DEPTH SENSORS IN SCREENING OF SCOLIOSIS

Péter Major, Dániel Ayhan, Péter Tamás
Budapest University of Technology and Economics Mechanical Engineering Faculty
axodox@outlook.com

1. Introduction

The aim of the presented project was to develop equipment for screening and early recognizing of scoliosis. The Moiré method is generally used in practice, but the correct processing of Moiré images is not currently solved. Depth sensors capture images which describe the distances of the observed surfaces from the camera. With the help of these sensors a 2.5 D model of the human back can be created. This model can be used to create an automatic or semi-automatic scoliosis diagnosis system. The paper describes the development, structure and internal mechanisms of such a measurement method.

2. The theoretical background of the Kinect sensor

The Kinect has four main parts: an infrared projector, an infrared camera, a color camera and a signal processor. The infrared projector projects a structured pattern of points to the environment.[1] It uses a special pattern of points (lower left corner of *Figure 1*) to provide local position information on the infrared image which is captured by the infrared camera. This type of structured light depth mapping is patented by PrimeSense.[2] Because infrared light is utilized the effect of external lighting usually does not disturb the measurement. However, the device cannot be used in direct sunlight as the infrared light of the projector is not bright enough to give the necessary amount of contrast. If the reflectivity of the observed surfaces is too strong or weak, the Kinect may not be able to calculate their true distance from the camera. The depth map is calculated by triangulation, as the position of the emitted and captured rays of IR light is known, the position of their intersections can be determined. For this method it is necessary to have enough distance between the IR projector and detector, which causes shadows on the depth image. The optical zoom equipped depth camera achieves a working distance of 0.6 to 8 meters.

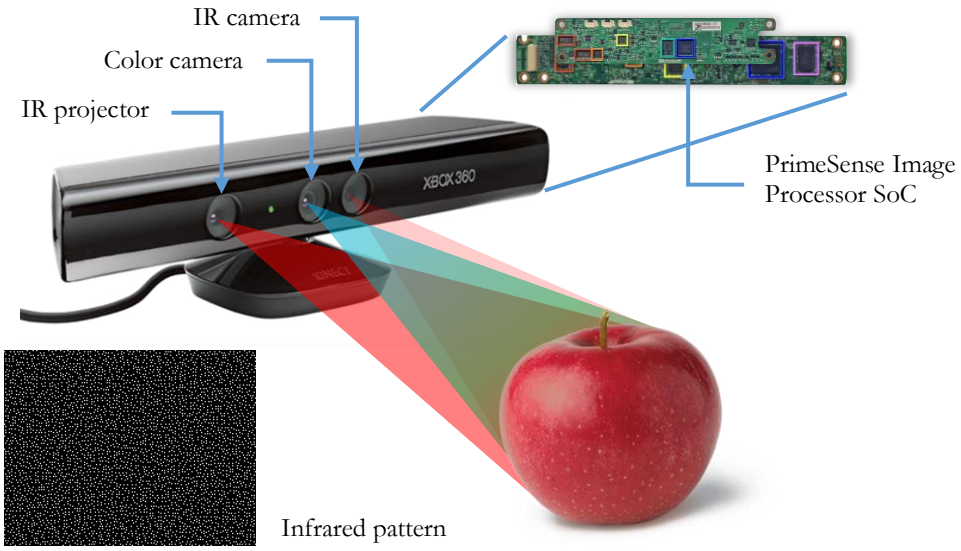


Figure 1. The Kinect sensor

The third main part of the Kinect is a color camera. The color images may be used to create textured models. The resolution of cameras is 1280 × 1024 pixels, but the bandwidth of the USB 2 connection limits the video to 30 frame per second at 640 × 480 pixels while transmitting the depth and color images simultaneously (*Figure 2.*). Because the projected grid of points cannot be used to calculate the depth for every pixel, the Kinect uses interpolation to fill the gaps. The distance measurement has a bit depth of 11 bits.

Figure 2. The color and IR images created with the Kinect

All cameras are calibrated during the manufacturing, but this calibration is not precise enough for 3D scanning. We have created an original calibration method to solve this problem.

3. Calibration

Before performing the measurements we had to calibrate the Kinect, for this the theoretical model of the measurement had to be established.

