Simple Gait Parameterization and 3D Animation for Anonymous Visual Monitoring Based on Augmented Reality

Piotr Szczuko

Faculty of Electronics, Telecommunications, and Informatics,
Gdansk University of Technology
szczuko@sound.eti.pg.gda.pl

Abstract. The article presents a method for video anonymization and replacing real human silhouettes with virtual 3D figures rendered on a screen. Video stream is processed to detect and to track objects, whereas anonymization stage employs animating avatars accordingly to behavior of detected persons. Location, movement speed, direction, and person height are taken into account during animation and rendering phases. This approach requires a calibrated camera, and utilizes results of visual object tracking. A procedure for transforming objects visual features and bounding boxes into gait parameters for animated figures is presented. Conclusions and future work perspectives are provided.

Keywords: visual monitoring; gait;privacy; augmented reality; computer animation

1 Introduction

Augmented reality (AR) is a technique of supplementing an input image or video stream with virtual elements, providing a user with additional graphical and textual objects important for particular application. The most popular are: location- and compass- based applications with image recognition, resulting in embedding labels and 3D models onto the image, e.g. describing points of interests in the real world [25][36][49]. Other approach is marker-based, and it requires a presence of a known visual pattern, fiduciary marker, to be held by the user or being located on the object of interest. Upon detection the marker is replaced by a 3D object located, rotated, and scaled accordingly to the marker orientation. Typically, a marker position and rotation estimations are enough for almost seamless presentation of rendered objects on a real background [38]. Advanced applications, e.g. television broadcast enriched with virtual objects, require also a correspondence between lights and shadows in a real environment (in studio) and in 3D virtual space [30]. Author proposes application of AR to a new domain – visual monitoring.

A number of cameras installed in cities is increasing rapidly, rising numerous privacy concerns. In London a person is captured by cameras dozens of times every day, and recordings can be used as an offense to such an individual. Therefore, anonymization methods are being introduced to monitoring systems, including automatic masking of image regions containing personal identifiable information [8]. Blurring, cutting out, and mosaicing are most common techniques, but can hamper understanding of a scene by the observer (e.g. by a monitoring center operator), making it difficult to determine number of persons, type of their activities, etc. Therefore it is proposed here to substitute real images of a monitored person with a virtual articulated figure, mimicking basic human actions: standing, walking, running [15][37]. Such a modification should work in real-time, to allow live observation, and should take into account a number of persons, their exact locations in the image, speed and direction of movement. An effort is described to animate 3D figure in accordance with a real walking and running motion.

2 Augmented reality for anonymous monitoring

For the AR-based anonymous monitoring application it is assumed that a random person can enter camera view and his pose should be recognized and mapped onto an animated figure. Such an avatar displayed on the screen is replacing the real image, but should provide sufficient situational awareness to the observer. In the presented work it is assumed that the location, size, direction and speed should be consistent with the object's features. This approach is similar to a domain of markerless motion capture, involving tracking body parts separately and acquiring accurate body pose and correct spatial orientation. Such task is complex and requires multiple cameras, lengthy optimization, and resolving numerous ambiguities [21].

The most popular markerless method is synthesis and matching [20]. It is based on generating numerous poses of a virtual model (by rendering a 3D image), and then checking correspondence between features of the model and the observation. It takes into account edges, regions, biomechanical constrains, plausibility of the estimated pose, and movement continuity [12][41][42]. State-of-the-art markerless motion capture methods are demanding, and computationally intensive. Survey of those methods can be found in [29]. Due to complexity, currently such an approach cannot be applied in widespread video monitoring.

In the presented work an assumption is made of simplifying the problem and reducing expected pose accuracy to achieve a real-time processing (at least 25 frames per second are expected). Specifically, only person's bounding box coordinates are extracted from the image to derive other motion features using simple biometry (person height, movement type) and estimation of gait features described in the article. Therefore, comparing to the real image, small mistakes in pose, speed, and location are allowed, as long as the virtual figure location matches an area occupied originally by the person. The aim is to provide viewer with anonymized video but still meaningful from the point of view of security. General awareness of number of persons and their global behavior are of interest, and the user should be able to reason about the monitored persons: "are they walking fluently"; "is the movement disturbed"; "are they stopping/slowing down in some particular location"; "where are persons gathering?"

Finally, naturalness and plausibility of the virtual figure motion are assured by using real motion capture recordings. Walking and running actions differ in pace, length, and the time of foot contact with the ground. To take those facts into account recordings of real captured motion were used, including walk and run sequences. On one hand, other actions can be added to such a library once the accurate classification of action type is integrated with the workflow. On the other hand, classifying and parameterizing other actions (bending, sitting, etc.) is a complex task, requiring e.g. optical flow calculation and dedicated classifier [4], or involving implicit expert user input [33]. Survey of such methods can be found in [18].

The purpose of this work is to propose a simpler method for one camera, and straightforward algorithm with a potential to be embedded into a camera. A potential application of this technique is a real-time processing of the video stream inside a camera, and transmitting anonymized video. Originals should be stored in encrypted form, and act as a reference in case of actual threats or for forensics.

3 Method outline

Input video stream is analyzed in real-time to detect areas occupied by moving persons, track their movements over time, and describe main features: location, area, orientation, and speed. Then, based on a camera viewpoint (calibration), and considering a human height (min, max)

and speed limits, a virtual 3D figure is positioned on the real background and animated to mimic the person movements (Fig. 1).

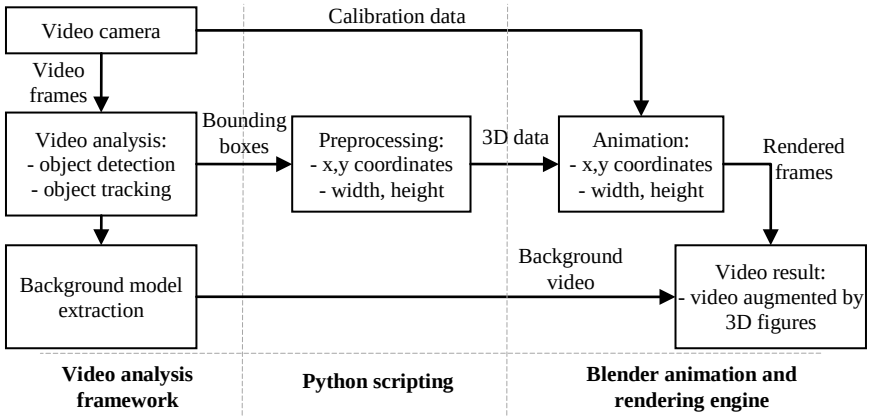


Fig. 1. Data flow, processing blocks and used engines

Video analysis stage is performed by a computer vision framework providing object detection and tracking implementations. This work was published and documented earlier [9][11][22][43] and is out of the scope of this article. For each video frame all bounding boxes parameters, and binary masks denoting objects positions are obtained.

