ACTIVE INFRARED MARKERS FOR AUGMENTED AND VIRTUAL REALITY

Evalds Urtans, Agris Nikitenko
Riga Technical University, Latvia
evalds.urtans@edu.rtu.lv, agris.nikitenko@rtu.lv

Abstract. Our research is proposing an algorithm and technical implementation of a system that can recognize positions and orientations of IR (Infrared) LED (Light Emitting Diode) markers that are invisible to a naked human eye. The system is made to work with Oculus Rift DK2 head-mounted display coupled with a 111 Hz IRLeap Motion camera. It adds functionality to these devicesby allowing them to track different kinds of objects using active IR markers. Up to now, the most common way for tracking markers for augmented reality were using fiducial markers that are visible to human eye orother static markers. Such marker systems require flat surface for attaching a sticker of a fiducial marker. With our system it is possible to create invisible markers and they can be attached to objects without flat surfaces. As aproof of the conceptwe made a virtual ping-pong game where the player uses a physical table tennis paddle fitted with active IR markers. At the time of the research, no such tracking systems were available in public domain. Currently similar commercial systems are in development by Oculus and other companies. Limitations of hardware that we found in our research might be one of the main reasons why the commercial product is not yet completed. In case of Oculus Rift's IRcamera, it has insufficient 30 Hz frame-rate to support active IR markers. We found that even with the capture rate of 111 Hz the proposed system works 3 times slower than conventional fiducial markers, but starting at this frame rate it can be applied for real-time applications.

Keywords: virtual reality, augmented reality, infrared markers, infrared tracker, signal processing, Oculus Rift, Leap Motion.

Introduction

In recent years there have been many research efforts in augmented and virtual reality using Oculus Rift headsets [1-2]. Our research utilizes the Oculus Rift headset and Leap Motion infrared stereo cameras together with custom built active IR markers on Arduino platform. Our system allows to experience augmented or virtual reality where the user can interact between physical and virtual objects. Leap Motion captures the surrounding space in real-time with stereoscopic video feed and Oculus Rift headset provides visual representation of augmented or virtual reality from this feed. In between processing pipeline our purposed system comes that can capture the position and orientation of objects that are fitted with our markers.

The system consists of a set of algorithms and a physical implementation that produce exact position of the infrared marker in 3D space relative to a viewer (see Fig. 1).

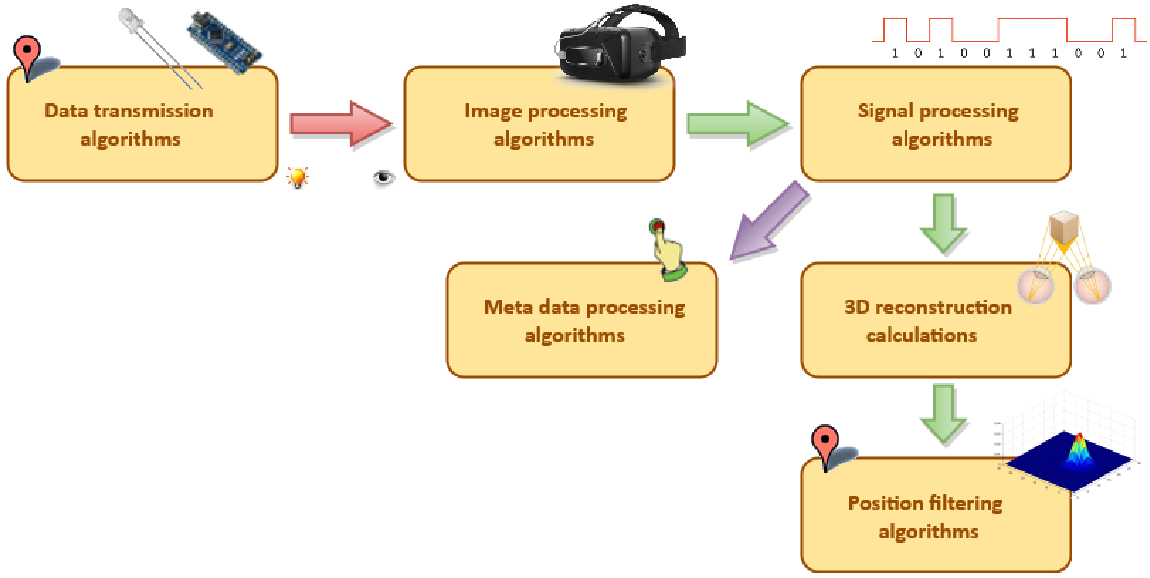

Fig. 1. **Activity diagram for the system of active infrared markers**

Ability to physically touch virtual objects has been studied extensively recently. Most widely spread method currently is to utilize 3D printing, but more dynamic methods have been proposed. For example, surface of pixels with dimension of height [3] or even levitating touchable pixels [4-5]. The goal is to provide haptic feedback from physical objects and apply virtual constructs over them for augmented reality. Our proposed system also can be used to simulate virtual overlay over physical objects even using different graphical representation. Active infrared markers can be placed on the object with any form factor. For example, a ball fitted with active infrared LED markers would have a better haptic representation of a virtual object than a box using fiducial markers.

