BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 61, No. 4, 2013
DOI: 10.2478/bpasts-2013-0094

Camera navigation support in a virtual environment

A. WOJCIECHOWSKI*

Institute of Information Technology, Lodz University of Technology, 215 Wólczańska St., 90-924 Łódź, Poland

Abstract. Artificial camera navigation is the most ubiquitous and principal interaction task within a virtual environment. Efficient and intuitive virtual scene navigation affects other tasks completion performance. Though many scientists have elaborated invaluable navigation techniques design guidelines, it is still, especially for novice users, the really challenging and demanding process. The user interface input hardware imprecision, interface operation cognitive burden put on users and deficiency of direct mapping between user physical movement and virtual camera navigation evoke discrepancy between user desired and actual camera position and orientation. The provided paper concentrates on the new potential field based camera navigation support method. Originally designed and exploited potential fields support not only collisions solving, but goal profiled attraction and camera manoeuvring as well. It works both in static and dynamic environments. It can be easily boosted by the GPU approach and eventually can be easily adapted for advanced or novice interface users for a miscellaneous navigation task completion.

Key words: virtual environment, camera navigation, potential field.

1. Introduction

Camera navigation within a virtual environment is a fundamental task implying satisfactory exploration of 3D scenes, where the virtual camera represents an observer (virtual character with a camera assigned to invisible body head at height of its eyes – h_c) who operates it by means of user interface. Operating the camera is understood as its position and orientation modification, with respect to the virtual body bounding volume (cylinder of height h_c and bottom radius R_1 – Fig. 1a). High quality navigation technique should let the user control the camera in an intuitive manner, with a proper camera-environment collisions solution and contextual point of view attraction when it is consistent with user's intentions [1].

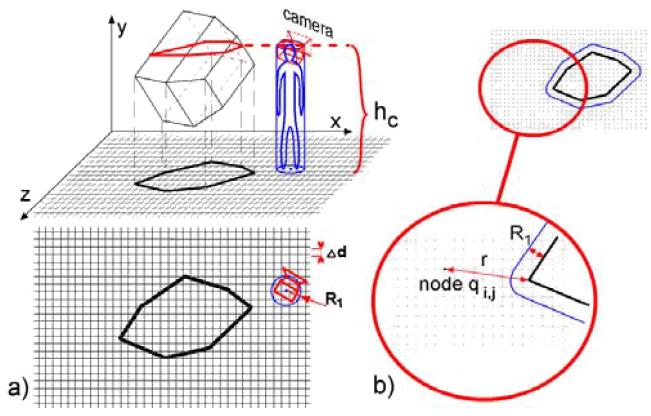

Fig. 1. a) Construction of an object reference contour for an actual camera height h_c b) oversized contour replacing camera margin R_1

Virtual environment exploration has usually no direct correspondence to relevant activities performed in a real world. By default navigational intentions of the user are transferred into an application by means of user input interface. The most efficient in this field seems to be direct communication between a virtual camera and a human brain, however it still becomes a developing solution with a considerable time lag [2]. More contemporary solutions exploit mouse, keyboard, data gloves, motion tracking systems, voice commands or vision-based hand gestures [3–9]. Eventual camera movements become a trade-off between user's intentions, interface equipment instability and navigation technique exploitation difficulties. Resulting camera transformations reveal frequently unintended discrepancy between user desired and actual camera position and orientation. Especially immersive hardware interfaces, powered by head mounted displays and cyber gloves put considerable burden on users while controlling the camera. For novice, inexperienced users, even desktop interfaces (mouse and keyboard) can cause considerable problems as performed action may not reflect expected camera behaviour.

It must be noticed that none of listed interface gadgets was especially dedicated for 3D scene navigation. Thus authors tend to compensate encountered difficulties twofold: correcting technical equipment instability and misinterpretation, or aiding user navigation technique improving movement efficiency and precision. Within first approach hardware retrieved signals are often tuned and filtered [10, 11] for more efficient camera operation. Usually appropriate hardware interpretation depends on artificial intelligence techniques like neural networks and expert systems or intelligent data extraction process [12–17]. Nevertheless the second approach, regarding navigation technique support, is discussed in this paper. This approach is the most crucial for user satisfaction while 3D scene exploration. The correspondence between users' intentions and their efforts fulfilment determines how they perceive navigation process efficiency.

*e-mail: adam.wojciechowski@p.lodz.pl

A new method of the camera navigation support, within information rich dynamic environment, is presented in the paper. The method exploits local potential field gradients for aiding camera control, allowing user to gather data in efficient way. Aiding virtual camera movements encompasses attracting camera towards local goals with respect to object-camera collisions and adjusting its view direction while approaching the aim. Though collisions solving for both static and dynamic scenes were presented in [18-20], the paper orders and summarizes all author's potential field theory components in a context of popular navigation tasks completion (from unconstrained up to fully controlled navigation). The quality of the method was measured by means of formal tests where users were asked to gather specified set of data from the spatial dynamic environment within possibly short time. Additionally paper provides general suggestions for potential field based, camera navigation method profiling, retrieved from extensive users' questionnaires.

2. Related works

According to Mine [21] and Bowman [22, 23] navigation is a key aspect of interaction. Camera navigation can be treated as a tool to achieve different temporal or permanent goals necessary for further object selection and manipulation. Darken [24] has suggested dividing navigation process into two stages. *Wayfinding* resulting in localization of a goal (both visually and mentally) and *travelling* understood as a motoric part of the movement. Simultaneously Bowman [1] has presented another motivation-based approach dividing the navigation process into:

- *exploration* – navigation when the goal is not specified,
- *search* – navigation with searching for the goal,
- *manoeuvring* – slight movements correcting the camera position and orientation.

In the context of provided navigation taxonomies presented method supports mainly dynamic aspects of camera movement aiding local *searching* and *manoeuvring* process. Navigation dynamic is understood as a virtual camera movement in certain directions or towards specified goal. Authors [25–29] have presented three main groups of methods for camera motion planning, assuming initial environment configuration with specified goal and obstacles:

- *roadmap* – where the path of movement is constructed from simple curves, omitting obstacles. The shortest possible and collision-free path is then chosen [30–34];
- *cell decomposition* – where the path of movement is composed of segments joining corners of obstacle free cells [35, 36];
- *potential fields* – where the shortest path of movement is calculated from the start position to the global potential minimum, recognized as a goal, avoiding local maximum representing obstacles [37–40].

Both roadmap and cell decomposition require long precalculation stage and are not suggested for dynamic scenes and scenes with volatile aims. Each scene reconfiguration results in time-consuming recalculation. Only potential field based approach lets calculate camera movement in real-time. If virtual camera reaches region of local goal influence, it is attracted and eventually manoeuvred towards specified point of interest. The approach is based on a theory of elementary particle behaviour in an electrostatic field.

The theory was developed by Egbert et al. [37] and Kathib [39], who assigned adequate potentials to goal and obstacles and let the camera simulation run. Authors introduced virtual camera movement rules, attracted by goal potential gradients and repulsed by obstacles. However none of the authors protected camera from getting stuck in local minimums, neither camera oscillations where potential fields counterbalance. Further development was performed by Xiao et al. [41]. Authors have modified purely physical potential field approach. They have introduced limited potential distribution and linear force affecting the user's position only when moving and facing towards field source. Additional care was taken for local extremes elimination and obstacle collisions detection.

Li et al. [42, 43] has extended Xiao method by a new potential field distribution, considering wider context of the movement. However the most extensive and interactive approach was presented so far by Beckhaus [44, 45]. Author proposed a voxel based method which supports multiple fluctuating goals with different level of importance, though voxel usage seems to be computationally demanding and vulnerable to constant object reorientation. Nevertheless the introduced method considered camera reorientation relatively to the goal and provided the user with more thorough control over the navigation process.

Potential field distributions originated from voxels are calculated in conformity with a physical potential field theory. Goal object or its part is placed in the minimum of a global potential field and the obstacles are situated in local fields' maximums. Finally algebraic superposition of the local fields is carried out. As a result global potential field distribution is formed and camera can follow the way from its initial position to a global potential field minimum (goal). The potential field distribution can be presented as a hyperbolic surface with a goal in its minimum and obstacles in maximums.

Nonetheless, potential field characteristics, prevailing currently in literature, have still many destructive drawbacks [35, 44, 46-48]. Local potential fields coming out of obstacles form unpredictable extremes which are situated in accidental places and they are blocking the camera on the way to the aim [45, 49]. Forces coming out of potential field distribution do not let the camera approach obstacles. As a result camera has to make a detour around obstacle on the way to the goal. Moreover prevailing gradients do not protect the camera from getting into the objects. Potential field distribution theory derived from physics (Eq. (1)), where potential value U is inversely proportional to the distance r to an origin of the field, causes numerous problems connected with its range, and superposition [50–52]

$$U \approx \frac{1}{r}. \tag{1}$$

Such potential field traps must be solved by additional correction algorithms [44, 47, 48].

