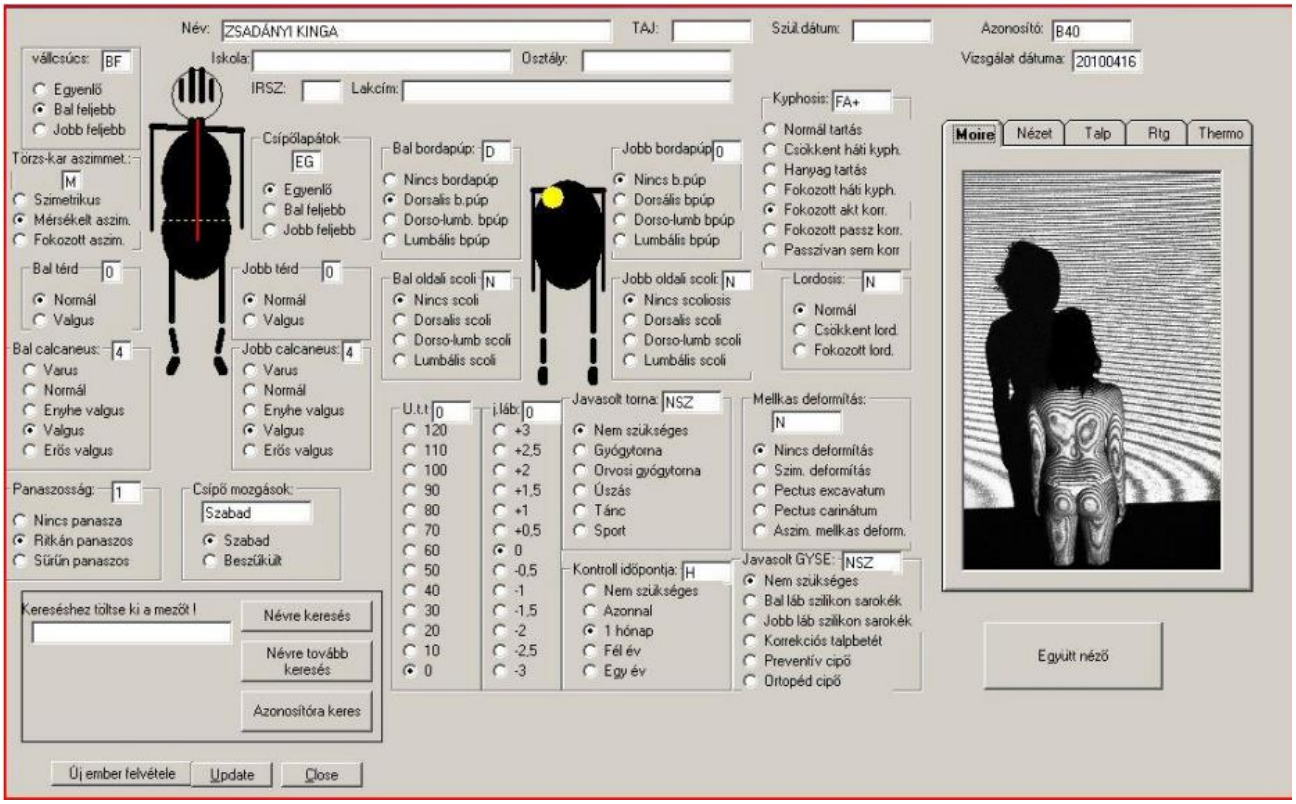
The design of the measurements in terms of methodology

Taking advantage of the screening opportunities we carried out actual screenings at several sites. In the case of manufacturing technology measurements we have only carried out experimental measurements so far.

The design of the methodology of the screenings

The general locomotor condition was assessed by a combination of different examination methodologies. The schools invited to participate in the screening programme permitted the locomotor screening to be carried out as part of the mandatory screening of 16 year-old students, in combination with the prescribed general medical screenings. To make sure that our data are verifiably documented for others as well and that they can be comparable in the case of repeated screenings, the data were recorded in several different ways. After recording the basic data on paper, the results were, in the first round, in ACCESS database and in Excel format. The data were entered in the database subsequently, after control by the chief physician. The on-line database that has been put in place had not yet been made available when much of the screenings were carried out. Of course those measurements are to be imported in the on-line database. The temporarily used - functionally perfect - data recording interface (Figure 2.1.1) already integrates all of the applied measuring techniques. (Changes will be made primarily on the basis of ergonomic considerations.) The standardised presentation of all of our software products is already being designed - this will include, in addition to the research and development organisations, the sponsor as well.



The temporary data recording interface.

The following is a description of the use of the innovative measuring technology applied in the screenings.

Screening examinations, manufacture preparation measurements

Visual examination

For visual examination by specialised doctors and for the evaluation of the range of movement and any visible alterations we applied Dr. Péter Marschalkó's evaluation and data recording method [61]. This system, which is used in the course of screenings, is supported by the Children's Section of the Hungarian Orthopaedic Association. The elements of the system include searching for and standardised description of various externally detected physical signs. Signs of backbone posture anomalies and structural illnesses of the spine can be identified by targeted examination. The so-called CVPTV system was applied in describing the condition of the legs and feet. This method provides a parametric description of the tilting of the heel bone and the knees along with any complaint, depending on frequency.

Coloured podoscope image of the feet

As was described in subsection 1.1.4 (Figure 1.1.25) the coloured podoscope image of the feet records alterations in pressure conditions. Conclusions can be drawn, from changes of multiple directions in the picture, concerning structural alterations of the spine or the feet (Figure 2.1.2)

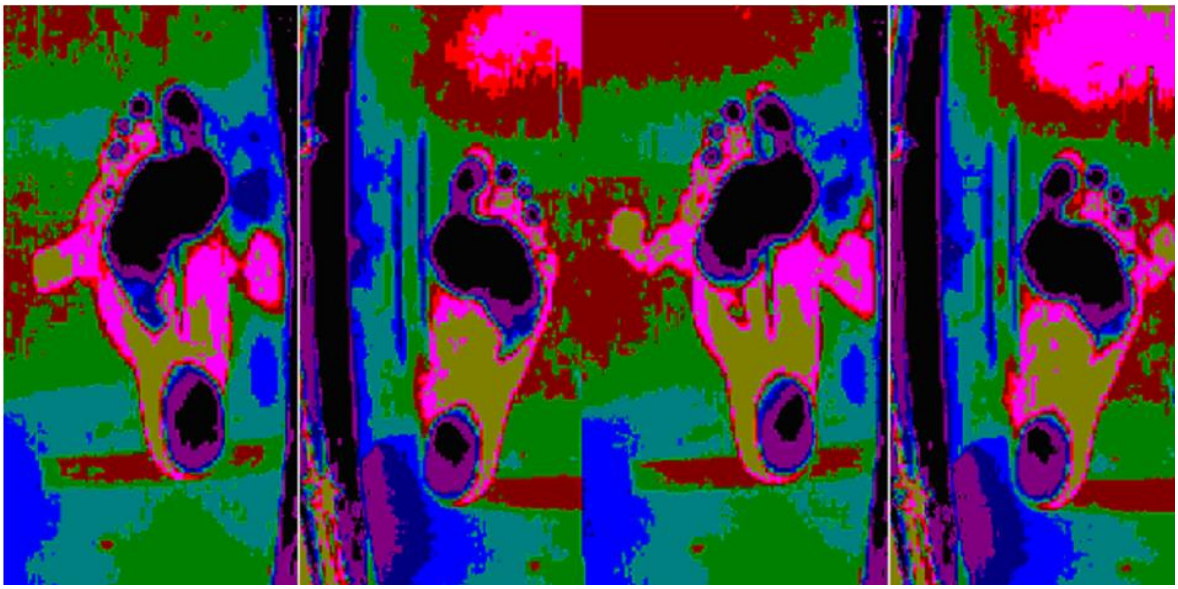


Figure 2.1.2

Podoscope images

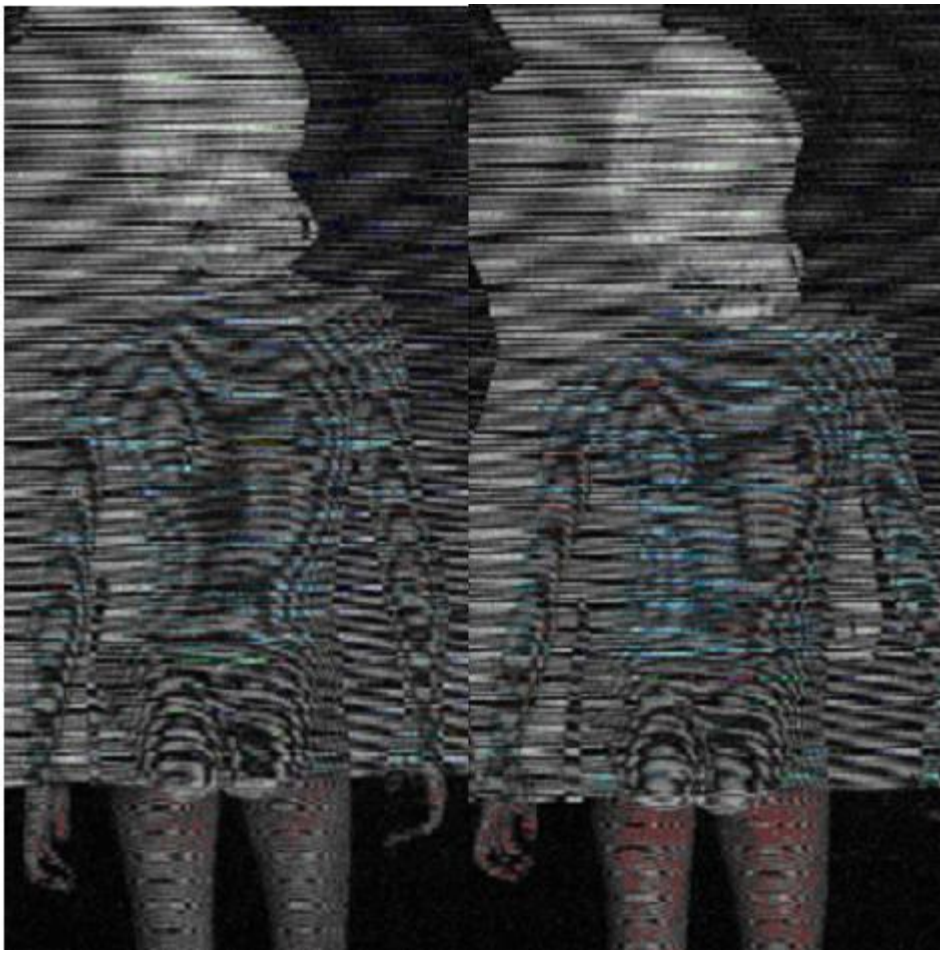
Since the most up-to-date podoscope with lifting function was available, equalisation was also possible in the case of individuals where a difference between the length of the two limbs was suspected. Those images were also stored. Figure 2.1.3a shows a moire image - unfortunately, of not the best possible quality - produced without equalisation, while Figure 2.1.3b shows the same case

with equalisation. As the patient stood on the podoscope, fully body native digital photos were taken even after limb length equalisation where it was necessary (Figure 2.1.4). If there was an X-ray image accompanying it then it was also presented (Figure 2.1.4b) along with a moire image (2.1.4c) and its Rainbow image (2.1.4.d) as well, despite the fact that no Rainbow calibration took place in the case of the majority of the screenings (the rainbow colour lighting also provides some basis of reference for the doctor to make up a diagnosis based on visual examination) and no two images based on different principles were recorded. Having been connected to the computer system providing for the interference streaks between the projected streaks and the recording lenses the moire image shows a topographic map of the trunk. Changes in the surface help making an assessment of the extent and nature of the alterations in the course of subsequent evaluation. Our system is being continuously upgraded to enhance the efficiency of the system.

Thermographic images

We also had an opportunity to test the thermograph developed by BME (the Technical University of Budapest). In addition to the native image of the back of the body we recorded the heat map as well, simultaneously with the examination. This image enables the identification of alterations in the distribution and types of muscle tensions.

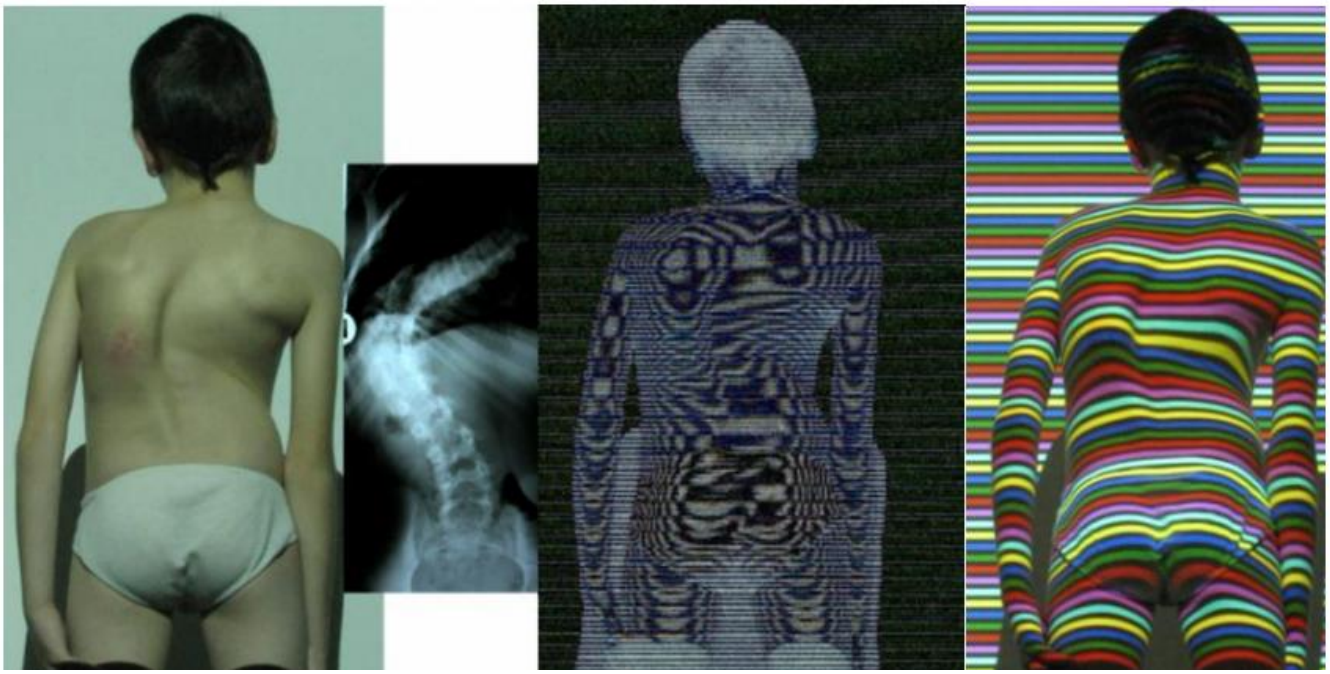
A number of parallel features can be identified between the heat map and the properties of certain diseases of the spine. In the course of the storage and further processing of the images we also explored possibilities for inclusion in the screening exercise. (Figure 2.1.5)



a. b.

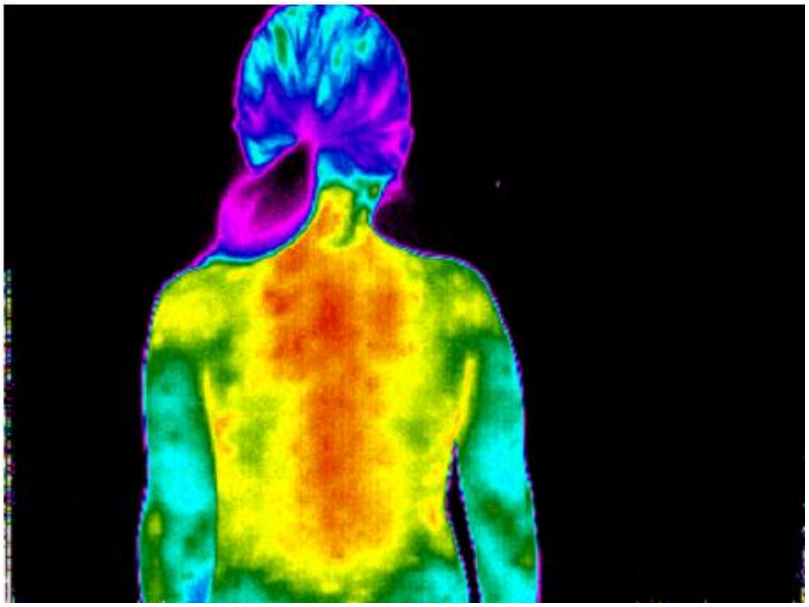
Figure 2.1.3

Podoscopic leg length equalisation images



a. b. c. d.

Figure 2.1.4
Equalised images



2.1.5. Figure
The heat map produced by the thermograph

Measuring methodology for the manufacturing technology

2.5D Scanner

In the organised event entitled Egészségliget (*Health Park*) on 6-7 June 2009 we produced 203 Rainbow images besides the moire images in rather unfavourable circumstances (in an army tent). Those images were used in working out our system's image processing methodology. The development of the software is being continued - as was noted in subsection 1.4 - in the direction of the portable 3D scanner.

3D scanner
143 images were produced, not for the purposes of corset manufacturing but only to build up experience concerning the operation of the device. (Figure 2.1.6.)

Figure 2.1.6
Test images

Measuring in a darkened room was found to accelerate processing and to improve the precision of processing. The led light sources of the calibrating sheet shown in Figure 1.4.6 and to be dimmed to enhance measurement accuracy. Disturbing structural reflections cannot be avoided if the device is not covered. Body hair is not a typical feature of girls but it is frequently encountered among boys and it disperses the laser rays. To avoid this, we purchased tight fitting clothes. The process of taking measurements could be accelerated and precision could be improved if instead of moving the camera by equal steps in some areas higher density resolution were applied in line with the height and shape of the body (between legs, in armpits), while in other areas it would be sufficient to record images less frequently (e.g. in the case of the thighs the time between triggers could be longer).

Methods of the statistical processing of the measurements

Identifying relationships between measurement data
The number of parameters is extremely large in the case of both the 3D body model detailed in subsection 1.4.6.2 - aligned to the requirements of the 3D scanner, illustrated in Figure 1.4.9 - and the 2.5D shape characteristic discussed in subsection 1.2.5, illustrated in Figure 1.2.24. Approximating the spatial point cloud (Figure 1.4.10) by body part - either with the Fourier regression discussed in subsection 1.2.4.3 or on the body parts without covered surfaces, the approximating

curves, with the approximation described in subsection 1.2.4.6 produce an immense number of data that are probably not independent from one another. Applying mathematical statistics, with the aid of correlation and covariance analysis, it is possible to scrutinise the extent of the covariance of the two variables and whether either of the data pairs can be regarded as the explaining variable. The covariance of the two probability variables is the expected value of the deviation of the variables from the expected values.

$$Cov(x_i, x_j) = M((x_i - M(x_i)) \cdot (x_j - M(x_j)))$$

and for n X_i, X_j value pairs,

$$\sum_{i=1}^{n} \frac{(x_i - \overline{x_i}) \cdot (x_j - \overline{y_j})}{n}$$

where X_i and X_j are the averages.