Fiducial markers as shown in the image below are used as a standard for augmented reality (see Fig. 2). These markers work well in most cases, but they have limitations. They are clearly visible and must be placed on top of an object. Our proposed markers can be placed under the cover of an object and can be invisible to a user. IR LED can be visible through many types of plastic using IR camera.

Fig. 2. **Traditional fiducial marker system used in augmented reality**

As a proof of the concept we have implemented a game of table tennis where using active IR markers it is possible to play tennis with a physical paddle, but using a virtual ball as shown in the image below (see Fig. 3). As an alternative to our system other static and dynamic IR marker systems have been made using stereo cameras or even a single camera [6-8]. These systems use special initialization procedure. Others use PnP (Perspective-n-Point) solver like the fiducial marker does where it is necessary to know geometric dimensions of markers before using them. In our purposed system initialization can be done while the system is in operation without need to resetit with some special initialization procedure. Another approach is to use features in an image as markers. These features in an image could be coloured spots, edges, corners, etc. Such systems usually use variations of SLAM (simultaneous localization and mapping) algorithms and are not as precise as prefabricated markers
[9-10].

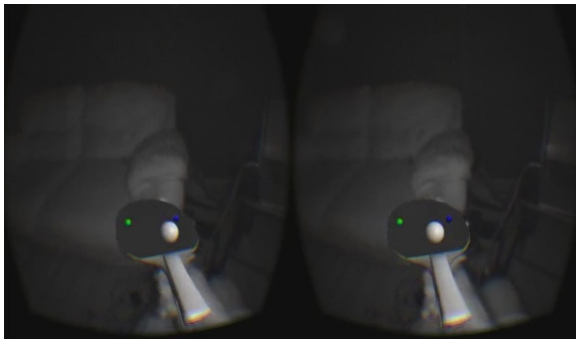

Fig. 3. **Table tennis paddle and a virtual ball in augmented reality using active infrared markers**

Physical implementation of IR markers

Physical implementation of the proposed system consists of IR stereo cameras that are mounted on to VR headset and trackable objects fitted with active infrared LED markers. We used Leap Motion

as IR stereo camera. It has a filter for 850 nm light waves and produces grayscale image of light intensities. As for active IR markers we have chosen 850 nmIR LEDs. Each object should be fitted with at least 3 LEDs in order to detect the object's orientation and position in space (see Fig. 4). In order to just find a position of an object, ignoring its orientation it is necessary to detect at least one marker suing both cameras (see Fig. 5). Intensity change or blinking interval of LEDs are controlled by the Arduino micro-controller that is sending identification codes of each marker with 37 Hz data-rate. Technology behind physical implementation is similar to TV remotes, but usually TV remotes transmit data with much higher frequency 33-40 kHz [11]. Another difference is that TV remote receiver is a simple light receiving diode, but in our implementation it is Leap Motion stereo cameras. Stereo cameras can be used to calculate the position of a marker in 3D space without using PnP solver. Data transmission program was implemented using timer interrupts to save battery.