Properly controlled camera should neither get into the objects nor be blocked in accidental places. Collisions solving (obstacle repulsion distance) should be precise enough to let the user reach each place in the scene not occupied by objects on the way to the goal. Influence of potential field gradients should be adaptable to the user experience. Such conditions are fulfilled by author's method of newly constructed potential field distributions forces superposition.

A newly presented solution is the most relevant to the Beckhaus method but it eliminates necessity of voxel usage and solves most of potential field problems. The presented method not only supports camera positioning and reorienting, but collisions solving and dynamic scenes fast adaptability as well. Depending on ingredient potential field influences, method determines the path of the camera movement and provides a user with an intention compatible movement, for temporarily approached goals. It can support gathering information while navigation process by means of collisions solving, obstacle avoidance or camera view direction management, like attracting the user's attention towards explicit goals.

3. Method

A new approach defines potential field distributions discretely and adjustably to the context of the camera and action the user is to perform. Functionality of the potential field based method might be subdivided into two main camera movement aspects: supporting the camera positioning and controlling the camera orientation. It might be useful to separate supervision of the camera position from the camera orientation. Then system designer, and finally user, may independently modify adequate camera affecting coefficients. In case of virtual environments, camera contexts worth aiding may comprise: camera-object collision, camera-object attraction and object-directed view modification. The first of them can replace costly traditional geometric calculations. The others, by attracting camera to the neighbouring aims and correcting camera view direction, can support volatile goals searching process or camera manoeuvring while approaching motion targets.

It was assumed that camera representing user's point of view is positioned at eyes' height (h_c) of an invisible virtual character body and character's trunk bounding volume is approximated with a cylinder of the same height and radius R_1 (Fig. 1). Cylinder (character with camera) moves horizontally over the ground and just surrounding objects occupying common elevation range (objects or their parts which fall in between two parallel planes $y = 0$ and $y = h_c$) are considered. It does not limit the method generality. In case of a rough terrain or a volatile camera altitude (i.e. if character is squatting or jumping) additional computation considering changeable cylinder dimensions or altitude (camera height) should be considered. For instance, cylinder representing squatting character is lower then cylinder for upright character and set of objects or their parts clipped with planes $y = 0$ and $y = h_s$, where $h_s < h_c$, is smaller and should be considered individually. Eventually several additional potential field distributions, representing different camera navigation styles, encompassing different parts of the scene, should be generated.

Following above assumptions, spatial problem of the camera (cylinder) movement within a 3D environment can be reduced to planar movement of the camera (circle) within objects' 2D contours. Contours should be generated by means of orthogonal projection of virtual environment elements which fall into subspace bounded by two parallel planes ($y = 0$ and $y = h_c$ in a Fig. 1a).

Camera margin (R_1), representing its distance that should be preserved from objects' contours (R_1 should be greater than camera viewing volume near clipping plane value) can be interchangeably assigned to contours and then camera can be treated as a point moving within oversized (by R_1) contours (Fig. 1b). Oversized contours approach is used in the method for potential field distribution instead of the oversized camera.

The presented method bases on an empirically shaped, contour based, potential field distributions (attractive and repulsive), for which classical definition (Eq. (1)) became just an inspiration. For discussed camera navigation contexts, newly constructed potential field distributions become a source of functional gradient values.

For camera management convenience, influence of attractive field distributions U_{att} and repulsive collision field distributions U_{col} are considered separately. For static objects, all selected type of distributions can be respectively superposed into average attractive distribution U_{avg_att} and average collision distribution U_{avg_col}, as field sources relative position and orientation do not change. For dynamic objects, distributions cannot be combined in a preprocessing stage but in real-time, while application is running. Consequently averaged and individual dynamic field distributions form sources of potential filed gradients (forces), which in weighted manner may influence the camera position and orientation.

For a default camera height (h_c) an adequate potential field values U are defined in discrete nodes ($q_{i,j}$) of a planar uniform square mesh with a certain density (Δd in Fig. 1) parallel to the ground surface (Fig. 1). Additionally, the technical vicinity (R_1) around object's contour is defined. It estimates the distance the camera should keep away from the object (radius R_1 of camera circle).

Depending on the type of influence designed for the camera, the potential field distribution may receive different values.

3.1. Potential field distributions for camera-object collisions.

In case of pure collisions just the vicinity and interior of the contour should be managed (Eq. (2)). Moreover interior potential field distribution gets exponential negative values as to protect camera from tunnelling effect.

$$U_{col}(q_{i,j}) = \begin{cases} 1 & \text{if} \quad r \cdot pos(q_{i,j}) > R_1 \\ 1 - \max\left\{1, \left\lfloor \left(1 - \frac{r}{R}\right) \cdot antie\left(\frac{R}{\Delta d}\right) \right\rfloor^{1.5}\right\} & \text{if} \quad r \cdot pos(q_{i,j}) \leq R_1 \end{cases} \tag{2}$$

where r is a distance between contour and node q_{ij}; $pos(q_{i,j})$ is a function which value is equal to 1 for nodes q_{ij} outside the contour and -1 for inner nodes; R_1 is a predefined vicinity of the contour that provides adequate distance between the camera and the object; Δd is a mesh density reflecting minimum acceptable object's thickness; R is a radius of the field influence (object bounding sphere radius).

Interior negative U_{col} values can be limited by integer range $[-32768$ to $32767]$. According to Eq. (2), exemplary discrete, collision potential field distribution U_{col} is presented in Fig. 2.

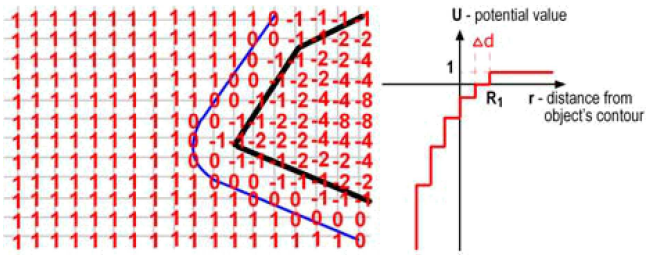

Fig. 2. An exemplary contour fragment and its potential field distribution devoted to collisions solving

3.2. Potential field distributions for goal attraction. When camera attraction to the object's contour is needed, additional potential field distribution may be generated according to Eq. (3)

$$U_{att}(q_{i,j}) = \begin{cases} 1 & \text{if} \quad r \cdot pos(q_{i,j}) \geq R \\ \max\left\{1, \left\lfloor \left(1 - \frac{r}{R}\right) \cdot U_{\max} \right\rfloor\right\} & \text{if} \quad r \cdot pos(q_{i,j}) \in [R_1, R) \\ 0 & \text{if} \quad r \cdot pos(q_{i,j}) < R_1 \end{cases} \tag{3}$$

where r is a distance between contour and node $q_{i,j}$; $pos(q_{i,j})$ is a function which value is equal to 1 for nodes $q_{i,j}$ outside the contour and -1 for inner nodes; R_1 is a predefined vicinity of the contour that provides adequate distance between the camera and the object; Δd is a mesh density reflecting minimum acceptable object's thickness; $U_{\max}$ is a maximum potential value assigned to object; R is a radius of object's bounding sphere considered as a scope of an object assigned, potential field generation.

Corresponding fragment of an attractive potential field distribution U_{att} is presented in Fig. 3.