Bounding box size, location, and changes of location in time are used to control global animation of the avatar – its size is based on height of bounding box, its location on current box coordinates, and rotation is based in speed vector. The bounding box speed is used to control local animation: from the velocity a type of gait is derived, influencing movement of limbs (standing, walking, running), and additionally the avatar color changes to better reflect the pace (green for walking, red for running).

For the purpose of this application several assumptions were made:

- Movement is in contact with the ground – bottom side of bounding box lies on the floor, and its *z* coordinate is 0m in 3D space.
- The ground is flat and positioned on *z*=0 plane
- Video camera is calibrated, its parameters are known, and used for positioning and configuration of virtual camera used for rendering. Camera does not introduce fisheye distortion (deformations will be considered in the future work).

In this approach a virtual environment is created, with ground plane on *z*=0 height, with camera orientation corresponding the real one, with uniform, omnidirectional white light used for rendering. For this purpose a common metric and orientation system is established for the real scene and virtual one. The unit is one meter in the real world. Due to the camera calibration, every video image pixel can be translated to *x,y,z* location in meters in the real world [40], and to coordinates in virtual 3D environment.

Described research and implementation utilizes publicly available datasets and additional software, namely:

- video S1-T1-C3 from PETS2006 [34] - publicly available reference video with walking persons (camera calibration data available), 3020 frames, 120 seconds long,
- 3D graphics rendering and animation software, Blender3D [31] – used for virtual figure modeling, walking and running actions preparation, rendering of result augmented video.

- Python programming language interpreter, built into Blender 3D [1] – used for camera and ground plane positioning in 3D space, for translating bounding boxes parameters into figure animation directives.

4 Video analysis

Rendered virtual figures should be positioned, and moved in accordance with persons captured on the video. Therefore, such individuals should be first located, and their movements tracked, to provide input parameters for the animation module. In literature numerous approaches can be found [2][3][14][16][17][39][45]. For example, particle filters are used to efficiently track object of interest [26][13]. A successful attempt in tracking object on a calibrated plane was reported by Roth et. al. [35].

Here, background subtraction exploiting Gaussian Mixtures Model for pixel colors and Kalman tracking are used [11][28].

4.1 Object detection

For the purpose of object detection a background subtraction approach was used [10][44]. Pixel color changes are analyzed over time, and expressed by a statistical Gaussian model of the color. The model is updated with each frame, and could contain several modes taking into account various phenomena: slow changes of light, shadows, and cyclic changes (e.g. waving foliage). Pixel-wise differences between the modeled background and a current frame indicate locations of pixels not belonging to the background, implying foreground objects. Foreground is presented as a binary mask (Fig. 2b), and further processed by morphological operations to remove noise (separated white pixels), smooth silhouettes, and close holes [10]. Unconnected regions are considered as distinct objects, and described by bounding boxes – vertically aligned rectangles circumscribed on the silhouette.

4.2 Object tracking

Tracking of objects is based on Kalman filtering [11] of bounding box parameters: x, y, $height$, $width$, and their changes: Δx, Δy, $\Delta height$, $\Delta width$. Generally, the filter analyses noisy (imprecise) input parameters and estimates correct values, by taking into account a few previous frames and assuming inertia. Kalman filtering results in more fluent changes of locations and size, and helps in resolving objects collisions, partial obscuration, short-term disappearances [11]. Therefore, behavior of bounding boxes obtained in the detection phase is more accurate – the filtered result is called “tracker”.

a)  b)  c) 

Fig. 2. Object detection and tracking process: a) original frame, b) objects’ masks, c) assigned trackers

Ineffective object tracking results in incorrect position and size of the avatar. In this work it is assumed, that tracking results are accurate. The tracking algorithms are developed by many

researchers, and their progress would eventually influence the accuracy of the presented method.

5 Coordinate systems transformation and calibration

Results of the video processing (Sec. 4) are taken as an input to the animation and rendering module, and necessary transformations between image pixels, world coordinates, and 3D space coordinates are performed.

One required transformation is made between coordinates of objects pixels and virtual world 3D coordinates of virtual figures. For this purpose a projective model was applied [27], able to map quadrilaterals to other quadrilaterals. Such projection mimics phenomena of visual perspective, and image acquisition by human eye and video camera [7] (for the purpose of this work it was assumed that fisheye distortions are not introduced). It is described by 3×3 transformation matrix (1).

$$T = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \tag{1}$$

To obtain *u*, *v* coordinates in virtual space respective to given *x*, *y* of a pixel, following calculations are performed:

$$u = \frac{u_p}{w_p} \tag{2}$$

$$v = \frac{v_p}{w_p} \tag{3}$$

where,

$$\begin{gathered} [u_p \quad v_p \quad w_p] = [x \quad y \quad 1] \cdot T \\ u_p = Ax + By + C \\ v_p = Dx + Ey + F \\ w_p = Gx + Hy + I \end{gathered} \tag{4}$$

and:

$$\begin{gathered} u = \frac{Ax + By + C}{Gx + Hy + I} \\ v = \frac{Dx + Ey + F}{Gx + Hy + I} \end{gathered} \tag{5}$$

To determine coefficients of transformation *T*, sample points must be provided, i.e. at least four pairs of coordinates *u*,*v* in meters in real world space, and corresponding *x*,*y* in image pixels. Once matrix *T* is known, inverse matrix can be computed, and transformations in both directions are available.

For the considered scene, 8 pairs of *u*, *v* coordinates were available (Fig. 3c). Altering camera orientation, location or zoom influences transformation coefficients, therefore calibration should be repeated.

Fig. 3. Image and coordinate space transformations: a) original image, b) after transformation from meters to pixels (every 72 pixels represents 1 meter on the ground), c) ground plan and keypoints coordinates

For the purpose of correct rendering a virtual camera must be calibrated as well. Parameters of image distortions, focal length, and position above the ground must be adjusted. A well known method of Tsai calibration can be performed [46]. In case of the used dataset the real camera calibration was available, therefore the virtual camera was set up with regards to provided parameters. This assured a correct perspective and scaling of rendered objects, to match the one introduced by the real camera.

For the purpose of validation of 3D camera calibration four elements of the real scene were recreated as 3D objects, positioned with regards to the floor plan (Fig. 3c), and result *x*,*y* coordinates on the rendered image were compared, confirming correct calibration (Fig. 4a).

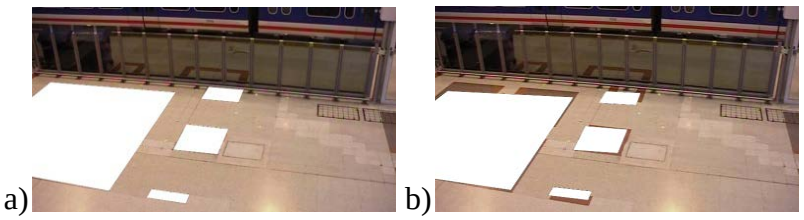


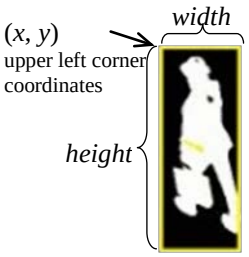
Fig. 4. Recreation of the scene in a virtual environment: a) virtual objects on the ground plane resembling scene elements, correct calibration, b) results of incorrect calibration

6 Bounding box parameterization

Previous video processing stages of object detection, Kalman filtering, object tracking, and coordinates transformation, return:

- object ID (incremented numeral),
- frame number,
- pixel coordinates of upper left corner of the bounding box (*x*,*y*), with assumption that pixel (0,0) is in upper left corner of the image,
- *width* and *height*.