If components X_i and X_j are given the correlation between them

$$r_{i,j} = \mathrm{cov}(x_i, x_j)$$

and

$$cor(x_i, x_j) = \mathrm{cov}(x_i, x_j) \cdot \sigma_i \cdot \sigma_j$$

were the σ_i and σ_j are the standard deviation values of the X_i and X_j variables. It follows from the formula of the calculation of correlation that its value falls between -1 and 1. The correlation so calculated inherits the symmetric nature of covariance, i.e. $cor(X_i,X_j)=cor(X_j,X_i)$. Moreover, $cor(X_j,X_i)=1$, i.e. a variable perfectly covariates with itself, in the same direction. And if two variables are completely independent from one another, then their correlation value is zero. A -1 correlation value also indicates perfect covariance but in this case the variables change in the opposite directions, i.e. a decrease in the value of one of the variables entails an increase in the value of the other variable. As was witnessed in the case of covariance, the correlation values can also be presented with the aid of a matrix. We are planning to carry out a correlation and covariance analysis of our measurement data characterising geometry and in the most closely linked cases we carry out regression analysis. The purpose of regression analysis is to set up a formula describing the relationship between two or more parameters (variables) under scrutiny, from the available experimental (measurement, observation) data. According to the mathematical model, a function of a type (linear, quadratic, logarithmic, trigonometric etc.) characterising the phenomenon under scrutiny is to be established from the number of m measurement data (P_1,P_2,…P_m) generated for the coordinates of the **(n+1)** dimension **P(Y, X_1 , X_2 ,... X_n)** vector representing the variables:

$$\mathbf{y=f(X_1,X_2,\ldots,X_n)=f(X),}$$

whose values recorded at the positions $X_k=(X_1,X_2,\ldots,X_n)$ $\hat{y}_k = f(X_k)$ either match the relevant measured value in which case we are talking about interpolation or we look for the case where $\hat{y}_k = y_k$ $e_k = (\hat{y}_k - y_k)$ differences meet some minimum criterion (regression). This can be calculated with the least squares method, that is.

$$\sum_{i=1}^{m} e_i^2$$

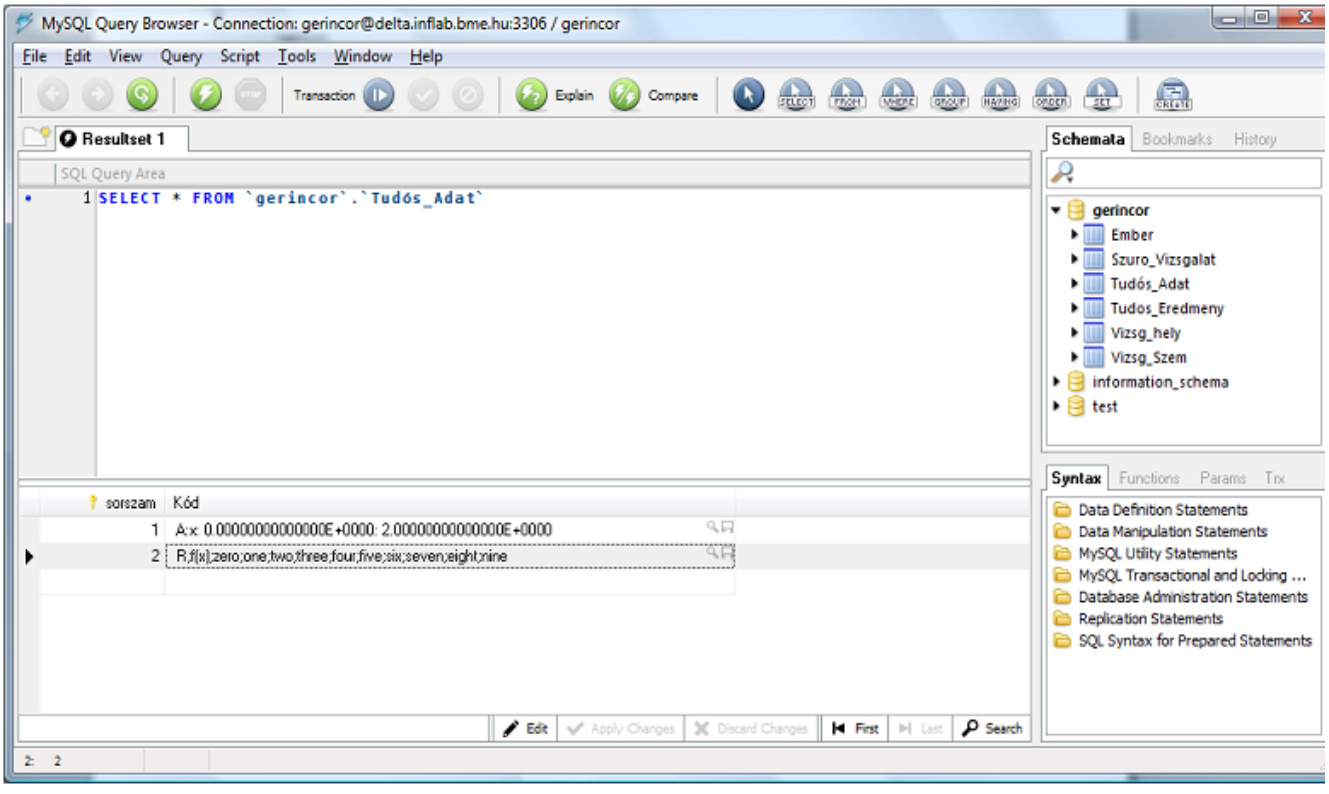
Adjustment of the body's model based on shape characteristics. The first modification of the shape characteristics is discussed in subsection 1.4. Further modifications do not take place before statistical analyses and the 'training' of the conclusion system.

Designing the database of the relationships between the data

The data storage system applied in the Tudós (*Scientist*) program to be described in subsection 4.2 is a text-encoded one. The data may be stored on real intervals (at the time of supplying data the program calculates the limits in which case the system stores the name of the Data x and the limits of its interval [0.2], as well as the name of conclusion R f(x) and its type, if it is listed, then with the list of the cases. The stored conclusions can also be found in the descriptive data file (after the ### marks).

A:x: 0.00000000000000E+0000: 2.00000000000000E+0000
R;f(x);zero;one;two;three;four;five;six;seven;eight;nine
###
1.0 ;one
0.0 ;zero
2.0 ;four

Creating the database of the relationships between data
The conclusions system can easily be integrated in the MySQL database discussed in subsection 1.5, just like by storing the descriptive records. Integration with the data tables does not take place before training.



2.2.1. Figure
Scientist data table

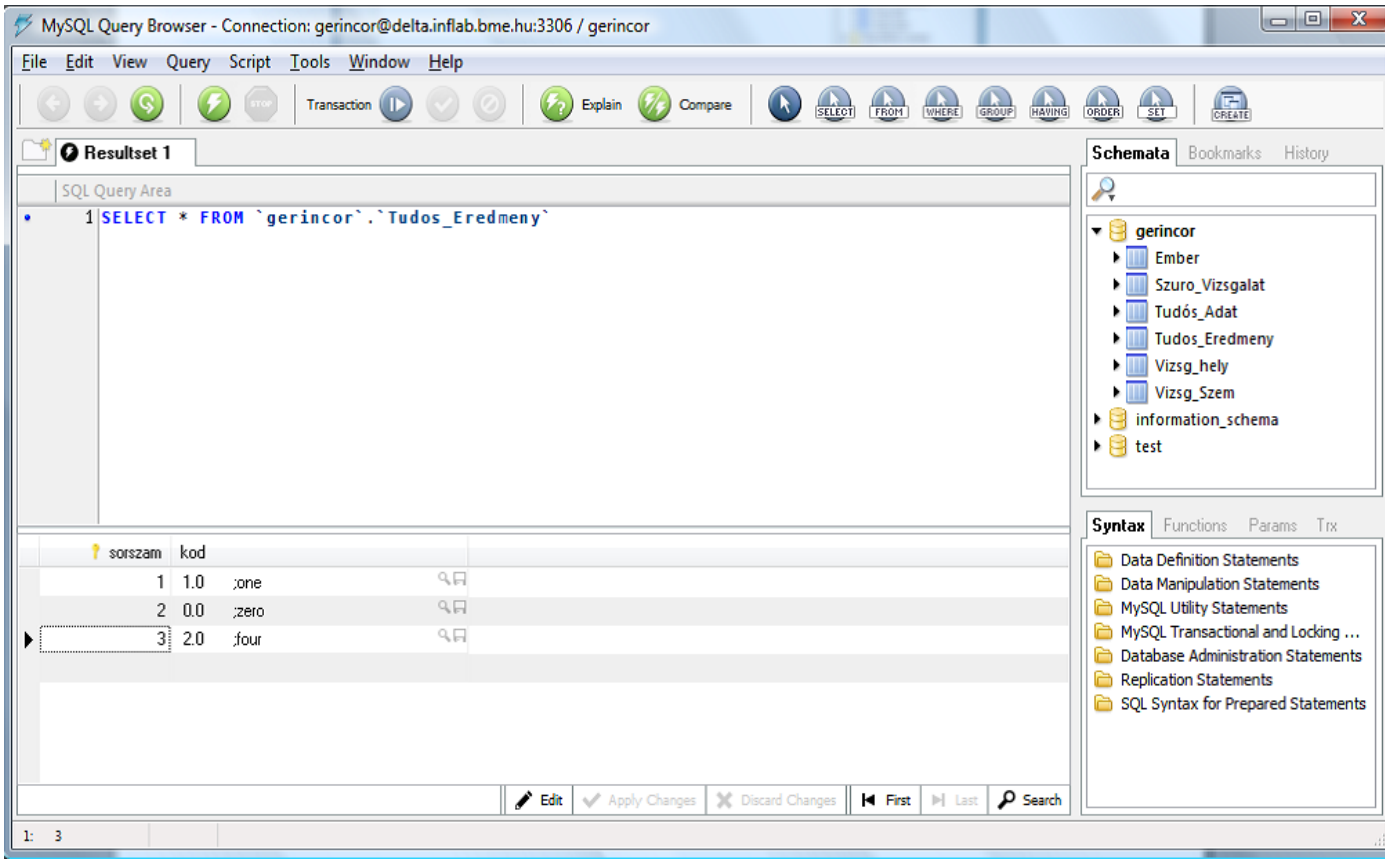


Figure 2.2.2
'Tudós' table of results

Actual downloading of the tables is part of the prospective tasks to be carried out in the course of the project.

Large numbers of measurements

The purpose of the screening devices that have been developed is to diagnose illnesses early on. Since our screening devices had been finished, we actually put them to use. With the manufacturing technology measurements we are (in accordance with the project plan) still only in the phase of gathering experience.

Mass screening examinations

Our express objective is to take out measurement with the moire method from the range of medical visual inspections. To this end, conventional X-ray images were also produced in relation to the 546 moire measurements carried out by Salus Kft., which we then recorded in a digital form as well. Digital X-ray images were made in 63 cases while in another 33 cases digital and conventional X-ray images (digitised subsequently) are also available. In the course of the screening examinations we recorded and evaluated data of 1561 children in 2010. The findings came as a real let-down.

Trunk-arm triangle asymmetry
The so-called trunk-arm triangle asymmetry can indicate the problem's suspicion even at the most minor degree of scoliosis. This is an alarming sign showing that other alterations should also be checked. This was detected in 44.1 % (!) of the children.

Scoliosis
We examined the frequency of the occurrence of scoliosis. Since the purpose of the screening is to identify stages between 'banal' and 'extremely urgent' cases, this data includes both lateral curves with a frontal plane that are not very dangerous but it also includes cases of structural scoliosis involving severe real structural distortions as well. The frequency of its occurrence is 46 % (!). After eliminating cases qualifying as 'banal' (functional scoliosis and static scoliosis caused by differences between the length of limbs), we arrive at the proportion of the cases of real scoliosis. The selection of these is made easier by the presence of what is called rib hump. There is no rib or muscle hump in the cases of banal scoliosis while in the case of structural scoliosis there is always a level difference on one side when the individual is bent down, at the height of the back, the interim section or the lumbar spine. Cases where there is no rib hump and limb shortening are categorised as functional scoliosis.

Rib hump
The frequency of the occurrence of rib humps was 31.7 % (!). Since the members of this 16 year-old group have practically completed the bone-growing phase of their growing and maturing process, this ratio is a reflection of the locomotor alterations that can be detected at the end of childhood and that affect the individual's adulthood as well. The fact that rib humps (and muscle humps as well, in the lumbar section) were found in such high proportions during the physical examination, raises a number of thoughts, which will be elaborated in the conclusions section.

FSC. Functional scoliosis
Its ratio was 4.1%. There is no rib hump, muscle hump or shortening either in such cases. This condition may improve spontaneously, but sports, physical exercise or therapeutic gymnastics help accelerate the process of improvement. It is regarded as a banal frontal plane curve.

STSC: Static scoliosis:
It occurred in 11.2% of the children we examined, where the tilting of the hip when standing upright is caused by a shortening of one of the legs. This is a result of conditions of known origins, leading to shortening, but it can also be detected without any other structural illnesses as well. Its importance is based on the fact that the curving of a compensating spine can be seen in the frontal plane. Owing to its separable and banal nature it should definitely be separated from real scoliosis.

There is no rib or muscle hump, but there is shortening!
It should be followed up, but this is not an urgent condition. Its origin needs to be clarified. If there is no diagnosable cause, then there is a need for propping the heel with a silicon insert, or in the case of a simultaneous structural disorder of the heel and the foot/leg an orthopaedic shoe correction or height increasing insoles are recommended for adolescents.

Kyphosis anomalies:
Such condition was found in 43.5% of the children and the cases are categorised as follows.

Modest:
This is found in 12.4 % of the cases, the importance of which is that there is a need for a 'healthy' thoracic curvature for a resilient and strong spine.

FA+ Increased but the individual is capable of actively correcting it:
This was found to be the case in 19.7 % of the children examined. It is characteristic of many children of a poor posture, showing the shape of a 'question mark' when standing up, a condition that has so frequently been described in literature, that they can 'brace themselves up' if they want to do so but they find this poor posture to be more comfortable. Sports, physical exercise, dancing, physical activities during spare time or therapeutic gymnastics to substitute these, can improve the condition.

FP+ Increased thoracic curvature which can still be corrected by the help of the examiner:
This condition was found 8.2% of the cases. The importance of this condition lies in the fact that this is the 'anteroom' to the M. Scheuermann's disease, this is not yet rigid but there is some problem with the structure. Continued examination is crucial (!).

FP- Excessive thoracic curvature that can no longer be corrected even by the examiner.
The curvature found in 2.3 % of the cases may point to an already existing Scheuermann's disease. Urgent further examination is necessary (!).

HT Poor posture
Instead of the 'stigma' that used to be attached to all children with a careless posture, today we use this term in relation to the 2.2 % of all cases, those that are characterised by a symmetric chest concavity. This is the reason why we established this condition in a significantly smaller percentage of cases than before. The structural distortion of the chest is fixed as a consequence of the looseness of the abdominal muscles and owing to the forward tilting of the shoulder girdle. This shape cannot be rectified, no matter how hard the child tries to straighten up. Sports, physical exercise, therapeutic gymnastics may help.

Frontal chest deformities:
These are found in 8.1% of cases. Among them, 2.2 % is made up of the symmetric chest deformity linked to poor posture, its asymmetric version makes up 3.8 %. Pectus carinatum and pectus excavatum is found in 0.2 % and 1.1 % of children. In the case of the last three groups examination by a specialised doctor is required in view of the unity of chest and spine, along with medical therapeutic gymnastics, where necessary. In order for the development of the key spine deformities to be detected in time and for the necessary effective treatments to be started, a general overview should be formed of every single child at the age of 10-11.

Screening for structural deformities of the foot:
The position and condition of the heel was categorised according to the CVPTV system, an approach that has been increasingly widely adopted in the field of orthopaedic shoe making s well, during the recent period.

Accordingly, an over 20º valgus position of the heel bone: CV4, increased heel valgus position: FSV. Over 30º: high degree heel valgus, NSV. In both groups there is a need for treatment with a suitable medical aid, preferably corrective orthopaedic shoes, and from adolescence with tailor-made orthopaedic instep-raisers. Emphasis should also be lid on strengthening the feet by sports, balancing and walking barefoot. In the cases involved in the screening we found 19.7 % CV4 and 2.4 % CV5, the two categories together making up 22.1 %. Screening should be carried out at the age of 6-7 to identify structural alterations of the feet and to make it possible to start effective corrective treatment!

Findings, conclusions and questions

A number of specialised doctors participated in the examinations. In the course of manual examinations - having reviewed the observations of the experienced specialised doctors newly joining the programme - we found no statistically identifiable differences in the judgements of the conditions of scoliosis, rib hump, thoracic curvature or feed deformities. In the case of a rib hump the text books prescribe that the nature of the process must always be clarified with the aid of X-ray images. Differences between the planes of the ribs may indicate only a difference of a few Cobb degrees but it may also indicate a major distortion in the background with significant bone alterations. X-ray images must be made of the spine, though it should be done according to a carefully planned schedule. To enable a comparison of the Moire image and the X-ray picture the camera must, by all means, be fixed in a vertical position. Our experience showed that as a result of the large number of on-line input data the measurements require a high power computer with ample memory storage capacity. The high proportion of children with structural foot problems requiring treatment also raises a lot of questions! Why do one in five children have to need special doctor's help and medical aids at the time when body growth is practically completed? The proportion of the detected deformities may be the result of the success of a 'consistent treatment system' starting from a materially higher proportion, in the course of 'today's effective specialised orthopaedic services and orderly screenings', or it may be the other way around, so that owing to the defective service providing system and the shortcomings of the screenings even the alterations that could have been treated and remedied earlier on, have developed into serious cases. We have no sufficient data for comparison. How is it possible to build our future on these young people with such poor spine and foot structures? Our conclusion is that the status assessment carried out at the age of 16 highlighted that general structural examinations of the spine and feet carried out at the ages of 6, 9 and 12 would make it possible identify the most important orthopaedic malformations in time so that they can be treated effectively. In our view the population's general status survey should be carried out in a concentrated way in the above age groups. Of course after categorising the identified cases by degree of severity phased and scheduled orthopaedic examinations are indispensable, in view of the findings of which the provision of means and capacities for effective treatment is also crucial!

Development of the Mobile device, this development has made automated diagnoses possible

Introduction

The aim of my paper is to describe the possibilities of utilising depth sensors for medical purposes. Depth sensors that carry out an imaging function similar to the mapping process performed by regular photo cameras with the difference that they specify the distance of the spatial dots mapped by the various pixels relative to the image plane, can be used in a wide range of fields of the health sector. They enable the production of three dimensional models of the human body surface, allowing automated or semi-automated diagnoses of a variety of deformities (e.g. scoliosis), the designing and production of aids, special clothing items and braces closely aligned to the body and a lot of other useful solutions. Their advantage lies in the fact that imaging can be carried out quickly, without inconvenience for the patient, it needs no preparations and can be repeated any times.

Although devices suitable for depth imaging have been available for quite some time now, a few years ago these had been usually expensive and in many cases not quite matured solutions. Laboratory prototypes had been under development as early as in the 70s and 80s, one of their first areas of use was engineering where the new technology enabled quick and fairly precise quality control checks of some of the components. Another important area was the navigation of robots, since with the aid of depth sensors they could adequately detect obstacles in their way. From the 2000s on significant progress was being made in the development of an increasing number of new concepts, resulting in compact, real-time, relatively precise and inexpensive solutions with little computing capacity requirements. The increasing and every wider availability of depth cameras enabled the development of a wide array of new devices. Gesture controlling, real-time model

generation, mobile three dimensional scanners, 2.5D photography/filming and the modelling of the environment may perhaps be highlighted as the most important ones among the large number of new fields of application. Depth sensors make it possible to produce three dimensional models of the environment without the sensor's operation disturbing people or animals present and subject to the applied technology even rapid movements can be recorded and high depth resolution can also be achieved. These methods are still in a stage of continuous development, primarily with the aim of reducing the measurement noise and at increasing resolution. Other important goals include reduction of the size of the sensors and their power uptake so as to make them suitable for incorporation in mobile devices.

.

I wish to introduce two devices in this paper that are suitable for two different medical tasks: the first one is intended to help diagnosing spine problems, while the other one is aimed to assist the design and manufacture of special, tailor-made shows and shoe-lasts, by producing three dimensional models of patients' feet. I will describe the structure of the measuring system developed for these applications, the operation of the program processing the measured data, along with problems encountered in the course of the program's implementation and the solutions worked out. Moreover, the study contains a detailed discussion of the calibration of the applied depth sensors, the creation of the model of their mapping function, the software packages that can be used and the associated difficulties. In this way, in addition to describing two concrete examples, I also give a general overview of the application of depth sensors. In the first part of the paper I describe the devices and instruments used and the available choices and this will be followed by a detailed account of the calibration and mapping of the measuring systems. In the main part of the paper I will describe the above mentioned practical solutions and the closing part will contain an analysis of the results and an outline of the possibilities for further development.

2.1 A review of the depth mapping techniques

The calibration of depth sensors makes it possible to compose functions describing their imaging on the basis of which the inverse of mapping can also be carried out and it is also possible to produce a three dimensional model of the mapped objects, from the depth image. Of the environment depth sensors 'see' only the surfaces towards the camera, so the model should be regarded more s a 2.5 dimensional one. Nonetheless, solutions can be worked out with the aid of which, by combining multiple images, true three dimensional models can also be produced. The sensor I used maps the entire field of view in a single step - rather than scanning - making it possible to produce real-time images. Since the various images can be treated as simple pictures, they are easy to store and process at a later stage. One disadvantage however, is that the solution produces a noise and low resolution output, another is that depth sensitiveness decreases rapidly with increasing distance.

There are a several different methods for the production of depth images, from among which I will describe the most important ones below [14]:

a) Laser triangulators: These machines comprise a laser of deflectable rays and a camera. The laser lights the object in one point or along a line and the camera detects the light returning from the object. The three dimensional position of the point then can be determined by simply triangulation in view o the system's parameters. The advantages of this solution include high precision and insensitiveness to surface texture, reflectivity and lighting. Since the surface is mapped by dots along lines, it is not highly suitable for taking real-time measurements though with the aid of a high speed camera and a similarly rapidly scanning laser even this is not impossible.

b) Structured light: these sensors project a pre-defined pattern onto the surface to be measured and then they determine the spatial positions of the dots by triangulation, as in the case of the above solution. Since these methods scrutinise the entire surface at the same time, they are suitable for

real-time operation as well. The patterns projected by the device may be various grids, point matrices etc. If the various elements of the pattern can be identified in the image recorded by the camera, their spatial position can also be determined.

c) Stereo vision: this solution may, from a certain aspect, be regarded as the passive version of the above solution, as in this case no pattern is projected onto the model, but its natural texture is used and it is observed with two cameras. Recognising the typical elements of in the images recorded by the cameras and matching them on the images of the two cameras their distances can be determined. The drawback of the method is that it can only be applied successfully if the surface under scrutiny has a sufficiently large number of clearly identifiable details. The method may be supplemented by a projector projecting a pattern onto the surfaces to create a suitable texture on the surfaces.

d) Photogrammery: This method is similar to the previous one, but in this case a three dimensional image is produced on the basis of pictures taken by one camera from different positions. This requires profound knowledge of the optical parameters of the camera used an and the relative changes in the position of the camera between the various pictures. This then leads to a task similar to the stereo mapping. This method cannot be used in real time, since it requires relative movements of camera and environment.

e) Measuring the time it takes for light to travel: in this case the object to be measured is lit with laser impulse, and then the time it takes for light to return to the camera is measured pixel by pixel. This method is suited primarily for larger distances, because in the case of smaller distances it takes time measuring circuits of very high resolution. One disadvantage of the method is that it manages reflecting surfaces very poorly. Also, the strength of the laser beam needs to be limited to protect life.

f) Interferometry: This method projects periodical patterns - changing in space or time - onto the surface, and then the reflected light is demodulated with the aid of a suitably selected reference grid. In the image resulting from the procedure the phenomenon of interference makes the differences between the distances of the dots of the surface visible. The most important advantage of such systems is their high degree of accuracy.

g) Moiré contours: this method lights the object with a beam of light through a grid and then monitors the returning light with a camera sideways from the projector, with an identical grid before the camera as well. The resulting interference makes depth measurements possible.

h) Based on focusing: the blurring of details indicates distance from the focal distance.

i) Based on shading: pictures are taken while the source of light is moved around the object and changes in the reflected light are monitored. The geometry of the surface can be worked out in view of the rules determining the reflection of light from the surface of the given object (Lambert etc.).