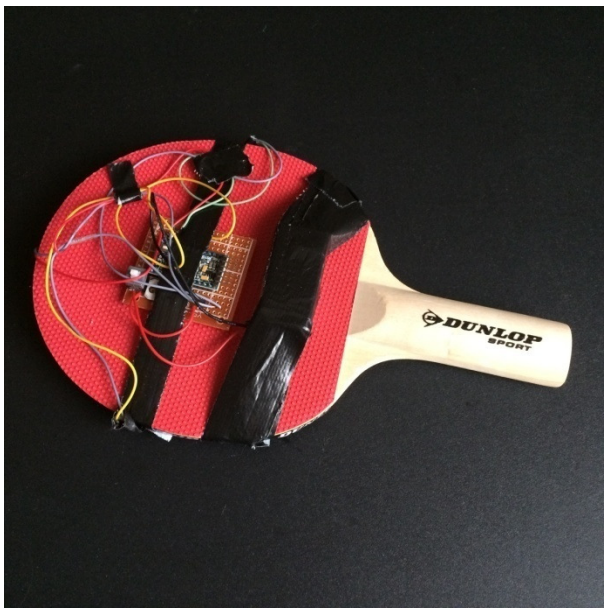

Fig. 4. **Table tennis paddle fitted with IR active markers**

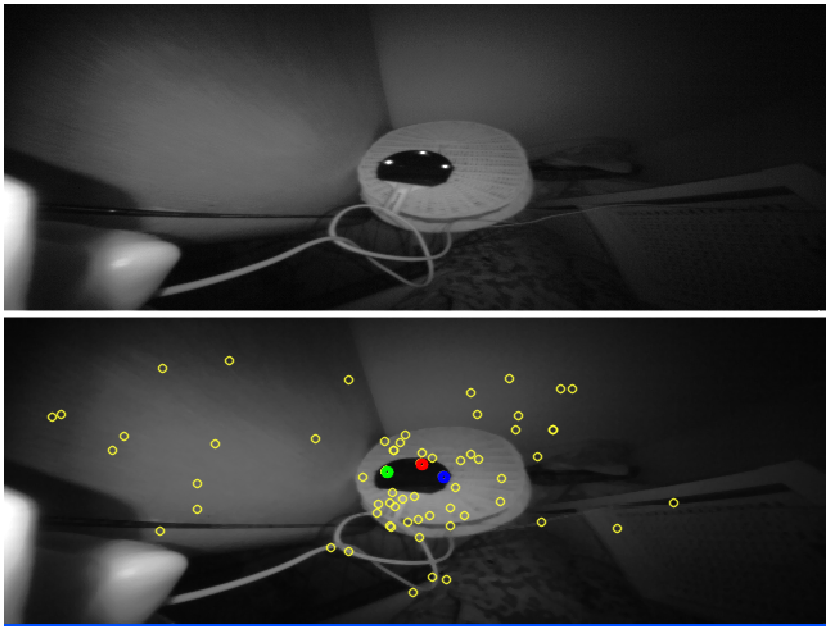

Fig. 5. **Identification of markers in captured image from Leap Motion stereo cameras**

As a proof of the concept we fitted our system also inside a plastic toy and were able to track its position. This implementation is invisible to a user, because IR LED can be visible through the plastic only using IR camera (see Fig. 6).

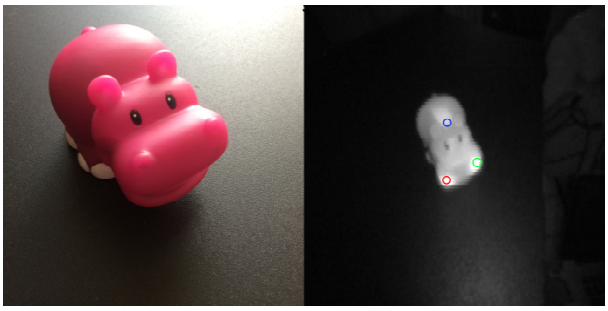

Fig. 6. **Identification of markers in captured image from IR camera**

Method for IR camera image processing

Image processing algorithm is executing the following steps.

1. Each camera stores a buffer of images.
2. Resolutions of images are reduced by the factor of 2 to improve performance.
3. Pixel blobs are found representing blinking IR LED markers.
    1. In order to eliminate lighting conditions iteratively filter images by slices of intensity of spectra from 5-255 with a step of 10 intensity units (see Fig. 7).
    2. Blur each slice of image using Gaussian filter to reduce noise and graininess.
    3. Use Median filter to make the image sharper.
    4. Find centres of blobs.
    5. Store state of blobs in between data frames. If blob is existing at same location store the value of 1, but if it is not store the value of 0.
    6. For each potential blob apply Manchester decoding algorithm that ensures time independent data transmission in Arduino implementation.
    7. Apply error correction algorithms – FEC, SEC, DEC (described in detail in the section about signal processing algorithms).
4. Classify blobs as markers using previously known identification codes.
5. Calculate positions of each marker in 3D space using geometric transformations.
6. Calculate the position of the object that is fitted with a set of markers (at least one marker detected).
7. If possible, calculate orientation of the object that is fitted with a set of markers (at least 3 markers detected).
8. Apply transform to positions of markers. Transform is acquired from the VR headset tracking system in order to predict the marker position after movement of head.
9. Apply Kalman filter to positions and orientation of markers.
10. Display the virtual object over the physical object or use positional data of an object.

In order to correctly segment markers it is very important to slice intensities of the image using a binary filter. It is necessary to iterate through intensities of each pixel image in a range 5-255 with a step of 10 units. All unfiltered intensities of pixels within each iteration are stored with value 1 and all filtered pixels are stored with value 0.