These data are represented as an XML structure, inspired by a common video ground truth description format [47], and are supported by further processing stages (Fig. 5).




```
<object framespan="12:180" id="1" name="Person">
  <attribute name="Name">
    <data:svalue value="P_1"/>
  </attribute>
  <attribute name="Location">
    <data:bbox framespan="12:12" height="49" width="90"
x="547" y="523"/>
    [...]
    <data:bbox framespan="333:333" height="106" width="126"
x="656" y="464"/>
  </attribute>
</object>
[...]
```

Fig. 5. Object bounding box parameterization

From these values other features are calculated and processed by the animation module: speed vector, object location and orientation (Sec. 7.2).

7 Virtual figure animation

To animate a virtual figure in a manner resembling real behavior of a person, several aspects are taken into account:

- changes of location on the scene (*global* movement of the whole body, change of *x*, *y* coordinates)
- changes of orientation to the camera (*global* movement of the whole body, change of rotation along vertical axis)
- changes of limbs position (cyclic *local* movement, independent on global movement, portraying actual action)

First two are obtained by positioning master bone of the virtual figure. The local movement is performed by playing animated movements (stand, walk, run), with a time scale and type of movement adjusted to the real person movement speed.

7.1 Animated figure structure

The purpose of the application is neither to allow discrimination between people nor personality recognition, but to offer anonymized monitoring providing general impression how many people are present on the scene and where, and how fast they are moving. Therefore, a figure used as a substitute to images of real persons, is modeled in 3D as an average-sized humanoid, without any gender, racial, or age features (Fig. 6). The skeleton was created with respect to BVH standard common in character animation and motion capture [6]. It comprises of hierarchically connected bones, influencing geometry of a simple 3D mesh describing model surface. Movement or rotation of a bone located higher in the hierarchy (called parent), changes locations and orientations of lower hierarchy bones (called children). Bone rotations are limited to biologically correct ranges (e.g. elbow angle from 0 to 180 degrees).

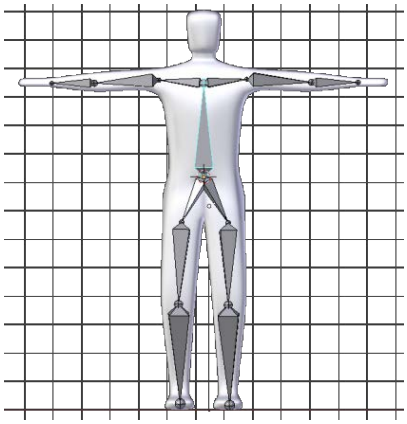


Fig. 6. 3D model and used skeleton

Root bone (pelvis) is a child of (is attached to) a master bone, located on the floor, used only to position and orient the whole model on z=0 plane, matching movement of the real person bounding box. Master bone is also scaled to reflect size of the object. It is assumed that proportions are fixed (no difference between child or adult body).

7.2 Global animation parameters

Object size estimation is not a trivial task, as a new object usually enters the scene at the screen border (depending on scene configuration, one of exceptions is an entrance in the frame center, e.g. a door) and is not totally visible for a few frames. Depending on the camera setup the object could be also partially visible during the total observation period. In such conditions the object parameters are inaccurate, and its size and location cannot be directly processed. Therefore the algorithm deals with all possible cases of object location, partial visibility, and obscurations leading to fragmentation.

7.2.1 Object size estimation for partial visibility

In the first stage the object is partially visible at the screen border. This can last for very short period (going through vertical border, e.g. left edge and quickly to the image center) or for whole observation (going from bottom left to bottom right corner). The object is assumed to be partially out of the screen if the bounding box is in contact with the screen border:

$x=0$	object still at the left side
$x+width=screen\_width$	object still at the right side
$y=screen\_height$	at the lower side
$y-height=0$	at the top side

If at least one of the above is true, then an average heighted (1.7m) virtual animated figure of gray color is presented on the screen, informing the operator, that some person of unknown height and location, and imprecise speed and walking angle has entered the scene. Three such cases are possible (Fig. 7):

- if the object is at the **left** or **right part** of the screen, the avatar is positioned that its legs match the middle point of bounding box base.
- if the object is at the **bottom part** of the screen the avatar is positioned in such a manner, that its head matches the middle point of bounding box top part.
- if the object is at the **top part** of the screen the avatar is positioned that its legs match the middle point of bounding box base

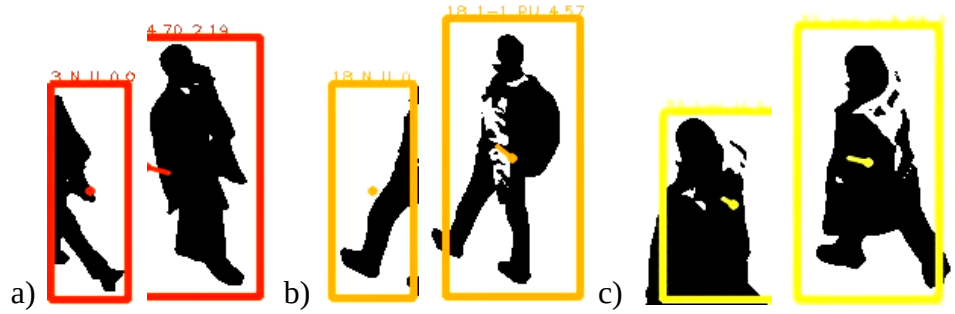


Fig. 7. Cases of partial visibility compared to full visibility. Objects entering through: a) left screen edge, b) right screen edge, c) bottom screen edge. All images are presented in the same scale

Beside partial visibility caused by the screen edges, the person can be obscured by scene elements (handrails, banisters, pillars, chairs, luggage, etc.) or by other persons. In such cases it is assumed that the scene geometry is known (all movement is on the flat ground leveled on z=0), and person *height* is fixed for the whole observation period.

On the other hand the object location is obtained by applying Kalman filtering of bounding box parameters (Sec. 4.2) therefore it takes into account short disturbance of object size caused by obscuration, and yields corrected values.

7.2.2 Object size estimation for full visibility

Once the object is fully visible and none of border contact criteria is true, it is assumed that current *width*, *height*, x and y of bounding box are correct and can be processed further. Moving average of *height* is updated for first 50 frames of full visibility (6), and then is used as a result (Fig. 8). The final *height* influences scale of the virtual figure, and its color is set to green/red (depending on movement speed – Sec. 7.3.1).