To resolve the task on hand I used Kinect applying the structured light principle from the above list but the bulk of the processing algorithm does not depend on the mode of the generation of the depth image thus the majority of the methods presented here can be used with other depth sensing techniques as well. The advantages of using structured light include the following:

- the required hardware is simple and is not expensive to implement,
- it enables real-time processing,
- it enables the creation of compact-size sensors,
- it works even in the dark,
- most hardware elements transmit depth images therefore there is no need for ‘machine-intensive’ processing operations for producing this.

A noisy output is, unfortunately, a drawback, just like the low resolution (in depth as well as in parallel with the image plane).

2.2 The applied software and hardware tools

Most of the image processing program was written in the C# language in the Microsoft .Net framework system [22], along with some of it written in C++ and C++/CLI. Image processing takes place almost entirely on the GPU, which I programmed in the HLSL language [18, 21] with the help of DirectX API. Since image processing is carried out by the GPU, the CPU load is relatively low. The graphic card was accessed from the .Net environment through the XNA framework system which maps the DirectX API in the .Net classes. I produced the program with the open source code freenect driver of the OpenKinect project in communicating with Kinect. Much of the development was carried out in the Microsoft Visual Studio environment. For calibration I also used the Camera Calibration Toolbox module of the MATLAB software.

The Microsoft Kinect depth sensor was used, which I was provided with by the Department of Mechatronics, Optics and Mechanical Engineering Informatics. The hardware of the rotating scanner is being developed by László Basch and it was also him who supplied the images produced with the tool. The main hardware requirement of the running of the rotating scanner is to be met by the graphic card in the case of which full support of the DirectXShader Model 3 or a later version is required. The required CPU and GPU capacity can be met by the average machines used today, because processing is highly optimised: few operations are carried out by the CPU and although the number of the operations performed by the GPU is relatively high, but the capacity of today's cards exceeds this requirement many times over. The memory requirement is about 70-200MB depending on the way of usage, the bulk of which is made up of the large amount of image and model information used in the course of image processing

.

3 Kinect calibration

3.1 Hardware structure

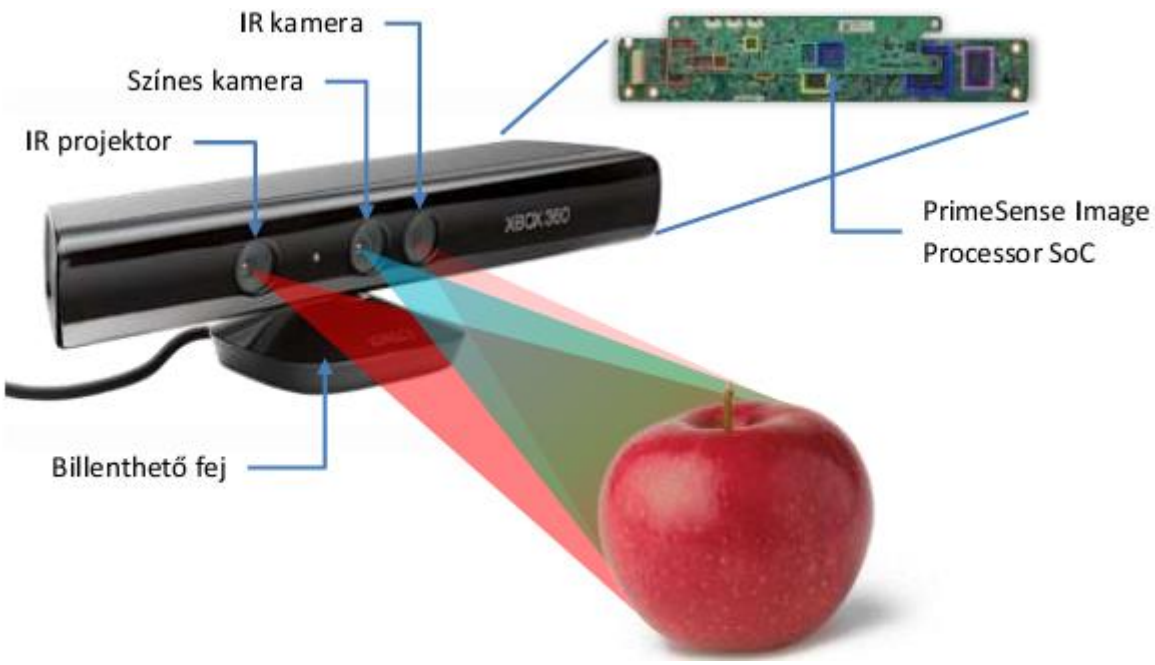
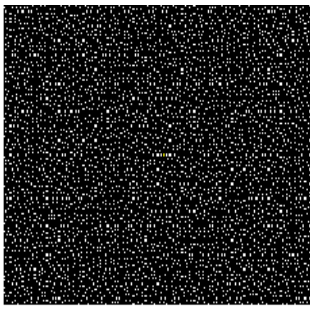


Figure 1 **The structure of the Kinect sensor**

From the aspect of depth measurement the Kinect is made up of four important modules. It contains and infrared projector which leads laser light through filters to create a pattern made up of dots [1]. Though the dots are set out on a regular grid their local arrangement is not regular. The population of projected dots is clearly made up of 3x3 segments, within which the arrangement of the dots is identical. Based on another patent of the company PrimeSense it is likely that only one pattern is generated and then it is optically copied [2]. Since the pattern is repeated only three times in both directions, this can be used for determining the positions of the various dots within the pattern. The Company named this technique Light Coding. The pattern is binary so the various dots can be clearly highlighted from the picture which is also helped by the fact that the light areas are a lot smaller in the pattern than the dark ones. The projector operates in the infrared domain, as a result of which its operation, or rather, the visibility of the pattern projected by it is not affected by environmental light conditions nonetheless, this method is difficult to use outdoors where sunlight is strong enough in the infrared domain to make the picture projected by the projector invisible in comparison to the sun's light. The pattern can be best detected when reflected from dull surfaces because if the surface reflects light, it blinds the camera looking at the pattern if it is facing the surface nearly perpendicularly. Besides reflection, problems are also caused by surfaces that do not reflect enough light or are nearly parallel with the optical axis. The various segments



.... is a dot of a diameter of (Figure 4.). These are the original dots of the system whose detection is a lot more reliable - on account of their higher power - than that of average dots.

Another important element of the instrument is an infrared camera providing an image of various shades of grey this camera monitors the
image projected by the projector. Since the projector's optical axis is different from that of the camera, shadows may appear in the image of the infrared camera. The reason for this is that the camera sees even surfaces that are not visible to the projector. This phenomenon is particularly strong when an object is close to the Kinect and the background is a lot farther back, when as a consequence of the large shaded area much of the background is not visible (Figure 3.). The light beams are not highly visible. The range of depth vision is limited at the same time by the absorption of light to about 6-8 meters. The infrared camera has a focusing mechanics as

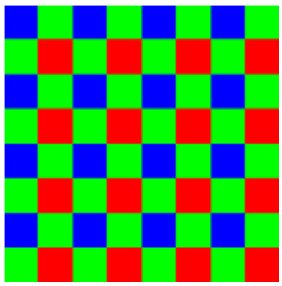
3. ábra Az infravörös projektor és kamera széttartása a távolság növekedésével láthatóvá válik, eltérő helyzete miatt árnyékok keletkeznekazonban a kamera vetítése miatt ez a kimeneten csak balra a szék háttámlájának léceitől

well, to enable it to operate in various distance ranges
Depth sensing has not only an upper but a lower limit as well because in the case of objects close to the instrument the strength of the light emitted by the infrared projector blinds the infrared camera. This phenomenon may be improved by attaching filter to reduce the strength of light (but in this case the upper limit will also become shorter). The experiments showed that in the case of highly non-reflective and dark coloured surfaces a minimum distance of 40 cm needs to be kept but up to 60 cm the output may still contain spots that could not be sensed.

Figure 4 The outputs of the Kinect's colour and infrared camera at a resolution of 1280x1024, even some of the elements of the 9x9 grid can be seen in the infrared picture

The third important element of the Kinect is the colour camera with the capabilities of a top category web camera. Though the colour camera is not involved in the sensing of depth, the importance of its role lies in the fact that it enables the creation of suitable textures as well to go with the geometric models. Though both the infrared and the colour cameras are high resolution cameras (1280x1024 or higher), in this resolution the information cannot be transmitted between the device and the processing computer at speeds over 7-10 FPS owing to the limitations constituted by the USB 2. Moreover, the system clearly visibly increases the exposure time in this case, so the picture will show blurred streaks. Another problem is that the content of the image may be changed while being transmitted, thus for instance

.

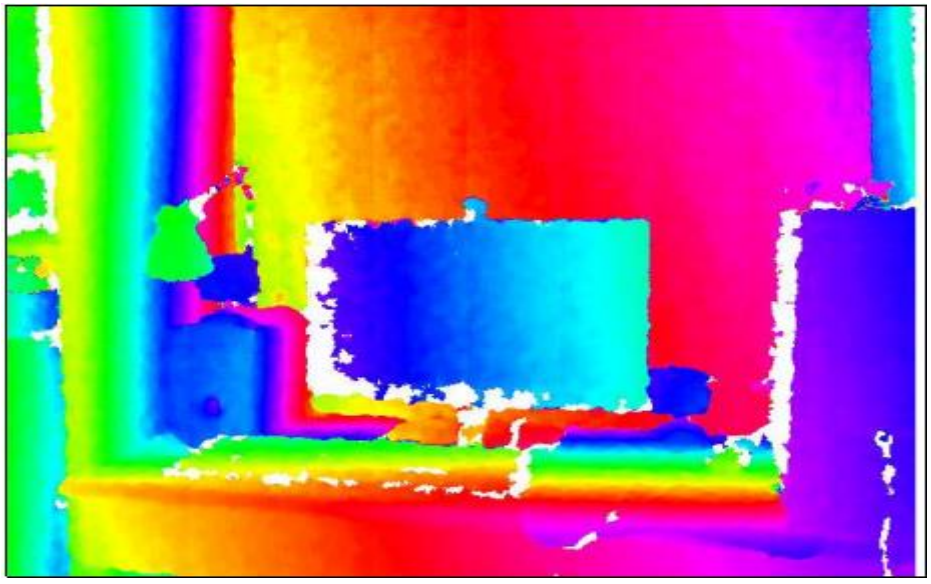


in the course of the rotation of the camera the environment 'tilts' since when going from top down the various lines were captured at different times. The video can be transmitted at a suitable speed (30FPS) only in a 640x480 resolution (depth and colour together). Even in this case there is a need for compacting, e.g. the colour video does not include the colour components separately for each pixel, instead, it encodes the colours according to the so-called Bayern pattern (Figure 5), as a consequence of which the image will be less sharp.

As has been noted, in the base case the Kinect transmits colour information of the encoding used in the case of depth and pictures.

The depth information is generated by the processing unit built into the Kinect, where the processing unit compares the pattern seen in the picture taken from the infrared camera to a reference pattern and it observes the shifts of the dots resulting from projection onto uneven surfaces. Since the resolution of the net of dots is low and the dots are not evenly distributed, therefore Kinect performs a process of interpolation which enters less error in the case of more smooth surfaces while along edges it produces a strongly jagged pattern which is made even worse by random noise s well (Figure 6). Since the Kinect fails to receive information continuously on quite a number of dots, therefore the system tries to fill these hiatuses. The measured values are transmitted by the system in a 11 bit form of a non-linear scale. Vertical streaks of equal widths are clearly observed in the depth image the cause of which is not known, owing to the confidentiality of the implementation of the system. These make smooth surfaces slightly groovy (their effects is visible particularly in the case of larger distances owing to the system's characteristic curve, see below), thus they need to be smoothed. What is unfavourable is that these keep moving and their width may vary widely in the different frames. The cause of this is not known either, but it may likely be related to the system's autofocus algorithm, if the content of the viewed picture does not change the movement of the vertical streaks comes to a halt after an initial settling period

.



6. Figure The depth output with rainbow colour shading of a 50 cm periodicity and averaging 32 frames

The Kinect's internal signal processing unit therefore provides a depth image of a resolution of 640x480 and a bit depth of 11 bits. In the course of the manufacturing process Microsoft calibrates the sensors, writing the calibration results in the unit's permanent memory. The information is read

out by the drivers thus it is possible to determine the three dimensional position of the detected dots. The problem with this solution is that the purpose of the Kinect is to implement gesture controlling, thus the calibration that is carried out is not accurate enough for three dimensional measurements. To eliminate this problem I worked out my own calibration method which is precise and can be relatively quickly executed.

.

3.2 Kinect's mapping model

The first step of calibration is to work out the model of the device to be calibrated. To this end, one needs to get to know the mapping rules applied in the Kinect imaging process, which rules I modelled with the aid of a pinhole camera model and Brown's distortion model. First I describe the mapping model of the infrared camera required for depth imaging. Let us take a three dimensional object, and one surface point on it (P) The position of point P! should be known in the coordinate system of the camera ($P_W(X_W,Y_W,Z_W)$) which is a right-angled coordinate system. The origin of the coordinate system is positioned at the aperture of the imaginary camera. The system of coordinates and its axes are positioned horizontally and vertically in line with the picture plane (their direction is determined in accordance with the digital storage of the image to make it easier to work with, thus in the picture they point from left to right and from to top to bottom), the Z axis is aligned to the camera's optical axis. In this system of coordinates every point has three coordinates. Let these be , and ! As the first step of the mapping process normaled coordinates can be derived from these ($\mathbf{P}_n$ (X_n,Y_n)), which are thus positioned in the Z =1 plane. This step also illustrates that the objects at larger distances appear to be smaller. This is followed by the modelling of image distortions.

The image distortions are made up of two components, in accordance with the Brown model. In the case of the radial distortion the distance of the points of the image relative to the optical axis changes according to

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

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

Where:

- $\mathbf{P_d}$ (X_d,Y_d): the coordinates of the distorted point
- $\mathbf{P_n}$ (X_n,Y_n): coordinates of the normalled point
- *ki*: radial distortion coefficients
- *dt*: tangential distortion

From among the coefficients in the formula it is only the first two that are used most frequently, while in the case of wide optic angle the first three coefficients are used. It should be noted that most calibration methods can only determine the third or subsequent members only with a degree of uncertainty comparable to their values, thus their use can increase the level of error. Another distorting component is the so-called tangential distortion which is usually small in the case of today's modern cameras but it is not always negligible

.

Tangential distortion can be described with the following formula:

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

Where the t coefficients are the tangential distortion coefficients. The sum of the two distortions describe the Brown distortion model. The definition of the distortions is followed by the second part of the camera's perspective projection transformation (where the first one was normaling with the component (**Z**). Though this transformation takes place between coordinate systems in one

plane, owing to the homogeneous coordinate form its description needs a 3x3 matrix. This matrix contains the system's vertical and horizontal focus distances in pixels along with the distance between the point where the optical main axis crosses the picture plane and the central point of the picture. This latter can also be viewed as though the CCD chip constituting the picture has shifted in comparison to the optical system. This new transformation is described s follows:

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

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

Where:

- **A**: is the camera matrix
- fx,fy : the focus distance and direction
- *Cx,Cy*: the coordinates of the point where the picture plane is crossed by the optical axis
- *S*: describes the deviation of the axes from the perpendicular, here
- ***P*** *i* : the position of the mapped point in the picture

This step results in the position of a real spatial point defined in the camera's coordinate system as mapped in the picture plane. The relationship of mapping is not unambiguous in the reverse direction owing to the picture distortion: though the distortion formula assigns only one distorted point to an undistorted one, but it does not work in the opposite direction. In this case more than one points could belong to one pixel but only one of them is the true solution, the others originate from the model. In the opposite direction the mapping formula turns into a system of non-linear functions the solution of which is more complex and it takes more calculations as only a numerical solution is possible. As will, however, be shown later on, owing to the operation of the graphic card the reverse process will not have to be carried out in order to eliminate distortion.

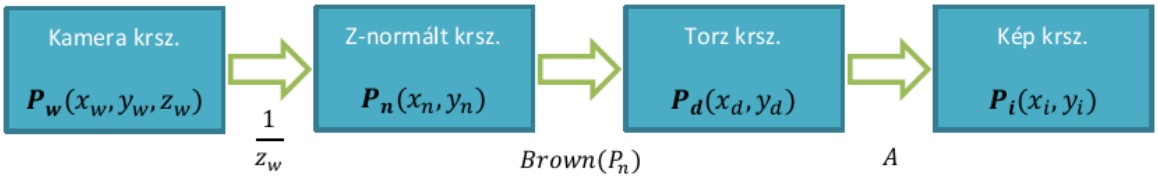


Figure 7. The camera's mapping model

This model is suitable for the description of both the infrared and the colour camera. Since the cameras have profoundly different attributes, the features describing the transformations and distortions can also be rather different. As the two cameras view the world from different points of view, connecting the points in the pictures they have mapped takes further calculations

.

3.3 The calibration process

3.3.1 Taking calibration pictures

can be identified in contrast pattern which makes it possible to identify the constants describing mapping based on the knowledge of their sizes. The task at this point is to explore the relationship between the two systems of coordinates. One system of coordinates is attached to a chessboard (e.g. to one corner of it) while the other is the picture's system of coordinate the origin of which is to be found in the upper left corner while its axes are positioned - in line with the storage of pixels - from left to right and from top down. Since the positions of the points are known both in the chessboard

and the picture coordinate system, and the transformation functions that can be defined between them are available in a parametric form as well, it is possible to calculate those parameters. Since the positions of the pairs of points are loaded with noise, it is only possible to produce a reliable and accurate result if a sufficient number of them is available. As the transformation is nonlinear and the task is overdefined, it can only be managed with complex calculations.

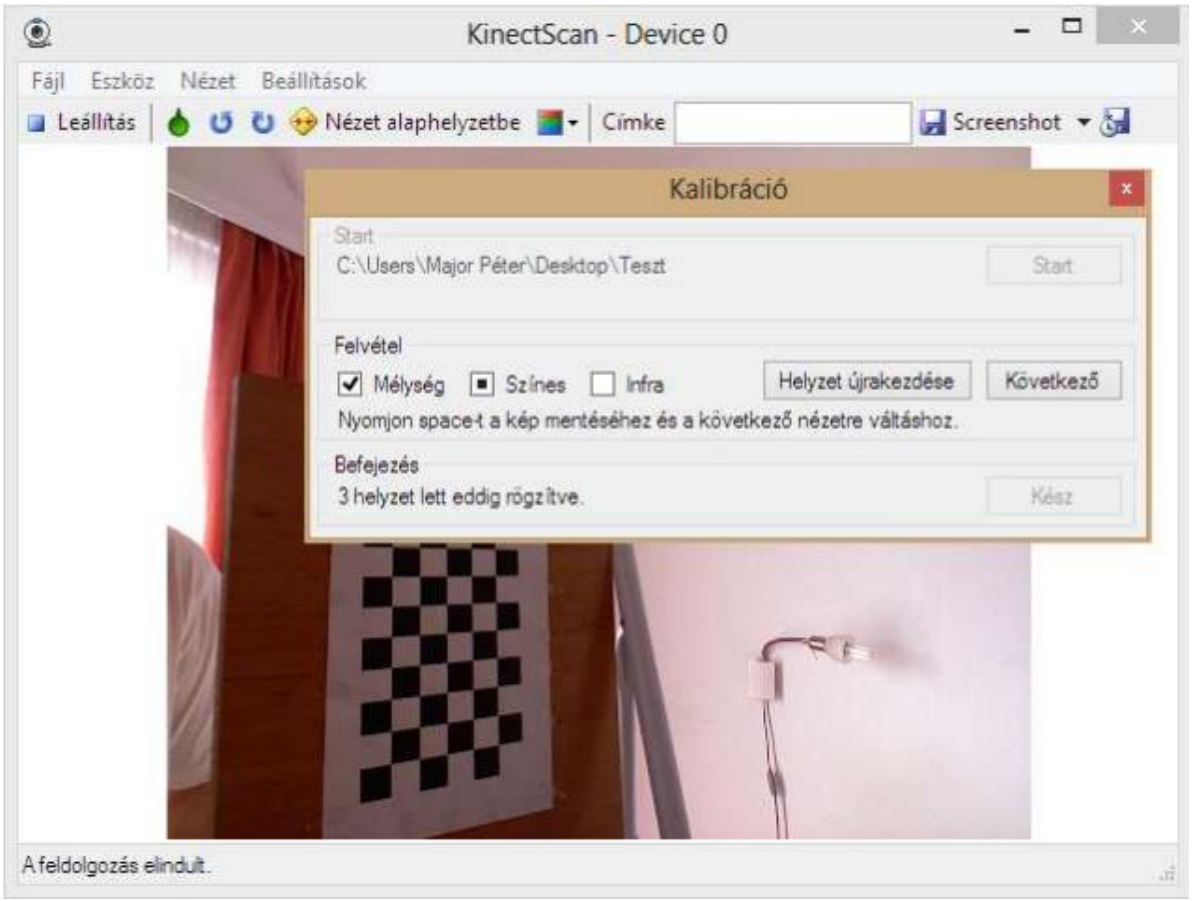


Figure 8. The operator interface of the KinectsScan user interface in the calibration mode

I calibrated the cameras with the aid of MATLAB. The freely accessible Camera Calibration Toolbox [6] developed for the program package implements a number of calibration methods and can be suitably tailored to the user's needs. Since this is an open source code module, it can be adequately modified and enhanced. I prepared my own MATLAB script for calibration which calibrates the Kinect with very little user input. Pictures have to be taken from the chessboard from different positions, for the purpose of calibration. The method I have worked out requires colour, infrared and depth images to be recorded in each position. Owing to the Kinect access these cannot be produced simultaneously because the Kinect's mode of operation needs to be altered during the process, and that is a time-consuming process, it may take up to several seconds, and after these switches it is not possible to take pictures right away because the device needs to set the exposure time as well. This is crucial particularly in taking colour pictures where up to several seconds may pass during the initial setting. I used the 1280x1024 resolution colour and infrared and the 640x480 depth codes for the capturing of the images. Their sizes differ from that applied in the case of taking measurements but

that is not a problem because owing to the proportionate enhancement the coordinates can easily be converted. Owing to the limited band width these modes can only be activated one after another but not simultaneously (indeed, the system's processing unit does not even enable parallel reading of the colour and infrared channels, only the colour and depth combination can be used). The production of the pictures is assisted by a specific aid tool built into the program changing the program's setting to the correct settings for the duration of the calibration process (Figure 8). In taking pictures it the raw data that are saved, no distortion, correction or reflection takes place since the parameters of these operations should be established. In the course of the production of the infrared images problems are caused by the built-in infrared projector which automatically switches on when the infrared camera is started. I found two solutions to this problem:

- The machine is calibrated in the presence of daylight, e.g. next to a window, to allow sunlight to clearly light the board. If the sunlight is strong enough, the pattern projected by the projector will not be visible, so it is enough to simply save the pictures. If sunlight is not strong enough, the infrared projector is to be covered. (9. Figure)
- The infrared projector is converted into a homogeneous light source, for example by folding up a transparent plastic foil and placing it before the sensor. The laminated foil will reduce the strength of the light only marginally but it produces a relatively homogeneous light source. The light strength of the projector is sufficient in this case to deliver enough light to the chessboard alone. I recommend this solution only if there is no infrared light source at hand because lighting will be inhomogeneous and even a modest noise will become visible. (Figure 10)

- An external artificial infrared light source may also be applied

This is the optimum solution but at the same time with a red camera and natural infrared light source this requires separate equipment.