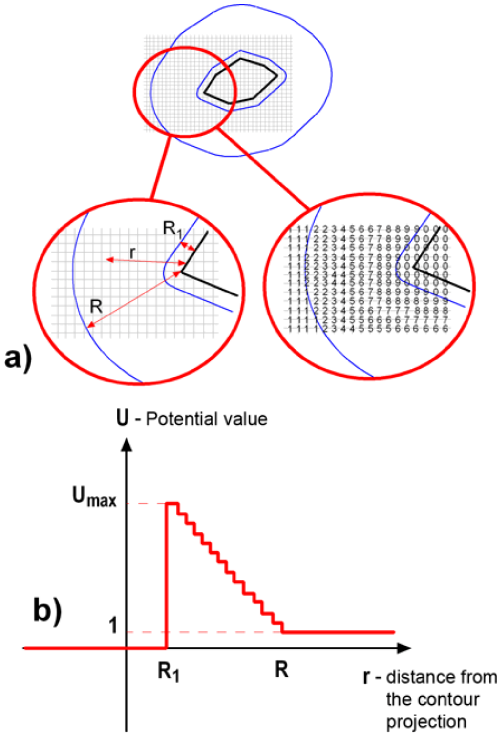

Fig. 3. Generation of an attractive potential field distribution for an exemplary object fragment; a) U_{att} potential field distribution; b) U_{att} distribution chart; (Eq. (3) for $R = 11^{*}\Delta d$, $U_{\max} = \left\lfloor \frac{R}{\Delta d} \right\rfloor$ and $R_1 = 2^{*}\Delta d$)

3.3. Local task oriented potential field modifications. Analogically specialized local potential field distributions can be elaborated for more sophisticated tasks. The hitherto presented approach considered potential field gradients derived from whole objects, whereas their specified parts or elements can be also privileged. For instance specified face or view of the object might be more interesting than others so the application developer may want to form local potential field maximum attracting the camera to this place. Such field distribution may be derived from a manually selected mesh node used as a source of a local attractive potential field distribution. Local potential field mesh nodes values U_{loc} can be calculated according to Eq. (4).

$$U_{loc}(q_{i,j}) = \begin{cases} \left\lfloor \left(U_{ob} \cdot \frac{R_{ob} - r}{R_{ob}}\right) + 1 \right\rfloor & \text{if} \quad r \leq R_{ob} \\ 1 & \text{if} \quad r > R_{ob} \end{cases} \tag{4}$$

where U_{loc} – local potential field value in node q, U_{ob} – maximum potential field value for a specified point/object, r – distance from the origin of the local potential field, R_{ob} – range of local potential field influence.

In order to prevent local potential field from influencing unintended regions (i.e. in case of thin object like inner walls, potential field distribution may overlap distribution of an opposite side of the object) it should be appropriately clipped (Eq. (5))

$$U_{loc\_clip}(q_{i,j}) = \begin{cases} U_{loc}(q_{i,j}) & \text{if ray casted from the origin of } U_{loc} \text{ to node } q_{i,j} \text{ DOES NOT intersect object} \\ 1 & \text{if ray casted from the origin of } U_{loc} \text{ to node } q_{i,j} \text{ intersects object} \end{cases} \tag{5}$$

If line segment connecting node $q_{i,j}$ with origin node of the local field crosses the object contour, node's value is set to 1. Otherwise values are calculated according to Eq. (4) (Fig. 4).

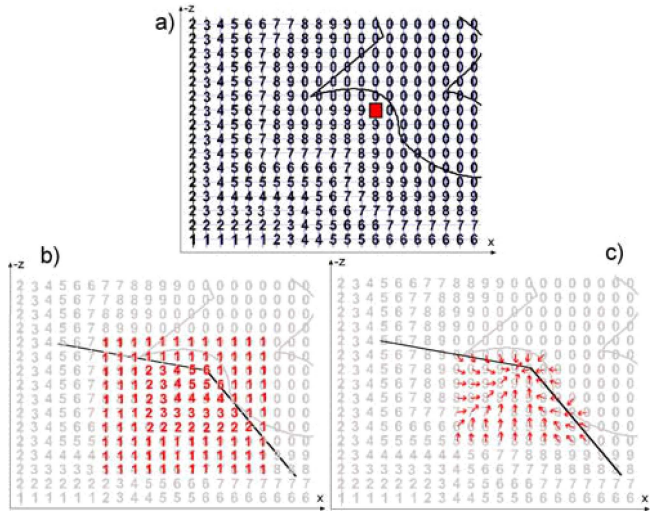

Fig. 4. Local potential field distribution a) origin node marked with red square b) local potential field distribution clipped with object c) corresponding potential field gradients (Eq. (4) for $R = 5^{*}\Delta d$ and $U_{ob} = 5$)

3.4. Potential field superposition. It may happen that several potential field distributions overlap each other (in figure 4b and 4c local potential field overlaps general attractive potential field distribution). As they may come from both static and dynamic objects they must be considered separately. Moreover, the author suggests considering separately distributions devoted to collision and attractive distributions, both local and global, as it lets to flexibly and individually balance attractive and repulsive forces in final camera steering process.

In case of static attractive objects' field distributions they can be superposed in a pre-processing stage according to Eq. (6).

$$U_{att\_avg}(q_{i,j}) = \begin{cases} \max\{U_{att\_s}(q_{i,j})\} & \text{if } \forall s \in \{1,2,...,k\} \; U_{att\_s}(q_{i,j}) \neq 0 \\ 0 & \text{if } \exists\, s \in \{1,2,...,k\} \; U_{att\_s}(q_{i,j}) = 0 \end{cases} \tag{6}$$

where k – number of overlapping ingredient potential field distributions, $U_{att\_avg}(q_{i,j})$ – is a final attractive potential value in $q_{i,j}$ mesh node, $U_{att\_s}(q_{i,j})$ – is a potential value of an elementary attractive or local field distribution s in node $q_{i,j}$.

Static attractive potential distributions superposition flatten all elementary distributions into one static potential field distribution choosing maximum potential value from the corresponding mesh nodes and leaving zero values inside objects' oversized contours. Such method of potential field superposition eliminates local unpredictable extremes formation. Besides extremes associated with objects or marked as their local points of interest, may not occur within superposed potential field distribution.

In case of collisions, static potential field distributions superposition, due to negative potential field values inside contours, should be performed by means of choosing minimum value from the corresponding nodes (Eq. (7)).

$$U_{col\_avg}(q_{i,j}) = \begin{cases} \min\{U_{col\_s}(q_{i,j})\} & \text{if } \exists\, s \in \{1,2,...,k\} \; U_{col\_s}(q_{i,j}) \neq 1 \\ 1 & \text{if } \forall\, s \in \{1,2,...,k\} \; U_{col\_s}(q_{i,j}) = 1 \end{cases} \tag{7}$$

where k – number of overlapping ingredient potential field distributions, $U_{col\_avg}(q_{i,j})$ – is a final average collision potential value in $q_{i,j}$ mesh node, $U_{col\_s}(q_{i,j})$ – is a potential value of an elementary collision field distribution s in node $q_{i,j}$.

Adequate superposition can be calculated in advance before running the application, however technically potential field distributions superposition can be easily achieved in real-time due to contemporary powerful graphics processing units (GPU). Both potential values and even gradient forces (Eq. (8)), represented as pixels (coded with pixels' colours) of dedicated textures can be immediately superposed with *fragment shaders* – small programs performed on GPU [18].

If camera cannot be limited to one predefined height (like h_c at the beginning of chapter 3) several additional potential field distribution layers should be generated covering the camera possible height range. Eventually several discrete planar potential field distributions, representing different camera altitudes, can be trilinearly interpolated as to obtain continuous potential value retrieval.

In case of potential field distributions, associated with dynamic, transformed objects, they should have three dimensional representations of the similar resolution, reflecting objects' complexity, in all three dimensions. Depending on the objects' orientation theirs spatial distribution may be questioned for valid potential field values. Additionally, for each individual dynamic object, their potential field distributions should be stored separately, as potential field values are retrieved in run time depending on an actual camera-object relation. Technically such retrieval process can be boosted due to 3D texture sampling mechanism in *fragment shader* on GPU [18].

Some readers might be overwhelmed with a number of data, connected with potential field distributions, that should

be stored for the proper method functioning. Then it must be remarked that contemporary computers are equipped with powerful graphics cards, which are able to load and, due to GPU, real-time perform a huge amount of data stored in a form of 3D textures. Contemporarily, available textures dimensions considerably exceed the required average method resolutions (256×256×256). Potential field distributions may be even stored with a rough, limited density and eventually interactively filtered (i.e. Gaussian filter) when necessary.

The most valid task that should be discussed now, is an adequate potential field gradient influence on the virtual camera while navigation process.

3.5. Potential field gradient. The potential field planar distribution can be treated as a source of gradient vectors $\vec{G}$ calculated discretely for each node $q_{i,j}$ considering its eight neighbours according to Eq. (8). Gradient vectors represent discretely defined forces that may influence camera position and orientation. Discrete gradient values are usually precalculated in order to reduce the run time operations number.

$$\begin{aligned}
\vec{G}(q_{i,j}) &= \langle G_x(q_{i,j}), G_y(q_{i,j}), G_z(q_{i,j})\rangle,\\
G_x(q_{i,j}) &= [U(q_{i-1,j+1}) - U(q_{i-1,j-1}) + U(q_{i,j+1})\\
&\quad -U(q_{i,j-1}) + U(q_{i+1,j+1}) - U(q_{i+1,j-1})],\\
G_y(q_{i,j}) &= U(q_{i,j}),\\
G_z(q_{i,j}) &= [U(q_{i+1,j-1}) - U(q_{i-1,j-1}) + U(q_{i+1,j})\\
&\quad +U(q_{i-1,j}) + U(q_{i+1,j+1}) + U(q_{i-1,j+1})],
\end{aligned} \tag{8}$$

where i, j are uniform mesh nodes' indexes.