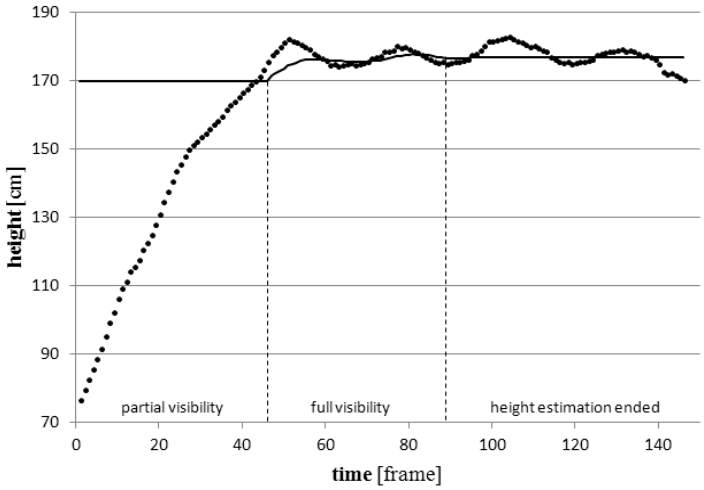


Fig. 8. Estimated height of a person (dotted – bounding box height, line – estimation): 1) initially partial visibility, average height of 1.7m; 2) height updated for 50 frames, using moving average of last 5 samples; 3) estimation completed with a result of 1.77m

$$height_{target}(t)=0.2\cdot[height(t\text{-}4)+height(t\text{-}3)+height(t\text{-}2)+height(t\text{-}1)+height(t)] \quad (6)$$

7.2.3 Movement features processing

For each detected object the sequence of bounding box locations and sizes is processed. First a transformation to real world coordinates is made (3), influencing values of *height*, *width*, *x,* and *y*. For the purpose of clarity it is assumed that all following calculations are in transformed coordinates, i.e. meters in the real world. Due to applied Kalman filtering parameters are considered to be noise-free, and smoothed in time.

Current object description is processed to determine movement *speed* (in meters per one video frame), *direction*, and object *location* (middle point of bounding box base):

$$\begin{aligned} V_x(t) &= \big(x(t) - x(t-1)\big) \\ V_y(t) &= \big(y(t) - y(t-1)\big) \\ V(t) &= \sqrt{V_x^2(t) + V_y^2(t)} \end{aligned} \tag{6}$$

$$\theta(t) = \begin{cases} acos\left(\frac{V_x(t)}{V(t)}\right) \cdot \frac{180}{\pi} \cdot sign\left(V_y(t)\right) & for V_y(t) \neq 0 \\ 0 & for V_y(t) = 0 \;\; and V_x(t) \geq 0 \\ 180 & for V_y(t) = 0 \;\; and V_x(t) < 0 \end{cases} \tag{7}$$

$$\begin{aligned} x_{loc}(t) &= x(t) + \frac{width}{2} \\ y_{loc}(t) &= y(t) + height \end{aligned} \tag{8}$$

7.2.4 Global animation implementation

A module called Non-Linear Animation is available in Blender3D, providing tools for combining and adjusting animations in a controlled manner, scriptable by Python language [1].

Global movement of the virtual figure on the scene is governed by current values of parameters *x*, *y*, *height*, and θ. Created script for current frame reads those inputs and sets the master bone on the floor (z=0) at coordinates of *x*, *y*, with its bone local axis *x* pointing to angle θ, axis *y* pointing upwards, and the master bone size equal to *height* expressed in meters (influencing scale of all bones and, as a result, size of 3D mesh).

7.3 Local animation parameters

Local animation parameters such as changing a color of the avatar, moving its limbs with appropriate speed and in a proper way reflecting type of gait are controlled independent of the global animation.

7.3.1 Color adjustment

It is assumed that the color of the model texture reflects measured speed:

- gray objects – standing still or just entering the scene (object of undefined size – Sec. 7.2.1),
- green objects –casual walk with a speed of 1.2m/s
- light green objects – typical brisk walk of 5km/s = 1.39m/s
- red objects – persons running with a speed up to 3m/s.

For easier color adjustment a hue, saturation, value (Hue, Sat, Val) color space is used (Fig. 9). Following calculations are used:

Val=120	fixed value for all speeds
Sat(V(t))=0	for speed=0, gray color, for standing
Sat(V(t))=159+27·V(t)	for speed∈(0, 3m/s) for walk to run
Hue(V(t))=81–27·V(t)	for speed∈<0; 3m/s) for stand to run

This result in continuous changes of the color along a trajectory: from light green to bright red (Fig. 9b). Light red (with high saturation) is used to demonstrate an alertness, and abnormal state. Saturation of green is lower; therefore walking avatars do not stand out in the image, comparing to light green of full saturation.

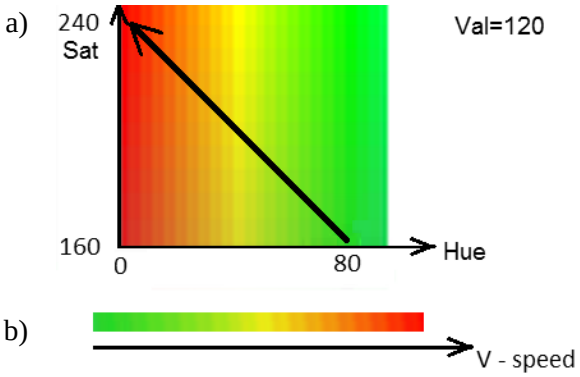


Fig. 9. Virtual figure color: a) color trajectory in HS space for fixed V=120, b) result colors (composition of given HSV) for increasing speed values

Clear presentation of color changes is assured by the tracker movement speed processed by Kalman filter, i.e. V(t) values are smoothed in time. Finally, the approach results in dim colors (medium saturation and value are used to not induce eyestrain), and for sequences of walking, standing, and running these color changes slowly, to give impression of continuity of motion and identity of the virtual figure.

7.3.2 Local movement animations

Three actions were prepared by importing and editing BVH motion capture recording to adapt it to looped playback.

Walk cycle. Animation cycle for walking human was created by editing one of a freely available BVH recordings (www.cgspeed.com). The cycle is 25 frames long, resulting in 1 second of animation. The average gait length (two steps) observed on video recordings is ca. 1.35m per cycle [5][32][48]. The stride can be adjusted during animation by setting *action playback scale* parameter. It was observed that in detail:

- 1.60m tall person walk cycle is 22-24 frames (880ms-960ms), gait length is ca. 1.25m,
- 1.75m tall person walk cycle is 27-29 frames (1080ms-1160ms), gait length is 1.45m.

Therefore a source animation was made for average avatar:

- 1.7m tall avatar, walk cycle 25 frames long (1000ms), gait length is 1.35m.

To achieve target gaits for other avatar height a linear regression is used (Fig. 10).

Run cycle. Running cycle was also created by editing motion capture file containing natural running action, and then applying a time scaling, slowing down the action to last 25 frames. Result source walk and run are synchronized, while played with a scale 1.0. It was observed that:

- short person running cycle is ca. 16 frames (640ms)
- tall person running cycle is 18-20 frames (720ms-800ms).

In the source animation for average avatar a run cycle is therefore 25 frames long, with a stride 1.40m.