Figure 9. Calibration pattern: a) colour, b) infra-

.

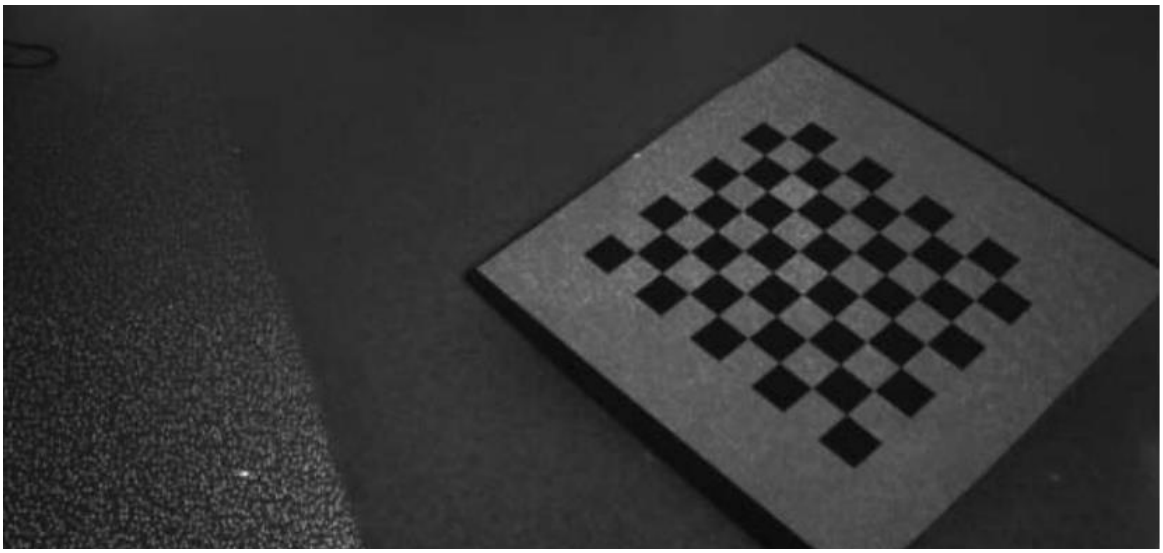


Figure 10. Homogenising of the light of the infrared projector by passing the light through layers of transparent foil, in this picture the projector is only partly covered - the normal intensity is seen on the left hand side, while on the right hand side the result of the application of the foil layers can be seen. Nonetheless, the effect of shading is not perfect since lighter spots can also be seen on the right hand side

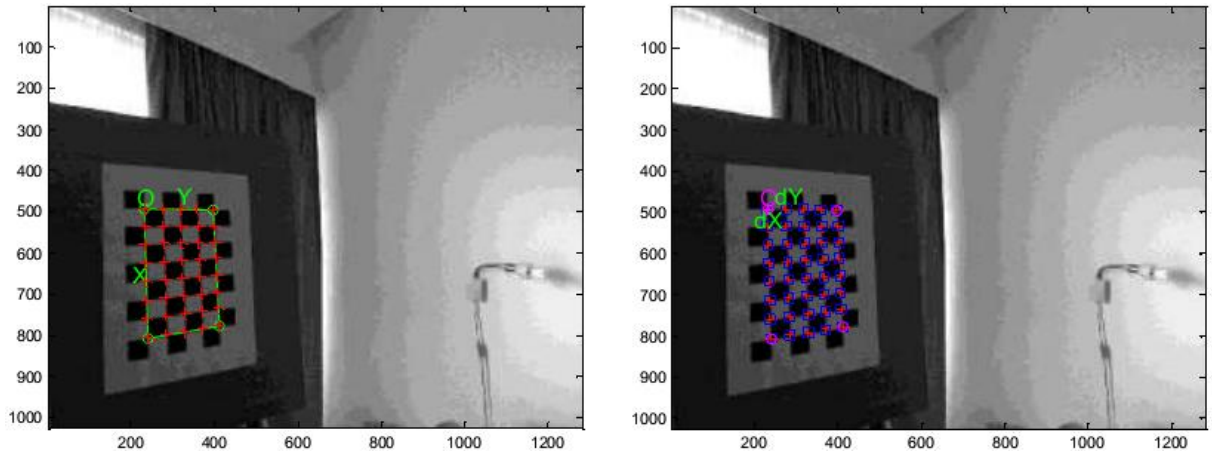
I applied the first two solutions. Since the automated system is prone to apply too long exposure times when reading infrared pictures in high resolution, particular attention must be paid to securing the position of the camera (which can only be achieved by using a suitable support structure so no hand-held sample can be used). Colour pictures are made in a way similar to infrared pictures Care must be taken because the two cameras have two different fields of vision, thus part of the pattern may be beyond the edge of one picture while it is perfectly inside the visible zone. The infrared camera has a smaller field of vision therefore in the course of calibration the system starts recording a situation by taking a depth image because if the pattern fits in that picture then it will do so in the case of the other operational modes as well (and in this way it can be checked whether the depth sensing works well in the given distance).

Making depth images takes somewhat more care. The reason for this is that in contrast to infrared and colour pictures the depth output is rather noisy. Accidental noise may be alleviated in two ways: in time and in space (rather, in a plane). A picture will always have numerous points whose in-depth visibility is uncertain thus not all frames have information on them. To eliminate this anomaly I wrote an averaging algorithm which compares the data of multiple frames for every single pixel and if there are available values, they are aggregated and then averaged. This method ensures that it is enough for a given point to be present in even only one of the frames out of the entire series. Since Kinect marks depth on a non-linear scale, therefore I convert the depth values with the aid of the parameters produced in the course of the calibration of the reference Kinect into real distance and then before saving the data I convert them back using the same formula. Though the ratio of vibrating areas - that are sometimes visible, sometimes not - is significantly reduced by averaging, this does not fully eliminate the problem of noise on the pixels. Therefore before the further processing of the images I carry out a Gauss filtering operation as well. Since the surface of the chessboard is smooth, the Gauss filtering does not distort its shape but it eliminates the horizontal blocks that are typical of the Kinect in the depth image, along with the extremes of the data averaged from only a small number of frames.

The pictures are recorded in the Windows Bitmap format, except for those of the depth images, where the information is stored in the 32-bit floating point format owing to the more accurate values resulting from averaging. The calibration module leads the user along the various steps of calibration and helps him or her in producing images. In selecting the various positions it is worth taking care to produce images in which the pattern can be seen substantially from one side (of course each field of the chessboard must still be identifiable), because more accurate results can be drawn from these. It is also important that information should be gained from the entire surface of the image, i.e. the pattern must be projected to all parts of the picture. It must be made sure that the pattern should be adequately plane so if it has been printed on a sheet of paper, it must be glued onto a more rigid board. When at least 20 positions have been recorded, it is time to start the calibration process in MATLAB

.

3.3.2 Calibration with the aid of MATLAB



a b

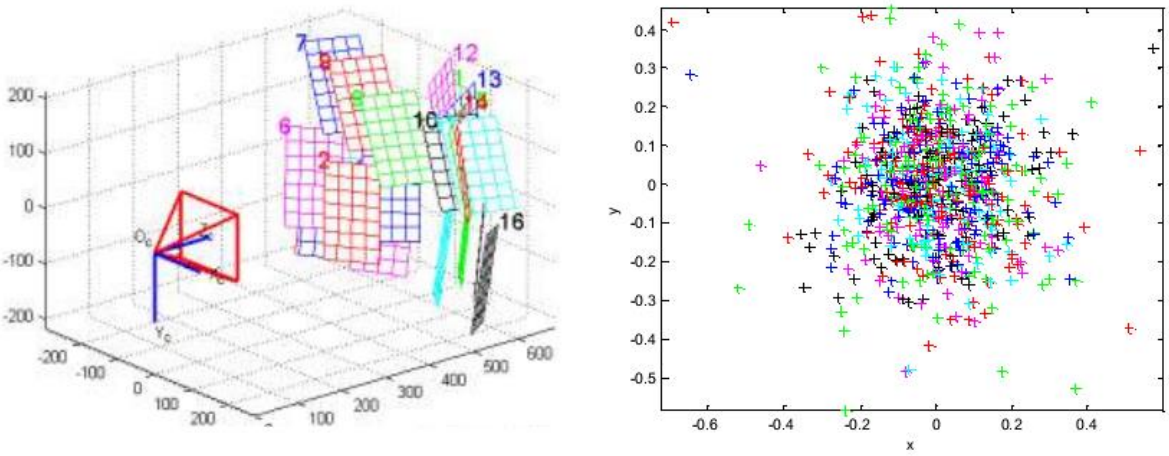
Figure 11. Searching for the nodes of the chessboard: a) corner positions predicted on the basis of the manual designation of the corner points, b) the precise locations of the corner points after the search operation.

The place of the images that have been produced - where they have been put by the calibration module - must be specified first. The colour, the infrared and the depth images are placed here in separate folders. These are read-in and then listed by the MATLAB script that I have written The matching of the images belonging together is ensured by the use of suitable file names. Thereafter the sizes of the grid have to be specified. I used not one but two calibrating chessboards. Initially, I used a 6x9 sized pattern with 25x25 mm fields, thereafter I used a 8x9 sized one, with 50x50 mm fields. In the next step the size of the area in which the system should be seeking for the nodes in comparison to the expected positions is to be specified. This should be set to the size of the squares in the pictures. The reason for this being of importance is that the four corners of the pattern must be marked by the user in the pictures which may be a bit of a problem if it needs to be carried out very accurately. The number of fields needs not be specified because they are automatically calculated by the system. In view of the four corners and the number of fields the system seeks for the corners in the picture, in the course of which the size of the given search window is used again. In the case of cameras with extreme distortion an estimated value is to be given for the radial distortion coefficients for the purpose of finding the corners but in the case of Kinect this was not necessary (the presence of radial distortion is visible to the naked eye but it is still acceptable). The sequence of the designation of the corners ensures clear definition of the position of the pattern for MATLAB therefore care is to be taken that it is always done in the same sequence. In the picture the system indicates the positions of the corners it has found. If in certain pictures the corners are not correctly designated, it can be corrected even automatically (see below). I implemented the search for the corners in a way as will ensure their correct handling even when the black or the white squares merge in the picture (depending on light conditions).

After finding the corners the system calculates the camera's distortion parameters, the focus distances and the position of the optical main axis and it also provides information on the uncertainty of these values. In view of these the points can be projected back to the pictures by the system and the average pixel error can be established, identifying the distance by which the perfect pattern points shifted after they were projected back in comparison to the originally found corner points. Projection back can at the same time be used for improving the designation of the points, since it provides new points of departure for the search operation. For example, if the user's hand slipped in

one picture and failed to accurately designate the corners, then after the first calibration calculations the system can rectify the designation of the points. This repeated projection plays an important role in the case of greater distortions as well, since the points of departure for searching projected back with distortion will be closer to the actual positions. The return projection error can be illustrated in a chart as well (Figure 12/b.) showing the quality of calibration: if there are many points with great return projection error, calibration is not sufficiently precise. If there are only a few extreme points, it usually means that in those cases either the search for the points produced wrong results (e.g. owing to noise or because the user failed to correctly designate the corner points), or the calibrating pattern was wrong. As can be seen in the figure, projection back showed little error in this case. Running the calibration process once again, more accurate, optimised results can be obtained on the basis of the repeated projection and the preliminary results. The colour and the infrared cameras are calibrated in the same way. After the calibration of the various cameras I store the results in different files. They are used in the course of stereo calibration and they can be separately examined after running the Kinect calibrating script

.



a) b)

Figure 12/ a) Identification of the spatial position of the chessboard in the camera's coordinate system (mm) b) Difference between corner point positions projected back in the calibration parameters and the actual positions identified in the picture (pixels)

So far we know the separate parameters of the colour and the infrared cameras but we cannot link them together. Therefore I also had to use the stereo calibration algorithm of the toolbox. A transformation matrix is calculated in the course of stereo calibration with the aid of which we can move from the system of coordinates of the infrared camera into that of the colour camera. In practice, this requires, of course, the depth information as well, since without them the spatial position of the points seen in the picture are defined only to the extent of a half-line.

.

The results obtained from stereo calibration:

Az infravörös kamera intrinzik paraméterei	
Fókusztávolság	[1161.95960, 1169.64938] ± [6.70019, 4.94887]
Főtengelypont	[639.86540, 521.46052] ± [5.49222, 6.91659]
Torzítás	[-0.04421, 0.06934, 0.00181, -0.00114] ± [0.00450, 0.00781, 0.00180, 0.00153]
A színes kamera intrinzik paraméterei	
Fókusztávolság	[1051.45009, 1053.78104] ± [5.82529, 4.30443]
Főtengelypont	[641.15443, 521.79049] ± [3.16070, 3.81499]
Torzítás	[0.14354, -0.24478, 0.00384, -0.00032] ± [0.00595, 0.01021, 0.00122, 0.00080]
Extrinzik paraméterek	
Forgatási vektor	[0.00271, 0.00793, 0.00667]
Eltolási vektor	[-25.95501, -0.03472, 5.28277]

Table 1. The results of camera calibration

The format of the results:

- Focus distance [F_x,F_y]: the focus distances in horizontal and vertical directions in pixels.
- Main axis point t[C_x, C_y]: the position of the main axis point in horizontal and vertical directions in pixels.
- Distortion [K_1,k_2,t_1,t_2]: the first two radial (k) and tangential (t) distortion coefficients according to Brown's model.
- Rotation vector [r_x,r_y,r_z]: the vector describing the rotating component of the transformation from the system of coordinates of the depth camera into that of the colour camera which can be converted into rotation matrix with the Rodrigues formula
- Shifting vector [t_x,t_y,t_z]: the vector of the shifting component of the transformation from the system of coordinates of the depth camera into that of the colour camera.

The uncertainties apply to a 99.7 % probability (3 σ). It is possible to attach a texture to the model generated on the basis of the depth image in view of the results of stereo calibration for which, however, the characteristics of the depth must also be identified first.

.

Before depth calibration I had to examine the depth characteristics of Kinect, for which I used the jacket of a cylinder of a dull surface with its axis parallel with the image plane. By suitably positioning the cylinder I made sure that the constituents of the cylinder are in a vertical position in the picture. I determined the distance of the cylinder measured by the Kinect by averaging the depth values long the nearest of the cylinder's constituents visible in the depth image. This latter ensured a suitable noise suppression. The application of the cylinder jacket offers is also suitable for instance because it is difficult to ensure that a plane board is perfectly parallel with the image plane, while averaging along the line guarantees the elimination of the accidental and potential position-dependent permanent error. (Of

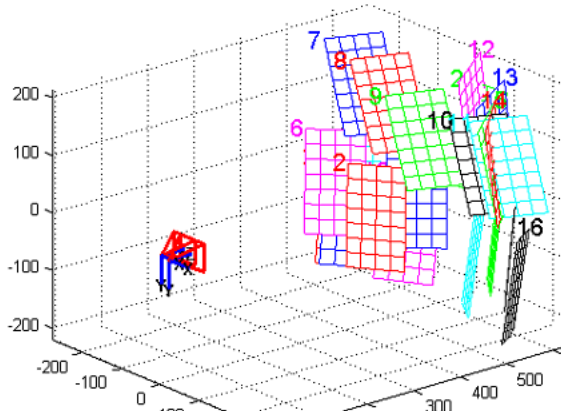


Figure 13. The result of stereo calibration (distances in mm)

course, averaging in time was also carried out.) Meanwhile, I also measured the actual distance of the cylinder. Thus I recorded the following points
:

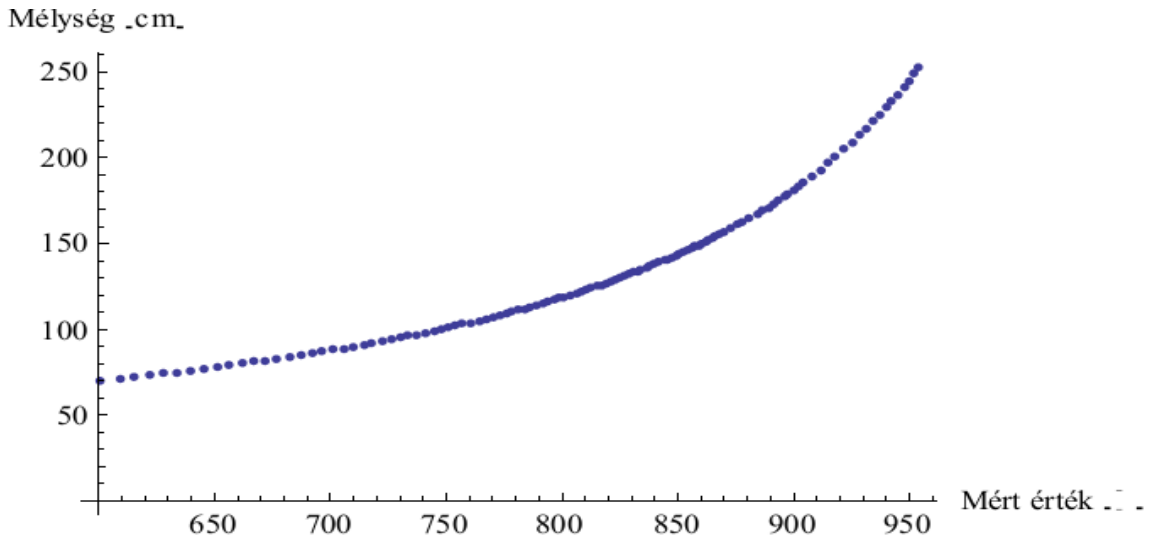


Figure 14. Values obtained in the course of depth measurement by the Kinect

It is clear that the relationship is strongly non-liner, thus the question arose as to what parameters it should be described with. I contemplated the following solutions (I even gave a formula received by fitting after the various different solutions - the function fitting was carried out with the Wolfram Mathematica software):

- The relationship can be simply described with the aid of a polynom and the coefficients can also be easily calculated with the aid of the least squares method. The disadvantage is that it only provides reliable result for a distance from which data were available. It is, however, not suitable for extrapolation which is a bit of a problem because of the difficulties of providing for large distance differences in the course of calibration (there is not enough room, different sized patterns at different distances etc.).

$$p(x) = -1879.44 + 8.19864x - 1.16564 \cdot 10^{-2}x^2 + 5.66001 \cdot 10^{-6}x^3 \qquad (6)$$

ver function but the fitting experiments show that fitting cannot be properly carried out in the case of larger distance differences. In the second case I tried to use an exponential curve which provides good results in the upward section but it does not do so on the initial flat section. At this point I chose the sum of a constant, a primary and an exponential member, with which the set of points can be very well approximated. Nonetheless, such a characteristic seemed to be rather unlikely, in view of operation.

$$e(x) = 5.40092 + 0.0106353\, e^{\frac{x}{100}} + 0.101291\, x \qquad (7)$$

- Eventually, the sum of a shifted hyperbolic member and a constant became the final and applied model. The reason for this choice is that it can be perfectly fit to the points and by checking with measurements I found that it remains accurate even in the case of extrapolation. Non-linear function fitting cannot be carried out with the aid of the least squares method, and initial values also had to be specified.

$$h(x) = -2.33458 - \frac{35498.8}{x - 1093.09} \qquad (8)$$

In the course of calibration it had to be answered where the singularity of the hyperbolic member will be in the case of the writing of a 11-bit depth which may take up values in the 0..2047 range. It became clear in the course of the trials that the values rarely exceed 950 because Kinect does not see that far, and where it does not see it assigns the value of 2047. In the course of the initial trials the singularity fell in the 1000..1100 range, from which it can be concluded that it should be at 1023, and the 11th bit is a flag showing whether the system sees anything in that point. The results obtained in the course of the fittings, illustrated in a chart:

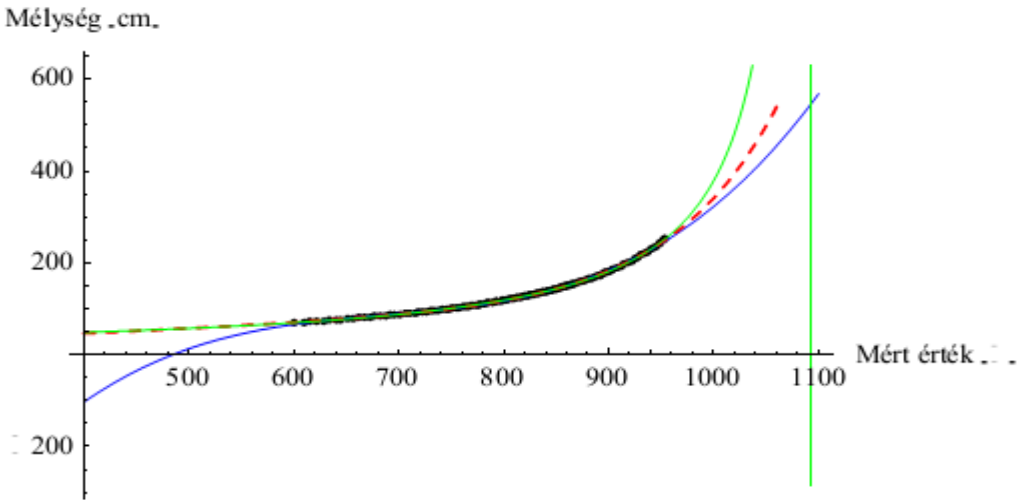


Figure 15. The depth functions scrutinised: blue) polynom function, red) exponential and linear member, green) hyperbolic function

As is clear from the chart each of the three functions give a good approximation in the examined range but for practical considerations (simple and quick to calculate) I chose the hyperbolic function. Kinect users apply several different solutions according to on-line sources, from among which the most widely applied one is the hyperbolic solution. The drawback of manual calibration is that it does not measure distance from the same point of reference as is applied by stereo calibration. To eliminate this problem I supplemented my MATLAB program calibrating the Kinect by producing a depth function.

