

(51) International Patent Classification:
G06T 7/20 (2006.01)(21) International Application Number:
PCT/EP2010/059222(22) International Filing Date:
29 June 2010 (29.06.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2,670,367 29 June 2009 (29.06.2009) CA(71) Applicant (for all designated States except US): **INSTITUT PASTEUR** [FR/FR]; 25-28, rue du Docteur Roux, F-75015 Paris (FR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **DE CHAUMONT, Fabrice** [FR/FR]; 74, rue de la Colonie - Appart. 6E, F-75013 Paris (FR). **OLIVO-MARIN, Jean-Christophe** [FR/FR]; 3, Boulevard Colbert, F-92330 Sceaux (FR).(74) Agent: **TEXIER, Christian**; Cabinet Regimbeau, 20, rue de Chazelles, F-75847 Paris Cedex 17 (FR).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

— with international search report (Art. 21(3))

(54) Title: METHOD FOR TRACKING OBJECTS

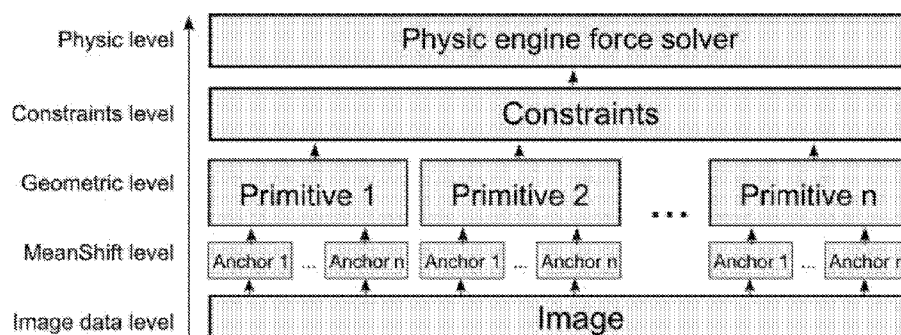


FIG. 2

(57) **Abstract:** The present invention relates to a method for tracking multiple objects. More particularly, the present invention relates to a method of tracking articulated objects being displaced, deformed or undergoing frequent contacts in a video sequence. There is provided a computer-implemented method for generating a model of a physical object based on a source image and providing a physics engine to cause the model to be transformed according to physical constraints or the like, in order to render a rich geometric information about the object at the articulation level, to distinguish objects when they collide, and to provide a near real-time simulation of the moving physical object.

5

1

METHOD FOR TRACKING OBJECTS

FIELD OF THE INVENTION

The present invention relates to a method for tracking objects. More particularly, the
10 present invention relates to a method for tracking one or more articulated objects,
which typically undergo frequent contacts in a video sequence.

BACKGROUND OF THE INVENTION

The tracking of multiple objects is for example useful for extracting behavioral statistics
15 from the co-habitation of animals, such as laboratory mice. Such work may require
identifying particular events occurring between the mice and is conventionally done
from a video recording and with a timer. An important number of playbacks on the
same sequence of the video recording are necessary to label the different events.
Moreover, these events having a very short duration, they are often labeled in over-
20 sized time frames. Thus there is a need for an improved system for tracking such
objects and events.

Single and multiple object tracking in video sequences is a field of active research in
computer vision, with numerous applications to scene surveillance. For example, known
to the Applicant are: (1) B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-
25 time computer vision system for vehicle tracking and traffic surveillance" Transportation
Research: Part C, vol. 6, pp. 271-288, 1998; (2) I. Cohen and G. Medioni, "Detecting
and tracking moving objects for video surveillance" in Proceedings of the IEEE
Conference on Computer Vision and Pattern Recognition, 1999; (3) D. Comaniciu, V.
Ramesh, and P. Meer, "Kernel-based object tracking" IEEE Transactions on Pattern

5 Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003; and (4) T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1208-1221. September 2004.

Throughout the literature, two sets of approaches have been studied, namely two-step
10 approaches and direct approaches. Two-step approaches perform a preliminary detection of the objects on each sequence frame, and then reconstruct the objects trajectories using various association techniques. For example, known to the Applicant are: (1) P. Willet, R. Niu, and Y. Bar-Shalom, "Integration of Bayes detection with target tracking," IEEE Transactions on Signal Processing, vol. 49, no. 1, pp. 17–29, January
15 2001; (2) M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002; and (3) N. Chenouard, I. Bloch, and J.C. Olivo-Marin, "Feature-aided particle tracking," in Proceedings of the IEEE International Conference on Image Processing, San Diego, Oct. 2008, IEEE, in
20 press.

Such methods are however not adapted to track articulated objects such as animals, as they focus more upon the object trajectory rather its geometrical conformation. Direct approaches perform both detection and tracking simultaneously by iteratively adapting user-defined models to match the objects, and repeat this process on each frame to
25 follow all targets. Popular candidates in this approach include mean-shift and active contour techniques. The following references teach mean-shift techniques: (1) Comaniciu et al.; and (2) C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005. The following references teach active
30 contour techniques: (1) A. Blake and M. Isard, Active contours, Springer, 2000; and

5 (2) C. Zimmer and J.-C. Olivo-Marin, "Coupled parametric active contours," IEEE Transactions Pattern Analysis Machine Intelligence, vol. 27, no. 11, pp. 1838–1842, 2005.