3.1 The mathematical model of the Kinect

The Kinect was modeled as a pinhole camera and the distortion was described by the Brown model. Let a 3D point be on the surface of an observed surface ($\boldsymbol{P}$)! Let assume that we know the position of P in the world coordinate system ($P_w(x_w, y_w, z_w)$)! The origin of camera is at the aperture of the lens, X is the horizontal axis, Y is the vertical axis of the picture plane and Z is perpendicular to X and Y. In this coordinate system every point has three coordinates (x, y, z). From this we create Z-normalized coordinates ($\boldsymbol{P_n}(x_n, y_n)$). After this the non-linear distortion of the cameras is applied.[2-3] The radial distortion is represented by (1) and (2) equations.

$$r_n{}^2 = x_n{}^2 + y_n{}^2 \tag{1}$$

$$\boldsymbol{P_d} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + k_1 r_n{}^2 + k_2 r_n{}^4 + k_3 r_n{}^6) \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \boldsymbol{d_t} \tag{2}$$

Where $\boldsymbol{P_d}(x_d, y_d)$ is the vector of the distorted point, $P_n(x_n, y_n)$ is the Z-normalized vector, $k_1..k_i$ are the coefficients of radial distortion and d_t is the tangential distortion vector. Usually only the first two or three coefficients are used in practice. The tangential distortion is defined by (3).

$$\boldsymbol{d_t} = \begin{bmatrix} 2t_1 x_n y_n + t_2({r_n}^2 + 2{x_n}^2) \\ t_1({r_n}^2 + 2{y_n}^2) + 2t_2 x_n y_n \end{bmatrix} \tag{3}$$

Where t_1, t_2 are the coefficients of tangential distortion. The perspective projection of the camera has to be applied to the distorted vectors. The projection is defined with the help homogenous coordinates by (4) and (5) equations.

$$\boldsymbol{A} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \tag{5}$$

Where $\boldsymbol{A}$ is the camera matrix, f_x and f_y are the focal lengths in the direction of X and Y axes, c_x and c_y are the coordinates of the intersection point of the optical axis and the picture plane. The skew factor (s) describes the bias of axes from perpendicular case (0). With this transformation the projected point ($\boldsymbol{P_i}(x_i, y_i)$) in the picture plane is defined. The flowchart of the projection model is shown on *Figure 3*. The projection is unambiguous in the direction of arrows in *Figure 3*. However, in the other direction it is non-linear and has multiple solutions due to the non-linear distortion. Fortunately with the GPU (Graphical Processor Unit) we can remove the non-linear distortion form the image, so a distortion free, linear model can be used.

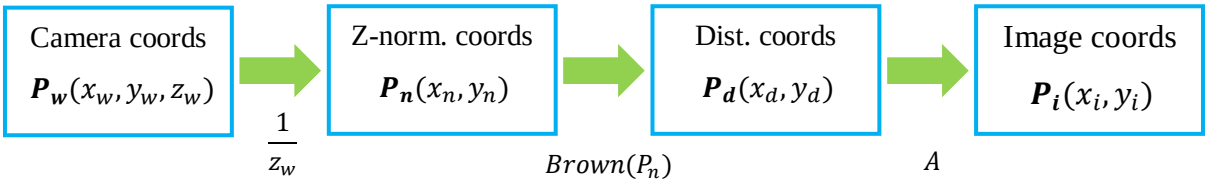


Figure 3. The projection model

This model was used for the color and the IR (depth) camera too, but first we had to estimate the parameters in the transformation matrices and distortion equations.

3.2 The calibration process

There are a lot of different methods for the calibration of the Kinect.[4-6] The calibration of IR and color camera is based on Zhang's method, which is implemented by the Camera Calibration Toolbox in MATLAB.[7] In this method multiple images are taken from different positions from a known sized chessboard pattern. Using the known relative positions of the corners of the

pattern and their position on the image an over constrained equation system can be defined, from which the value of parameters can be approximated by numerical optimization.