If a marker candidate is detected, then its position and bit status are stored for particular frame and for both images of a pair. Empty bit is stored for the candidate if no blob has been detected at the position of a previously detected blob at specific intensity range. Afterwards these candidates in series of frames are used for signal processing. Most of series of blobs do not contain any data from markers, but in the midst of noise data streams from markers can be detected.
In order to detect contours of blobs S.Suzuki algorithm was used [12]. Other algorithms for detection of blobs also can be applied [13-16]. Position of a marker is calculated using thegeometric centre of a blob from a bounding box. These algorithms are well established and widely used in OpenCV image processing library.

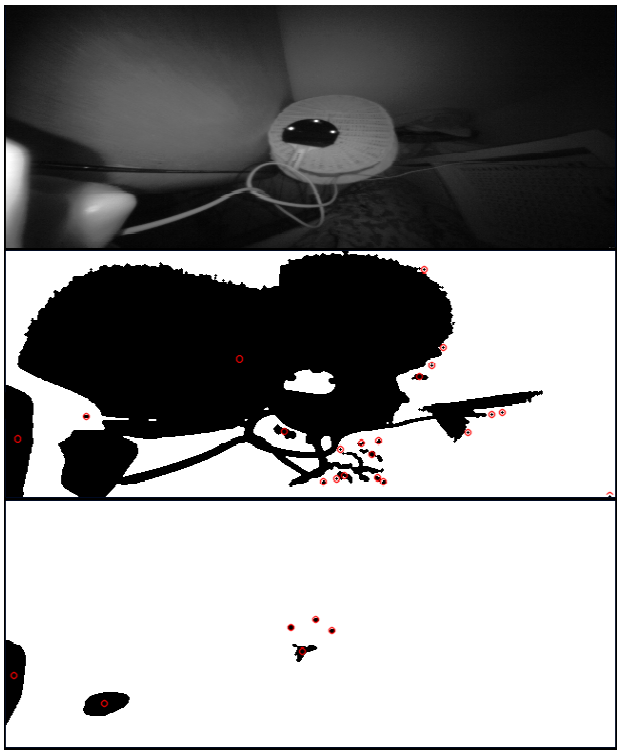

Fig. 7. **Binary filter with thresholdranges for tracking LED markers**

Method for active IR marker signal processing

Identification codes of markers were sent over the air using IR LEDs at 37 Hz data-rate. These codes where received with Leap Motion IR stereo cameras with unstable 111 Hz frame-rate. Transmission is omni-directional. It is coming from the transmitter to the receiver. For transmission auto correction code – FEC (Forward Error Correction) was used. Each 3 data slots contain 1 bit of data that can be automatically corrected if the signal has been noisy.

Table 1

FEC using 3 data frame slots

Transmitted data frames	Received data frames	Bit state
000	000	0
000	001	0
000	010	0
000	100	0
111	110	1
111	101	1
111	011	1
111	111	1

Another aspect that must be taken into an account is the Leap Motion cameras. They do not produce steady 111Hz data stream that would give image frames with 9ms delay. Instead it produces frames with variable delay of 7-21 ms [17]. Luckily, it is possible to get time-stamp from each frame that can be used for error correction. Using this time-stamp it is possible to detect hidden bit values and assign them as either "1" or "0" depending on known bits in 3 bit frame.

Table 2

Hidden bit value caused due irregularities in frame-rate

Delay between frames	9ms	8ms	9ms	18ms	7ms	9ms
Bit	1	0	1	1 **?**	1	0

It is also important to accumulate shifted time delays in order to detect a point at which FEC will fail. When it happens, the system can still try to use the received data and correct it.

Table 3

Time shift of data frames due irregularities in frame-rate

Delay between frames	**7ms**	**7ms**	**7ms**	**7ms**	**7ms**	**9ms**
Bit	1	0	1	1	**Error**	**Reset**

Especially because of the problem described above it is not possible to reliably transmit long sequences of repeating bit values, because it will be very difficult to find out the exact number of same bits in a row. For example, "111101" could easily be read as "11101". In order to get a precise number of sequential the same number of bits Manchester encoding should be used [11].

Using simple XOR calculation each bit is synchronized with a clock frequency that takes 2 data frames. Only at frames where bits are flipped in data sequence the transmitted bit changes. Because states are constantly oscillating, time shift of the same bit value does not affect transmission anymore.

It is important to keep in mind that Manchester encoding is implemented on top of FEC algorithm. It means that each bit in Manchester encoding is represented by 3 physical data frames.