Mean-shift based techniques (e.g. (1) Comaniciu et al.; (2) C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure", in
10 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005) are widely used to track whole objects in real-time in natural environments thanks to their computational efficiency. In biological imaging, such methods have been employed to track multiple deforming cells throughout a sequence (e.g. O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker, "Tracking of migrating cells under phase-
15 contrast video microscopy with combined mean-shift processes," Medical Imaging IEEE Transactions on, vol. 24, no. 6, pp. 697-711, 2005). Yet, mean-shift approaches have two important limitations. Firstly, tracking ambiguities quickly arise when similar objects come into contact, where concurrent models are attracted toward a common local maximum and fuse until the objects separate again. This association problem is
20 common in many other applications, and additional methods must be employed to solve these ambiguities (e.g. A. Genovesio, T. Liedl, V. Emiliani, W.J. Parak, M. Coppey-Moisan, and J.-C. Olivo-Marin, "Multiple particle tracking in 3D+T microscopy: methods and application to the tracking of endocytosed quantum dots", IEEE Transactions on Image Processing, vol. 15, no. 5, pp. 1062-1070, 2006). Secondly,
25 mean-shift and CAMShift approaches (e.g. G.R. Bradski, "Real time face and object tracking as a component of a perceptual user interface", IEEE Workshop on Applications of Computer Vision, pp. 214-219, 1998) are able to track objects as a whole, accommodating global object transformations such as scale, orientation or pose changes, but do not quantify local shape information and thus cannot monitor
30 deformations at the articulation level. Despite their efficiency for multi-object tracking, this remark also holds for active contour methods (e.g. C. Zimmer and J.-C. Olivo-

- 5 Malin, "Coupled parametric active contours", IEEE Transactions Pattern Analysis Machine Intelligence, vol. 27, no. 11, pp. 1838-1842, 2005) , which are also computationally more demanding.

Mean-shift and active contour methods require a higher acquisition frame rate to allow each model to adapt locally to the new object position, and are well suited for objects
10 undergoing frequent shape or pose changes. Unfortunately, these methods only provide global localization and conformation of the objects, and fail to provide shape information at the articulation level. This is a limiting factor when tracking multiple touching object in behavioral studies, where in-depth shape information is necessary to extract objects interaction events. Debeir et al. teaches a mean-shift-based method to
15 track multiple cells observed in phase-contrast microscopy, exhibiting a particular two-phase intensity profile (bright center and dark boundary). This pattern was modeled using multiple mean-shift kernels stitched together in a rigid way, and the displacement of the global model was driven by the combination of all the underlying local mean-shift processes. While the method could distinguish touching cells efficiently, the model
20 rigidity failed to provide shape information about the tracked cells.

Hence, in light of the aforementioned, there is a need for an improved system which, by virtue of its design and components, would be able to overcome some of the above-discussed prior art concerns.

25 SUMMARY OF THE INVENTION

According to an aspect of the present invention, there is provided a computer-implemented method for tracking a physical object composed of one or more physical primitive(s). The method comprises the following steps: (a) receiving source image data being representative of the physical object at a time t_n ; (b) identifying, in the source

5 image data, the object and said one or more physical primitive, and associating a source geometrical primitive to each of said one or more physical primitive, thereby defining a model of the object according to a source configuration; (c) receiving target
10 image data being representative of the physical object at another time t_{n+1} ; (d) identifying, in the target image data, the object and said one or more physical primitive, and associating a target geometrical primitive to each of said one or more physical
primitive, thereby defining the model of the object according to a target configuration; (e) associating to each of said source geometrical primitive a corresponding target
15 geometrical primitive; and (g) transforming said model, according to predetermined rules being representative of physical forces, from the source configuration to the target
configuration, to simulate the real-life movement of the physical object between the
times t_n and t_{n+1} .

In embodiments of the present invention, limitations of the art are addressed by proposing a novel approach for multi-object tracking using physics-constrained models. The principle is to define a model of the target using geometrical primitives that are
20 preferably linked via constraints, being representative of physical constraints. Each primitive of the model is then linked to the image data using an appropriate technique such as a standard mean-shift process or quadtrees, and the global model transformation is performed in real-time or at least near-real-time, preferably using a physics engine solver. Such engines can be used to perform object tracking in video
25 sequences. By representing the objects using physically constrained models, and by linking these models to the image data, the engine solver is able to track the evolution of all objects simultaneously, even if they come into contact over time

As a result, the proposed approach is able to, according to embodiments thereof:

- a) track objects in a very robust way;

- 5 b) maintain the identity of multiple touching objects;
- c) provide local shape information at the articulation level through each of its geometrical primitives.

Other features and advantages of the invention will be better understood upon a reading of preferred embodiments thereof, given for the purpose of exemplification
10 only, with reference to the appended drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a physics model of a mouse. (a): description of the model bodies (solid lines) and the boundary of their attractive maps (dashed lines). Grey bodies are not allowed
15 to overlap with grey bodies of another model. (b): final model matching a mouse in a sequence frame.

FIG. 2 is a diagram representing the system of the method according to an embodiment of the present invention.

FIG. 3 shows matching mice in close contact without (a, b) and with (c, d) proximity
20 handling. With proximity handling (c, d), the overlap between attractive maps of same nature is removed from the computations, yielding a correct adjustment of the models.

FIG. 4 is shows a sequence of eight (8) images taking advantage of motion prediction. On each frame, the grey model is the current matching model for the frame, and the white model is the initialization which will be used for the next frame.

25 FIG. 5 is a block diagram showing steps involved in the tracking method according to an embodiment of the present invention.

5 FIG. 6 shows the tracking of two sporozoites observed in reflection interference contrast microscopy. (a): sample frames of the original sequence. (b): physics model of the sporozoite.

FIG. 7 shows tracking results on the sporozoite sequence. Each line represents one sporozoite. Displacement values are given in pixels for each frame.

10 FIG. 8 shows force vectors computed for each model anchor. (a): original image. (b, c): force vectors super-imposed on a gradient map (b) and binary map (c) of the image. (d): vectors and boundaries of their attractive maps.

FIG. 9 shows a mouse with a contracted posture (a) and with the head on the side (c). (b, d): final mouse model super-imposed on the original images.

15 FIG. 10A is an image of a mouse (original image).

FIG. 10B is the image of 10A, with a model of the mouse shown super-imposed on the original image.