To simplify the capture of the necessary depth, color and IR images, we have created an easy to use interface, as shown on *Figure 4*. The captured images are processed automatically by our MATLAB script. First the corners of the checkerboard pattern are extracted with human assistance from the IR and color images (see *Figure 5*.). After that the first approximation of the camera matrix, the distortion parameters, and the position of the pattern is calculated. The 3D positions of the corners are projected back to the image plane, and a new, automatic corner selection is performed. Than the approximation is repeated, to achieve better results. The reprojection errors also show the quality of the calibration (*Figure 6*.) After this we perform the stereo calibration of IR and depth cameras, the results are used for texturing the models.

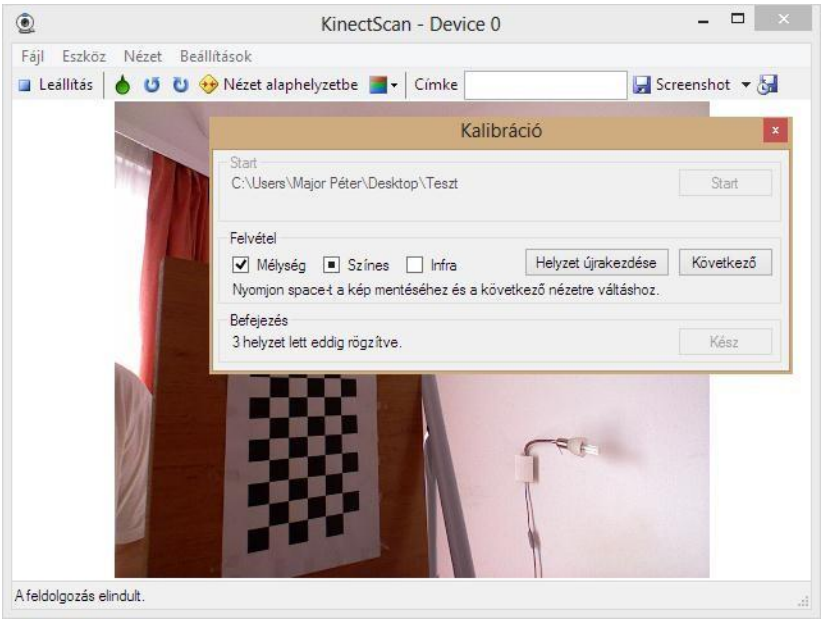
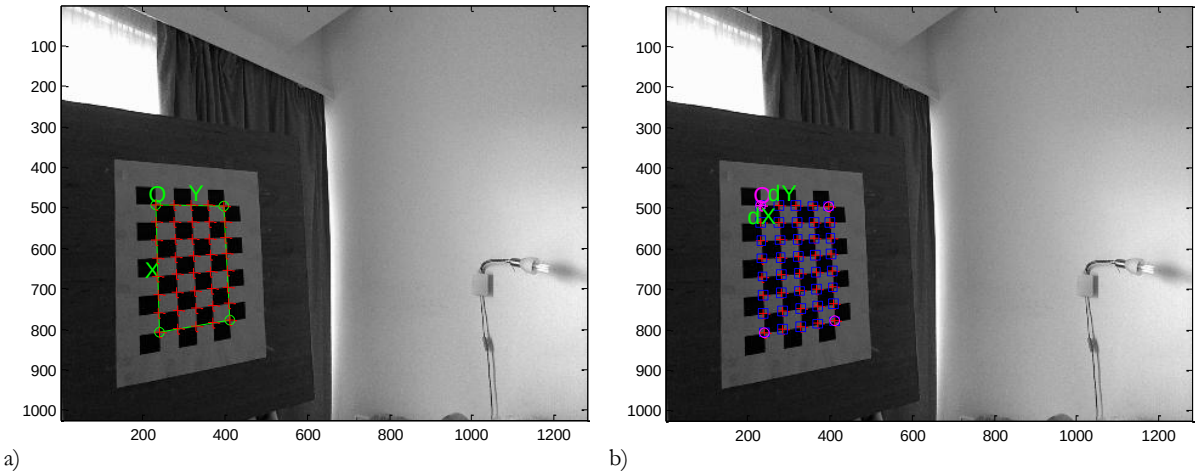
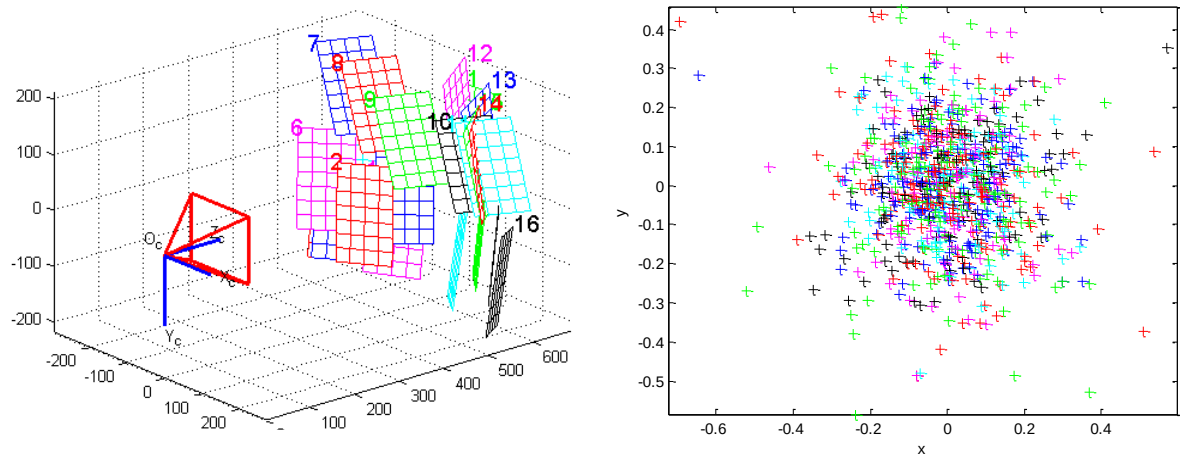