Actions blending. While the avatar movement speed increases (descreases), the action playback scale decreases (increases, respectively), resulting in speeding up the walk, elongating the gait, and gradually changing the walk into a run. Due to same lengths of run and walk cycles, these two actions for any *playback scale* are always synchronized, facilitating combining these together. Fading of one action into other is made by setting an *influence* variable in Blender's NLA (Fig. 11).

Standing animation. For movement vector V<0.5m/s the standing animation is applied. It is also a cyclic motion, with upper body subtle movements (head rotating sideways, arms waving, center of gravity swaying). Length of the cycle is equal to 25 frames as well, synchronized with a walk and run cycles. Blending a walk into a stand is performed by setting the playback scale to 2.7 (average walk of 1.35m/s is slowed down to 0.5m/s). As V is very low, the orientation angle is imprecise, therefore it is assumed to remain unchanged from walk/run phases with distinct movement direction.

Table 1. Summary of action playback scales and cycle lengths for standing, walking and running actions for three person heights

person height [m]	1,6		
	speed [m/s]	scale	cycle length [ms]
stand	1.25m/2.7s = 0.46	2,7	2700
walk	1,25	1	1000
run	1.25m/0.7s = 1.78	0,7	700
person height [m]	1,7		
	speed [m/s]	scale	cycle length [ms]
stand	1.35m/2.7s = 0.50	2,7	2700
walk	1,35	1	1000
run	1.35m/0.7s = 1.93	0,7	700
person height [m]	1,75		
	speed [m/s]	scale	cycle length [ms]
stand	1.45m/2.7s = 0.54	2,7	2700
walk	1,45	1	1000
run	1.45m/0.7s = 2.07	0,7	700

Summarizing, to generate local movement (limbs rotations), the script modifies variables influencing movement style:

1) `action playback scale` (time stretch) changes speed of local movement of limbs,
2) degree of `influence` of the action on the character, is used to fade one action into other, i.e. smoothly blend actions and change walking into running or walking into standing still and back. It uses weighted average between available action recordings.

These variables are controlled by values of V(t) (current movement speed in m/s in real world units)(Fig. 11).

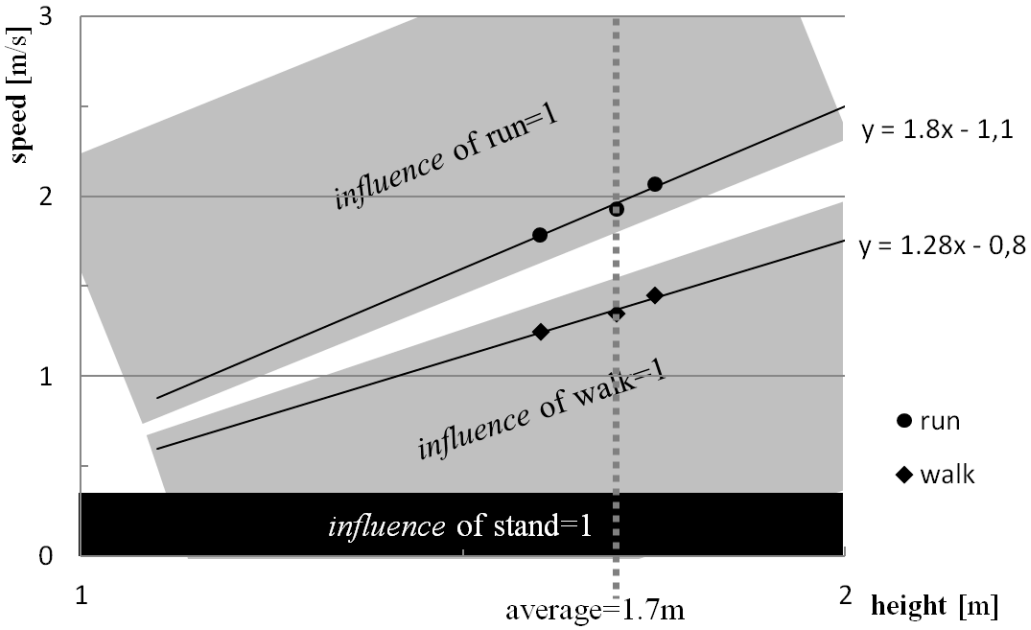


Fig. 10. Overview of dependence of person height and speed for walking and running actions. Linear regression is used to hypothesize for other heights. Regions of applying particular animation type with full *influence* are marked.

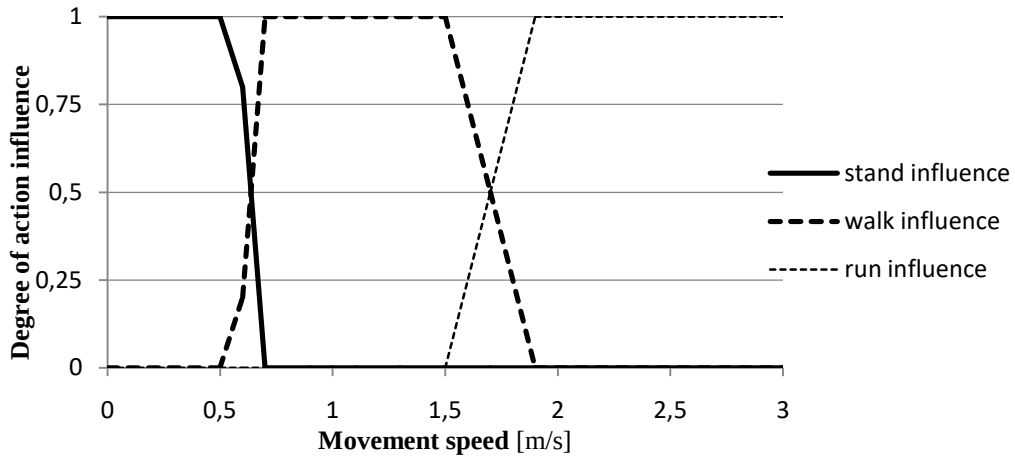


Fig. 11. *Influence* for given speed for average person height=1.7m (marked as dotted vertical line on Fig. 10)

It is assumed that movement speed accelerates/decelerates slowly (as a result of applied Kalman filtering), therefore changes of aforementioned parameters can be made on the fly without decreasing animation quality.

Current approach provides animated actions that are not synchronized with the actual motion, yet align with the location, speed and orientation of the real person. The future work will focus on detecting the motion phases and synchronizing the animation accordingly.

8 Output video generation

Calibrated virtual camera is used to render an animated figure on a transparent background. Result images are embedded on video frames of the background without moving objects (Fig. 12). In case of a constantly changing background a current image is obtained from Gaussian model – as current background is updated in the model with every frame.