FIG. 11A is a first of a sequence of image frames showing two (2) mice.

20 FIG. 11B is the image frame shown in FIG. 11A, further displaying corresponding models super-imposed on the respective mice.

FIG. 12A is a second of the sequence of image frames.

FIG. 12B is the image frame shown in FIG. 12A, further displaying corresponding models super-imposed on the respective mice.

FIG. 13A is a third of the sequence of image frames.

25 FIG. 13B is the image frame shown in FIG. 13A, further displaying corresponding

5 models super-imposed on the respective mice.

FIG. 14 is a timeline of the labelled events found on a 40-second sequence (530 frames). In this particular sequence, one can note that the "head 1-genital 2" event never occurs.

10 FIG. 15 is a schema showing a computation according to three (3) different quads. The main final computation: a) using quadtrees, b) standard method, c) quadtree example over a large anchor. The main final computation is made from each biggest quad of the quadtree instead of all pixels.

15 FIG. 16 is a schema showing a scale propagation of an attractive map at different scales (original map: 512x512): A) Representation of ψ_i (with intensity rescale); B) vector representation of $b_{s,i}$ for each possible quad of the main quadtree.

FIG. 17 is a schema showing a cumulative map, each number representing the number of anchors owning the colored area. A quadtree is computed for each different area.

FIG. 18 is a graph showing the number of computation needed to compute f_a by the radius of the anchor.

20

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

In the following description, the same numerical references refer to similar elements. The embodiments mentioned and/or geometrical configurations and dimensions shown in the figures or described in the present description are embodiments of the present
25 invention only, given for exemplification purposes only.

5 Broadly described, the present invention provides a computer-implemented method for tracking one or more physical object composed of one or more physical primitive.

Methods according to embodiments of the invention may be used in a variety of contexts. In one example, the method may be applied to the study of behavioral statistics from the co-habitation of animals, such as laboratory mice. With reference to
10 FIG. 1, each mouse is considered to be one physical object **O** and articulated body parts of the mouse being considered as a distinct physical primitive **P**, as will be explained further below. In another example, the method may be applied to track one or more objects composed of a single deformable physical primitive such as cells or the like from microscopic observations. Numerous other applications could be envisioned,
15 such as, to name just a few, the tracking of players on a sports playing field, the monitoring of traffic, the studying of the movements of a dancer, etc.

In one of its embodiments, the present invention tackles the problem of tracking multiple articulated objects undergoing frequent contacts in a video sequence. Conventional tracking methods usually fail to distinguish objects during contact,
20 implying the use of disambiguation techniques to recover the identity of each object. Moreover, such methods do not provide detailed shape information at the articulation level. The inventors address these limitations by proposing a novel approach to track multiple articulated objects by defining a model of each object using geometrical primitives and physical constraints in a physics engine. Discussed herein is a method
25 applied to track mice observed with a webcam and to track two sporozoites observed in reflection interference contrast microscopy, highlighting the flexibility and genericity of the framework.

Physics engines are software or hardware toolkits dedicated to the realistic simulation of physical systems governed by Lagrangian dynamics. They first appeared over 30
30 years ago, but only received extensive attention in the last decade. Due to technical

5 limitations, early engines were tailored for very specific applications, and had therefore very limited interest. Nowadays, their evolution is mostly driven by the computer-gaming industry (e.g. J. Garca de Jaln and E. Bayo, Kinematic and dynamic simulation of multibody systems. The real-time challenge, Springer, 1994), striving to produce even more physically realistic video games. As a result, numerous commercial and free APIs
10 have been developed in recent years for the simulation of highly complex systems, addressing delicate problems such as collision detection between articulated objects in an efficient way. Many engines are now integrated within modern graphics cards, benefiting from multiple hardware optimizations and parallel implementations which are crucial for real-time applications. Known to the Applicant is I.A. Kakadiaris, D. Metaxas,
15 and R. Bajcsy, "Active part decomposition, shape and motion estimation of articulated objects: a physics-based approach," in CVPR, Jun 1994, pp. 980–984, wherein physic considering rotation and translation are used to learn and track the articulation of a 2D-represented human arm. The present application generalizes the tracking procedure and shows how to exploit such engines to track articulated objects in video sequences.

20 The method according to embodiments of the invention first includes a step of receiving source image data being representative of the physical object O at a time t_n . Preferably, the source image data is extracted from a still frame of a video sequence recording movement of the object O . As will be readily understood by one skilled in the art, the source image data may include one or more physical object to track, and the method
25 described herein can be used to track multiple objects in a same video sequence by performing the steps below for each such objects.

The method next involves defining a model 20 of the object O according to a source configuration (i.e. an initial position and/or shape) by identifying, in the source image data 10, the object O and its physical primitive(s) P or member(s), and associating a
30 source geometrical primitive 12 to each physical primitive P . Physical primitives P may

5 be for example, in the case of a mouse: a body **B**, a head **H**, legs **A1**, **A2**, **A3**, **A4**, etc. Basic geometrical primitives include rectangles, circles and other convex polygons, while more complex objects can then be constructed as combinations of these primitive bodies.

Similarly, target image data representative of the physical object at another time t_{n+1} is
10 received and target geometrical primitives are associated to each physical primitive identified therein, thereby defining the model **20** of the object according to a target configuration. Each source geometrical primitive **P** is associated to a corresponding target geometrical primitive and the model **20** is transformed, according to predetermined rules being representative of physical forces, from the source
15 configuration to the target configuration, to simulate the real-life movement of the object between the times t_n and t_{n+1} . Thus the model appears to move realistically (as per the predetermines rules) between the images taken at times t_n and t_{n+1} . Preferably, t_{n+1} becomes the new t_n and the method is thus repeated for a subsequent pair of images frames, and so on an so forth for a sequence of image frames obtained from a video
20 recording.