After implementation of Manchester encoding, optionally SEC (Single Error Correcting) or DEC (Double Error Correcting) codes can be implemented to furthermore improve stability of data transmission. From our results that are described in more detail in next chapters we found that such error correction layers are advisable only in a noisy environment like in fog or snow. Whereas in normal room conditions no interference can occur. Usually in room conditions data are received immediately as soon as the marker is in a direct line of the sight of a camera. If SEC is necessary it can be implemented using the parity bit pattern. This pattern adds additional bits (parity bits) to sequence at positions of 2^k in a resulting sequence. Depending of a length of sequence different numbers of parity bits are added. Algorithm is also called as Hamming Encoding [18]. Number of parity bits k needed to send a string of data bits m must comply with Hamming rule (1).

$$2^k \geq k + (m + k) + 1. \tag{1}$$

In order to encode data sequence of 4 bits D using SEC, parity bits b_1, b_2, b_4 must be added. It would satisfy the rule above.

$$2^3 \geq 2 + (4 + 2) + 1. \tag{2}$$

$$D = \{0,1,1,0\}. \tag{3}$$

Below are given sequences of bits with SEC parity bits added.

Table 4

Sequence of bits with SEC parity bits added

1	2	3	4	5	6	7
b_1	b_2	0	b_4	**1**	**1**	0

For calculation of parity bit values, it is necessary to store in a matrix indexes of positions that are holding the bit value of "1". In the given example these indexes are 5 and 6. Then these indexes should be stored in binary format to corresponding columns like shown in Table 5

Table 5

Calculation of parity bits using matrix and XOR

Indexes in resulting sequence (state "1")	b_1	b_2	b_4
5 (decimal) = 101 (binary)	1	0	1
6 (decimal) = 110 (binary)	1	1	0
XOR to find b_k	0	1	1

Rows in a matrix are the same number as values of 1 in data sequence. For example, if bit 7 would be with a state "1" then 3 rows of values would be subjected to XOR. But using the given example the following data sequence is produced and transmitted.

$$D' = \{0,1,0,1,1,1,0\}. \quad (4)$$

Table 6

Sequence of bits with SEC parity bits added

1	2	3	4	5	6	7
0	1	0	1	**1**	**1**	0

On the receiver's side parity bits are also calculated from data bits and compared with those that are received along with data bits. If both sets of parity bits match, then no error is present in a sequence. If they do not match XOR between both sets of given values of a syndrome, then it points to a wrong value of an index in a data sequence. If a bit is flipped then the new syndrome value is equal to 0. Sequence D'' with an error at bit index 6 is given below.

$$D'' = \{0,1,0,1,1,0,0\}. \quad (5)$$

Table 7

Sequence of bits with SEC parity bits added and one error bit

1	2	3	4	5	6	7
0	1	0	1	1	**0**	0

Calculation to find an error bit is shown below in a matrix.

Table 8

Calculation of parity bits using matrix and XOR to find a syndrome

Indexes in resulting sequence (state "1")	b_1	b_2	b_4
Received states b_k	0	1	1
Calculated b_k	1	0	1
Syndrome (XOR)	1	1	0
Index value 110 (binary) = 6 (decimal)			

Syndrome in an example above points to an index value 6. It means that after flipping the bit at this index, error is automatically corrected and $D'' = D'$.

In order to apply DEC an additional bit should be added in order to use extended Hamming distance algorithm [18]. Bit is added at the end of a sequence and it is calculated by XOR between all values in SEC sequence. For example, D' can be encoded with DEC as shown below by adding a bit with index 8.

$$b_8 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1. \quad (6)$$

$$E = \{0,1,0,1,1,0,1\}. \quad (7)$$

Then on the receiver's side extended Hamming distance bit is also calculated and the following set of rules are applied in order to find 2 error bits.

Table 9

SEC-DEC rule set for 4 bit data transmission

Rule		**Conclusion**	
Extended bit	**Syndrome value**	**Number of errors**	**Description**
$b_8 = b_8'$	0	0	No errors
$b_8 \neq b_8'$	0	1	Error in b_8
$b_8 = b_8'$	$\neq 0$	1	Error indicated by syndrome value
$b_8 \neq b_8'$	$\neq 0$	2	Must sequentially test one by one each bit 1..7 until the number of errors reduces and apply rules

Notice that FEC, SEC and DEC are applied before Manchester encoding. These auto correction codes become more effective when data sequences are longer. It is not very important for data being

transmitted in our proposed system that uses 4 bit unique identification codes. In production systems it should use longer identification codes that could be generated from time-stamps to reduce a possibility of interference between identification codes. In our implementation the length of codes was restricted by frame-rate of Leap Motion stereo cameras.

Transmission algorithm for each LED is shown in the following steps.

1. Create an identification code (sequence of 4 bits).
2. Optionally apply SEC (sequence of 7 bits).
3. Optionally apply SEC-DEC (sequence of 8 bits).
4. Optionally add start of transmission code (for example "000") if identification codes could collide in continuous sequence of repeated codes.
5. Optionally apply Manchester encoding (sequence of 16 bits).
6. Apply FEC encoding (sequence of 48 bits) by 37 Hz data-rate.
7. Continuously repeat transmission.