Figure 4. The calibration process



a) b)

Figure 5. Corner extraction



a) b)

Figure 6. The backprojection

For the depth calibration of Kinect we have used the estimated position of the pattern and compared it to the measured depth values. *Figure 7* shows the measured values. We have used non-linear curve fitting to approximate the depth function, and we have found the following formula:

$$h(x) = -2.33458 - \frac{35498.8}{x - 1093.09} \tag{6}$$

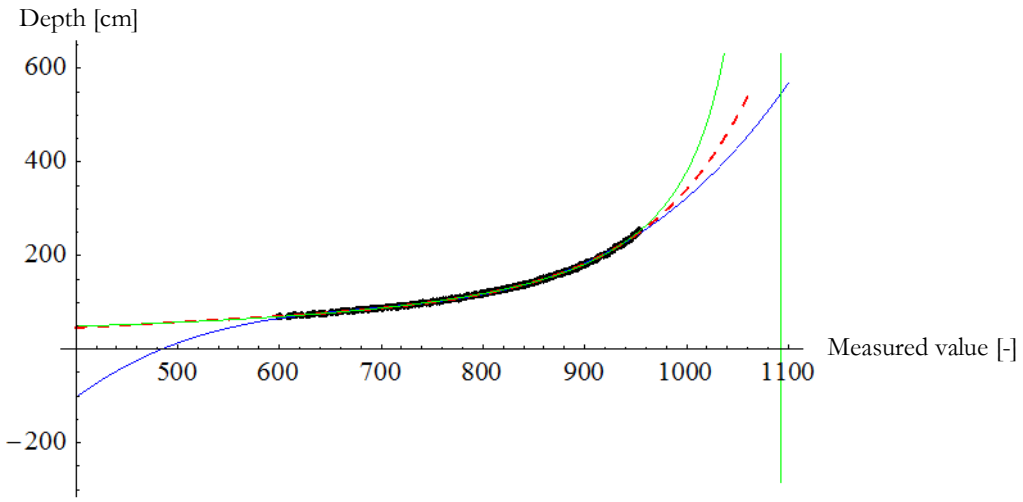


Figure 7. The depth values

4. The image processing pipeline

The result of following process is the textured 3D model of the observed environment. The process is entirely hardware accelerated by the GPU, so it works in real-time.

4.1 Control of processing

We have used the OpenKinect driver[8] because of its functionality and small resource usage. The structure of the image processing pipeline is shown on *Figure 8*.