By "model", it is meant a systematic representation of a physical object that shares important characteristics with the physical object. Such a model may be material, visual, mathematical, or computational. In the embodiments, described herein, the model is provided graphically in the form of the above-mentioned geographical
25 primitives.

Moreover, the "geographical primitives" may be represented graphically or in any other suitable data form depending on the particular application, as can be understood by a person skilled in the art. Indeed, it may be desirable to compute the model in data form and to output to a user, only selected information on certain events occurring to this
30 model.

5 Also, by "movement", it is meant a displacement, a trajectory, a deformation, a change in configuration, a reorientation and/or the like.

The term "physical forces" refers to any force existing in the physical world.

Preferably, the transforming step is performed by a physics engine. Physics engines define a model M in a world W . A model $M_{O,C}$ is defined by a set $O = \{O_i\}_{i=1..n}$ of basic
10 geometrical primitives, each characterized by a mass, a damping (i.e. friction) factor, and an anchor where displacement forces will be applied. The geometrical primitives are then linked together, preferably pair-wise, by constraints 14 called *joints*, describing the degree of freedom of the articulation considered. In classical physics engines, one may use joints of at least four types: distance, angle, elastic and slider. Any other type
15 of suitable joint may be provided, as can be understood by a person skilled in the art. Collisions between geometrical primitives of a given model and between geometrical primitives of distinct models are both handled by the physics engine, using various built-in algorithms that either favour computation time or physical realism.

There are traditionally two types of physics engines, depending on how the bodies are
20 handled. *Rigid-body-engines* consider bodies through their centers only, yielding simple management, but requiring an additional rotation parameter to account for all types of movements. *Mass-aggregate-engines* handle bodies as a set of little masses (i.e. a box is described by 8 masses representing its corners). Using this latter representation, all movements can be described without any additional parameter, though more
25 computations are required to induce a correct displacement of all components of the body.

In the following section, the inventors describe how the initial tracking problem can be expressed as a set of local mean-shift processes applied to individual bodies, and combined into a global physically-constrained system to exploit the physics engine

5 solver, as represented in the diagram shown in FIG. 2. To the best of the inventors' knowledge, this is the first time that such an approach is taken, thereby bridging the gap between physics engines and image-based object tracking.

Physics-constrained mean-shift

An image-based physics model

10 In order to exploit image data to attract the physics model toward the position of the physical object (i.e. the target configuration), the inventors define a relationship between each body anchor and the image data. An anchor a is thus defined by a point (or position) p_a , as previously mentioned, and the associated attraction map M_a is computed from the original image I . Depending on the image information that is
15 exploited, one can define multiple attraction maps, and use a different map for each anchor. Each map is then restricted to a sub-region of the image centered on the anchor.

An image-drive attraction force is then computed for each of its individual geometrical primitive(s), and the physics engine solver combines these forces with the physical
20 constraints imposed by the model joints in order to compute the final displacement, shape, deformation and/or configuration. Firstly, there is provided a pool of attraction maps $M = \{m_i\}_{i=1..n}$ which may be either computed from the original image using a filter-like method, a quadtree method, or as in traditional mean-shift, given by the look up table of a predefined feature space. In the present example, only two attraction maps
25 are used: a binary mask of the image is used to attract the center of the model toward the center of the image object, and an edge map is used to attract peripheral bodies toward the object boundary. The maps are computed once for each sequence frame and are not updated during the model deformation. The attraction maps data is then converted into physical forces and applied to the model through its anchors. Each

5 anchor a (defined by its position p_a) is assigned an appropriate attraction map m taken from M . Then, this map is restricted to a subset of pixels centered on the anchor (noted Q_a). In the present case, this subset is defined as a simple disk of arbitrary radius around the anchor position, although more complex gating kernels may be employed.

Finally, for each anchor a , one computes the resulting attraction force $\vec{f}(a)$ as follows:

$$10 \quad \vec{f}(a) = \sum_{q \in Q_a} q \overrightarrow{p_a q},$$

where q describes all pixels within the attractive map.

Preferably, once all the forces have been computed for each anchor, they are sent to the physics engine, which in turn combines these forces with the physical constraints of the model. The deformation is an iterative process, therefore attraction forces are
 15 recomputed at each iteration until a stop condition is reached (i.e. to detect when the system reaches equilibrium). A classical strategy is to detect when the system reaches a steady-state, where the speed of all models falls below a small value ϵ , according to:

$$\left\| \sum_{a \in \mathcal{M}} \vec{f}(a) \right\| < \epsilon,$$

where ϵ is a substantially small constant.

20 If each model can be converted to a probabilistic distribution, another way to detect convergence is to measure the distance between each model distribution and some reference distribution, e.g. using the commonly-used Bhattacharyya distance (e.g. Zhao et al).

5 Example: Physics model of the mouse

Referring now to FIG. 1, a physics model 20 of a mouse **O** is defined as follows: the bodies or members **H**, **B** and **N** are the head, the belly and the neck of the mouse (i.e. physical primitives), respectively. As the mouse can elongate or contract itself, the inventors allow the neck to slide along the belly, constrained between parts **C** and **D** to
10 define the degree of freedom. A sliding joint is added to bound the distance between the belly and the neck. The head is a simple circle and is linked to the neck by a distance joint 14. The model thus has two degrees of freedom: a) one for the sliding neck simulating mouse elongation or retraction, and b) one for the head rotating at the end of the neck. In order to prevent overlap between different models, the inventors
15 impose that all bodies colored in gray may not overlap with similar bodies of another model (see FIG. 8 for an illustration of mice contact situation). Anchors **H** and **A1** through **A4** exploit an attractive map based on intensity gradients, while **B** exploits an attractive map that is a binary image highlighting dark-image regions (i.e. reflecting the presence of a mouse. Dashed circles depict the boundary of the attractive maps 18.
20 Finally, **T1** to **T4** are bodies defining the mouse tail, and are linked by a distance joint. In the current model, they are not linked to any attractive map. The final model is shown super-imposed on a mouse image in FIG. 9. Similarly, an image of a mouse is shown in FIG. 10A, the same image being shown in FIG. 10B, with a model being super-imposed on the mouse.

