maximum and minimum error, a good way to compare various methods with one another. It is also good to keep in mind the required calculation times since many applications require fast processing times. The application designers are then required to chose the best pose estimation technique for their specific application.

7. Conclusion(s)

Can the Posio pose estimation application be adapted to use stereo vision and if so, in what way?

This is a very good possibility. As was described, multiple methods may be used to realize the use of stereoscopic vision in Posio. The approach opted here is to estimate the user’s pose in both images separately. Then, using the known camera setup parameters and the parameters describing the camera properties, the two resulting estimated poses are correlated with one another.

Although the implementation of stereo vision usage in Posio has a high computational load, the results show that this approach does indeed yield results. Sadly, it was not possible to perform an evaluation of the stereoscopic vision system because of the lack of validation data which should be used to perform a comparison between multiple approaches.

Which methods can be applied to realize the use of stereo vision in this context?

In section 4, two plausible methods for stereoscopic vision are described. The first method, which has also been opted here, is to estimate the user’s pose in both images of a stereo image pair and then combining the information in those two poses. The second method is to extract the feature points using correlation of the stereo silhouette pair and thus execution pose estimation one time in stead of three. A plausible alternative is via pixel wise determination of the stereo correlation in the stereo image pair and using the depth information in the resulting disparity map to estimate the user’s pose. However, this approach would deviate too much from the starting point of this research: the approach described in [Pop04].

How will the shortcomings of Posio be improved upon by the usage of stereo vision?

Better pose estimation using stereoscopic vision is obtained. To what extend this improves upon the original Posio application is not known for the lack of validation data needed to execute an evaluation between the two approaches. Issues as depth detection and occlusion, which occur in each application using a single camera, are reduced but not necessarily solved by the usage of two cameras.

To what extend will the stereo vision enhanced version of Posio improve upon the usage of the original Posio system?

This question can not be answered since no system evaluation was possible. It is possible to use both the original Posio version and the stereo vision extended version in the Screpe ambient intelligence application for user pose detection purposes. The latter approach, which has been described in this paper, is likely to yield more accurate measurement results which will improve the realism of the virtual mirror.

8. Recommendations

Researching whether an approach in which immediate stereo image pair correlation is applied yields better results than the approach described here. This should at least have less computational load since only one pose estimation process needs to be executed in stead of three. Also I expect the results to be almost similar to the approach described in this document. This because in both cases, assuming the use of Posio as a basis, the approach will remain more or less the same. The difference will start to occur after the skin colors in the silhouette have been detected and the process of extracting the feature points starts.

It should be examined if the extra computational load required in a stereoscopic vision version of Posio applied in the Screpe ambient intelligence system is worth its results. An evaluation test with users could result in a obvious choice here. Such a test can be performed by defining two user groups and by confronting each group with on of the two versions of Posio in combination with the Screpe system.

Implementing the described solution to the assumption made in Posio that the human root, shoulders and hip feature points are in a plane perpendicular to the image plane. This should be done in such a way which is also applicable in a real world camera setup. It should also be investigated if there might be better methods to overcome the problems caused by this assumption.

ACKNOWLEDGEMENTS

For Susan.

I would like to thank Mannes Poel and Rutger Rienks for their support during the course of this research and my friends from the Human Media Interaction Project “Screpe”; Koen, René and Luc.

REFERENCES

[Pop04] R. Poppe. *Real-time pose estimation from monocular image sequences using silhouettes.* University of Twente, Enschede, 2004

[WA97] C. Wren, A. Azarbayejani et. al. *Pfinder: A Real-time Tracking of Human Body*. MIT Media Laboratory, 1997.

[Tom02] B. Tomlinson et al. *Leashing the AlphaWolves: Mixing User Direction with Autonomous Emotion in a Pack of Semi-Autonomous Virtual Characters.* MIT Media Lab, 2002.

[MBP95] P. Maes, T. Darrell, B. Blumberg, A. Pentland. *Wireless, Full-body Interaction with Autonomous Agents*. MIT Media Laboratory, 1995.

[Moe01] T. Moesland. *A Survey of Computer Vision-Based Human Motion Capture*. University of Aalborg, 2001.

[MP00] R. May Oral, M. Perez-Ilzarbe. *Competitive Hopfield Neural Network for Stereo Vision Correspondence*. University of Navarra, 2000.

[SB98] S. Se, M. Brady. *Stereo Vision-based Obstacle Detection for Partially Sighted People*. University of Oxford, 1998.

[FP03] D. Forsyth, J. Ponce. *Computer Vision – A Modern Approach*. Universities of California and Illinois, 2003.

[HAW] Humanoid Animation Working Group. *Specification for a standard humanoid*, 2004. Online: http://www.h-anim.org/

[HS97] J. Heikkila, O. Silven. *A Four-step Camera Calibration Procedure with Implicit Image Correction*. University of Oulo, 1997.

[Oja99] H. Ojanen. *Automatic Correction of Lens Distortion by Using Digital Image Processing*, 1999.

[OCV] Intel Cooperation. *OpenCV, Open source Computer Vision Library*. Online: http://groups.yahoo.com/group/OpenCV/, http://www.intel.com/research/mrl/research/opencv and http://sourceforge.net/projects/opencvlibrary/.

[Bou] J-Y. Bouget. *Camera Calibration Toolbox for Matlab*. Online: http://w3.impa.br/~pcezar/3dp/original/CVL_html/appPage/calib_doc/