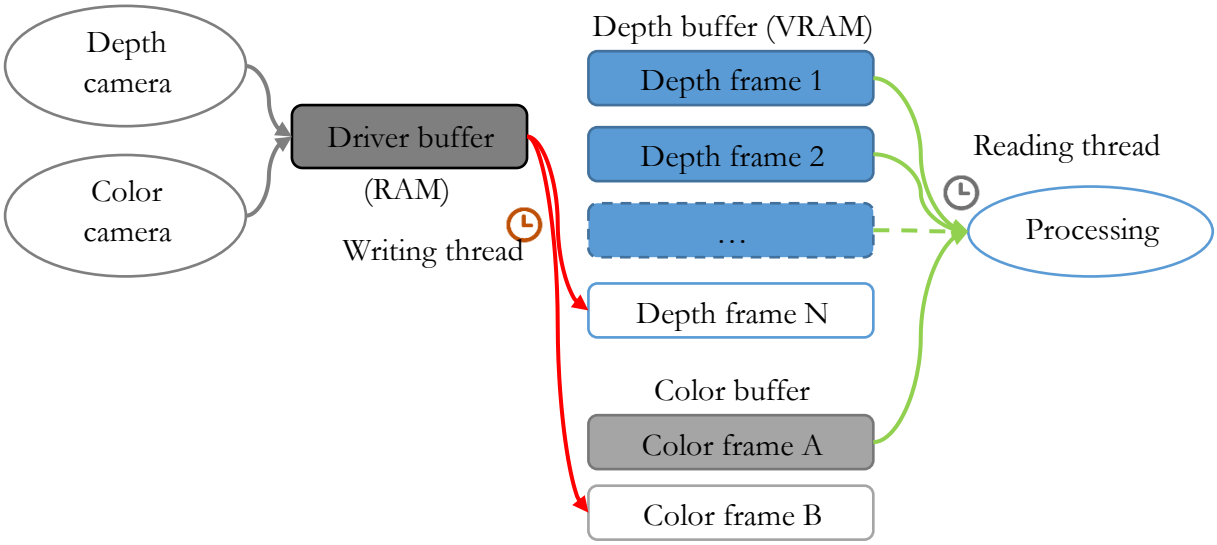


Figure 8. The image processing pipeline

As the depth and color images are captured asynchronously, first the depth and color images are uploaded to a double buffer in the video memory, where they stand ready for further processing. From this point every image processing related calculation is performed by the GPU, while the CPU controls the processing.

4.2 Hardware accelerated image processing

The application development has been done with Microsoft DirectX in the .Net environment by using the XNA framework. This allowed us to utilize the raw power of the graphics card by writing every image processing function as a HLSL shader, and achieving rapid development with the help of the .Net framework.

As depth images are rather noisy, we have applied filtering on depth images. For static and quasi-static measurements we have used temporal averaging. For dynamic measurements averaging cannot be used, so we have also implemented Gaussian filtering for spatial averaging.

From the distortion parameters we have created distortion maps for depth and color images. With these maps we remove the distortion of the images by a special shader in real time. After removing distortion the color image is ready to use as a texture.

We calculate the vector of the represented light ray for each pixel of the depth image using the inverted camera matrix of the depth camera, with the constant depth coordinate of 1. After that the real depth for every pixel is calculated using the hyperbolic formula. Than the vectors are multiplied with the associated depth values, and we get the position of the points in the coordinate system of the depth camera.

These points can be projected on the screen using Z-normalization and a virtual camera matrix. Before that the user can also define a variety of transformations by the world matrix. With this the user can rotate, scale and translate the created model. The points can also be projected to the color image. This way the texture coordinates for each vertex are defined. The whole scheme of these calculations is shown on *Figure 9*.