25 Relation with the mean-shift approach

The principle of mean-shift tracking (e.g. Zhao et al) is to move a model toward a local maximum defined by a kernel. In one embodiment of the inventors' method, the model corresponds to a body of the inventors' physics model, and the associated kernel is analogous to the attraction map associated to the body anchor. The iterative process

5 that seeks for the local maximum leads to a translation vector that is transformed into a force for the physic system. The model of the physical object is then moved by the physic engine which preferably considers the whole forces and constraints of the model instead of the simple translation vector given at the mean-shift level. It is this step which gives the control of the mean-shift results to the physic engine. The present approach
10 incorporates physical constraints between bodies of a same physics model, thereby improving object detection and tracking using a higher level of semantic information.

Application to multi-object tracking

The appealing interest of the physics model approach is that an arbitrary number of models can be defined in the same world and deformed simultaneously using the
15 physics engine solver. One can thus perform concurrent multi-object tracking, where each model follows a given target in the image according to the displacement of each of its constitutive objects. In case two objects come into contact, ambiguities related to model overlapping are automatically detected and handled by the physics engine through efficient collision detection and feedback strategies.

Dealing with object proximity

While model overlapping is forbidden, it may occur that anchors of same nature are sufficiently close such that their attraction maps overlap. Consequently, an identical portion of image information is exploited to compute the movement of concurrent
25 anchors, potentially attracting the models toward wrong positions, as illustrated in FIG. 3, wherein (a) shows the tracking of mice being in close contact, via corresponding models. In images (a) and (b), no proximity handling is used. With proximity handling, as shown in images (c) and (d), the overlap between attractive maps of same nature is removed from the computations, yielding a correct adjustment of the models. The inventors remove the ambiguity by masking all attractive maps of same nature such

5 that all pixels shared by two or more maps are not involved in the computation of $\vec{f}$. For any given anchor a_i , the inventors compute the intersection between its attractive map Ma_i and the attractive map of all other anchors a_j (provided the maps are of same nature), and exclude this intersection from Ma_i . Thus, the ambiguity is removed by weighting the influence of each computed vectors, according to the equation below, by
 10 introducing $\delta(q)$ such as it represents the number of anchor sharing q and having the same attractive map nature.

$$\vec{f}(a) = \sum_{q \in Q_a} \frac{q \overrightarrow{p_a q}}{\delta(q)}$$

Motion Prediction

For simple applications where the object does not move significantly from a frame to
 15 the next, tracking is performed straightforwardly by using the resulting model on a given frame to initialize its position in the next frame. Nonetheless, if the acquisition frame rate is not sufficient, the object may move fast enough such that its new position is no longer reachable by the model, as it is no longer visible within the attractive maps.

To cope with this situation, and more generally to reduce convergence time, the
 20 inventors take advantage of mass-aggregate engines to predict the position of the model for each frame but the first. The inventors recall that in such engines the whole object undergoes complex movement as a result of the constrained linear motion of each component. The inventors exploit this motion information by computing the speed vector for each anchor from frame $t - 1$ to frame t , and extrapolate this speed to obtain
 25 a predicted position on frame $t + 1$.

The motion prediction is preferably performed under the supervision of the physic engine. The inventors aim at finding the next position $t+1$ of the target knowing its

position at time t and $t-1$. The new object position is sought considering the object prolongs its previous motion (alternatively, a probability map of the next position may be generated). To do so, mass-aggregate engines are used. As explained, in such engines, the whole object undergoes complex movements as a result of the constrained linear motion of each component. FIG. 4 illustrates a sequence of eight (8) frames with a model using motion prediction. Frames (a), (b) and (c) illustrate rotation prediction while frames (c), (f) and (g) illustrate neck elongation caused by the head motion prediction which creates a pulling force on the neck. This motion information is used to compute the speed vector for each anchor from frame $t-1$ to frame t , and this speed is then applied to the model at time t in order to obtain the constrained predicted position on frame $t+1$. Physics engines offer the ability to bypass the speed computation made with forces and allow one to set directly a computed speed to anchors (*i.e. which find its motivation in impulse*). At the end of the convergence iterations, the following is applied to each anchor:

$$\forall a \in \mathcal{M}, \vec{v}(a_t) = \overrightarrow{a_{t-1}a_t}$$

Then the physic engine is iterated once. To ensure the current inertia will not interfere with the next image to process, each anchor's inertia is set to null vector. FIG. 5 shows the steps involved in the tracking method, according to an embodiment of the present invention.

RESULTS

Sporozoites tracking

Referring now to FIG. 6, an experiment consisting in tracking plasmodium berghei sporozoites observed in reflection interference contrast microscopy will be described. Although the sporozoites do not exhibit major shape changes, the difficulty with such

5 images is that local structures may not be visible, causing local detection methods to fail. In the present case, the inventors rely on the prior knowledge that this type of sporozoite moves in a pseudo-circular fashion. To improve the tracking performance, the inventors incorporate this shape information directly in the physics model, thereby denoting the flexibility of the framework. There is shown in FIG. 6, images of two
10 sporozoites being observed in reflection interference contrast microscopy, namely (a) sample frames of the original sequence and (b) a physics model of the sporozoite.

The sporozoite model thus comprises a set of anchors linked to each other by a distance and an angular joint, imposing a constant curvature to the model during its displacements. The sporozoites are then matched by computing attractive maps for
15 each anchor from a wavelet transform of the image (e.g. 10. J.-C.Olivo-Marin, Extraction of spots in biological images using multiscale products", Pattern Recognition, vol. 35, no. 9, pp. 1989-1996, 2002). As a result, the global model successfully tracks the whole sporozoites, even if some parts are invisible. FIG. 7 plots the measured sporozoites displacements in pixels for each sequence frame. Each line represents one
20 sporozoite.