In some situations, where vertical influence of gradient vector $\vec{G}$ should be reduced or eliminated its orthogonal projection $\vec{G}_\perp = \langle G_x, 0, G_z\rangle$ should be considered. Then gradient vectors affect camera only horizontally.

Exemplary fragments of attractive local (U_{loc_clip}) and attractive general (U_{att}) potential field gradients ($\vec{G}$) distributions are presented respectively in Figs. 4c and 5. For general, object attracting, field distribution (U_{att} – Fig. 5) nodes' forces are directed towards object within its surrounding (R) and rapidly change their influence within object close vicinity (R_1). Such potential forces distribution can attract camera within considered surrounding R but eventually protects it from getting into object by means of repulsive gradient. For the local potential field distribution (Fig. 4c), nodes' forces are directed towards field origin (point of interest) within certain range (R_{ob}).

Potential field gradients should be calculated for each ingredient potential field distribution. These are static averaged global (U_{att_avg}, U_{col_avg}) and local ($U_{loc\_clip\_avg}$) distributions and all individual dynamic objects spatial distributions (U_{att}, U_{col}, U_{loc_clip}). All elementary gradients derived from potential field distributions can influence the camera navigation.

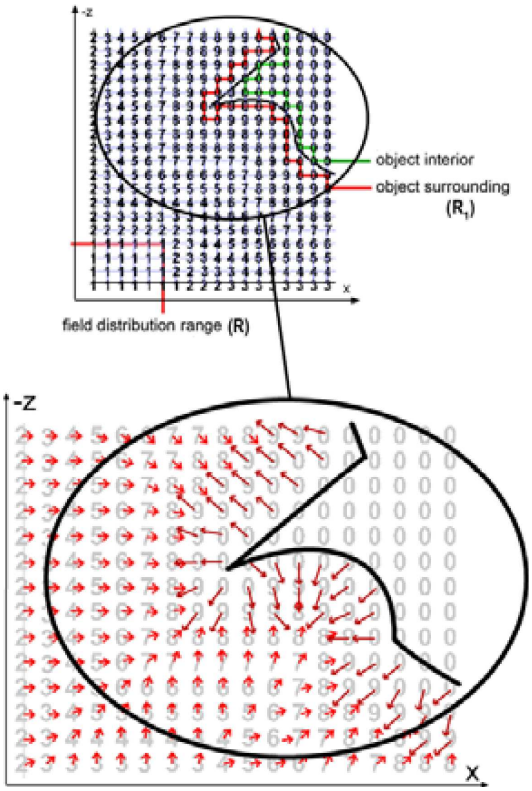

Fig. 5. Potential field gradient attracting virtual camera towards object and eventually protecting from getting into it

3.6. Camera navigation in a virtual environment. Technically virtual camera is represented with three independent vectors determining its position ($\vec{pos}$), view direction ($\vec{dir}$) and up orientation ($\vec{up}$) in a global coordinate system. Besides camera characteristics also user interface output and potential field gradients, are represented as vectors, so they can be mathematically summed in order to modify the camera final state. It is a crucial task to provide appropriate weights for component vectors affecting camera. Their proportion influences the camera navigation process, and consequently affects perception and conformity with user intentions.

Basically camera position and orientation can be upgraded with time-scaled, user interface generated, translation vector $\vec{K}$ along viewing direction $\vec{dir}$ (Eq. (9)) or rotation vector$\vec{L}$ (Eq. (10)) (perpendicular to $\vec{up}$ and $\vec{dir}$ vectors) (Fig. 6).

$$\begin{aligned}
\vec{pos}' &= \vec{pos} \oplus \vec{K}\cdot\alpha_1(t),\\
\text{where} \qquad \vec{K} &= \frac{\vec{dir}}{\|\vec{dir}\|},
\end{aligned} \tag{9}$$

$$\begin{aligned}
\vec{dir}' &= \vec{dir} \oplus \vec{L}\cdot\beta_1(t),\\
\text{where} \qquad \vec{L} &= \frac{\vec{dir}\otimes\vec{up}}{\|\vec{dir}\otimes\vec{up}\|},
\end{aligned} \tag{10}$$

where $\alpha_1(t)$, $\beta_1(t)$ are time dependent scalar coefficients influencing $\vec{dir}$ and $\vec{pos}$ vectors, $\otimes$ and $\oplus$ represent respectively cross product and vector sum operators.

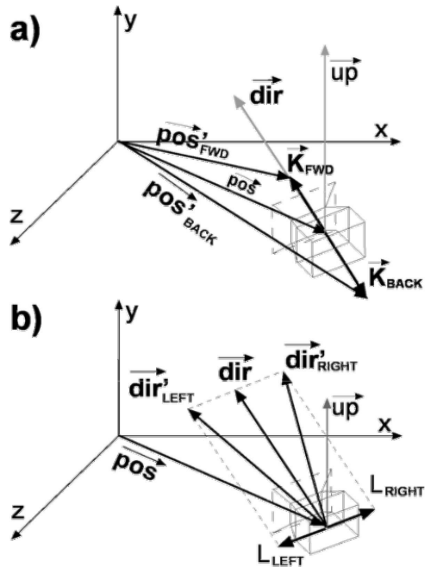

Fig. 6. Virtual camera modified by user interface generated vectors; a) K – translation vector; b) L – rotation vector

As it is mentioned in previous chapters potential field distributions might be responsible for global target or local points of interest attracting, obstacle avoiding or collisions solving. As a result, the camera position and orientation can be modified by each individual averaged gradient vector retrieved from the potential field distributions affecting its actual position. The calculation of the camera position ($p\vec{o}s$) corresponding averaged gradient vector ($\vec{G}_{avg}$), for discretely defined potential field, is performed according to Eq. (11).

$$\vec{G}_{avg}(p\vec{o}s) = \sum_{s=i-1,t=j-1}^{s=i+2,t=j+2} \left[\vec{G}(q_{s,t}) \cdot \exp^{-\text{distance}(p\vec{o}s;q_{s,t})}\right], \tag{11}$$

where s, t are indices of potential field discrete nodes, *distance* is an Euclidean distance from the camera position to a $q_{s,t}$ node and Σ is a vector sum operator.

For an actual camera position ($p\vec{o}s$) exponentially weighted mean of 16 neighbouring mesh nodes' gradients is calculated. Each gradient is scaled by exponential coefficient reflecting its node distance to the camera current position. Empirical studies have shown that such approach smoothens better camera transformations in comparison with simple algorithm of the closest neighbours' arithmetic mean value.

Additionally it was assumed that, for attracting forces, the more the camera is directed at the potential field origin, the stronger the potential field influence should be. In bounding condition, camera should not be attracted by a source of field located behind it. Such reasoning should not be applied for camera collisions as they should be solved independently from camera view orientation. It is worth reminding that attractive and collision based gradients are considered separately so such diversity is available. Consequently the camera position can be modified due to attractive forces according to Eq. (12).

$$p\vec{o}s' = p\vec{o}s \oplus (di\vec{r} \cdot \alpha_1) \oplus (\vec{G}_{\perp avg}(p\vec{o}s) \cdot \alpha_2 \cdot \alpha_c) \tag{12}$$

where

$$\alpha_c = \cos \angle(di\vec{r}_{\perp}, \vec{G}_{\perp avg})$$

for

$$\angle(di\vec{r}_{\perp}, \vec{G}_{\perp avg}) \in \langle -90^{\circ}, 90^{\circ} \rangle .$$

The potential field gradient is taken into consideration only if the angle between view direction projection ($di\vec{r}_{\perp}$) and field gradient vector ($\vec{G}_{\perp avg}$) is smaller than 90 degrees. Potential field gradients should not also influence the camera when it is not modified by the user. Then $\alpha_1 = \alpha_2 = 0$, coefficients are equal to zero and the camera does not change its position. For collision based distributions, calculations are performed according to modified Eq. (12), where α_C coefficient is constant and equal to 1.

In case of static objects they form one superposed potential field distribution with gradient value retrieved each time adequately to an actual camera position. In case of dynamic objects potential field gradients should be actualized not only due to camera replacement but due to object transformation as well. Such problem can be solved twofold. The object translation should be considered as if the camera had additional translation equal to negative object translation [19]. Analogically object rotation results in the same potential change as if object was stable and the camera rotated around object pivot by the same angle but in opposite direction [19]. The resulting camera position may fall in between spatial potential nodes distribution so tri-linear nodes' gradients interpolation can be applied. In consequence Eq. (11) should be extended to 64 neighbours, or for boosting calculation, the closest two dimensional potential field distribution should be chosen.