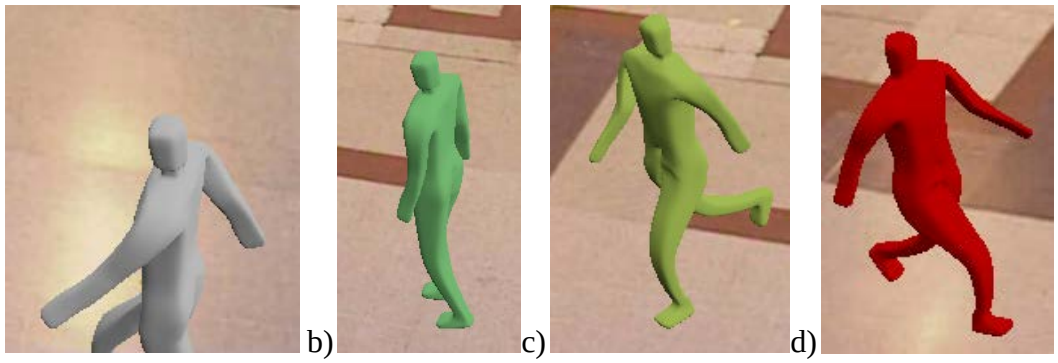


Fig. 12. Frames from result sequence: a) person entering the area; b) walking person; c) brisk walk; d) running person

The described approach is capable of working in real time, as the video processing of PAL resolution takes ca. 70% of one CPU core clocked at 1.8GHz, and rendering 1–4 silhouettes (no anti-aliasing, one directional light) takes 0.05-0.2s. The result time would be shorter once the rendering is performed on a GPU instead of CPU, or pre-rendered silhouettes library is used instead (images of several intermediate motion types and phases seen from various orientation read from memory).

9 Results analysis

By comparing binary masks of the person and the substituting avatar (Fig. 13), known objective metrics can be formulated. The number of correctly matched and mismatched pixels and the objects areas are taken into account:

True Positives is the number of correctly matched pixels of both masks:

$$TP=\| mask_{\text{person}} \wedge mask_{\text{avatar}} \| \tag{9}$$

False Positives is the number of avatar pixels not matching the person:

$$FP=\| mask_{\text{avatar}} - mask_{\text{person}} \| \tag{10}$$

False Negatives is the number of person pixels not matched by the avatar:

$$FN=\| mask_{\text{person}} - mask_{\text{avatar}} \| \tag{11}$$

Positives is the total number of person pixels expected to be matched by the avatar:

$$P=\| mask_{\text{person}} \| \tag{12}$$

Negatives is the area outside the person, that should not be matched by the avatar silhouette:

$$N=width \times height - \| mask_{\text{person}} \| \tag{13}$$

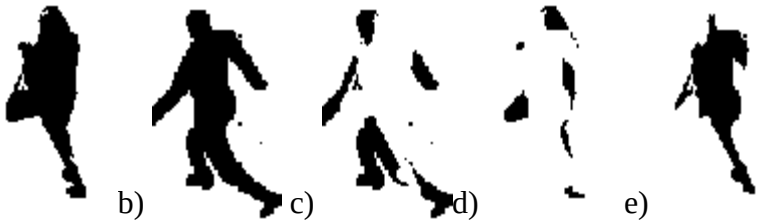
Recall or *True Positive Ratio* is a metric expressing ratio of *true positives* to all *positives*:

$$TPR=TP/P = \|(mask_{\text{person}} \wedge mask_{\text{avatar}})\| / \| mask_{\text{person}} \| \tag{14}$$

Precision or *Positive Prediction Value* is a metric expressing ratio of *True Positives* in all hypothetical positives of avatar silhouette:

$$PPV = TP/(TP+FP) = \|(mask_{\text{person}} \wedge mask_{\text{avatar}})\| / \| \text{mask}_{\text{avatar}}\| \tag{15}$$

where: *mask* – binary matrix of pixels
$\| \cdot \|$ – counting number of non-zero values
$\wedge$ – pixel-wise logical conjunction



a) b) c) d) e)

Fig. 13. Sample masks: a) true positives, a) true positives+false positives, c) false positives, d) false negatives, e) true positives. Obtained values indicate acceptable match: *TPR*=0.75, *PPV*=0.5282

For perfect coverage of the person silhouette by the avatar silhouette the *TPR*=1. Assuming the sizes (numbers of pixels) of both silhouettes are the same, *PPV*=*TPR*=0 indicates no coverage, and *PPV*=*TPR*=0.5 is half coverage.

The *PPV* and *TPR* can be calculated separately for each person, however this can result in omitting cases of improperly rendered avatars not assigned to any real object. Therefore it is preferred to take into account all persons and avatars in total, by calculating binary masks over the whole image.

For the *TPR* combined for all objects over the whole sequence the mean of TPR_{mean}=0.5161 was acquired, TPR_{max}=0.9512, and 44% of frames with TPR larger than the TPR_{mean}, and 88% of frames with TPR>=0.3 (Fig. 14). No case of total misplacement was detected. The asynchronous movements of limbs is reflected by TPR oscillating in range ca. ±0.1. For the *PPV* combined for all objects over the whole sequence the mean of PPV_{mean}=0.5235 was acquired, PPV_{max}=0.8747, and 62% of frames with *PPV* larger than the PPV_{mean}, and 88.7% of frames with *PPV*>=0.3 (Fig. 15). No case of total misplacement was detected. The asynchronous movements of limbs is reflected by *PPV* oscillating in range ca. ±0.1.

In the anonimized videos cases of *TPR*<<1 and *PPV*<<1 usually indicated lack of proportionality between person and avatar sizes, occurring mainly when a person enters and exits the camera view.

The *TPR* and *PPV* would increase for avatar animation synchronized with the person limbs movement, what is the next step in the presented work.

Other objective metrics were considered, such as movement direction and speed errors, and size error. These are not implemented, because their results depend only on the performance of object tracking, which is developed independently of this work.

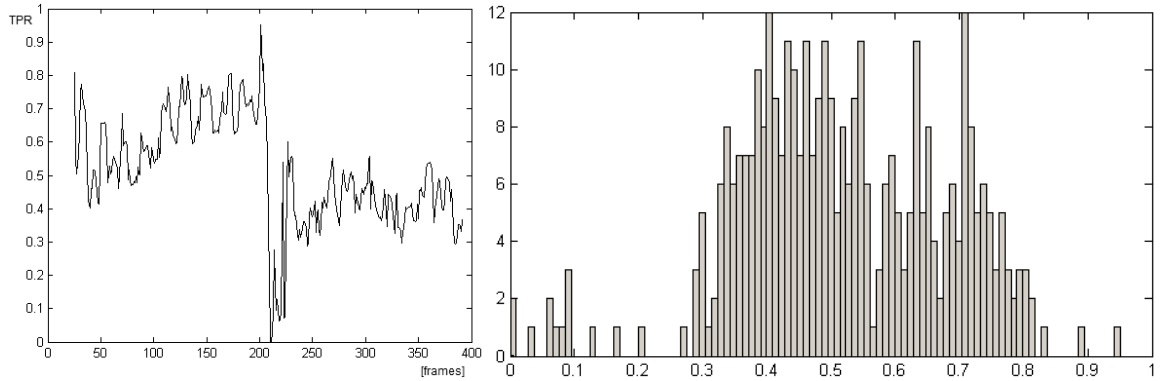


Fig. 14. Sample sequence metrics: a) True Positive Ratio, b) *TPR* values histogram