Multiple mice tracking

As previously mentioned, the present invention may be applied, according to an embodiment thereof to extract behavioral statistics from the co-habitation of two mice evolving in a rectangular cage observed with a webcam. More particularly, the inventors
25 focus on labeling three types of events: oral-oral contact and oral-genital contacts (in both directions). To do so, the inventors must be able to determine at all times the position and orientation of each mouse. As previously mentioned, this quantification was previously done manually with a timer, and an important number of playbacks on the same sequence were necessary to label the different events. Moreover, these
30 events having a very short duration, they are often labeled in over-sized time frames.

5 FIGs. 11A to 13B illustrate frames of a recorded sequence, showing two mice moving from one frame to the other, each frame being shown as an image only (11A, 12A, 13A) and with corresponding models shown super-imposed on the respective mice detected (11B, 12B, 13B), according to an embodiment of the present invention.

Moreover, illustrative movies are available online at the following address:
 10 www.bioimageanalysis.org/8722. A first movie illustrates steps of convergence for each frame, and the motion prediction step (slowed down for viewing purpose). The second illustrates the whole sequence, according to a preferred embodiment of the present invention.

Validation

15 As mentioned earlier, convergence is detected on each frame when the model(s) reach a steady-state. It is however possible with mean-shift approaches that a model converges to a wrong position, e.g. right in the middle of two local maxima. To evaluate this behavior, a matching score function $s(M)$ ranging from 0 to 1 is defined, for each model, by:

$$20 \quad s(\mathcal{M}) = \frac{\sum_{a \in \mathcal{M}} \overline{Q_a}}{N_a(\mathcal{M})},$$

where $\overline{Q_a}$ is the average value of the pixels forming Q_a , and $N_a(M)$ is the number of anchors in the model. In case the model matches perfectly with the object, then $s(M) = 1$, this value decreasing otherwise.

25 The average matching score for our 530 frames sequence is 0.9 for each mouse, with a standard deviation of 0.1, showing that the mice were correctly followed all along the sequence. It is noted that the physical constraint layer increases the robustness of the

5 mean-shift process, since more information is combined during the deformation process, thereby reducing the risks of driving the model towards a wrong position. Additionally, if the attractive map linked to the anchor is the look up table of a feature space, then $s(M)$ can also be computed as the model and reference distributions, e.g. using the Bhattacharyya distance (e.g. Comaniciu et al.).

1.0 Results

Tracking results are illustrated in FIGs. 3, 4 and 8 to 13. The proposed method allows automatically detecting the position and orientation of the mice even when they are in close contact, and computing the proximity of the head and posterior of each mouse to label the events in an efficient way. The inventors have applied the method on a 530
1.5 frame sequence, acquired at a rate of 12.5 fps (frames per second), i.e. about 42 seconds long. The final timeline of events for this sequence is presented in FIG. 14. By virtue of the computational efficiency of the method according to the present embodiment, the method is able to process the whole sequence is processed in 55
seconds, i.e. yielding a processing rate of approximately 10 fps², which is very close to
2.0 real-time in the present case.

Computational performance

In order to compute the overhead of the physics layer on the overall performance, the proportion of processing time spent was measured, and it was found that only 3.5% of the time was consumed by the physics engine to solve all constraints and deform the
2.5 models. The main bottleneck of the method is the iterative computation of the forces to apply to each anchor which is almost solely dependent on the number of mean-shift processes necessary to solve (i.e. function of the number of models, of anchors per model, and of pixels per local attraction map). Though the physics constraint layers add a negligible computational overhead with respect to its numerous advantages, it may

5 be desirable to enhance the computational algorithm of the resulting force for each anchor. In the following section, an alternative embodiment of the present invention is described.

Multi-scale decomposition and quadtrees for improving the anchor computation

At each physics iteration step of the above described method, the attraction vector of
 10 the anchor must be re-computed, making this computation the bottleneck of the algorithm since there is an average of 30 computation steps per image to reach the system equilibrium. In such a process, less than 5 percent of the CPU is taken by the physics engine, and 80 percent is taken by the anchor's resulting force computation.

The anchor computation is slow since it computes the sum of all the vectors starting
 15 from the center of the kernel, using, as previously mentioned the following equation:

$$\vec{f}(a) = \sum_{q \in Q_a} q \overrightarrow{p_a q},$$

where q defines pixels (preferably all pixels) within the attractive map belonging to Q_a and p_a is the center of the anchor.

As also explained, in order to handle proximity detection, the influence of each
 20 computed vector is weighed by the number $\delta(q)$ of anchors sharing the same coefficient of attractive maps of same nature, according to:

$$\vec{f}(a) = \sum_{q \in Q_a} \frac{q \overrightarrow{p_a q}}{\delta(q)}$$

Both of the above equations require a computation which is needed at each single
 move of an anchor. This computation is very heavy since it does a full loop over the
 25 data at each computation of the final force.

- 5 There is shown in FIG. 15, a main final computation made from each biggest quad of the quadtree instead of all pixels, namely (a) the use of quadtrees, (b) the standard method, and (c) a quadtree example over a large anchor.

Thus, according to an alternative embodiment of the present invention, the speed of the computation of the anchor's final vector is improved in an exact manner and able to
 10 natively manage conflicting anchors. The key idea is to minimize the number of vector computations by creating a multi-scale attraction map, where the final anchor's vector is computed from the largest scale to the smallest one.

Building scales

It will be shown below that all scales are redundant in data, offering the ability to query
 15 the barycenter of a whole area at each different attraction-scale. The next section explains the use of extracting barycenter maps and intensity maps from attractive maps, as they can be extracted from the original vector computation.