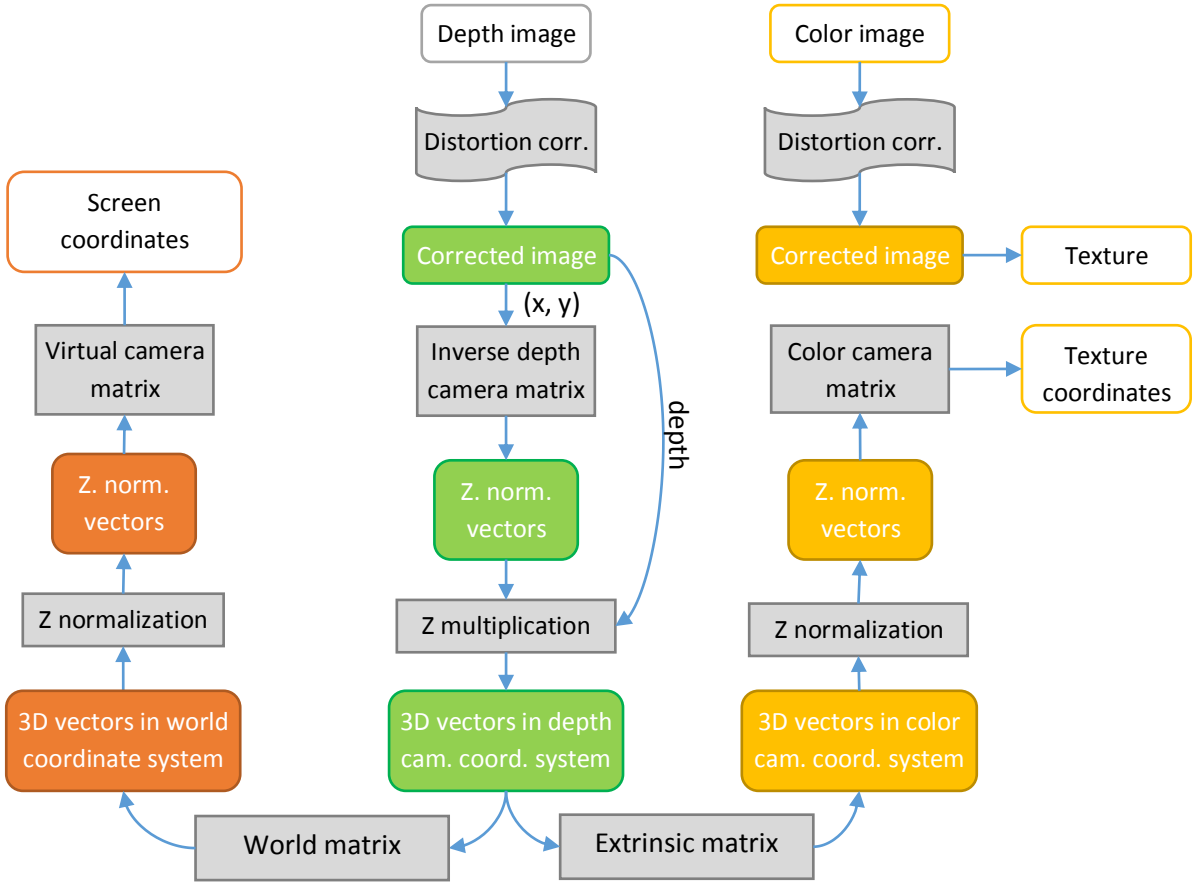


Figure 9. The flowchart of modelling

To show the 3D model we use the depth image as a displacement map for a grid of triangles. As the Kinect cannot see behind objects, it is necessary to remove some of the triangles which are facing nearly perpendicular to the depth camera. For this purpose the surface normals are also estimated, these can be used for visualization too.

5. Visualization

After the vertex processing numerous visualization modes are possible by applying different pixel shaders. It is possible to put the original texture on the model (first picture on *Figure 10.*). The surface can be also colored by using the depth from the depth or the virtual camera. By applying a sinusoidal function to the depth we get false-Moiré images (second picture on *Figure 10.*), which can be used for traditional diagnostic methods. We can specify a color scale for depth as shown on the third picture in *Figure 10*. By using the depth as the hue value in HSL color space we get rainbow shading (fourth picture on *Figure 10.*).