Finally problem of several overlapping potential fields affecting camera simultaneously should be considered. As each potential field distribution is a source of field gradient vectors they can be geometrically added. For camera position ($p\vec{o}s$) final actualization, horizontal gradients $\vec{G}_{\perp}$ coming out of all fundamental gradients sources should be algebraically summed (Eq. (13)).

$$\begin{aligned} p\vec{o}s' = {} & p\vec{o}s \oplus (\vec{K} \cdot \alpha_1(t)) \\ & \oplus(\vec{G}_{\perp att\_avg}(p\vec{o}s) \cdot \alpha_2 \cdot \alpha_{C,2}) \\ & \oplus(\vec{G}_{\perp loc\_clip\_avg}(p\vec{o}s) \cdot \alpha_3 \cdot \alpha_{C,3}) \\ & \oplus(\vec{G}_{\perp col\_avg}(p\vec{o}s) \cdot \alpha_4) \oplus \sum_i \vec{G}_{\perp dyn_i} \end{aligned} \tag{13}$$

where $\alpha_{C,k} = \cos \angle(di\vec{r}_{\perp}, \vec{G}_{\perp avg})$

for $\angle(di\vec{r}_{\perp}, \vec{G}_{\perp avg}) \in \langle -90^{\circ}, 90^{\circ} \rangle$ and $k = \{2, 3\}$

and $\vec{G}_{\perp dyn_i} = \vec{G}_{\perp att_i}(p\vec{o}s) \cdot \alpha_{2_i} \cdot \alpha_{C,2_i})$

$\oplus(\vec{G}_{\perp loc\_clip_i}(p\vec{o}s) \cdot \alpha_{3_i} \cdot \alpha_{C,3_i})$

$\oplus(\vec{G}_{\perp col_i}(p\vec{o}s) \cdot \alpha_{4_i}).$

For camera view direction $\vec{dir}$ actualization adequate gradients should be taken into consideration (Eq. (14)).

$$\begin{aligned}\vec{dir}' &= \vec{dir} \oplus (\vec{L} \cdot \beta_1(t)) \\ &\oplus(\vec{G}_{att\_avg}(\vec{pos}) \cdot \beta_2 \cdot \beta_{C,2}) \\ &\oplus(\vec{G}_{loc\_clip\_avg}(\vec{pos}) \cdot \beta_3 \cdot \beta_{C,3}) \\ &\oplus \sum_i \vec{G}_{dyn_i} \qquad (14)\end{aligned}$$

where $\beta_{C,k} = \cos\angle(\vec{dir}, \vec{G}_{avg})$

for $\angle(\vec{dir}, \vec{G}_{avg}) \in \langle -90°, 90° \rangle ; k = \{2, 3\}$

and $\vec{G}_{dyn_i} = (\vec{G}_{att_i}(\vec{pos}) \cdot \beta_{2_i} \cdot \beta_{C,2_i})$

$\oplus(\vec{G}_{loc\_clip_i}(\vec{pos}) \cdot \beta_{3_i} \cdot \beta_{C,3_i})$.

3.7. Task oriented potential fields. Presented above, set of potential field distributions, becomes a source of gradients that might be applied for the camera navigation tasks support. Potential field gradients solving collisions and potential field gradients attracting viewer to predefined objects or their specific points can be smartly superposed for achieving camera navigation scenarios.

When collisions gradients are only applied the user can freely (without path constraints) move throughout a navigational space solving collisions with all objects came across on the way. Collisions are solved independently whether these are static or dynamic objects. This is an equivalent of the *exploration wandering* mode specified by Bowman where no goal is clearly specified.

Application user, sightseeing virtual environment, may interactively specify temporal point/object of interest and as a consequence dedicated goal gradient distribution can be launched with an appropriate influence affording the camera to reach a specified goal as a global or temporal goal. This corresponds to Bowman's *search mode* where one or several goals are targeting the user towards their origin. Not only goal forming objects can be switched interactively but their influence can be adjusted to context and user actual preferences.

Appropriately prepared potential field distributions may also influence the camera at a considerable distance. The virtual ambient field distribution, not necessary connected with any scene object may form some kind of a global target affecting the camera at a certain rate. Besides one specified ambient goal, several virtual goals may form so called waypoints forming navigational paths.

The above applications exploit mainly gradients affecting the camera position, however camera view direction influencing gradients become also very useful, for instance for Bowman's *maneuvering mode*. The camera approaching goal of navigation is facing very rarely directly to the goal, whereas usually further interaction process requires comfortable, front side view. Both local and global attractive gradients influencing both the camera position (Eq. (13)) and the camera view direction (Eq. (14)) may help in an appropriate view targeting. Especially for novice users, not familiar with burden put on the user by immersive interfaces, such a solution might be helpful.

On one hand, a considerable number of coefficients specified in Eq. (13) and (14) may suggest the method complexity but on the other hand, it justifies the method flexibility and user preferences adaptability. User impressions may depend not only on method possibilities as such but on user interface functionality. For instance, during experiments, some users have found it very helpful to interactively launch *maneuvering mode* by pressing of mouse button whereas others preferred constant view modification influence as they easily adapted the interface functionality.

Experiments on potential field based camera influences provided conclusions on camera elevation changes support. It was experimented that winding stairs can be easily explored by means of appropriately prepared potential field spatially winding path pushing camera out of handrails and forcing adequate elevation increase or decrease depending on move direction (ascending or descending stairs).

Experiments have also proved that spatial potential field distributions can be exploited for building object interrelationships. Objects can physically attract or repulse each other and simulate physical interactions. In fact the proposed solution is burdened with its discrete potential field distributions, however there are several applications where substantial component of physics engine can be substituted with potential field gradients interrelationships.

4. Method profiling

The most important aspect of the method application is a context dependent method profiling. Though several potential field distributions may exist within a scene simultaneously, their role and importance may be diversified by means of scalar coefficients (from Eq. (13) and (14)) and potential field distribution range (potential value).

Initially, for a considered scene, minimum object thickness Δd should be set. It affects the potential field distribution density represented with uniform mesh node size (Fig. 3). Parameter Δd should let represent the object roughness and all important, from navigational point of view, objects' artefacts. Performed experiments have shown that, for average objects complexity (Fig. 7), distributed over moderately cluttered surface, scene could be represented with a mesh of 256 nodes size in each direction. The resolution of 512 nodes was absolutely sufficient for all considered objects' features representation.

In a second step maximum speed of camera movement (in Δd units) should be specified. Within tested scenes movement speed did not exceed 0.7 Δd and rotation speed was limited by 1.0 Δd. Both maximum speed values were time dependently cumulated. Their values increased 0.03 Δd per frame what caused that the user input was cumulated and interpreted

with a certain latency and the maximum camera transformation speed was achieved after about half a second (assuming about 60 fps frame rate). Such a solution protected the user from unwilling rapid camera changes.

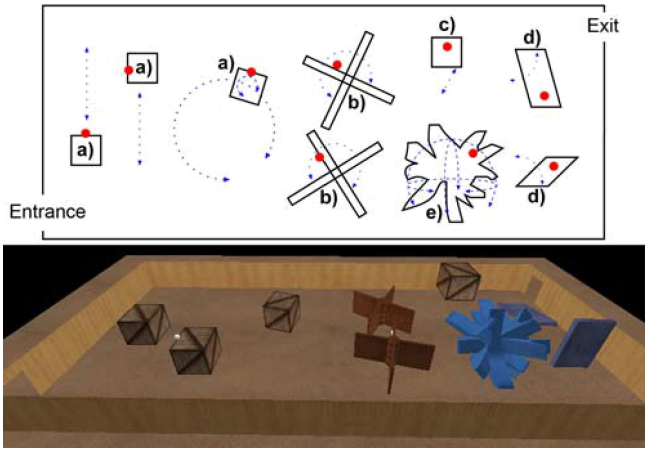

Fig. 7. A tested scene comprising dynamic objects with marked (red dots) positions of plates comprising numbers; Object types: a) translated horizontally; b) rotated around vertical axis; c) translated vertically; d) rotated around horizontal axis; e) rotated simultaneously around all three Cartesian axis

Consequently potential field influence range R and maximum potential values $U_{\max}$ were specified. As average objects' size in Δd units fell between 20 Δd units for simple box (Fig. 7a, 7c) and 90 Δd units for turnstile (Fig. 7b), for tested objects influence range R should be set to about 50% of average component objects grid size in Δd units (mean radius of objects' bounding spheres). Indifferent to objects size their attractive influence should be comparable. Thus for all objects maximum potential value $U_{\max}$ was set to 99 and influence range to 35 Δd units. In this way gradients assigned to different objects and goal are comparable (potential field distribution has similar steepness). Local points of interest, for better comparability, have assigned the same maximum potential value and equal range of influence, which for tested scenes was 35 Δd units. The whole tested scene was encompassed within a space of 200 by 500 Δd units.