Proof of reconstruction of the final vector by scales

There is provided an anchor $\vec{F}$ which is decomposed at point $\vec{X}_0 = (x_0, y_0)^T$, wherein
 20 the column vector $\vec{X}$ has Cartesian coordinates $[x, y]^T$.

$$\begin{aligned}
 \vec{F}(\vec{X}_0) &= \sum_{\vec{X} \in A} \left(v(\vec{X})(\vec{X} - \vec{X}_0) \right) \\
 &= \sum_{\vec{X} \in A} \left(v(\vec{X})\vec{X} - v(\vec{X})\vec{X}_0 \right) \\
 &= \sum_{\vec{X} \in A} \left(v(\vec{X})\vec{X} \right) - \sum_{\vec{X} \in A} \left(v(\vec{X})\vec{X}_0 \right) \\
 &= \sum_{\vec{X} \in A} \left(v(\vec{X})\vec{X} \right) - \vec{X}_0 \sum_{\vec{X} \in A} v(\vec{X}) \\
 &= \vec{B} - P\vec{X}_0
 \end{aligned}$$

5 where $P \triangleq \sum_{\vec{X} \in A} v(\vec{X})$ is the sum of the intensity values for all the pixels $\vec{X} = [x, y]^T$ in the path A , and
 $\vec{B} \triangleq \sum_{\vec{X} \in A} (v(\vec{X})\vec{X})$ is the mass center of the patch (weighed with intensity values).

This shows that: a) if the map B and P can be computed, the final vector from these maps and a center point X_0 , can be directly computed, and b) both maps B and P are
 10 computation-expensive since they loop over the pixels of the patch. Thus, according to the proposed embodiment, those maps are computed using multi-scale areas containing both B and P and queried through a quadtree algorithm.

Scales Computation

According to the present embodiment, dyadic scales are used, since they fit adequately
 15 with a quadtree exploration. The scales are computed from the original (i.e. "source") image at time t . Each scale $S = \{s^j\}_{j=1..n}$ is defined by a set of values: a) the value of the current pixel at scale i p_{s^i} which is the sum of all the previous scales values; b) one barycenter $b_{(x,y)}$ by scale-pixel defined by b_{s^i} .

The barycenters and the sums of the pixels are propagated from one scale to the next
 20 so that computing the final vector from c_a to a block of points is the same as computing it directly from the biggest quadtree as soon as it includes all the underneath pixels. As the barycenter and the mean-value of the coefficients are propagated, each scale contains the same information as all the sub-scales within a different quad-size as shown in FIG. 16.

25 The first scale S^1 is the copy of the original image, where each barycenter is the center of each pixel. Then, from one scale S^i to the next one S^{i+1} the computation of the underneath value is:

25

5

$$P(x,y)_{Si} = \sum_{u=\frac{x}{2}}^{\frac{x}{2}+1} \sum_{v=\frac{y}{2}}^{\frac{y}{2}+1} P(u,v)_{Si-1}$$

The barycenter $b_{(x,y)}$ is the barycenter of the 4 barycenters of the previous scale, and are defined for scale i by:

$$b_{(x,y)_{Si}} = \frac{\sum_{u=\frac{x}{2}}^{\frac{x}{2}+1} \sum_{v=\frac{y}{2}}^{\frac{y}{2}+1} P(u,v)_{Si-1} \times b_{(u,v)_{Si-1}}}{\sum_{u=\frac{x}{2}}^{\frac{x}{2}+1} \sum_{v=\frac{y}{2}}^{\frac{y}{2}+1} P(u,v)_{Si-1}}$$

10

Since all quads of each scale contain the mean barycenter and the mean pixel of all subsequent scales, one can compute the final vector from a point (x, y) directly from the set of the biggest quads the mask contains.

Fast computation of the force vector for each anchor

A map for conflicting area

15

For a given attractive map, a new mask-map is computed, which integrates all the current masks of all the anchors using the same kind of attractive map. The final result of this map is the number of concurrent anchors per pixel (FIG. 13). This information is used to weigh the attractive force computed in those areas. This cumulative map CM^m where m is the nature of the attractive map, is computed with equation:

$$P_{(x,y)_{CM}} = \sum_{q \in Q_a^m} P_{(x,y)_a}$$

20

To compute the final resulting force of an anchor, one resulting vector per weight is

- 5 calculated. Each vector computed for each weight-area is then divided by the area value in the cumulative map.

Computation of the final resulting force of an anchor

- Next, the points concerning the anchor a are extracted from the cumulative map CM^m of type m . This extraction provides the map a_{CM} obtained using $a_{CM} = CM^m \cap \omega a$, where
 10 ωa is the set of points belonging to a .

- The final vector can now be computed by launching a quadtree for each possible value of a_{CM} . The bounding box B of the mask a_{CM} is delineated by $B_{(x_{min}, y_{min}, x_{max}, y_{max})}$. Each scale is scanned starting from the biggest to the smallest in order to fill the mask with the minimum number of quads necessary. For each value of $p_{(X,Y)_{si}}$ of each scale, a
 15 corresponding quad is computed in the original image which is $Q_{(x^*, y^*, x^*+l, y^*+l)_{si}}$. The inclusion test is $I(p_{(X,Y)_{si}}, a_{CM})$. If this test is true, the concerned quad's barycenter is added to the vector computation and is removed from the mask. If the quad was in a_{CM} , the return value of I is 1, otherwise it is 0. Then, building the final vector of the anchor a , centered on c_a , is given by:

$$\vec{f}(a) = \sum_{i=i_{min_S}}^{i_{max_S}} \sum_{XY} I(p_{(X,Y)_{si}}, a_{CM}) \times \frac{c_a \times b_{(X+i, Y+i)_{si}}}{p_{(X+i, Y+i)_{si}}}$$