Since the Camera Calibration Toolbox is not capable of managing depth cameras, therefore I carried out the encoding of the calculations to be carried out here. To this end, I relied on the fact that in the case of every picture to be calibrated the system specifies the lengths of the rotation and shifting vectors required for the conversion between the camera and the system of coordinates fixed to the pattern. In view of the other calibration parameters this makes it possible to calculate where the various corner points will fall in the depth image. First the algorithm takes up the positions of the various corner points in the pattern's coordinate system and transforms these into the camera's coordinate system. Thereafter we carry out the distortions and the projection as well, producing the image coordinates to be used in the infrared image. Thereafter one should only read out the values taken up in the relevant positions of the depth image. This however, would not be accurate in this way, since in the course of the analysis of the system it was found that the depth and infrared pictures cannot be matched together by simple resizing calibration. The reason for this is that the Kinect cuts off the edges of the infrared image before generating the depth output. The studies also found that the widths of the parts cut off on the different sides are also different, and a slight magnification (not keeping the original proportions) also takes place. Since Microsoft does not publish the precise implementation in detail, the actual cause of this is not known. In the existing calibration method these shifting and magnification parameters are established manually by aligning

together the depth and infrared images taken from the same position. (In the case of increased accuracy we used for this a spatial chessboard as well, where the relevant fields stand out. The ample detail of this spatial chessboard and its pictures taken from multiple angles help the accurate establishment of the relationship between the infrared and the depth images.) Thereafter the system - taking this adjustment into account - queries the depth values belonging to the corner points. Since the calculated positions will not be whole in the picture, the querying takes place with the aid of bilinear interpolation. The system collects the resulting value-pairs in two vectors. The actual values and the values measured by the Kinect create the following shape if presented in a chart:

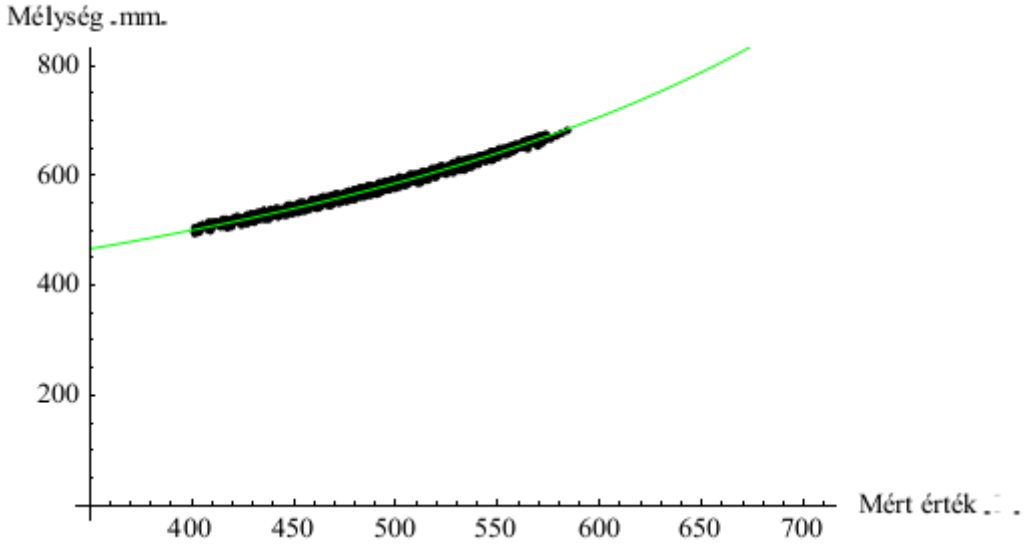
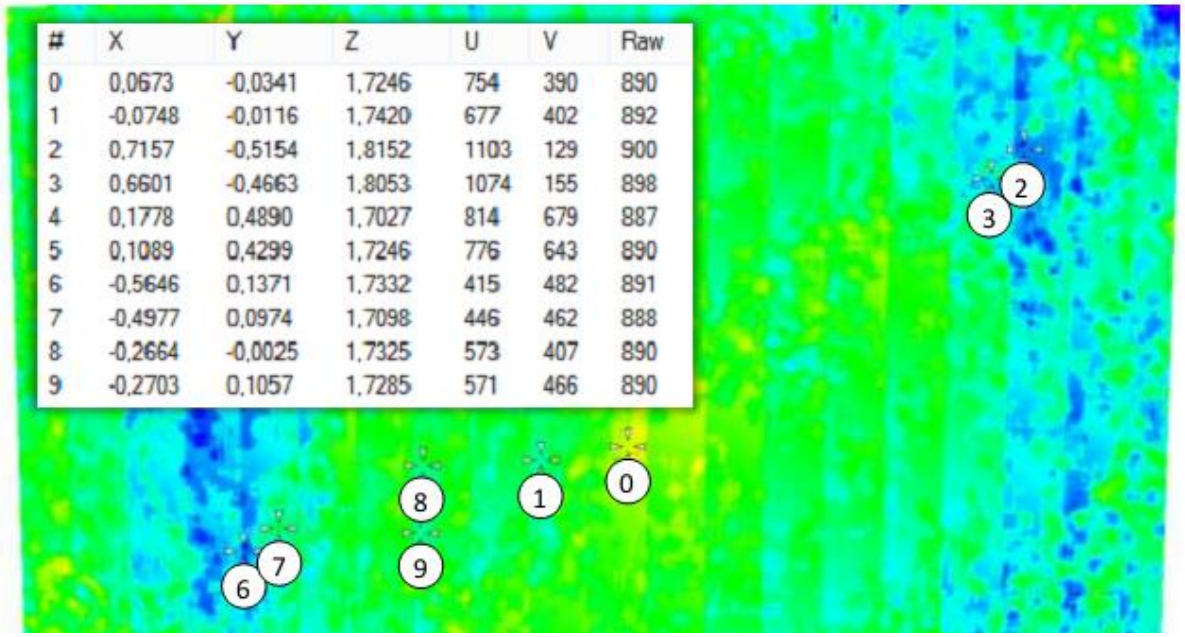


Figure 16. Depth function when MATLAB is used

In the course of calibration with MATLAB I found that the three dimensional identification of the position of the pattern is most accurate if it is at a wide angle with the image plane. If the pattern is nearly parallel, then when the points obtained on the basis of the given image are staked out, in extreme cases the curve falls apart into horizontal curve sections positioned one over another. As is indicated by the chart, the points are dispersed in a broader patch. My analyses showed that this is caused by the Kinect's autofocus system and in the case of larger distances even minor breaks appear in the curve as a consequence. Thereafter the MATLAB program stores the results in an XML file, which can even be managed by the processing program.

3.4 Analysis of the measurement results

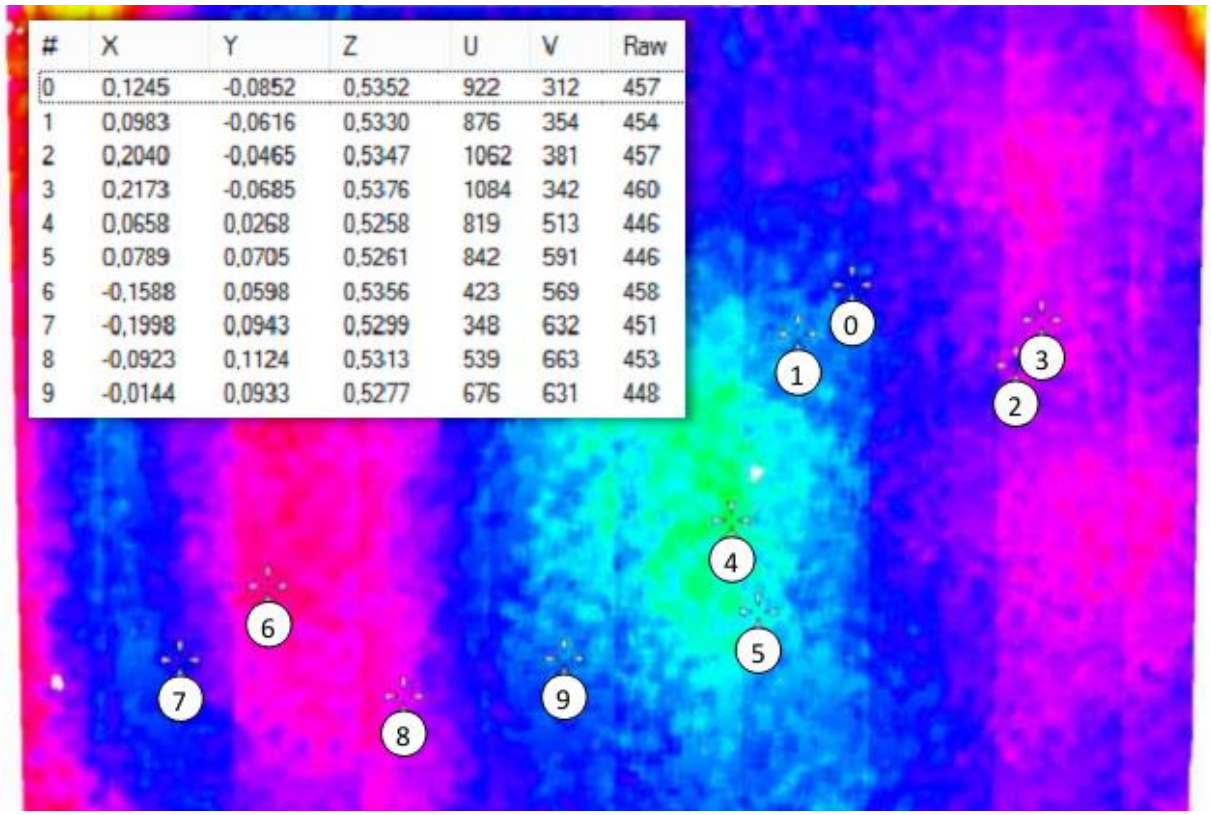
The subject of the analysis of the measurement results is to check the depth measurement, since this is the direction in which the models that have been created are the least precise. Perhaps the simplest way to do so is taking pictures of a smooth surface with the Kinect from different distances and then analysing those images. As a consequence of the Kinect's characteristics we expect precision to dramatically deteriorate with the increase of distance, along the upward section of the hyperbolic curve. This phenomenon has been observed by many in the course of calibration [15, 17]. I developed a 'probes' function for the measurements, with the help of which it is possible to query the positions of the points of the model generated in real time with probes 'stuck' to the surface of the model.



#	X	Y	Z	U	V	Raw
0	0,0673	-0,0341	1,7246	754	390	890
1	-0,0748	-0,0116	1,7420	677	402	892
2	0,7157	-0,5154	1,8152	1103	129	900
3	0,6601	-0,4663	1,8053	1074	155	898
4	0,1778	0,4890	1,7027	814	679	887
5	0,1089	0,4299	1,7246	776	643	890
6	-0,5646	0,1371	1,7332	415	482	891
7	-0,4977	0,0974	1,7098	446	462	888
8	-0,2664	-0,0025	1,7325	573	407	890
9	-0,2703	0,1057	1,7285	571	466	890

Figure 17. The precision of the Kinect in the case of large distances tested with the aid of a wall of smooth surface, the columns of the table: X, Y, Z: the position measured in the camera coordinate system (m), U, V: the position selected in the picture (pixel), Raw: the value produced by Kinect.

Clearly, the measurement accuracy deteriorates significantly as the distance grows: while at a 40 - 50cm distance even a 0.5mm accuracy can be achieved, at a distance of 2 metres up to 3-4 cm surface errors appear. It is also observed in the pictures that the Kinect's depth characteristics are not fully identical on the entire surface of the picture: rather, it slightly changes subject to the distance of the optical axis. The effect of this in the case of measurements in short distances, such as those presented in the applications, is less substantial (if adequate filtering procedures are applied) but the management of these could be included among the options for the further development of the software. A lot more disturbing than this is the presence of the abovementioned vertical streaks, also causing more problems at larger distances. Since the position and width of those streaks vary from time to time, they could be removed only with some special algorithm. In the case of smaller distances the error is also a lot smaller, in this case unevennesses will not be larger than a few millimetres in size. It has been experienced that more precise measurements can be carried out primarily in the 40-60 cm range, beyond which the error grows gradually and in the maximum possible distance of 6-7 metres the sizes of surface errors may be as large as up to 10 cm. In this range the system can only be used for gaining approximate information. There is a so-called macro-mode in Kinect's variant produced for use with PCs with which it is possible to produce more precise images from even closer. If the Kinect is placed closer to the object of measurement, the field of vision will include a gradually decreasing part of the object, limiting the possibilities for the application of this method. I reduced surface unevennesses by averaging over time and Gauss filtering.



#	X	Y	Z	U	V	Raw
0	0,1245	-0,0852	0,5352	922	312	457
1	0,0983	-0,0616	0,5330	876	354	454
2	0,2040	-0,0465	0,5347	1062	381	457
3	0,2173	-0,0685	0,5376	1084	342	460
4	0,0658	0,0268	0,5258	819	513	446
5	0,0789	0,0705	0,5261	842	591	446
6	-0,1588	0,0598	0,5356	423	569	458
7	-0,1998	0,0943	0,5299	348	632	451
8	-0,0923	0,1124	0,5313	539	663	453
9	-0,0144	0,0933	0,5277	676	631	448

Figure 18. The precision of the Kinect in the case of small distances, tested with a smooth surface

4 The steps of image processing

This chapter will describe the operation of the Kinect and the basic processing of the data measured by it. The series of operations described here leads to the production of the surface model of the observed environment. These operations are executed in real time, enabled by hardware acceleration, with the aid of which the system could manage several times the frame speeds enabled by Kinect. Since the graphic card is necessary for effective three dimensional presentation, the implementation of hardware acceleration does not entail additional resource requirements.

4.1 Controlling of image processing

4.1.1 Driver selection

Despite the fact that Kinect was initially designed as a game controller, the developers found other potentials in it too. Since initially Microsoft did not release drivers for the system, users set about creating those themselves. It was owing to the initial lack of support that at least three different drivers and associated programming interfaces are available at present to go with the devices, each with rather different strengths. I tested each of the three drivers in the course of the development process and I drew the following conclusions:

- OpenNI driver [7]: this driver was produced by a consortium and by PrimeSense manufacturing Kinect itself. This driver is part of middleware, which is capable of operating Kinect together with other devices for identifying gestures. its drawback lies in its high ‘machine-intensity’ while using the Kiect, and that the initialisation of the hardware elements is complicated and not flexible enough in its framework system. Its advantages

include that with this system the Kinect can be accessed by multiple programs simultaneously and the settings are also relatively easily accessed.

- Microsoft driver [8]: since Microsoft also identified the opportunities offered by Kinect in terms of development for purposes other than games (and the market for such developments), they also released a n official driver. The advantages of this driver include, for example, the ease of distribution (automated installation from the driver database of Windows Update), more reliable support and more stable operation. Its drawbacks include for example the increased machine capacity requirement entailed by the use of Kinect (probably owing to gesture recognition and the continuous running of image adjustments) and that it does not enable access to many of the functions of Kinect, from among which the most important for me is that the image of the infrared camera is not accessible with it whereas it is necessary for my calibration method.
- OpenKinect driver [9]: this driver was developed by the free developers' community on the basis of tapping and analysing the communication between Kinect and the XBOX. This driver is a multiplatform one and is accessible for a variety of operating systems. Its advantages include its very little resource use and that it provides access to many of the hardware elements (infrared camera, acceleration sensor, status indicator LED, further output modes etc.), that are not accessible with other drivers. This was the first driver with which Kinect's 640-480 resolution depth output could also be accessed which had not been enabled by the other solutions at the beginning of the development process.

After the completion of tests the choice fell on the OpenKinect driver, since that was the one best suited to the requirements. Using the driver required downloading its source code, and then after acquiring the required further modules (e.g. libusb, phthreads) it had to be compiled. Installation can be easily carried out with the aid of the tool manager after the selection of the position of the driver. When Kinect is connected, Windows automatically downloads and installs Microsoft's driver, thus for the time it takes the Internet connection needs to be broken (or the Windows Update's automated driver download function is to be inhibited). When the installation of the driver is completed, it is no longer manipulated by Windows.

In this section I sum up some comments concerning the hardware to help users of the program submitted together with the paper. It is possible to work with several Kinects in parallel on a given machine. The possibilities are limited, however, by the fact that every Kinect needs to be connected to a separate USB root hub in the computer. Without this latter only one Kinect can be used from among those present on the given root hub and the band width of the connection may also be too little (discarded frames). It must be noted here that in normal mode the Kinect uses the entire effective USB 2.0 band width, thus it is not possible to connect any other tool requiring any considerable band width to the USB controller that is managing the Kinect without the loss of frames. The USB controllers of certain computers (e.g. many notebooks) cannot provide the band width required for Kinect, and the images will get stuck (another reason for this may be that other internal peripheries are also connected to the USB controller concerned, which are not visible from the outside, such as web cameras, network adapters). One of the weaknesses of the Kinect communication protocol is that it tries to post every single frame, even if the band with is not sufficient. Consequently, if the band width is little, the end of the frames will almost always get lost and not even one image is obtained in its complete form. Depth images enjoy preference over colour pictures and thus if the band width is not sufficient for both, the user receives only depth images for the most part (with full speed) with very few colour ones (e.g. in the case of one notebook I tested I received one every 4-6 seconds, randomly), or none at all. Accordingly, the Kinect cannot be used with very narrow band widths (not even a low frame speed can be obtained) because the new frames break the process of the transmission of the previous ones. The Kinect appears as three different hardware elements for the computer which need to be installed separately:

Kinect NUI motor (this is the control unit, and the others are not visible until it has been installed), the Kinect NUI camera and the Kinect NUI Audio. Only one type of Kinect driver can be installed on a given Windows installation and for proper functioning the old one needs to be deleted before the installation of the new driver.

4.1.2 The management of the Kinect

Although the Kinect has several different drivers with different functionalities, the schemes of the controlling of the sensor are practically identical. The first step is querying the number of the Kinects that have been connected, to find out the number of Kinects that can be used. The Kinects are identified on the basis of their serial numbers. Before putting to use a Kinect needs to be initiated, which can only be carried out successfully if it is not being used. This operation sets the Kinect in its base position. After use the connection with the Kinect is to be broken by calling the relevant function, otherwise it remains in an undefined state (in this case the cameras and processing do not stop on the hardware side, resulting in wasting energy). The Kinect's head can be tilted forward and backward with a motor, but this motor was not designed for continuous use, thus the number of settings should be limited. In the course of initialising the Kinect tries to take up a horizontal position (which is assisted by the built-in acceleration meter) during which it must be fixed in a stable position. If the connection has been successfully established, the various hardware units become accessible. The cameras and the central data processing unit are controlled by the user's selection of the required mode of operation on two channels, the depth and the video channels, which do not, in fact, directly correspond to the hardware elements, but instead have the following functions:

- The images of both the colour and the infrared camera can be accessed through the video channel. These can be used in two different levels of resolution: 1280x1024@7Hz and 640x480@30Hz. The frame speed belonging to the higher resolution can only be achieved if only this channel is active.
- The depth channel provides only depth information, in the following modes: 640x480@30Hz or 320x240@30hz.