Additional care was put to objects' vicinity R_1 (contour margin) which differentiated according to their movement specification. Transformed horizontally objects had it specified at a level about 5% of object size whereas vertically modified objects required a distance buffer at a level about 10% of the object size. This mechanism protected the camera from visually crossing objects' mesh.

Moreover vertically transformed objects required adequate spatial potential field distributions. Spatial blob (Fig. 7e) had discrete potential distributions every 30 degrees rotation around two horizontal axes. There are 12 potential distribution cross sections around each of two axes. The vertical axis rotation did not require separate distributions as potential gradient was retrieved algebraically basing on previous ones. Higher resolution of spatial field distributions was suggested for extensive big (long) objects that may fall on user's head. For assuring smooth camera escape from under the "falling bridge" (Fig. 7d) and "falling box" (Fig. 7c) consequently about 24 and 20 distributions were suggested.

Possible fluctuations in gradient vectors lengths may suggest necessity of their normalisation. It must be remarked that constant, for all objects, maximum potential field value $U_{\max}$, fading within different ranges R of diversified in size objects, evokes different resulting field gradients. However vectors lengths were left unchanged as they reflect importance of each field influence and might be still tuned by scalar coefficients. The role/importance of each potential field depends on the implementation and configuration of the system and, in this paper, author decided to moderate and test gradient influence by means of scalar coefficients adjustment.

5. Tests

According to Nielsen and Gabbard [53-55] usability engineering methodology, tests were performed both formally and informally. It means process data and bottom-line data were collected during testing stage. Formally time devoted to a specified task completion was measured and informally general impressions and problems were recorded with previously elaborated questionnaire.

The testing environment was designed to comprise wide spectrum of objects complexity (Fig. 7) undergoing different translations (Fig. 7a, 7c) and rotations (Fig. 7b, 7d, 7e). Gathering information effectiveness was tested by means of randomly generated plates comprising small numbers (Fig. 8), users were asked to read and confirm both verbally during the test and by specifying the final sum at the end of the test.

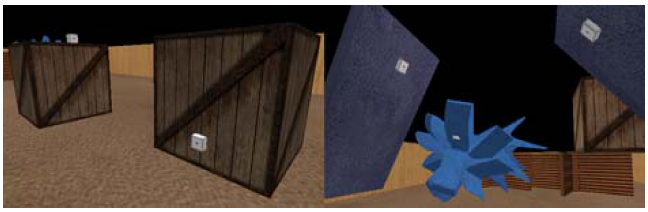

Fig. 8. Exemplary white plates with randomly generated numbers attached to different places of transformed objects

Additionally users were asked to cover the whole distance from the entry to the exit within possible short time, and visiting all number plates, what was measured. The camera position was controlled with a keyboard buttons (WSAD) as well as the camera orientation (keyboard arrows). The number of plates localised on the objects and their distribution throughout the scene was presented on a map to the user in advance of the test. The order of number plates visiting was not forced on the users however each of them was allowed to perform time unlimited training period and they were able to prepare their own strategy. After completion of training stage and before formal tests users were asked to fill out the form questioning users' gender, level of experience, satisfaction and perceived effectiveness of camera aiding process.

In the tested scenes two different configurations of potential field distributions, assigned to dynamic objects, were

tested. Within first configuration all attracting forces were disabled and just collisions gradients were launched. For Eqs. (13) and (14) adequate scalar coefficients corresponding to mentioned categories of objects (Fig. 7a–7e) were implemented (Table 1).

The tested configuration considered just user-based camera influence with time-dependent speed accumulation, and experimentally selected collision coefficients. Coefficient $\alpha_4 = 1.6$ gave more than twice of maximum camera translation unit (0.7 Δd). Just for relatively thin objects like rotating doors scalar coefficient was strengthened. Coefficient $\alpha_4 = 2.4$ gave about 3.5 times of maximum camera translation unit (0.7 Δd), what assured greater (adequate) distance keeping.

The second tested configuration has concentrated on the camera position ($\vec{pos}$) and view direction ($\vec{dir}$) support. For clarity of field influence on the camera position, attraction gradients were exchangeably eliminated. Several preliminary tests have shown that points of interest assigned to extensively moving objects (group a and b from Fig. 7) may require some considerable attraction while others do not. That's why some noticeable position attraction was assigned to selected objects' goals (Table 2).

Influence coefficient $\alpha_3 = 0.16$ for previously specified local potential filed gradients (Eq. (4) for $U_{ob} = 99$; $R = 35\Delta d$) and cosines function scaled, fell in between a range of about $0 \leq \alpha_3 \cdot \alpha_{C,3} \leq 4 \cdot \Delta d$ translation units. For the camera view direction ($\vec{dir}$) general attraction gradients influence was reduced to a margin level. Coefficient $\beta_2 = 0.01$ for specified local potential field gradient (Eq. (3) for $U_{\max} = 99$; $R = 35\Delta d$) and cosines function scaled, fell in between a marginal range of about $0 \leq \beta_2 \cdot \beta_{C,2} \leq 0.2 \cdot \Delta d$. At the same time considerable $\beta_3 = \{6, 15\}$ possible values evoked substantially huge influence on the camera view direction ($0 \leq \beta_3 \cdot \beta_{C,3} \leq 270 \cdot \Delta d$ for Eq. (3) with $U_{ob} = 99$ and $R = 35 \cdot \Delta d$). As a consequence corresponding component of Eq. (14) affects both horizontal and vertical camera view direction.

The hypothesis of the tests was to prove that scene with target oriented navigation support provides more effective and user friendly navigation than unconstrained camera movement without support.

21 volunteers were invited (2 women and 19 men) to participate in the scene exploration. Testers' age fell between 21 and 28, all of them were right handed and none of tested participants declared problems with colour perception. Consequently the level of experience in 3D interfaces exploration was questioned. 12 participants specified their experience as high (frequent users of First Person Perspective games) and 9 testers declared themselves as moderately experienced. Concluding, experience of tested people was more than average, and it can be stated that the tested method was confronted among users with strong habits derived form classical WSAD and mouse computer game interface.

The first set of invaluable conclusions was retrieved from questionnaires. It must be remarked that 81% of tested users have found potential field based vertical camera rotation very helpful and also 81% preferred automatic camera modification to manual one (19%). Automatic horizontal camera rotations were appreciated by 72% whereas only 53% of testers stated that automatic horizontal rotations are better than manual.

Table 1
Set of scalar coefficients assigned to first dynamic scene configuration supporting only collisions within a specified group of objects form Fig. 7

	group a	group b	group c	group d	group e
$\|\vec{K} \cdot \alpha_1(t)\|$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$
$\alpha_2 \cdot \alpha_{C,2}$	0	0	0	0	0
$\alpha_3 \cdot \alpha_{C,3}$	0	0	0	0	0
α_4	1.6	2.4	1.6	1.6	1.6
$\|\vec{L} \cdot \beta_1(t)\|$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$
$\beta_2 \cdot \beta_{C,2}$	0	0	0	0	0
$\beta_3 \cdot \beta_{C,3}$	0	0	0	0	0

Table 2
Set of scalar coefficients assigned to second dynamic scene configuration supporting both collisions and camera view orientation and position modification for: $\cos\angle = \cos\angle(\vec{dir}, \vec{G}_{loc\_clip})$ and $\cos_\perp\angle = \cos\angle(\vec{dir}_\perp, \vec{G}_{\perp_{loc\_clip}})$

	group a	group b	group c	group d	group e
$\|\vec{K} \cdot \alpha_1(t)\|$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$	$\in (0; 0.7)$
$\alpha_2 \cdot \alpha_{C,2}$	0	0	0	0	0
$\alpha_3 \cdot \alpha_{C,3}$	$0.16 \cdot \cos_\perp\angle$	$0.16 \cdot \cos_\perp\angle$	0	0	0
α_4	1.6	2.4	1.6	1.6	1.6
$\|\vec{L} \cdot \beta_1(t)\|$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$	$\in (0; 1)$
$\beta_2 \cdot \beta_{C,2}$	$0.01 \cdot \cos\angle$	$0.02 \cdot \cos\angle$	$0.01 \cdot \cos\angle$	$0.01 \cdot \cos\angle$	$0.01 \cdot \cos\angle$
$\beta_3 \cdot \beta_{C,3}$	$6 \cdot \cos\angle$	$15 \cdot \cos\angle$	$6 \cdot \cos\angle$	$6 \cdot \cos\angle$	$15 \cdot \cos\angle$

The next interesting conclusion comes out of unintended camera behaviour within supported and unsupported scene. Further statistics revealed that 33% of tested people have noticed unintended camera behaviour in the supported scene while exploration, when unsupported scene has left only 19% of users lost, within dynamic and fast changeable scene. It proves that the camera navigation support is very foreseeable and easy to learn. Most of users (67%) could understand the method and exploit it in the explored scene just after a short training period. It is only 14% trapped people more than for traditional unsupported environment.