Data retrieval algorithm for each Leap Motion stereo camera.

1. Receive image at 111 Hz data-rate.
2. Extract data from FEC.
3. Optionally extract data from Manchester encoding.
4. Optionally extract data from SEC-DEC.
5. Optionally extract data from SEC.
6. Check identification code.

Method for positioning markers in 3D space

After each marker in stereoscopic image has been identified, its position in 3D space can be calculated. The distance between cameras *b* and focus length *f* to image plane are known (see Fig. 8).

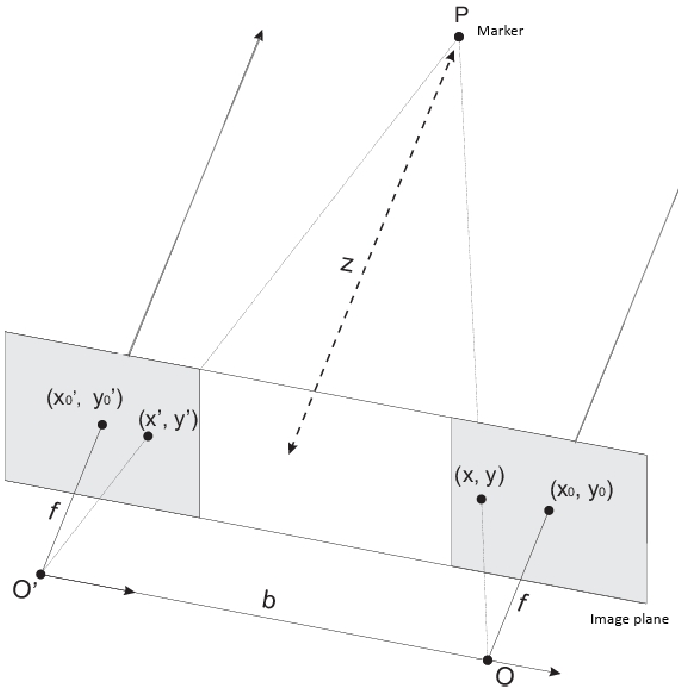

Fig. 8. **Marker positioning in 3D space using stereoscopic projections**

The task is to find coordinates of a marker $P_c = (x_c, y_c, z_c)$. To find the distance to marker z_c, the value of disparity d is introduced [19].

$$d = x' - x, \tag{8}$$

$$z_c = f \cdot \frac{b}{d}, \tag{9}$$

$$x_c = z_c \cdot \frac{(x - x_0)(x' - x'_0)}{2 \cdot f}, \tag{10}$$

$$y_c = z_c \cdot \frac{(y - y_0)(y' - y'_0)}{2 \cdot f}. \tag{11}$$

After finding the position relative to a camera P_c it is necessary to find it in the world space using the Oculus Rift's headset rotation matrix R and the translation vector T. These variables are calculated and constantly updated by the Oculus Rift's position capture system.

$$P_w = R^T \cdot P_c + T. \tag{12}$$

When at least 3 marker positions of an object have been detected the object's pose can be calculated using cross product between these positions. Pose of an object can be described with one or more normal vectors of a surface.

$$N_1 = (P_2 - P_1) \times (P_3 - P_1). \tag{13}$$

After finding the surface normal vector, it can be used to calculate Euler angles or the rotation matrix. Finally, to reduce noise in positions and orientations of objects Kalman filter [20] should be applied. In implementation of Kalman filter only positions of markers and their speed in between observations are used. It is used to avoid misidentification of a marker when its position jumps long distance away. Kalman filter ensures smooth performance of the system by reducing impact of such errors.

Results

Initialization of an object's position and orientation in 3D space can be done in less than 300ms using active IR markers. Using traditional fiducial markers, it can be done in less than 100ms. Detection speed of fiducial markers is limited by the processing speed, lighting conditions and camera resolution, but the detection speed of active IR markers is mostly limited by frame-rate of IR cameras. Positions of detected markers can be tracked by using gyroscope and positional tracking data from the Oculus Rift headset. It allows to predict positions of markers in the next frame by following markers and not repeating initialization. Implementation of Kalman filter also improves stability of the system.

From our testing we could see that on average receiving a message takes about 2 times longer than transmitting a message. These are expected results, because transmission is omni-directional without functions to synchronize data transmission. It was also possible to see that the length of identification code is proportional to transmission and reading time. From the length of identification codes, it is possible to calculate the minimal frame-rate of camera needed to achieve the required performance. In order to compare the systems performance to a traditional marker system in augmented reality, commercial XZImg fiducial markers were used. Same camera resolution and dimensions of fiducial markers were used. From the experimental results it was possible to conclude that both systems work stable within limited distance from the camera. Active IR markers were able to work up to 1.5 m from the camera, whereas fiducial markers worked at up to 3m distance. Size of markers layout was about 10x10 cm and camera resolution 640x240 pixels. Itmade pixels to became undetectable in a long distance.