It is also possible to use the two channels simultaneously, but owing to band width limitations images can only be obtained at the 640x480 resolution level (in this case t 30 FPS). IF the infrared camera is on, the infrared projector also starts up automatically, it cannot be separately switched on. After specifying the modes the data flows can be started but this may take up to several seconds, thus no quick switches are possible (e.g. for calibration). The data flows are automatically received and collected in a buffer by the driver. To receive the frames that have been delivered the delivered frames need to be requested in a cycle (polling). To minimise the required processor use I carry out this process in a separate thread which waits 5 seconds after each query to make it possible that every frame is received with a little delay but processor use is kept minimised. It follows from the mode of the operation of the system that upon a given call the driver delivers multiple frames up to the capacity of the internal buffer. The driver transmits the frames through a call-back on a separate thread (always on the same one). The colour and the depth images are not received in a synchronised way and completely different frame speeds can also easily occur (e.g. frames discarded in the case of the colour camera). Therefore the processing system I have written collects the frames in double buffers, separating the colour pictures from the depth images. The dual buffer that I have created is capable of storing more than two frames in the case of the depth channel, which may be used for averaging in the case of static or quasi static measurements. Another advantage of averaging is that the number of temporary hiatuses appearing in the Kinect depth image (on which there is no depth information) is significantly reduced by averaging. Averaging is not necessary in the

case of the colour camera because we do not take measurements with that one, and the noise in its output is also marginal. I read the contents of the buffers through a thread running asynchronously with the cameras:

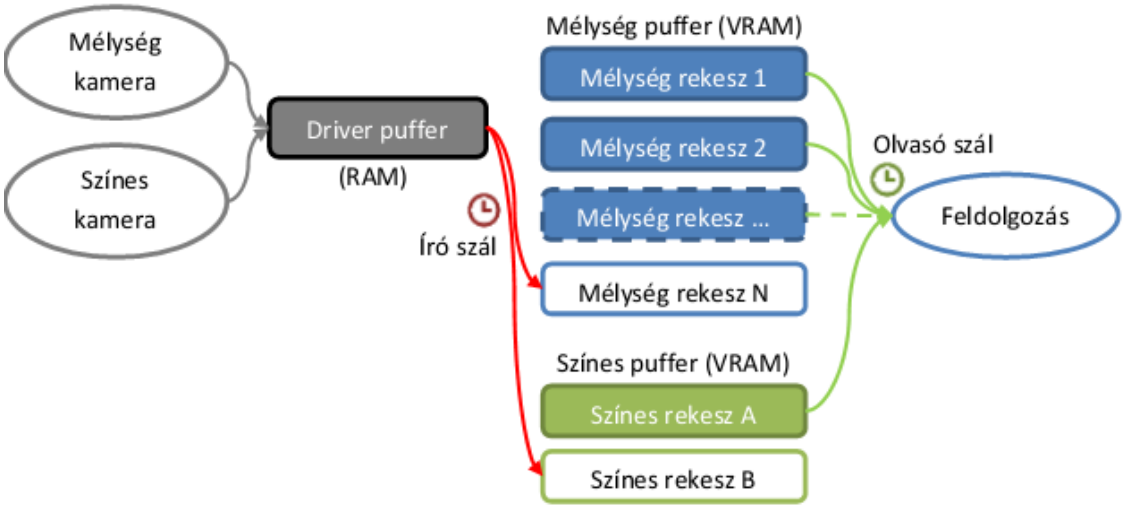


Figure 19. Management of the recorded frames

This solution enables parallel operation of multiple processing threads, but the additional performance so achieved is not significant because much of the processing takes place with the aid of a graphic accelerator which is thus already highly parallel in regard to data (SIMD). Since the writing/reading of the buffers takes place in parallel, the buffers being used need to be monitored on a continuous basis. The software keeps records of both the racks being read and those being written, selecting the targets of writing and reading operations accordingly. This is an important issue primarily when more than two frames need to be stored. In such cases writing takes place in a revolving manner, while in the course of reading I read all of the elements for averaging, except where the given element is just being written. The result of this latter is that the number of frames being averaged is fewer than the designated number by one frame. The actual processing of the images is started after the reading of the buffers.

4.2 Processing of the images - hardware image processing

The images are processed on the graphic accelerator, creating textured three dimensional models in my program, in real time, that can be displayed with various shading techniques from the raw colour or debt data received from the Kinect. To describe processing on the graphic card I will first give a brief overview of its architecture from the aspect of programming. Programming tools have been developed by now that make it possible to utilise the computing capacity of the graphic card for resolving non-graphic problems without having to convert the problems into graphic ones. These however, are not necessary in this case, since all of the tasks to be carried out are graphic tasks or similar to those. In the course of programming I used the XNA framework system giving access to the DirectX environment to the .Net system, which makes all functions - offered by DirectX - accessible from the aspect of programming, but in a clarified, well structured and object oriented form.

Before the description of the resolving of the various graphic tasks I wish to outline the solving of a standard drawing task in the DirectX system. Drawing requires a geometric model, which consists, in the base case, of a block of peaks and the relevant index block. Every peak is described by a structure containing, as a minimum, their position in space, but in most cases these are accompanied by texture coordinates, normal vectors and colour information as well. In addition to using the structures prepared in advance it is also possible for the user to create his own structures, thus it is possible to transmit any information that can be described in terms of numbers. The index block specifies the positions of the peaks in the series that belong to the various primitives to be traced out. The use of the index blocks is highly advantageous since in the case of most models a given peak is part of multiple primitives and since the peaks need to undergo transformations before drawing, it is enough to carry out those only once. Before it becomes possible to apply the blocks they have to be uploaded into the memory of the graphic accelerator, with the aid of index and vertex buffers. These buffers have to be created in advance and their fixed sizes must also be specified. It is an advantage however, that these buffers can be updated with new content at any time, as long as it fits in the buffer concerned. In addition to model information there is a need for texture information as well, for which an adequate size of memory capacity needs to be reserved - also in advance - in the graphic card. In the course of reserving memory both the size and the type of the texture need to be specified. Different types of different bit depths and channel numbers are available for the storage of the images, while at the same time in up-to-date cards even the floating-point description technique can be used, entailing a variety of new usage possibilities, since in the case of the whole types only the 16-bit channel with can be achieved (at best). The textures need to be assigned to samplers before use, which can read from them on the basis of the prescribed settings. Such settings include for instance the interpolation to be used (point, bilinear, bi-cubic) or the management of parts hanging off the texture (repetition, limitation, mirroring). Similarly to the index and verter buffers, the contents of the textures can also be updated. In the course of updating it is possible to update any given part separately. One advantage of the buffers is that the data required for the various frames can be accessed directly from the graphic memory, they need not be repeatedly transmitted for every drawing. The calculations have additional input parameters which can be used by the programs running on the graphic card. One important feature of these is that they cannot be changed in the course of drawing - their contents can be updated only between two drawing jobs. Such ‘variables’ define the transformation matrices, the vectors used in the course of the calculations, the logical settings etc. In addition to these the values of numerous registers also play an important role in influencing the mode of the operation of the card. These include for instance the settings of colour mixing, the hiding of triangles with their backs to the camera (triangle culling) and the specifying of the textures and buffers to be used. Some similar parameters can only be set in the course of the initialising of the card, but it cannot take place later on. These include, for instance, the size of the screen buffer, its colour depth, the type of the depth buffer, the sampling quality etc. Further important elements include the rendering target points which are memory areas not directly displayed on the screen on which it is possible to draw and can be used later on as texture.

.

The following is a description of the drawing process. The first step is the processing of the vertexes with the aid of the so-called vertex shaders. A vertex shader is a function both of whose input and output parameter are vertex structures (the type of the input may differ from that of the output). Vertex shaders carry out the task of mapping the peak points into the screen coordinate system. On modern cards vertex shaders access textures as well, playing a key role in many parts of the programme. Thereafter the primitives to be traced out are generated from the contents of the processed vertexes and the index buffer. These data then are transmitted to a rasterising unit which identifies the pixels covered by the given primitive and interpolates the information stored by the vertexes out to each pixel. Thereafter the vertex information that vary pixel by pixel, is received by the pixel shaders. The pixel shaders determine

.

Figure 20. The drawing process in the DirectX environment

the colour of the pixel to be traced for which they may use the textures as well, in addition to the interpolated vertex information. Thereafter the various pixels are transferred to the colour mixer which determines the colour of the pixel in the output, in view of the new colour to be drawn and the colour that used to be present in the pixel previously. In most cases if a depth buffer is also used, only those pixels are drawn that are closer to the camera. Since the Direct3D does not directly support two-dimensional drawing, therefore if only the pixel shaders are required then I draw an oblong composed of two triangles. The pixel and vertex shaders have to be arranged in so-called techniques. Every technique may comprise several rendering processes and every process comprises one vertex and one pixel shader. I did not use rendering of multiple rounds in the course of the project, as there was no need for it. Since one vertex or pixel shader can be used in multiple techniques, it is easy to implement various display modes for the same model.

Finally, I wish to describe some restrictions and limitations that needed to be taken into account in the course of the development of the program. The HLSL supports the use of functions but they are always inlined, thus they do not cause a drop in performance. Graphic cards today usually do not support ‘dynamic bracing’ because it would entail excessive overhead in the card’s scheduler. This means that the card always executes both branches but it utilises the result of only one. This is necessary because a modern card has 1500-3000 shading cores running the same program with strong paralleling and without this the scheduling of the calculations and their equal cycle numbers would not be provided for. Therefore so-called unroll needs to be carried out on the cycles, meaning that the machine code will contain repetition of the same commands instead of real cycle with jumps. Of course to make this happen there is a need to know how many times a cycle will be run, thus it is not possible to create cycles that can be run any number of times either. The length of the shading programs is also limited and the maximum number of the various operations (arithmetic, texture reading) is

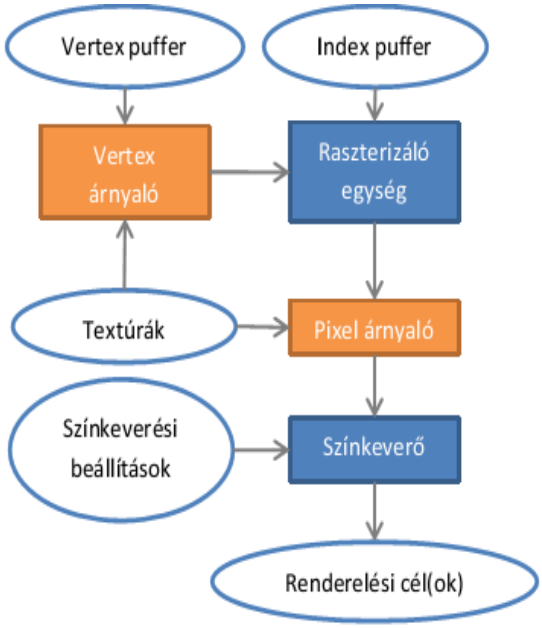


Figure 20. The drawing process in DirectX environment

specified in terms of units. Every command is worth a given number of units and their sum cannot be higher than the maximum determined by the version of the given shader. Despite the fact that DirectX was developed to enable the hardware products of different manufacturers to be managed in a standardised way, my practical experience shows that the operability of the shaders must always be checked on the different hardware elements (the nVidia and AMD cards I use showed extremely

large differences in certain cases in the interpretation of the various shaders, but it is usually possible to find implementations that work well on the cards of both companies). Moreover, since hardware products implement only a subset of the DirectX tool kit, the rest of the functions are CPU emulated which may result in massive deceleration. In the case of part of the features used in the program, software emulation is not available therefore it cannot be run on some older computers.

4.3 The procedure of the processing of images

The following is a description of how textured three dimensional models can be generated from the images produced by depth and colour cameras. It should be noted that the whole process takes place on the graphic accelerator, so it entails negligible CPU use.

4.3.1 Accessing the measurement results

The programming interface delivers the data received from the Kinect through asynchronous recalls, on which even some processing takes place, so that the data are converted into a form that is compatible with the graphic card. Accordingly, for instance the depth data's 11-bit pixels have to be transformed into 16-bit items and the colour images have to be turned from Bayer coding into simple RGB pictures. These operations are already taken care of by the driver, if the output formats are adequately specified. The data that have been received are uploaded to the graphic card via the recall threads created by the driver, to make sure that this does not delay the rest of the processing. The type of the target texture is of outstanding importance in uploading since this has a profound impact on the form in which the data will be available. In the case of the textures the data are always accessed in the floating point form, and regardless of the type of the texture each sampler returns a vector of four elements. If the texture contains fewer than four colour channels, the value of the unused channels will be zero. If the texture itself already had floating point description, there is no need for conversion. If storage is of the integer type, then the data are transcribed into the 0..1 domain. For uploading colour images they have to be enhanced to 32-bit colour depth, as the hardware supports only pictures with 1, 2 and 4 channels, but it does not support the 24-bit 3-channel version that is available as source. In the course of conversion a zero value is assigned the fourth channel - interpreted as transparency value - so the picture will be intransparent. In the case of the depth buffer there is a need for a 16-bit colour depth which could be best described with the BGRA4444 format from among the available formats, which has four 4-bit colour channels. Since the stored 16-bit value is broken down into four 4-bit ones, thus the original value has to be restored before use.

Since the colour and the depth frames come asynchronously, I decided that processing and the management of the incoming frames should be carried out asynchronously. (Of course if in a given case the user wishes to link processing to the receipt of depth or colour frames, it can also be easily carried out: in this case it is enough to call the processing function at the end of the recall). To carry out the task I created my own dual buffering system to enable parallel reading and writing of the data. If buffering is switched on, averaging also takes place later on in the course of processing. In this case the system uses a buffer with a number of sections corresponding to the number of frames to be averaged, where writing takes place in a revolving way. In the base case the recalls perform no other tasks apart from this. The program makes it possible to save raw data coming from the Kinect, which is also carried out in the same place. Serial pictures can also be taken but in this case the raw data are put into a separate buffer from which IO threads parallel with the processing threads write them onto the disk. Since writing on the disk is a slow process, therefore serial pictures would significantly increase the time spent waiting on the recall threads, which why there is a need for parallel writing on disk. The saving threads read the data to be saved from one line and when the line is depleted, I send them into a waiting status from which they are awakened by the main program when new frames are in to be written.

The raw saved data can - even without the presence of a real Kinect - be loaded into the program later on. This function was implemented only in a later stage of the development process, therefore it was important t make sure that only minor changes be necessary in the other parts of the program. Therefore I created an abstract Scanner class which can be used as a common interface in the case of any number of depth sensors (not only Kinect). The class mapping the original Kinect was then derived from this, and I also created a file scanner class, which can substitute the Kinect with the aid of raw depth and colour data. Since the program reaches the sensor though the shared interface, there is no difference for these between a real and a virtual tool, which is ensured by the polymorphism of the derived classes. Of course certain parts of the program continue to use the derived classes for specific functions (loading files in, head tilting etc.). Another benefit of this architecture is that it is possible later on, without any major change to the program, to apply scanners operating on the basis of different principles, provided the colour and the depth images can be generated. In using the file scanner it is possible to skip between elements of a given directory, improving the efficiency of its use.

4.3.2 Averaging

The scheduling of processing is carried out with the aid of a timer. Upon every single beat of the timer the processing process is run once. Since processing takes place on the graphic card, and thus it takes place very rapidly, there is no need for multi-thread processing (the graphic card performs massive paralleling in the first place). The first step is reading the next frame from the buffer storing the depth image frames. This may take place in one of two ways, depending on whether the user has switched on the averaging of the depth frames. If there is no averaging, simply the frame in the first position at the given point in time in the dual buffer is used. Otherwise all of the frames are read out from the frame buffer, except of course for the element just being written. The frames have to be aggregated first if they are to be averaged. The most obvious method for adding up the frames would be by setting the colour mixer to have it sum up the source and the target colour. Although it is truly possible to select summation for a colour mixing function but that cannot be used in the case of a floating point output. There is a need for floating point rendering goal because it is not possible to sum up any number of images in the other formats without major loss of data owing to the limited set of values. Since not only the sums but the number of the frames summed up must also be recorded (by pixel, owing to the sensing gaps), thus this requires two-dimensional vectors. To rendering goals are required for the addition, these should be marked *A* and *B*, and there is a need for an image to be added, which should be marked *S0*, *S1*..*SN*. The addition cycle takes place as follows: B = 0, A = S1 + B,B = S2 + A, A = S3 + B and so on.

Since the Kinect has strongly non-linear characteristics, before averaging the values it provided need to be converted into real depth values, which is carried out on the basis of the following formula
:

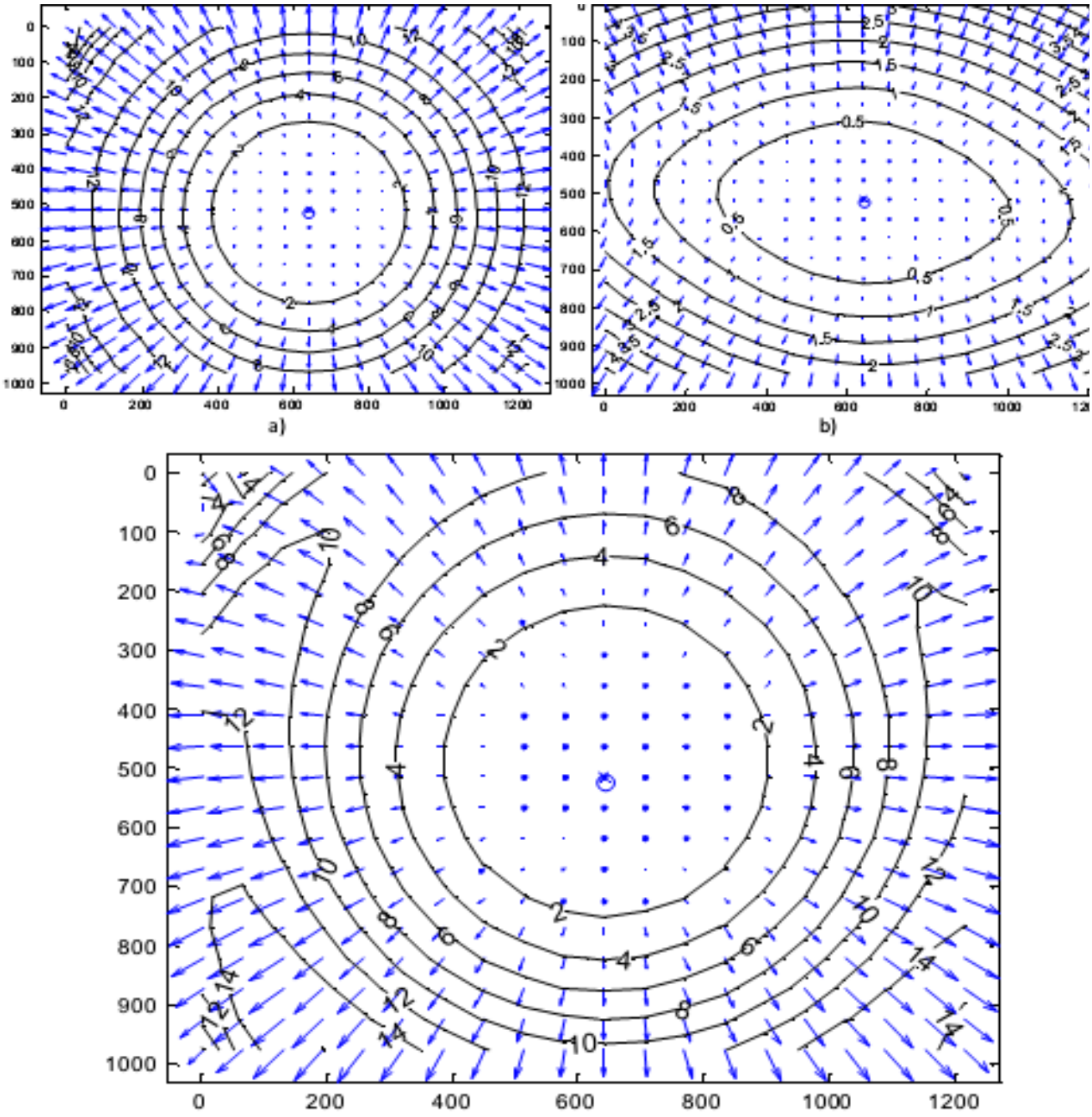
$$d(x) = \mathrm{a} - \frac{\mathrm{b}}{x - \mathrm{c}}$$

Where:

- *a,b* and *c* are parameters characterising the hardware concerned, specified in the course of calibration
- *d* depth in real units (m)
- *x* depth in Kinect units

In the course of summing up the values the system discards the invisible areas belonging to values 2047 and 0, and the user can also select a maximum distance for the field of vision.
.

After the running of the shaders aggregating the frames the result is averaged. The result itself may even be placed in two rendering goals depending on the number of summations. In the case of the shader averaging the result it is possible to set a minimum number of frames to be taken into account in averaging. This should be the case when the user intends to guarantee that every single pixel has a depth resulting from the averaging of at least n values. To enhance performance the shader carrying out the averaging function is fused with the distortion correction. In the course of distortion correction the system uses the predetermined distortion map. Distortion map i specifies which point of the undistorted picture is matched by which point in the distorted picture. The distortion maps are calculated upon the first start-up of the program or upon changing the relevant calibration parameters of the hardware. After the generation of the map I save it in the program's library so there is no need for calculating it when the programme is started up the next time. This then significantly accelerates the start-up of the programme itself. The process of the correction of distortions is as follows: in the course of the drawing of the given pixel I read from the map the pixel that is to be placed there and then I check whether it is still in the picture or perhaps it 'hangs off'. If it is on the picture, I write it in, otherwise I write a zero value. In this operation I utilise the interpolation capability of the card enabled by its sampler units, which provides for a nice and smooth output.



In the course of the correction of the distortions the aim is to make it possible after correction to work with the camera model as though its optical mapping were perfectly undistorted, without having any of the other parameters changed. Let the given object be in the real space, which can be mapped to the picture in three steps: a) z-normaling, b) distorting, c) multiplying with the intrinsic camera matrix. The aim is to alter the picture in a way that makes it possible to drop the distorting step. In this case since the observed reality and the other transformations remain the same, the only change is to be made to the picture. Though the model of mapping is written up moving from the real space towards the picture, and the model is unambiguous only in this direction, in practice the reverse route is also important since the three dimensional model has to be produced on the basis of the pictures.