Beyond any doubt became results of camera collisions solving success rate. All tested people (100%) could navigate throughout the scene without intruding to objects interior, and could keep sufficient distance to objects in order to read plates' numbers.

Additionally, during the formal tests period, time of covering the whole scene upon visiting each number plate, was measured. For evaluation clarity not only total time of scene exploration (T_i) was measured but also time of pure navigation (N_i) was recorded (Table 3).

Table 3
Descriptive statistics and Shapiro-Wilk test results verifying normal distribution of time measurements within dynamic scenes. T_{col} and N_{col} are respectively total time T and pure navigation time N within dynamic scene with just collisions. Analogically T_{sup} and N_{sup} represent corresponding time values for navigationally supported scenes

	T_{col}	T_{sup}	N_{col}	N_{sup}
Number of attempts (N)	21	21	21	21
Median (Me)	63.4	51.6	25.9	21.5
Arithmetic mean (M)	61.38	52.19	25.80	22.77
Standard deviation (SD)	10.15	11.51	4.84	5.12
Shapiro-Wilk test (p-value)	0.518	0.118	0.409	0.782

Unfortunately due to insufficient reliability level ($p > 0.05$), the hypothesis verification with, normal distribution-based, t-Student test, was not authorised. In return non-parametric Mann-Whitney test, for verification of time measurements medians relation, was applied. These tests show that with a considerable reliability level ($p < 0.008$) median of T_{sup} is smaller than T_{col}, and with also very satisfactory reliability level ($p < 0.029$) median of N_{sup} is smaller than N_{col}.

Provided results clearly and formally demonstrate that dynamic scenes with navigation support let more effectively gather information than in case of scenes without any support.

6. Method discussion

The most preliminary method's usage is the potential field based camera-objects collisions response. In such a mode the camera is navigated freely by the user interface and potential field distribution gradients (Fig. 2) block camera movements in case of approaching repulsive object's vicinity (R_1 in Eq. (2)). As potential values decrease almost exponentially with getting into the object, adequate repulsive gradient vectors consequently increase. It solves the discrete time based collision tunnelling problem users may experience on slower computers or computationally demanding 3D scenes. It is also important to provide appropriate user's camera translation vectors' length which should be at maximum as long as the shortest repulsing gradient vectors (Eq. (15)). It is achieved not only by means of adequate values of α_1 and α_4 coefficients proportion, from Eq. (14), but length advantage of $\vec{G}_{col}$ vector as well. Collision response gradient should be much stronger than user-derived camera translation vector (2 or 3 times in tests) and should not be camera view direction dependent as collisions should work even if camera moves backwards.

$$\left\|\vec{G}_{col}\cdot\alpha_4\right\| >> \left\|\vec{K}\cdot\alpha_1\right\|. \tag{15}$$

Perfectly designed collision forces should push the camera back outside object close vicinity range (R_1). Then translation vectors and collision gradients counterbalance and the camera stops before a virtual collision barrier. In case of poor framerate the camera may skip several repulsing rings and might test itself against one of internal repulsing gradient vectors but then, due to repulsion strength it would be rapidly translated back outside the object's oversized contour. Outside the close vicinity of the object (distance greater than R_1 in Eq. (2)) collision gradient vectors do not exist so the camera can be thoroughly controlled by the user.

The composition of potential field gradient vectors attracting the user towards the object or a specific point of interest (Fig . 4, 5) may be the next method's application. It can be especially useful in case of virtual environment exploration when a novice user tends to approach some temporal goals or points of interest. The exploitation of immersive virtual environments usually causes certain navigational disturbances due to specialised hardware operational inexperience [3]. Then appropriately added, attractive potential field influence (Eq. (13)), may help the observer reach specified goal (if camera is within goal's influence). The appropriateness is controlled with α_2 and α_3 coefficients which may depend on the user experience and their influence should be inferior to user intentions (Eq. (16)).

$$\left\|\vec{G}_{att}\cdot\alpha_2 \oplus \vec{G}_{loc}\cdot\alpha_3\right\| < \left\|\vec{K}\cdot\alpha_1\right\|. \tag{16}$$

In case of dynamic objects where several attractive vectors may influence the camera at one time their algebraic vector sum should also be scaled down, with scalar coefficient α_i as to respect superior influence of the user intensions (Eq. (17)).

$$\left\|\sum_i \left(\vec{G}_{att_i}\cdot\alpha_{2_i}\oplus\vec{G}_{loc_i}\cdot\alpha_{3_i}\right)\right\| < \left\|\vec{K}\cdot\alpha_1\right\|. \tag{17}$$

Author researches discovered that with growing experience constant background user aiding becomes unwilling but consciously and interactively launched, camera position modifications, can be smartly and eagerly exploited. As a consequence α_2 and α_3 should be diminished with experience gaining. Preferably it should be user controlled by means of some configuration panel or it should be interactively launched in a favourable context (i.e. by means of mouse button press).

If environment exploration designers plan to attract the user attention towards objects' specified elements/points of interest rather than the whole objects as such, local potential fields gradients $\vec{G}_{loc}$may be privileged (Fig. 4). Their influence, controlled with α_3, should be stronger than general field influence controlled with coefficient α_2. If local field gradient is to play a minor role it might be incorporated into general field influence which is controlled with α_2 coefficient. Then it forms global attractive influence with some local points of interest.

In spite of the camera position, potential field gradients might be used for camera view direction $(\vec{dir})$ modification (Eq. (14)). This subtle modification should be performed with a special care of user perception and navigational satisfaction. As view direction is a valid camera orientation vector it must be remembered that rapid, unintended vector direction changes may invoke loss of user spatial orientation [24]. That is why default (system controlled) view direction changes (Eq. (14)) should come out of local potential vectors $\vec{G}_{loc}$ rather than global attracting gradients $\vec{G}_{att}$. It should just help the user achieve requested view direction of approached point of interested.

Experiments (tests section) revealed that virtual camera view direction attracting mode can be triggered by the user while exploration and should cumulate the user's intention – it should be applied gradually achieving its maximum predefined value (β_1 coefficient in Eq. (14)) exponentially or at least linearly in time. Such approach eliminates rapid view direction changes and appears only when the user expects such camera behaviour. For example it can be triggered by any user interface output signal revealing intention of view direction modification (i.e. the user presses side keyboard arrows while navigation, marking his intention to modify the view direction).

Consequently the view direction can be also modified by both global and local attractive gradients respectively scaled with β_2 and β_3 coefficients. As global attraction seemed to play the minor role, local view modifications, especially vertical ones, were declared to be very useful. Conducted tests have combined both horizontal and vertical influence at the same level. Simultaneously with camera vertical expected adjustments users reported unintended horizontal camera modifications. Thus these both directions of influence should be separated and treated individually with superior role of vertical influence.

The above discussion consciously limits the camera altitude modification as the usability range of the presented method strictly depends on a computational power of the computer. Contemporary computers equipped with Graphics Processing Unit can easily perform necessary computations [18]. A greater range of camera height modifications causes the necessity of additional potential field distribution generation, their consequent storing (preferably as textures on graphics card) and processing. Such decision must be made under consideration of system requirements and computational power of dedicated computers. Nevertheless, there are many optimisations and improvements possibilities especially in implementation stage.

7. Conclusions

The new, potential field based method provides many flexible scenarios of the camera navigation support. It can aid camera collisions, object's specific element attractions or even user view modifications. Adequately balanced field gradients allow users for unconstrained camera movement with simultaneous obstacle avoidance and efficient information gathering process. The performed tests have justified not only very high method acceptance, but have proved its usability in the most frequent virtual environment navigational tasks like: search, exploration and manoeuvring.

The classification of tested users suggests that especially new virtual environment explorers may take great advantage of the method, whereas proficient navigators require special attention as the method may contradict their user interface habits. New virtual environment users like to be supervised (supported) more strongly than advanced ones. Virtual environment practitioners prefer contextual and self controlled, interactive method usage. Fortunately, the potential field-based method of a navigation support, presented in this paper, can be easily profiled and tuned according to specific user requirements and expectations. Due to clear and foreseeable coefficients it does not need to be programmed in advance, but can be adjusted, individually by users, in an application configuration stage.

REFERENCES

[1] D.A. Bowman, "Principles for the design of performance-oriented interaction techniques", in *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Ass., Mahwah, New Jersey, 2002.

[2] D.J. McFarland and J.R. Wolpaw, "Brain-computer interfaces for Communication and Control", *Communication ACM* 54 (5), 60-=66 (2011).

[3] E. Foxlin, "Motion tracking requirements and technologies", in *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Ass., Mahwah, New Jersey, 2002.