From the test results using error correction codes it was possible to conclude that SEC, SEC-DEC and even Manchester encoding are not necessary for setup used in normal room conditions (see Fig. 9). In a direct line of sight there is no notable noise in the captured images even in different lighting conditions. If noise occurs in most cases, it is filtered by image processing algorithms. Nonetheless, due Leap Motion's irregular frame-rate FEC implementation is necessary.

In order to test SEC we tested the system in a steamy environment to obscure and distort image of active IR markers. We found out that using SEC in such environment achieved the same performance as in normal room conditions (see Fig. 10). At the same time the algorithm corrected 3 times more errors, showing that SEC was working and obstructions were present (see Fig. 11).

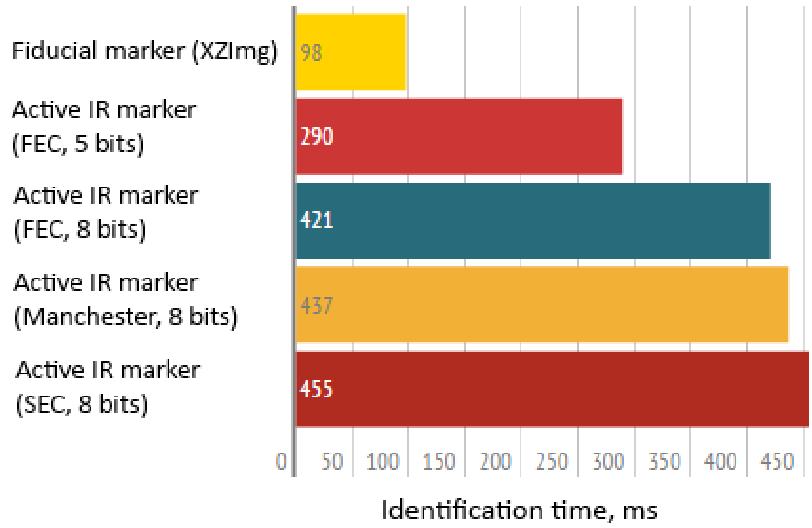

Fig. 9. **Performance of active IR markers with different types of error correction**

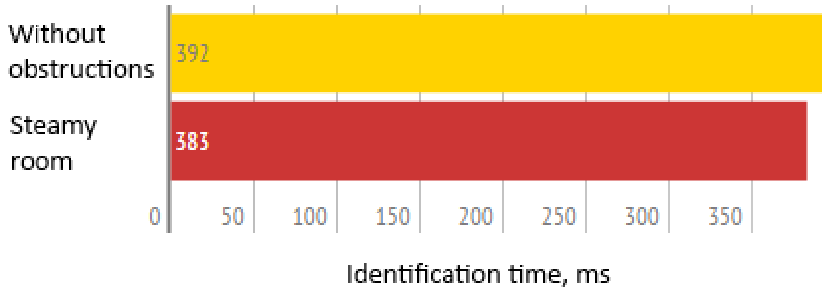

Fig. 10. **Comparison of SEC performance**

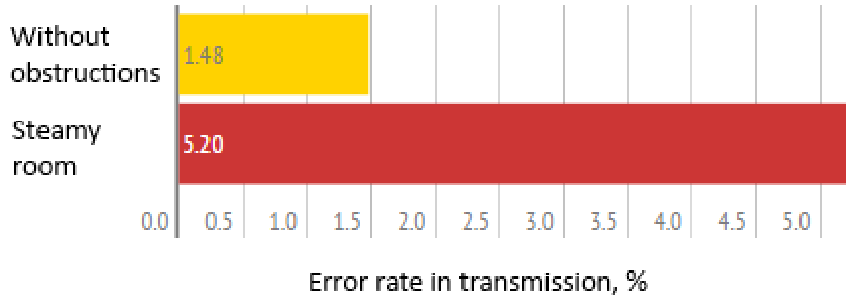

Fig. 11. **Comparison of error correction rate using SEC**

Conclusions

The proposed active IR marker system has been tested successfully. Current implementation due to technical restrictions is able to produce 3 times slower identification of markers than the traditional fiducial marker systems. Possibly this is one of the reasons why the commercial Oculus Touch system that uses a similar system to ours will not be available for Oculus Rift DK2. Oculus Rift DK2 has IR camera with only 30 Hz frame-rate. Most likely the Oculus Rift consumer version will come with higher frame-rate tracking camera. Nonetheless, we found the following advantages in our proposed system:

1. It isinvisible to a user under a cover of an object fitted with markers, but the IR camera could still track it.
2. It can be implemented in any form factor unlike fiducial markers that are rectangular 2D planes.