If we have a point in the picture, then in the reverse route the following steps need to be carried out if its spatial position is the opposite: a) to the inverse of the intrinsic matrix, b) re-distortion (solution of system of non-linear functions, multiple results, choice), c) multiplication by distance. In the course of distortion correction the picture has to be changed in a way whereby from the real space the Z-normaling and *the multiplication with the intrinsic matrix can get to the picture directly* . To this end - in contrast to the original concept - there is no need for solving a system of linear functions since GPU's imaging operates logically in the reverse direction: we know where we want to draw, and we need to determine what to be drawn there (in contrast to our model where what we want to draw is given and we have to decide where). From the Z-normaled space through distortion we obtain the picture supplied by Kinect, while without those we obtain the distortion-corrected picture. Therefore if we intend to trace out a given point of the undistorted picture, then the next in an unambiguous direction, c) we multiply with the intrinsic matrix.

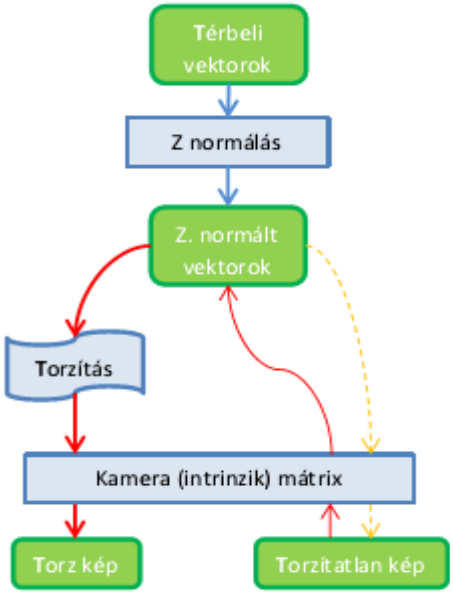


Figure 22

Figure 22. The distortion (we obtain which point of the Z-normaled space will mark transformation here while the yellow on dotted will be mapped here), b) we apply the distortion formula and the lines show the model of the resulting new system.

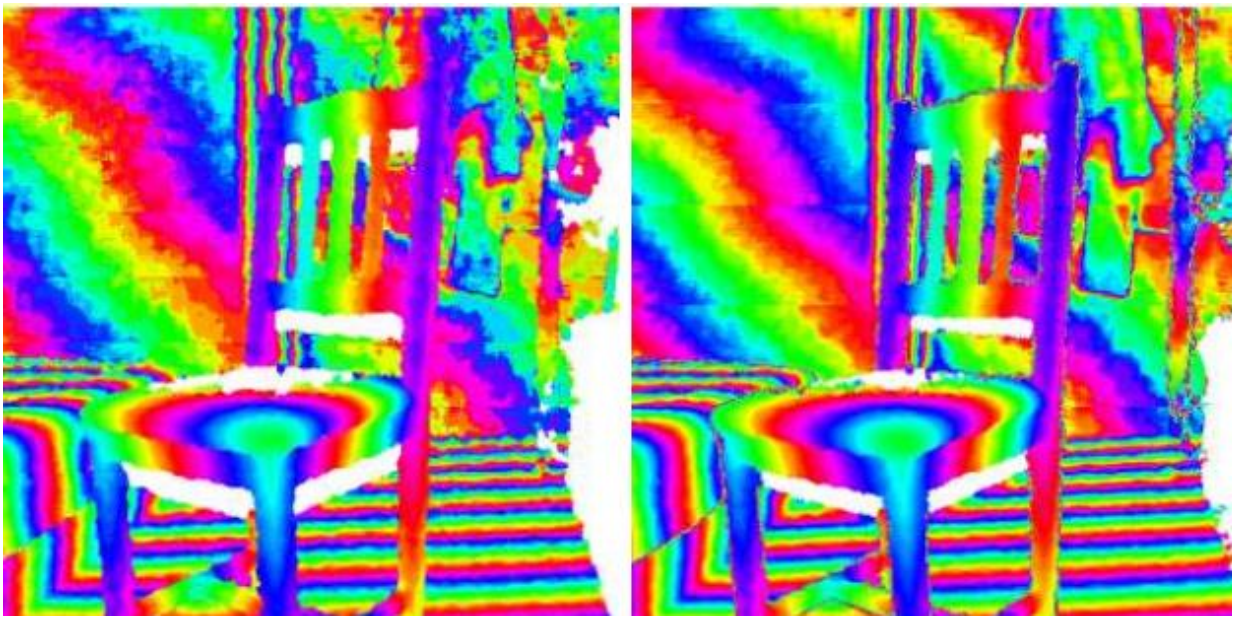
This series of operations specify which point in the distorted picture matches the undistorted point which then makes it possible to eliminate the distortion. Into each undistorted point we write the depth value obtained in the distorted point matching it, thereby we have eliminated the distortion of the picture:

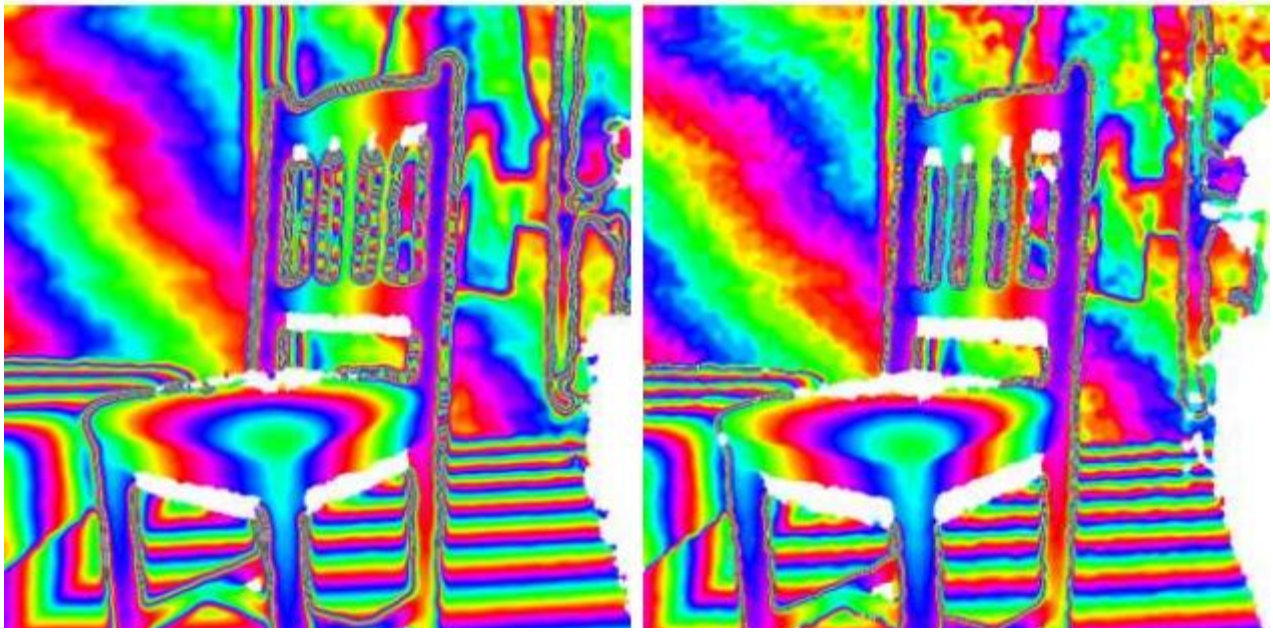
b) picture adjusted for distortion - the upper part of the picture shows the correction carried out by my program, while the lower part shows the results of MATLAB CCT, which are in overlap, as had been expected.

In the course of this operation there is no need for depth information, therefore the operation is independent of what is seen. Another advantage is that with adequate settings - by increasing the sampling frequency (multisampling) - the GPU can produce highly smooth pictures. This correction can be carried out in both the infrared and in the depth pictures with the parameters specified in the course of calibration. As has been noted, in the following phases the management of distortion can be dropped in the course of transformations, thus from this point on the camera is used as though it provided the distortion-adjusted pictures right away, and thus as if it were 'ideal'.

4.3.4 Gauss filtering

The next - optional - step of processing is the application of the Gauss filter. This filtering is useful primarily if the averaging function is not available. The Gauss filter averages the neighbouring pixels with weighting corresponding to the two-dimensional Gauss distribution thereby significantly improving the smoothness of the output. There is a need for the Gauss filtering because without averaging the Kinect provides a highly noisy output and owing to its mode of operation it has not only accidental errors but also input-dependent errors. This latter is a result of the depth calculation algorithm implemented by the machine's internal processor. The unevennesses of the surfaces sensed by the Kinect can be effectively smoothed by the Gauss filtering process. The key issue of the implementation of the filter is keeping the number of the texture reading and calculation operations low. Therefore instead of carrying out two-dimensional filtering with the required matrix convolution, I carry out the operation in two one-dimensional steps (once horizontally and once vertically). It can be mathematically proven that for instance instead of convolution to be carried out with a 9x9 convolution matrix we obtain two convolutions to be carried out with two vectors of 9 elements each. In this way only 18 texture queries need to be carried out by pixel, instead of the original 81, resulting in much more efficient operation. Further performance increase may be obtained by performing convolution not with one vector of a larger number of elements, but with several smaller ones, one after another. In this case the effects of the various filtering actions get multiplied and the resulting kernel size increases dramatically. The filtering process was implemented similarly to averaging, with the aid of two rendering goals (*A*, *B*), one of which always contains the result of averaging. In the course of the filtering process the result of the smoothing is transferred to *B*, and then that of *B* is transferred into *A*. In the course of each iteration I perform on horizontal and one vertical smoothing action. The width of the Gauss curve can be set in the system along with the distances between the sampling points. This latter should be set higher than the default value of 1 only in smaller capacity computers because although the degree of smoothing will increase, a variety of picture faults can occur in the case of values higher than 1. In the course of the calculation of the weighted average the filtering takes into account the missing pixels, thus in dividing by the sum of the weights it takes only those weights into account to which depth information was also attached. One disadvantage of the filtering is that it blurs the edges as well. In the current area of use this is not a problem but in some cases it may be. In such cases there may be a case for calculating the edge intensity values relating to the pixels (e.g. Canny's edge detection [10]), and for using this to smooth only the parts where there are no sharp edges. This method relies on the fact that the amplitude of noise is much lower than the relevant edges. The convolution vectors befitting the settings are calculated on the CPU and this is from where they are transmitted to the GPU memory.





c) d)

Figure 24. The effects of temporal average and spatial Gauss filtering: a) the original depth image, b) output with averaging, c) output with averaging and Gauss filtering, d) output with Gauss filtering (colouring period: 20 cm).

4.3.5 Creating three dimensional models

The most important element of processing is the creation of a three dimensional model. The task requires the creation of a three dimensional model from one texture, in accordance with the relevant rules. If the task were to present the depth image in a three dimensional chart as a surface, it could be easily carried out by first specifying an oblong in the in the graphic card's vertex and index buffers comprising pairs of triangles placed on the regular grid of the peaks and then by moving the positions of those peaks in the vertex shading phase of drawing in accordance with the height information drawn from the depth picture. In this case the depth picture is in fact managed as though it were the result of an orthogonal projection which of course, is not true. If the vertex positions are generated in accordance with the camera's mapping, we obtain a real and proportionate three dimensional model. The disadvantage of the method is that a lot of older graphic cards - primarily integrated graphic solutions - do not support the accessing of the textures from the vertex shaders without which this method cannot be implemented.

The following operations need to be executed for generating the model:

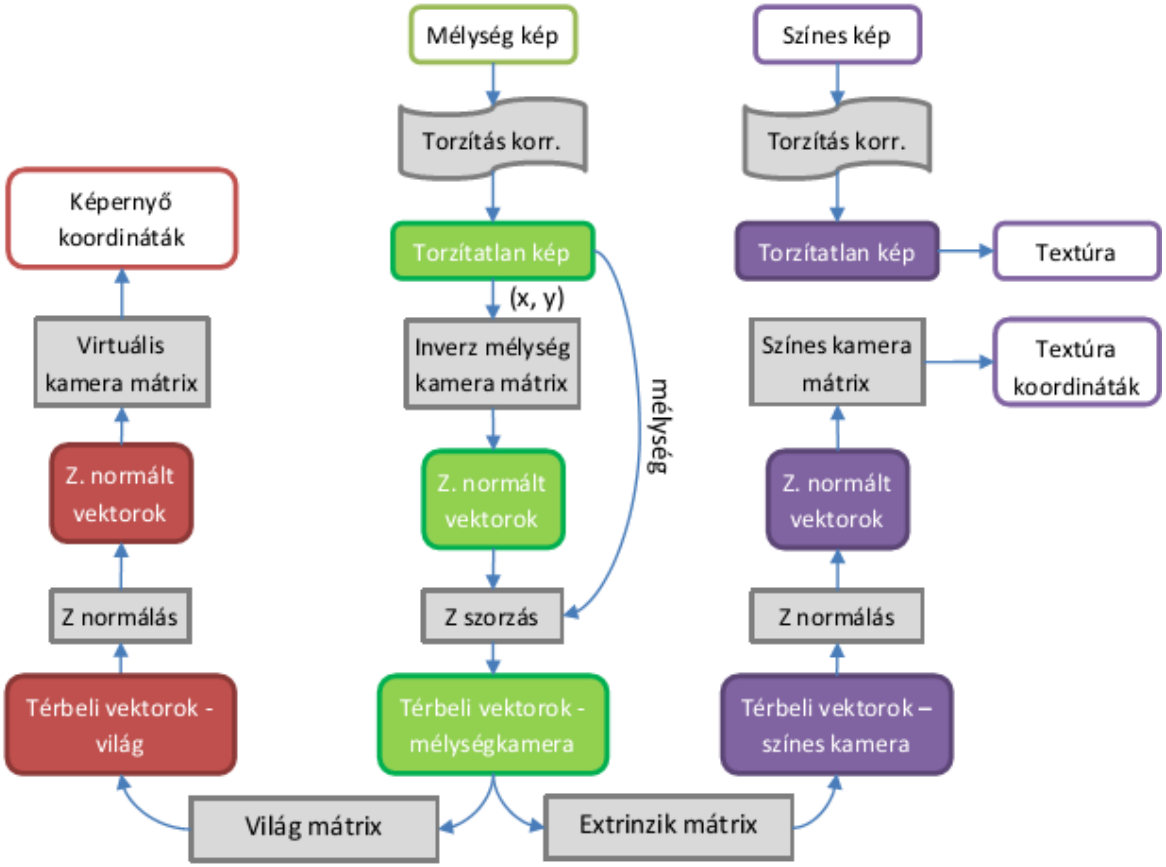


Figure 25. Schematic drawing of the operations required for generating the model

In creating the input image the points written up in the camera's coordinate system need to be Z-normaled and distorted, and finally multiplied by the camera matrix. Now the operations need to be carried out in the opposite order. Since there are no picture distortions in place any longer, they need not be taken into account. This means that Z-normaling can shifted after multiplication by the camera matrix, since this will not alter the result This is useful because in this way, carrying out the operations backwards, only one vector needs to be written up and then it is to be multiplied with the relevant matrix to generate the spatial coordinates.

As to where exactly we are, can be established from the texture position of the vertex to be processed, which in my case varies in the 0..1 range in both directions along the axes of the grid. Therefore the
following vector is to be written up:

$$Q_D = \begin{bmatrix} w \cdot u \cdot d \\ h \cdot v \cdot d \\ d \\ 1 \end{bmatrix} \qquad (10$$

Where the depth is given by the depth picture (*d*) in meters, while the values describe the position in the picture (0..1), and *w*, *h* is the width and height of the picture in pixels. It is at this point that multiplying by the inverse of the camera matrix is to be carried out. Since the pixels of the high resolution infrared pictures used in the course of calibration cannot be directly matched to the pixels

of the depth images (different resolution, different picture ratios, and there is a shift between the central points) thus the matrix received from calibration is to be modified as follows:

$$A_d = \begin{bmatrix} sx_{IR,d} & 0 & 0 & tx_{IR,d} \\ 0 & sy_{IR,d} & 0 & ty_{IR,d} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_{IR} \qquad (11)$$

Where:

- $sx_{ir,d}$, $sy_{ir,d}$: describes the magnification between the depth and the infrared picture
- $tx_{ir,d}$, $ty_{ir,d}$: describes the shift between the depth and the infrared picture
- A_{ir}: the intrinsic matrix of the infrared camera
- A_d: the depth 'camera's intrinsic matrix

After inverse multiplication the positions in the three dimensional space are available in real units:

$$P_D = {A_d}^{-1} \cdot Q_D \qquad (12)$$

Where P_D is the vector applied in the coordinate system of the depth camera. To obtain from this a three dimensional model that can be displayed on the screen, the vectors need to be projected into the picture plane. Before projection back it is possible to transform the points, e.g. to rotate them around a given point, to magnify or shift them. After completing these transformations the positions of the points in the screen coordinate system need to be calculated. Any projection parameter can be used for this; to simplify matters I used the camera matrix of the depth camera of the Kinect. The result of this is that by rotating the object we obtain a picture that is as though it were recorded by the Kinect from a different direction. In view of this I named the operation 're-projection'. Since all of the operations to be carried out are effected by multiplication by matrices, the re-projection can be resolved by a single multiplication by a matrix. The result will then have to be transcribed into the DirectX screen coordinate system, marking the screen's upper left corner with the (-1,-1), and the bottom right with the (1; 1) coordinates. Thereafter the new depth values need to be stored before Z division because that is required for the use of the Z buffer. Since the picture ratio of the screen buffer varies by the resizing of the window, the vectors of the resulting points need to be magnified in direction and then shifted. The factors applied here are determined at the time of the sizing of the screen buffer. If the model were to be textured as well, it is to be calculated where the points seen by the depth camera will fall in the picture of the colour camera. I use the transformation matrix obtained from the stereo part of calibration and the intrinsic matrix of the colour camera for this purpose, thus the necessary transformation is as follows:

$$Q_C = {A_C}' \cdot T_{D,C} \cdot {A_D}^{-1} \cdot P_D \qquad (13$$

Where:

- Q_C : the coordinate applied in the projection space of the colour camera (from which the picture coordinate is obtained after Z-normaling)
- A_C': the adjusted intrinsic matrix of the colour camera

- $T_{D,C}$: the transformation between the coordinate systems of the colour and the depth camera
- A_D: the intrinsic matrix of the depth camera

This transformation is applied between Z-normaled vectors. Since the picture ratio of the higher resolution picture used in calibration differ from that of the smaller resolution picture used in live operation, the intrinsic matrix of the colour camera also needed to be modified for obtaining a suitable result. The input of the re-projector vertex shader contained the vertex position (3D) and the texture's coordinates (2D) while its output contains the positions applied in the screen coordinate system (4D1), the depth (1D) and the texture coordinates (2D).

It is a problem that since the system is only a '2.5' dimensional one, the resulting surface will be continuous, without any gap, which means that false surfaces - that do not actually exist - appear in a direction nearly perpendicular to the optical axis. To remove these, the direction of the normal vector of the surface needs to be estimated. The problem is, however, that depending on the settings the input may be noisy and the methods performing high quality derivation take a lot of calculation power. To eliminate this I chose a simpler method, which simply calculates three dimensional positions of each of the four pixels (over, below, on the right and on the left hand side) at a given distance from the given pixel in the depth image, and then from this it produces the following sum:

$$\left(\frac{p_u - p_d}{d}\right)^2 + \left(\frac{p_r - p_l}{d}\right)^2 \qquad (14)$$

Where :

- P_U , P_R,P_D ,P_I : the positions of the neighbouring pixels above, below, to the right and to the left
- d : the distance of the pixel concerned, from the camera

If this sum exceeds a predetermined value, the shader sets the vertex position coordinates to the no number (NaN) value. Since the HLSL language used for the shaders does not enable the use of the no number constant, I write it from an external program code running on the CPU while it is running, into a global variable in the course of the initialisation of the card. As a result of the NaN value the rasteriser rejects the triangles concerned, thus the false surface segments disappear.