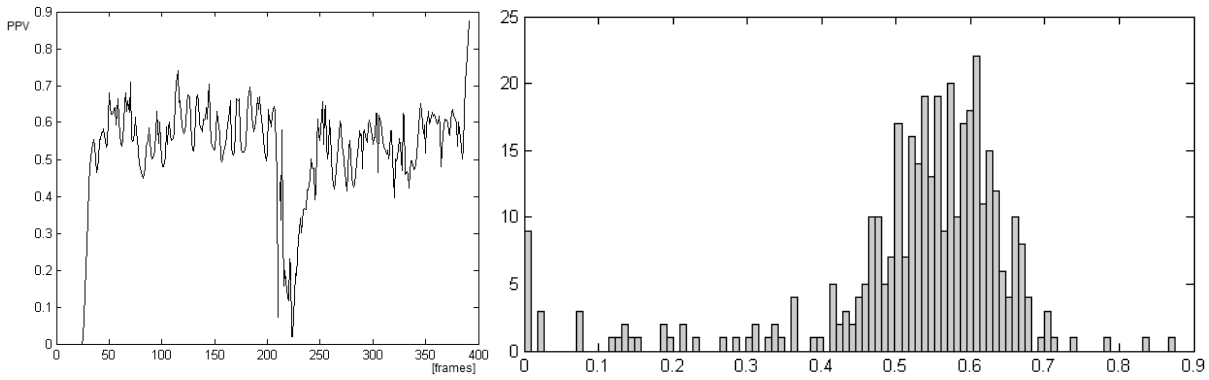


Fig. 15. Sample sequence metrics: a) Positive Prediction Value, b) *PPV* values histogram

10 Conclusions and future considerations

Presented work is a proof-of-concept of new method for video anonymization: substituting person image by a virtual 3D figure performing actions similar to the real one. Results underwent an initial assessment, involving intelligibility of the video [19], and objective measurements. Actions are recognized correctly by the viewer, and anonymized video provides general situational awareness. Used technologies for video analysis, and 3D image generation were adapted to suit the general workflow (Fig. 1). Data export from object tracking phase, importing, and parsing, coordinate systems transformation, and methods for animation parameters inference were created for the purpose of presented work. The dependency between person height, speed and movement type was explored and a definition of such relations was obtained.

Because of omission of **other objects**, such as luggage or trolleys the video lacks detail, and exact meaning is not conveyed. Regardless, the main goal of giving the information of number of people, their location, and movement, is fulfilled.

Currently the **scene geometry** is assumed to be a simple plane, located at z=0, with real points measured at the camera initialization. The camera is expected to be fixed, to allow object detection and tracking methods. The object tracking is based on background modelling, thus cannot handle global scene changes (result of panning the camera). If other approach is used to process moving camera videos and perform accurate object tracking [24], it can be integrated in the presented workflow, after adapting the output format according to Sec. 6.

Other actions can be added once the correct classification of action type is integrated with the workflow. Generally classification of actions (bending, sitting, etc.) is a complex task, requiring multiple cameras, and powerful processing units, and cannot be easy achieved in realtime. Main purpose of this work is to propose a computationally simple method for one camera, and straightforward algorithm with a potential to be embedded into the camera.
It is assumed, that tasks of effective object detection, tracking, conflicts solving, separation of the object from a group, choosing the right type of the object (person only) should be solved independently. This work deals with problems and issues of the last phase - presentation of clear and readable anonymized video, accurate enough to provide general situational awareness to the viewer (sample video can be viewed at http://youtu.be/Upy08IQzD0g).

In the future work it is intended to introduce generic 3D models for inanimate objects: e.g. represented as a 3D cuboid proxy. Moreover, combining the anonymization with shape **classification** would result in processing only human silhouettes. In such a case original images of trolleys, carts, luggage, bags will be still present in the image, assuring higher intelligibility of anonymized video.

Presented approach can be extended with an automatic **event detection**. Events could be defined as a presence of person/avatar in defined restricted area, or crossing defined edge in

prohibited direction (counter flow), as well as shouting or screaming [3][23][28][39]. In such a case the avatar can be visually marked by blinking bounding box, or distinct color. Moreover it is assumed, that while the operator is provided only with the anonymized video, the original one is securely stored in an encrypted form, and in case of confirmed threat can be streamed on demand to a high level authorized officer.

Acknowledgement

This work has been partially funded by the ARTEMIS Joint Undertaking and the Polish National Centre of Research and Development as a part of the COPCAMS project (http://copcams.eu) under Grant Agreement No. 332913.

Bibliography

[1] Anders MJ., Blender 2.49 Scripting, Packt Publishing, 2010.
[2] Atrey PK., El Saddik A., Kankanhalli MS., Effective multimedia surveillance using a human-centric approach. Multimedia Tools and Applications, Vol. 51, Issue 2, pp 697-721, Springer, 2011.
[3] Ballan L., Bertini M., Del Bimbo A., Seidenari L., Serra G., Event detection and recognition for semantic annotation of video. Multimedia Tools and Applications, Vol. 51, Issue 1, pp 279-302, Springer, 2011.
[4] Benmokhtar R., Robust human action recognition scheme based on high-level feature fusion, Multimedia Tools Applications, Vol. 69, Issue 2, 253–275, Springer, 2014.
[5] Bratt B., Rotoscoping. Focal Press, 2012.
[6] Biovision Hierarchy, http://en.wikipedia.org/wiki/Biovision_Hierarchy
[7] Cederberg JN., Projective Geometry. A Course in Modern Geometries, Undergraduate Texts in Mathematics, 213-313, Springer, 2001.
[8] Cichowski, J.; Czyzewski, A. Reversible video stream anonymization for video surveillance systems based on pixels relocation and watermarking. Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference. 1971-1977, 2011.
[9] Czyżewski A., Szwoch G., Dalka P., Szczuko P., Ciarkowski A., Ellwart D., Merta T., Łopatka K., Kulasek Ł., Wolski J., Multi-stage video analysis framework. (Ed. Weiyao Lin) Video Surveillance, Chapter 9, 145-171, Intech, 2011.
[10] Dalka P., Detection and Segmentation of Moving Vehicles and Trains Using Gaussian Mixtures, Shadow Detection and Morphological Processing. Machine Graphics and Vision, Vol. 15, No. 3/4, 339 – 348, 2006.
[11] Dalka P., Szwoch G., Szczuko P., Czyżewski A., Video Content Analysis in the Urban Area Telemonitoring System. G.A. Tsihrintzis et al. (Eds.): Multimedia Services in Inteligent Environments, 241-261, Springer-Verlag Berlin Heidelberg, 2010.
[12] Deutscher, J., Blake, A., Reid, I.D.: Articulated body motion capture by annealed particle filtering. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 126–133, 2000.
[13] Gao T., Li G., Lian S., Zhang J., Tracking video objects with feature points based particle filtering. Multimedia Tools and Applications, Volume 58, Issue 1, pp 1-21, Springer, 2012.
[14] Ghazal M., Vázquez C., Amer A., Real-time vandalism detection by monitoring object activities. Multimedia Tools and Applications, Vol. 58, Issue 3, pp 585-611, Springer, 2012.
[15] Goffredo M., Bouchrika I., Carter JN., Nixon MS., Performance analysis for automated gait extraction and recognition in multi-camera surveillance. Multimedia Tools and Applications, Vol. 50, Issue 1, pp 75-94, Springer, 2010.