3. It can transmit meta data like touch events by dynamically changing transmission codes from the transmitter side whereas fiducial markers are static.

From the testing results of data transmission the following conclusions can be drawn:

1. Active IR markers can be used for real-time tracking, but identification time is 3 times slower than for fiducial markers.
2. Identification times of active IR markers aremostly limited by the frame-rate of an IR camera.
3. SEC, SEC-DEC are not necessary when the system is used in normal room conditions.
4. Manchester encoding is not necessary when FEC is used.
5. FEC efficiently solvesirregular frame-rate of Leap Motion camera.
6. SEC efficiently solves data transmission in a presence of temporary visual obstructions.

In a near future there should be stereo cameras with higher frequency and combined IR, and visible light image capture. Such camera like thenext version of Leap Motion Dragonfly could utilize the proposed system. It would use image of visible light in tandem with IR image to create fully immerse and convincing augmented reality experiences using physical objects in 3D space fitted with covert IR marker system.

References

1. Bolas M., Iliff J., Hoberman P., Burba N., McDowall I., Phan T., Luckey P., Krum D.M., Open Virtual Reality, IEEE, 2013, pp. 183-184.
2. Kreylos O., An Eye-tracked Oculus Rift. [online] [01.01.2016] Available at: http://doc-ok.org/?p=1021
3. Follmer S., Leithinger D., Olwal A., Hogge A., Ishii H, inFORM: Dynamic Physical Affordances and Constraints through Shape and Object Actuation, UIST ‘13, 2013, pp. 417-426.
4. Ochiai Y., Hoshi T., Rekimoto J., Three-dimensional Mid-air Acoustic Manipulation by Ultrasonic Phased Array, PLoS ONE 9(7): e10252, 2013.
5. Lee J., Post R., Ishii H., ZeroN: Mid-Air Tangible Interaction Enabled by Computer Controlled Magnetic Levitation,UIST ‘11, 2011, pp. 327-336.
6. BurnettD., Coulton P., Using Infra-Red Beacons as Unobtrusive Markers for Mobile Augmented Reality, MobileHCI 2011, 2011.
7. Klein G., Murray D., Parallel Tracking and Mapping for Small AR Workspaces, ISMAR, 2007
8. Cutler M., Michini B., How J. P., Lightweight Infrared Sensing for Relative Navigation of Quadrotors, Unmanned Aircraft Systems (ICUAS), 2013 International Conference, 2013, pp. 1156-1164.
9. Davison A. J., Reid D. I., Molton D. N., Stasse O., MonoSLAM: Real-Time Single Camera SLAM, Volume: 29, Issue: 6, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, pp. 1052-1067.
10. Kanbara M., Yokoya N., Takemura H., Registration for Stereo Vision-based Augmented Reality Based on Extendible Tracking of Markers and Natural Features, Pattern Recognition, 2002. Proceedings. 16th International Conference, 2002, pp. 1045-1048
11. Gotschlich M., Remote Controls – Radio Frequency or Infrared,Infineon Technologies AG, 2010.
12. Suzuki S., Abe K., Topological Structural Analysis of Digitized Binary Images by Border Following,Computer Vision, Graphics, and Image Processing, vol. 30(1), 1985. pp. 32-46.
13. Zhang L., Chen G., Ye D., Che R., A Fast Center of Mass Estimation Algorithm for Coordinates of IR Markers,Young Computer Scientists, 2008. ICYCS, 2008. pp. 1120-1125.
14. Wu K., Otoo E., Shoshani A., Optimizing Connected Component Labeling Algorithms,SPIE Medican Imaging Conference 2005,LBNL-56864, 2005.
15. Li Y., Me K, Dong P., An Efficient and Low Memory Requirement Algorithm for Extracting Image Component Information,International Journal of Advanced Intelligence, vol. 3(2), 2010. pp. 255-267
16. Oliveira V.M.A., Lotufo R.A, A Study on Connected Components Labeling algorithms using GPUs,SIBGRAPI, 2010.

17. Guna J., Jakus G., Pogačnik M., Tomažič S., Sodnik J., An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking, Sensors 14', 2014. pp. 3702-3720.
18. Warren H.S., ERROR CORRECTING CODES, Hacker's Delight (2nd Edition), Chapter 15, 2012.
19. Souza A.A.S., Maia R., Gonçalves L.M.G, 3D Probabilistic Occupancy Grid to Robotic Mapping with Stereo Vision,ISBN 978-953-51-0660-9, 2012.
20. Wan J., 3D relative position and orientation estimation using Kalman filter for robot control,IEEE, 0-8186-2720-4, 1992. pp. 2638-2645