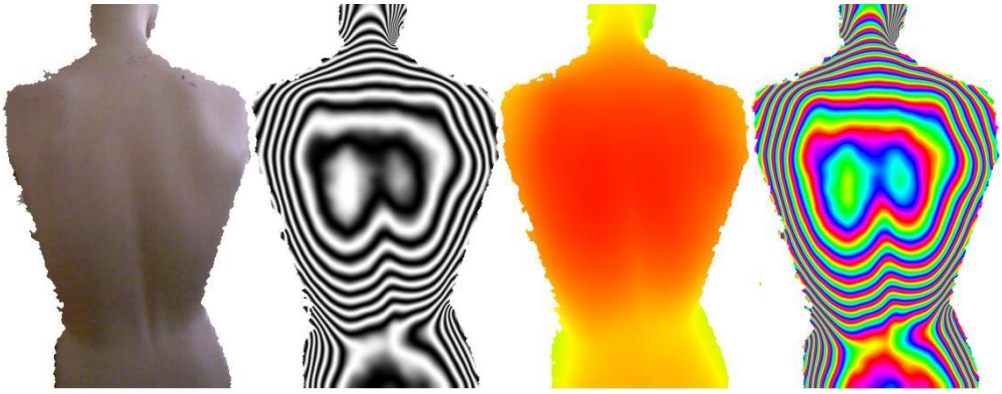


Figure 10. The visualization of the model

6. Position of vertebral column on the model

There is a hypothesis that the curve of the vertebral column can be estimated on a 3D model. In this case the 3D scanning works as a spinal mouse. In the first step we have to find the symmetry line of the back. This is computed by checking the cross sections of the back. The numerical model is based on the difference between the two parts as it shown in *Figure 11.* The minimum of the difference as a function of the position defines the symmetry point in every level.

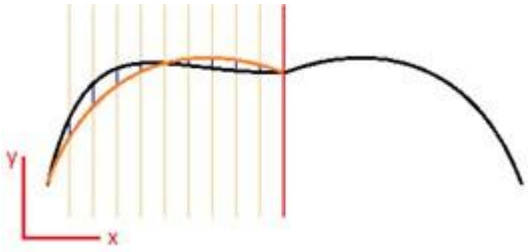


Figure 11. Searching for the symmetry curve

According to the hypothesis the symmetry point defines the position of the spine. (*Figure 12.*)

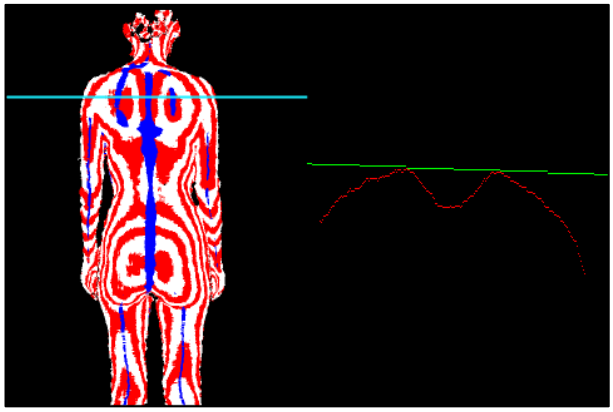


Figure 12. The estimation of the spine's position

7. Future work

The presented method can be a good basis of the following tests to prove the possibilities of the method.

REFERENCES

1. Freedman B, Shpunt A, Machline M, Arieli Y. „Depth mapping using projected patterns", United States Patent Application Publication, Pub. No.: US 2010/0118123, 2010
2. Brown DC. „Close-Range Camera Calibration", Photogrammetric Engeneering 1971; 37(8): p. 855-66
3. Vassy G, Perlaki T. „Applying and removing lens distortion in post production", The Second Hungarian Conference on Computer Graphics and Geometry 2003, Budapest, 2nd best speech.
4. Khoshelham K. „Accuracy Analysis of Kinect Depth Data", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-5/W12, 2011, ISPRS Calgary 2011 Workshop, 2011 Aug 29-31; Calgary, Canada. p. 133-38.
5. Smisek J, Jancosek M, Pajdla T. „3D with Kinect", 2011 IEEE International Conference on Computer Vision Workshops, Proceedings; p. 1154-60.
6. Herrera C. D, Kannala J, Heikkilä J. „Accurate and Practical Calibration of a Depth and Color Camera Pair", CAIP 2011, Part II, LNCS 6855; p. 437–45.
7. Zhang Z „A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000; 22(11): p. 1330-34.
8. „The OpenKinect project", http://openkinect.org/wiki/Main_Page, date of access: 2012. 09. 28.

We would like to thank to the National Office for Research and Technology (NKTH) of the Hungarian Government for their support since this study has been carried out commonly as part of the project GERINCO2 TECH_08-A1/2-2008-0121.