[16] Guo C., Liu D., Guo Y., Sun Y., An adaptive graph cut algorithm for video moving objects detection. Multimedia Tools and Applications, Volume 72, Issue 3, pp 2633-2652, Springer, 2014.
[17] Höferlin B., Höferlin M., Weiskopf D., Heidemann G., Information-based adaptive fast-forward for visual surveillance. Multimedia Tools and Applications, Vol. 55, Issue 1, pp 127-150, Springer, 2011.
[18] Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern 34:334–352
[19] ITU-T recommendation P.800: Methods for subjective determination of transmission quality (http://www.itu.int/rec/T-REC-P.800-199608-I/en), 1996.
[20] Kakadiaris, I., Metaxas, D.: Model-based estimation of 3D human motion. IEEE Tran. Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, 1453–1459, 2000.
[21] Kehl, R., Van Gool, L.: Markerless tracking of complex human motions from multiple views. Computer Vision and Image Understanding Vol. 104, No. 2-3, 190–209, 2006.
[22] Kotus J., Dalka P., Szczodrak M., Szwoch G., Szczuko P., Czyżewski A., Multimodal Surveillance Based Personal Protection System. Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 100-105, Poznan, 2013.
[23] Kotus J., Łopatka K., Czyżewski A., Detection and localization of selected acoustic events in acoustic field for smart surveillance applications. Multimedia Tools and Applications, Vol. 68, Issue 1, 5-21, Springer, 2014.
[24] Kriechbaum A., Mörzinger R., Thallinger G., A framework for unsupervised mesh based segmentation of moving objects. Multimedia Tools and Applications, Vol. 50, 7–28, Springer, 2010.
[25] Krolewski J., Gawrysiak P., The Mobile Personal Augmented Reality Navigation System. Man-Machine Interactions Vol. 2, Springer, 2011.
[26] Lalos C., Voulodimos A., Doulamis A., Varvarigou T, Efficient tracking using a robust motion estimation technique, Multimedia Tools Applications, Vol. 69, Issue 2, 277–292, Springer, 2014.
[27] Laveau S., Faugeras O., Oriented projective geometry for computer vision. Computer Vision ECCV, Lecture Notes in Computer Science Vol. 1064, 147-156, Springer 1996.
[28] Lavee G., Khan L., Thuraisingham B., A framework for a video analysis tool for suspicious event detection. Multimedia Tools and Applications, Vol. 35, Issue 1, pp 109-123, Springer, 2007.
[29] Moeslund T.B., Hilton A, Kruger V (2006) A survey of advances in vision-based human motion capture and analysis. Comput Vis Image Underst 104:90–126
[30] Moshkovitz M., The Virtual Studio: Technology and Techniques. Focal Press, 2000.
[31] Mullen T., Mastering Blender, Sybex, 2012.
[32] Novaes RD., Dourado VZ. Usual gait speed assessment in middle-aged and elderly Brazilian subjects. Brazilian Journal of Physical Therapy, Vol.15, n.2, p.117-122, 2011. doi: http://dx.doi.org/10.1590/S1413-35552011000200006
[33] Ntalianis K-S, Doulamis A-D, Tsapatsoulis N, Doulamis N, Human action annotation, modeling and analysis based on implicit user interaction. Multimedia Tools Applications, Vol. 50, 199–225, Springer, 2010.
[34] PETS 2006 Bemchmark Data, IEEE Conference on Computer Vision and Pattern Recognition 2006, www.cvg.rdg.ac.uk/PETS2006/data.html, 2006.
[35] Roth R., Koller-Meier E., Van Gool L., Multi-object tracking evaluated on sparse events. Multimedia Tools and Applications, Vol. 50, 29–47, Springer, 2010.
[36] Rumiński D., Walczak K., Creation of Interactive AR Content on Mobile Devices. Business Information Systems Workshops, Springer, 2013.

[37] Samangooei S., Nixon MS., Performing content-based retrieval of humans using gait biometrics. Multimedia Tools and Applications, Volume 49, Issue 1, pp 195-212, Springer, 2010.
[38] Schreer O., Kauff P., Sikora T. (Eds), 3D Videocommunication: Algorithms, concepts and real-time systems in human centred communication. Wiley, 2005.
[39] Simon C., Meessen J., De Vleeschouwer Ch., Visual event recognition using decision trees. Multimedia Tools and Applications, Vol. 50, Issue 1, pp 95-121, Springer, 2010.
[40] Szwoch G., Dalka P., Czyżewski A., Spatial Calibration of a Dual PTZ-Fixed Camera System for Tracking Moving Objects in Video. Journal of Imaging Science and Technology (JIST), Vol. 57, No. 2, 1-10, 2013.
[41] Szczuko P., Hierarchical Estimation of Human Upper Body Based on 2D Observation Utilizing Evolutionary Programming and "Genetic Memory". Multimedia Communications, Services and Security, Communications in Computer and Information Science, Vol. 149, 82-90, Springer, 2011.
[42] Szczuko P., Genetic programming extension to APF-based monocular human body pose estimation. Multimedia Tools and Applications, Vol. 68, 177-192, Springer, 2014.
[43] Szwoch G., Dalka P., Ciarkowski A., Szczuko P., Czyzewski A., Visual Object Tracking System Employing Fixed and PTZ Cameras. Journal of Intelligent Decision Technologies, Vol. 5, No. 2, 177 – 188, 2011. http://iospress.metapress.com/content/m5060n24tk125406/?p=2aa903da834b4371955e56c56b058b6b&pi=5
[44] Szwoch G., Dalka P., Layered background modeling for automatic detection of unattended objects in camera images. WIAMIS 2011: 12th International Workshop on Image Analysis for Multimedia Interactive Services, Preprint No. 50, Delft 2011.
[45] Tavli B., Bicakci K., Zilan R., Barcelo-Ordinas JM., A survey of visual sensor network platforms. Multimedia Tools and Applications, Vol. 60, Issue 3, pp 689-726, Springer, 2012.
[46] Tsai RY, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation, Vol. 3 No. 4, 323–344, 1987.
[47] University of Maryland, Guide to Authoring Media Ground Truth with ViPER-GT, http://viper-toolkit.sourceforge.net/docs/gt/
[48] Uustal H., Baerga E., Gait Analysis. In: (Ed: Sara Cuccurullo) Physical Medicine and Rehabilitation Board Review. Demos Medical Publishing, New York, 2004. http://www.ncbi.nlm.nih.gov/books/NBK27235/
[49] Wikitude, augmented reality platform, http://www.wikitude.com/