4.3.6 Shading of the model

Once the transformed vertexes get to the rasterising unit, the pixel shaders set to work. I wrote a number of different pixel shaders, which enable the following display forms:

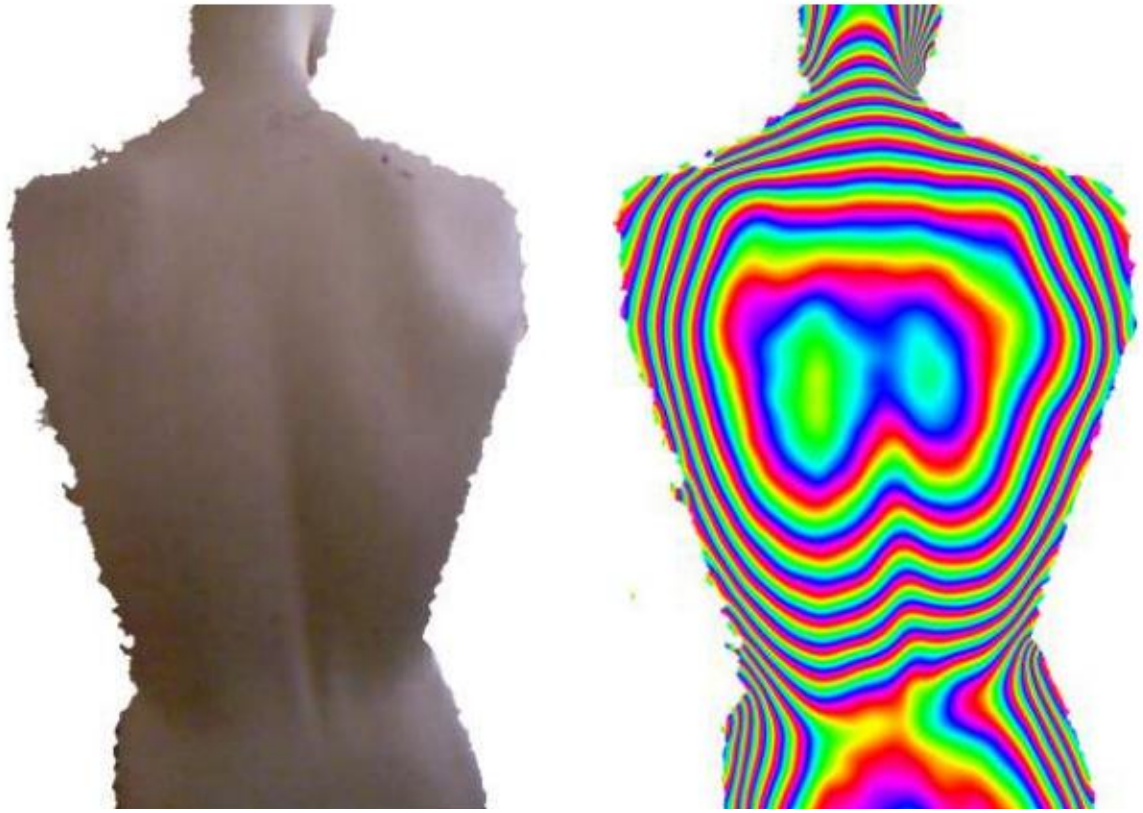
- Textured: the model is textured on the basis of the picture taken by the colour camera. The texture coordinates calculated by the vertex shader - for the peaks - are transferred to the pixel shader after interpolation (as all of the values that are stored in floating points). To display the texture the distorted coordinates belonging to the received undistorted texture coordinates need to be read out first, in order to be able to eliminate the distortion. I sample the colour picture in the position so obtained.
- Zebra: in this case the basis of shading is the new depth value obtained from re-projection, whose sinus function of the given period is the basis of colouring. The black and white streaks of the end result may be regarded even as a false Moiré image.
- Rainbow: in this case I calculate the residue of the depth for a period specified by the user, and then I divide it with that. In this way I obtain a value between 0 and 1, repeated in cycles with growing distance from the camera. Then taking this value as a colour shade I

carry out a HSL-RGB colour space transformation the result of which is a periodically repeated rainbow-like colouring pattern.

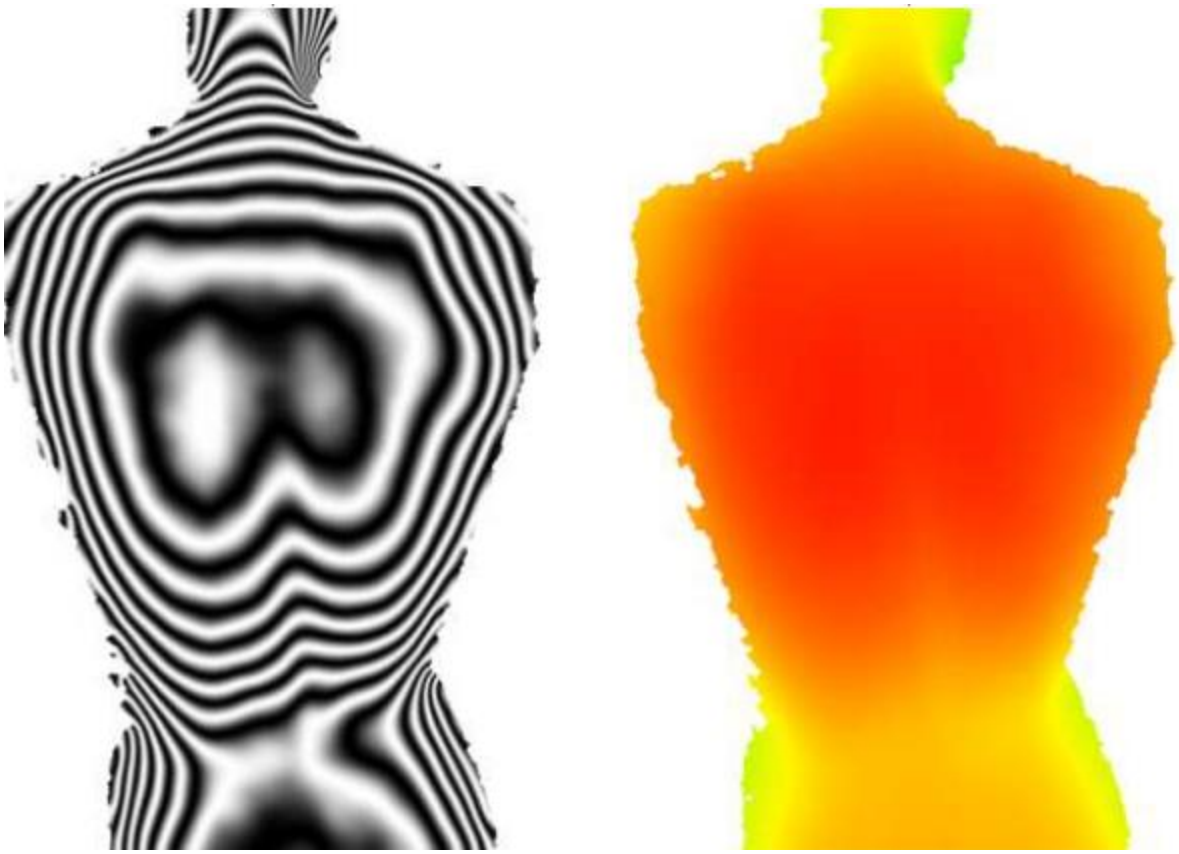
- Shaded rainbow: the only difference in this case from the above one is that in the course of the HSL-RGGB transformation the value of brightness will be the z component of the estimated normal vector of the surface. In the course of the estimation of the normal vector I proceeded in a similar way to the solution applied in eliminating the undesired triangles, but in this case I calculated the result of the cross multiplication of the distances between the horizontal and vertical neighbours and then I normalised that result. Taking the z component of the vector so obtained as brightness, a shading that is similar to topographic images can be obtained, in which the detailed texture of the surface can also be clearly seen.
- Palette: in this case I do not perform a HSL-RGB transformation, instead, I choose one colour from a texture comprised of one line with the value describing the colour shade, which the user can specify in advance in the form of a bit picture.

After shading buffer is ready for display. Drawing is not done directly on the screen but in the back buffer therefore after the completion of drawing it must be transferred by a command to the front buffer. If it was switched on in the course of the initiation of the graphic adapter, the drawing of the frames is synchronised with the monitor's picture update frequency. At this point the command triggering the drawing of the picture waits for the next update until which it stops the running of the program.

In addition to the three dimensional model it is also possible to view and save the raw outputs of the colour, infrared and depth modes as well. Since the content of the textures stored on the graphic card can be copied into the central memory, it is possible to export and save the results. Though the XNA system itself also offers methods for saving pictures, in some cases those produce unexpected results therefore I could not use them. The .Net framework system enables the writing of pictures of different formats through the GDI. To this end, the pictures have to be copied with pointers into the GDI picture object where attention is to be paid to the fact that the sequences of the colour channels differ and that the length of every line is a multiple of four in storage (they may end in unused pixels).



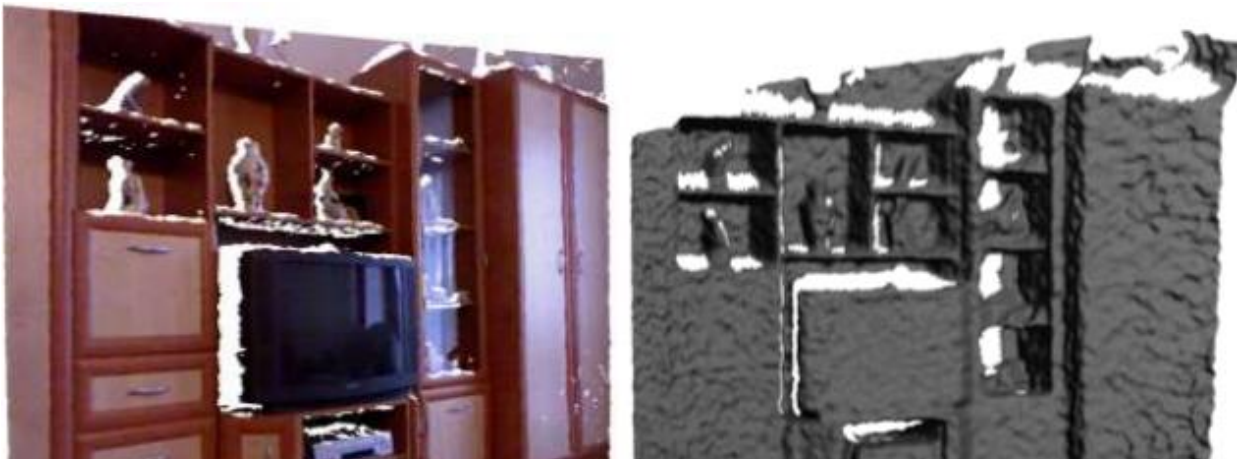
a) b)



c) d)

Figure 26. Images produced of the back of a puppet with different shadings: a) textured, b) rainbow (colouring period: 1cm), c) zebra (colouring period: 1cm), d) scale

a) b)



c) d)

Figure 27. Images of furniture: a) textured model, b) rainbow colour fill-out according to depth, c) model rotated from the original view, d) model rendered with the Autodest Maya software (clearly, the Klinect is less accurate in the case of larger distances)

The programme enables not only the saving of images but it also makes it possible to export images as models. A model can be exported in one of two ways:

a) as a block of three dimensional vectors, where every vector describes the spatial position of the corresponding pixel

b) in STL format, in which the coordinates of the peaks of the triangles are specified, in addition to the normal vectors of the various triangles

Besides the saving of models the textures are also saved.

The designing of the processing process was heavily influenced by the fact that the graphic card as a resource becomes inaccessible for processing in some cases, or it has to be initiated again in order to modify various settings. Since the object representing the graphic adapter is attached to the screen buffer and to the window enabling description, its resolution cannot be modified during the process - that requires the creation of a new object specimen. Windows virtualises the graphic accelerator therefore it can be used by multiple programs in parallel. Virtualisation also ensures that the various users cannot access each others' memory areas. This is true even if we link to the card several times within a program. When we get detached from the graphic card by discarding the relevant object, the contents of all used memory areas get lost. In this way upon resizing the window in the course of creating a new object specimen all of the data, texture etc. have to be uploaded onto the card. Meanwhile, processing is halted and can be continued only after the uploading is done. The data had to be reloaded also when the device 'got lost' (except for DeviceLost) which may be caused by a defect in the driver of the graphic card or by the appearance of the Windows Security Prompt (when the screen is darkened for security).

4.4 Application for diagnosing scoliosis

The programme can be used for diagnosing scoliosis as well, based on the three dimensional models and false Moire images taken of the back (Figure 26).

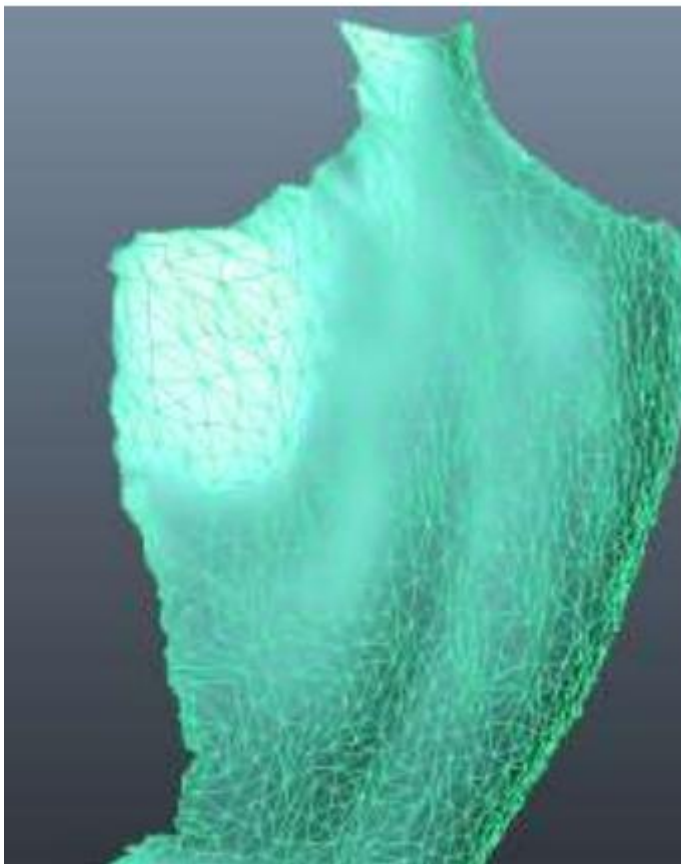


Figure 28. The model of the back after a 98 % reduction in the number of triangles

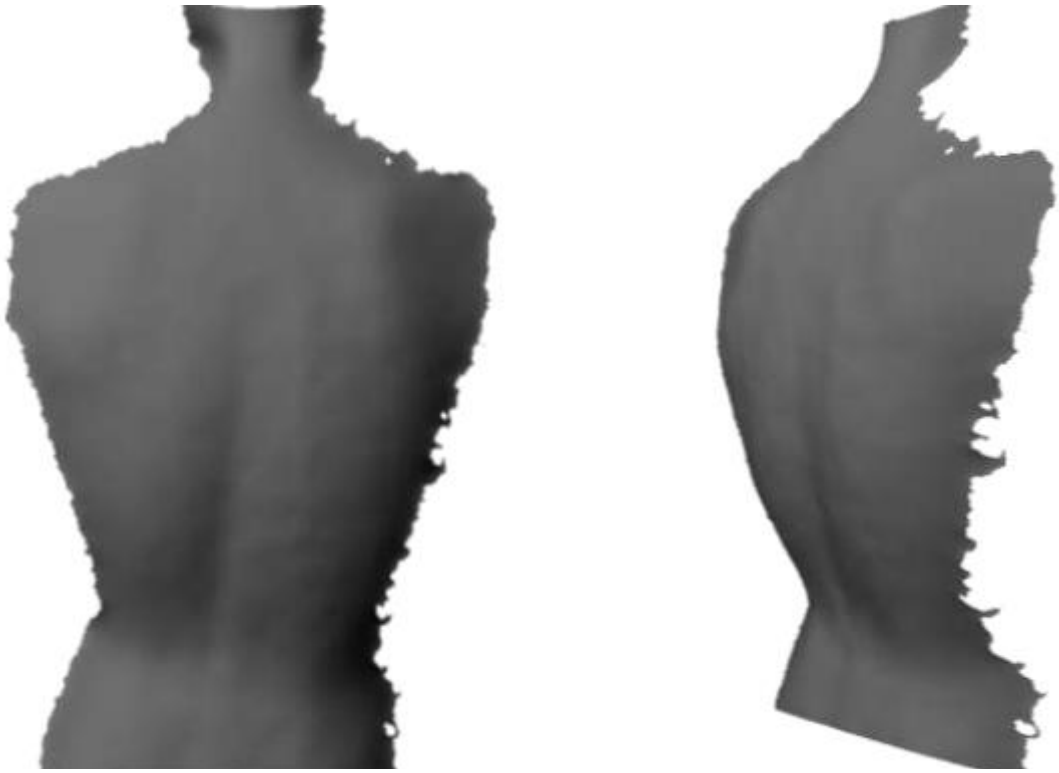


Figure 29. Rendering of the saved model with the aid of Autodesk Maya

Elaborated diagnostic methods are already available for false Moire images, while the three dimensional models enable the preparation of an automated diagnostic program

.

In 2012 we managed to connect the developments described and presented above and then we produced a suitable program and the automated diagnostic device. From this moment we reached the end of our research. Further development options

The development of the program is being very actively continued, thus I can describe a large number of directions for further development that are currently being worked on, including the following:

- Microsoft has recently released a new version of the Kinect development tool kit which is capable of - among other things - accessing the infrared video stream that used to be accessible only with the OptenKinect driver, and setting a lot of camera parameters (e.g. exposure time). This may make it possible to resolve problems caused by autofocus and for the self-blinding of the infrared camera of the Kinect. Another advantage is that hardware installation would also become easier.
- Full automation of the calibration of the Kinect, and instead of applying the MATLAB script, its integration in the programme.
- Part of the user interface of the programme - dealing with the processing of the recordings of the rotating scanner - is still only being used for testing, these must be made more user friendly before practical use.
- Increasing the program's hardware independence: at present the program works with newer graphic cards (those fully supporting the DirecX Shader Model 3 and the texture reading of the vertex shaders). In the case of older computers and those using integrated graphic processors (except for the latest generations) the programme cannot be started. To resolve this the parts running on the graphic card need to be changed to make them operate on a smaller set of commands and to make them shorter.
- The stereoscopic parts of the programme are not compatible with a lot of hardware elements, thus there may be a case for the use of a more widely spread program package that can produce all kinds of outputs. The development of integration with the TriDef Ignition software is currently underway.
- Further development of the Kinect model, e.g. reducing the impacts of quasi-static place-dependent picture errors, or analysis of the place-dependence of the depth function. In the course of development, of course, any programme defects are also eliminated as they emerge. The general parts and those aimed at spine diagnostics have been tested widely and for long, and they are already functioning highly reliably and without flaws (they are already even in live use). The new rotating scanner parts are still in the development/testing phase, the testing of program stability will follow the completion of calibration, connection with hardware and the final shaping of the operator interface, after which this part of the program will also be released. This latter is expected to take place in the first quarter of 2013. The programme is equipped with an automated on-line update module thus the various developments and patches are continuously delivered to users. Upon each start-up of the program the update module checks whether any new version is available and if it is, it offers its installation to the user. If the user allows the update, it is carried out automatically. An installer is also produced for the program enabling the suitable installation, the association of the file types and the required removal of program elements as required for installation.

Conclusion

In my paper I described the possibilities of the use of depth sensors for medical purposes. As was demonstrated, these sensors make it possible to efficiently produce high quality models without having to prepare the environment to be mapped. The small room requirement of the sensors make it possible to use them in portable devices as well. The models so produced can be highly useful in analysing deformities, injuries or other anomalies and for fixing their shape (dermatology), as well as for the manufacture of tailor-made medical aids. The resolution of the existing sensors is going to be increased substantially, which will make them suitable for use in other areas as well. In addition to resolution, there is also room for improvement in the noise level of the output of the sensors.

Form the aspect of the algorithm most difficulties are encountered primarily in matching the different three dimensional pictures and the construction of a model comprising all of the available images. These are also the operations that take the most processor time in the course of processing. At the same time, the results presented here also show that by working out suitable algorithms even today's processors make it possible to produce relatively high quality results. From an IT aspect the need for accelerating by hardware should be highlighted, because the real time execution of the operations to be carried out here could only be achieved with extremely powerful processors.

In addition to the general use of depth sensors this paper also demonstrated the resolving of a medical task. Such solutions have been worked out that could only be implemented in such an easy to use, fast and efficient way, with such little room requirement, only for very high costs without the use of depth sensors. The program used in examining scoliosis is already in live use in the Heim Pál Hospital and the Orthopaedic Clinic, while the rotating system is expected to be finished in the first quarter of 2013. Developments and patches are automatically received by the users, thanks to the on-line update system. The structured nature of the program ensures that it can be easily enhanced in further directions. This is ensured by the existing plug-in system as well, providing for the integration of various modules (e.g. rotating scanner) via regulated interfaces, making the code more visible and enabling multiple developers work together.

There are a lot more possibilities for the use of this technology. One great possibility is a special Dermatological use:

Medical Examination of Skin Cancer

In the last few weeks of further development with our basic project, we have reached further possibilities.

- The found the right usage for dermatological examination
- In further use we found out that the software what we use for the ortopedic diagnosis is in basically the base for this is already made for further eliorations to this way. So it will be possible to have the same device for both medical sectors, as different applications.
- We have recognized that this way we can create a device for home usage as well.

This project is not finished yet, and it needs further development of 12 months and 1,800 000 USD of funding.

The costs for the development are significantly part in the needed hardware base, because it is important to create a easily and inexpensively produceable device. For this we have our results in the research already in the experimental stage, but those results were already showing the right direction.

The usage and application of the finished device

The device can check the birthmarks and other spots and moles on the skin, it can analyze the negative, maling disorders. It will save the 3D image file of the body and the skin (urface of the body) so that way it gets possible to check back, and later it will be possible to compare if there are any further changes on the skin. That way it will be fast and easy to diagnose any further disorders on the body, and it will be amazingly easy to use for appropriate treatments. It will recognize also the smallest changes on the skin, evan if that is a small birthmark. It can recognize all three sorts of skin cancer.

- Based on epithelium part of the cell that is the starting from the basal cell carcinoma (carcinoma basocellulare)
- Spiny strata based spinalioma,which is located above the basal epithelial cells (carcinoma spinocellulare)
- Melanoma in the epithelium located pigment-producing cells (melanoma malignum)

In the language of medicine when we talking about skin cancer, basalioma and spinalioma is what we mean by that mostly, but althought these are carcinomas, they are usually not deadly diseases. Melanoma is frequent carcinoma, which is unlike the other types of skin cancer, is spreading from the pigment cells, which are not the part of the skin structure itself, but they are parts of the nerve cells. Melanoma is a very agressive disease, and potentially it is deadly, nevertheless it is healed in the beginning stage.

If someone recognizes the similar or exsact symptoms of Melanoma, should make the identification, and go as fest to the doctor as possible. But to make the right desizion, those persons, who has a tendency for any cancer type, should have a our device, to make regular control especially in sunny times.

The marketability of our devicee is endless, for the households.

As in the case of any medical technological development though the engineering work and scientific results are important, the most crucial requirement is that the resulting devices and techniques are as useful for people facing medical conditions s possible. Accordingly, I also hope that the project's completed and future developments can improve the lives of many of our fellow human beings.

Sincerely,

Zoltán Kalla

.