[4] J. Wilder, G.K. Kung, M.M. Tremaine, and M. Kaur, "Eye tracking in virtual environments", in *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Ass., Mahwah, New Jersey, 2002.

[5] W.T. Nelson and R.S. Bolia, "Technological considerations in the design of multisensory virtual environments: the virtual field of dreams will have to wait", in *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Ass., Mahwah, New Jersey, 2002.

[6] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.F. Ma, C. Kirbas, K.E. McCullough, and R. Ansari, "Multimodal human discourse: gesture and speech", *ACM Trans. on Computer-Human Interaction* 9 (3), 171–193 (2002).

[7] J.P. Wachs, M. Kolsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications", *Communication ACM* 54 (2), 60–71 (2011).

[8] A. Wojciechowski, "Hand's poses recognition as a mean of communication within natural user interfaces", *Bull. Pol. Ac.: Tech.* 60 (2), 331–336 (2012).

[9] G. Glonek and M. Pietruszka, "Natural user interfaces (NUI): review", *JACS* 20 (2), 27–46 (2012).

[10] K. Szabat, T. Orłowska-Kowalska, and K.P. Dyrcz, "Extended Kalman filters in the control structure of two-mass drive system", *Bull. Pol. Ac.: Tech.* 54 (3), 315–325 (2006).

[11] K. Guzek and P. Napieralski, "Measurement of noise in the Monte Carlo point sampling method", *Bull. Pol. Ac.: Tech.* 59 (1), 15–19 (2011).

[12] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks", *Proc. Int. Conf. Human Factors in Computer Systems CHI* 91, 237–242 (1991).

[13] A. Sandberg, "Gesture recognition using neural networks", *Technical Report*, KTH University, Stockholm, 1997.

[14] K. Symeonidis, "Hand gesture recognition using neural networks", *Neural Networks* 13, 5–10 (1996).

[15] A. Niewiadomski, "On finity, countability, cardinalities, and cylindric extensions of Type-2 fuzzy sets in linguistic summarization of databases", *IEEE Trans. on Fuzzy Systems* 18 (3), 532–545 (2010).

[16] D. Rutkowska, *Intelligent Computational Systems: Genetic Agorithms and Neural Networks in Fuzzy Systems*, PLJ, Warszawa, 1997, (in Polish).

[17] L. Rutkowski, *Methods and Techniques of Artificial Intelligence*, PWN, Warszawa, 2012.

[18] A. Wojciechowski and G. Wróblewski, "GPU calculated camera collisions detection within a dynamic environment", *LNCS* 6375, Springer, 2010, (in Polish).

[19] A. Wojciechowski, "Potential field based camera collisions detection within dynamically moving 3D environment", in *LNCS* 5337, Springer, Berlin, 2009.

[20] A. Wojciechowski, "Potential field based camera collisions' detection in a 3D environment",*Int. J. MGV* 15, 665–672 (2006).

[21] M.R. Mine, "Virtual environment interaction techniques", *Technical Report* 95-018, Department of Computer Science, University of North Carolina, Chapel Hill, 1995.

[22] D.A. Bowman and L.F. Hodgas, "Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments", *J. Visual Languages and Computing* 10 (1), 37–53 (1999).

[23] D.A. Bowman, "Interaction techniques for common tasks in immersive virtual environments", *Ph.D. Thesis*, Georgia Institute of Technology, Georgia, 1999.

[24] R.P. Darken and B. Peterson, "Spatial orientation, wayfinding and representation", in *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Ass., Mahwah, New Jersey, 2002.

[25] J.C. Latombe, *Robot Motion Planning*, Kluwer, Amsterdam, 1991.

[26] R.R. Murphy, *Introduction to AI Robotics*, MIT Press, Cambridge, 2000.

[27] I. Dulęba, *Methods and Algorithms of Motion Planning of Manual and Manipulative Robots,* Exit, Warszawa, 2004, (in Polish).

[28] I. Dulęba, "Impact of control representations on efficiency of local nonholonomic motion planning", *Bull. Pol. Ac.: Tech.* 59 (2), 213–218 (2011).

[29] Z. Hendzel, "Collision free path planning and control of wheeled mobile robot using Kohonen self-organizing map", *Bull. Pol. Ac.: Tech.* 53 (1), 39–47 (2005).

[30] K.Z. Haigh and M. Veloso, "Route planning by analogy", *Proc. Int. Conf. on Case-Based Reasoning* 160–180 (1995).

[31] T.Y. Li and H.K. Ting, "An intelligent user interface with motion planning for 3D navigation", *Proc. Conf. IEEE Virtual Reality 2000* 1, 177–184 (2000).

[32] C. Urdiales, A. Bandera, F. Arrebola, and F. Sandoval, "Multilevel path planning algorithm for autonomous robots", *Electronics Letters* 34 (2), 223–224 (1998).

[33] C. Urdiales, A. Bandera, E.J. Perez, A. Poncela, and F. Sandoval, "Hierarchical planning in a mobile robot for map learning and navigation", *Autonomous Robotic Systems – Soft Computing and Hard Computing Methodologies and Applications,* eds. D. Maravall, D. Ruan, and C. Zhou , Springer, Berlin, 2003.

[34] J.S. Zelek, "Dynamic issues for mobile robot realtime discovery and path planning", *Proc. Int. Conf. Computational Intelligence in Robotics and Automation* 1, 232–237 (1999).

[35] S. Bandi and D. Thalmann, "Space discretization for efficient human navigation", *Proc. Eurographics* 17, 195–206 (1998).

[36] S.M. Drucker and D. Zelter, "Intelligent camera control in a virtual environment", *Proc. Graphics Interface* 1, 190–199 (1994).

[37] P.K. Egbert and H. Winkler, "Collisions-free object movement using vector fields", *IEEE Computer Graphics and Applications* 16 (4), 18–22 (1996).

[38] S.S. Ge and Y.J. Cui, "Dynamic motion planning for mobile robots using potential field method", *Autonomous Robots* 13, 207–222 (2002).

[39] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *Int. J. Mobile Research* 5 (1), 90–99 (1986).

[40] M. Khatib and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions", *Proc. Int. Conf. Intelligent Autonomous Systems IAS* 4, CD-ROM (1995).

[41] D. Xiao and R. Hubbold, "Navigation guided by artificial force fields", *Proc. Int. Conf. CHI'98* 1, 179–186 (1998).

[42] T.Y. Li and H.C. Chou, "Improving navigation efficiency with artificial force fields", *Proc. Int. Conf. Computer Vision, Graphics and Image Processing* 1, 1–7 (2001).

[43] T.Y. Li and S.W. Hsu, "An intelligent 3D user interface adapting to user control behaviours", *Proc. 9th Int. Conf. Intelligent User Interface* 1, 184–190 (2004).

[44] S. Beckhaus, "Dynamic potential fields for guided exploration in virtual environments", *Ph.D. Thesis*, Otto-von-Gueicke Universitate, Magdeburg, 2002.

[45] S. Beckhaus, F. Ritter, and T. Strothotte, "CubicalPath – dynamic potential fields for guided exploration in virtual environments", *Proc. IEEE Int. Conf. Computer Graphics and Applications PG'00* 1, 389–457 (2000).

[46] R.C. Arkin, "Motor scheme-based mobile robot navigation", *Int. J. Robotics Research* 8, 92–112 (1989).

[47] B. Bederson, J. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, "Pad++: a zoomable graphical sketchpad for exploring alternate interface physics", *J. Visual Languages and Computing* 7, 3–31 (1996).

[48] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots", *IEEE Trans. on Systems, Man and Cybernetics* 1 (5), 1179–1187 (1989).

[49] S. Beckhaus, F. Ritter, and T. Strothotte, "Guided exploration with dynamic potential fields: the cubical path method", *Computer Graphics Forum* 20 (4), 201–210 (2001).

[50] J. Barraquand and J.C. Latombe, "Robot motion planning: a distributed representation approach", *Technical Report*, Department of Computer Science, Stanford University, Stanford, 1989.

[51] S. Beckhaus, G. Eckel, and T. Strothotte, "Guided exploration in virtual environments", *Proc. Int. Conf. Electronic Imaging'01* 4297, 426–435 (2001).
[52] D. Benyon and K. Hook, "Navigation in information spaces: supporting the individual", *Proc. Int. Conf. Human Computer Interaction: INTERACT'97* 1, 39–46 (1997).
[53] J. Nielsen, "Finding usability problems through heuristic evaluation", *Proc. Int. Conf. ACM CHI'92* 1, 373–380 (1992).
[54] J. Nielsen, *Usability Engineering*, Academic Press, New York, 1993.
[55] D. Bowman, J. Gabbard, and D. Hix, "A survey of usability evaluation in virtual environments: classification and comparison of methods", *Presence: Teleoperators and Virtual Environments* 11 (4), 404–424 (2002).