- 20 where $X = \{\frac{B_{x_{min}}}{i}, \dots, \frac{B_{x_{max}}}{i}\}$ and $Y = \{\frac{B_{y_{min}}}{i}, \dots, \frac{B_{y_{max}}}{i}\}$

Experiments and results

The method according to the above-described embodiment offers two interesting considerations about its computation: i) the computations in the above-equation are highly independent in data, which means highly parallelizable, in opposition to the

5 anchor computation of the previous embodiment wherein there is no parallelizing since the algorithm uses the same dataset (i.e. causing concurrency when one attempts to apply multi-threading), and preferably, a multi-threading implementation is applied, using one thread per quad generated; and ii) conflicting areas may be computed only once and as long as they remain conflicting, without requiring recalculation since they
10 represent the same barycenter and coefficient, thereby taking advantage of such conflicting areas to speed up the processing, whereas in the method according to the previous embodiment, the conflicting anchors slow down the processing.

The method is also able to provide good results using small datasets, but is even more efficient while using very large datasets, since representing the problem through
15 multiscales and quadtree brings a logarithmic complexity to the problem, whereas in the naive way the complexity is quadratic, as seen in FIG.13. Thus, this computation is substantially fast, parallelizable and does not require the creation of groups of pixels of same weight.

In the present application, the inventors present a novel approach to track multiple
20 articulated objects in 2D video sequence by way of geometrically constrained models and physics engines using an a priori physical knowledge of the tracked objects. The method describes the tracking problem a set of multiple local processes applied to individual bodies, which are then linked together into a physics model. The physics engine solver is used to solve all these local processes while handling user-defined
25 constraints such as collision between different models in a fast and fully automated fashion.

The above-described embodiments are considered in all respect only as illustrative and not restrictive, and the present application is intended to cover any adaptations or variations thereof, as apparent to a person skilled in the art. Of course, numerous
30 modifications could be made to the above-described embodiments without departing

5 from the scope of the invention, as apparent to a person skilled in the art.

5 **WHAT IS CLAIMED IS:**

1. A computer-implemented method for tracking a physical object composed of one or more physical primitive, the method comprising the steps of:
 - 10 (a) receiving source image data being representative of the physical object at a time t_n ;
 - (b) identifying, in the source image data, the object and said one or more physical primitive, and associating a source geometrical primitive to each of said one or more physical primitive, thereby defining a model of the object according to a source configuration;
 - 15 (c) receiving target image data being representative of the physical object at another time t_{n+1} ;
 - (d) identifying, in the target image data, the object and said one or more physical primitive, and associating a target geometrical primitive to each of said one or more physical primitive, thereby defining the model of the object according to a target configuration;
 - 20 (e) associating to each of said source geometrical primitive a corresponding target geometrical primitive; and
 - (g) transforming said model, according to predetermined rules being representative of physical forces, from the source configuration to the target configuration, to simulate the real-life movement of the object between the times t_n and t_{n+1} .
2. The method according to claim 1, wherein at least one of said source geometrical primitive has a constraint being representative of an articulation and an associated degree of movability between physical primitives of the object, and
25 wherein the predetermined rules of the transforming in step (g) take into account
30

5 said constraint.

3. The method according to claim 2, wherein at least one of said one or more source geometrical primitive has an anchor being defined by a position p_a and an attraction map M_a delineating an area about the position p_a in which said corresponding target-geometrical-primitive is expected, thereby defining attraction map data, and wherein the associating of step (e) is based at least on matching
10 the source-geometrical-primitive with one of said target-geometrical-primitive located in the attraction map M_a of the associated anchor.

4. The method according to claim 3, wherein the associating step (e) comprises performing a mean-shift process.

- 15 5. The method according to claim 3, wherein the associating step (e) comprises performing a multiscale decomposition of the attraction map data of the source image in quadtrees.

6. The method according to claim 4 or 5, wherein further comprising:
(f) computing, for each of said anchor, an attraction force according to:

$$\vec{f}(a) = \sum_{q \in Q_a} q \overrightarrow{p_a q},$$

20

where q describes pixels within the attraction map, and wherein the predetermined rules of the transforming of step (g) includes combining said attraction force with said constraint.

7. The method according to claim 5, comprising computing each of said map using
25 multi-scale areas and querying said map through a quadtree algorithm for computing a final attraction force vector of said anchor from a largest scale to a smallest scale thereby minimizing vector computations in the associating step (e).

- 5 8. The method according to claim 7, wherein the method is applied for handling proximity detection.
9. The method according to claim 7 or 8, wherein the computer-implemented method is performed by a multi-threaded implementation.
10. The method according to any one of claims 7 to 9, wherein the computing step
10 and the transforming step (g) are repeated iteratively until a stop condition is reached.
11. The method according to claim 10, further comprising, after said stop condition is reached:
- 1.5 (h) predicting a new target configuration of the model at least based on a computation of the source configuration and the target configuration, and the method looping-back to repeat steps (c)-(h), wherein the target configuration becomes the source configuration, and wherein the new target configuration is taken into account at the associating step (e).
- 20 12. The method according to claim 11, wherein the predicting of step (h) is at least based on an extrapolation of a speed vector calculated for at least one of said anchor, from the source configuration at the time t_n to the target configuration at the time t_{n+1} .
- 25 13. The method according to any one of claims 9 to 12, wherein the transforming step (g) takes into account proximity handling when two or more physical objects are in proximity based on the source image data.
14. The method according to claim 7, wherein the computing is provided according to:

$$\vec{f}(a) = \frac{\sum_{i=i_{min_S}}^{i_{max_S}} \sum_{XY} I(p_{(X,Y)_{Si}, a_{CM}})}{C_a \times b_{(X*i, Y*i)_{Si}} * p_{(X*i, Y*i)_{Si}}}$$

5

where $X = \{\frac{B_{xmin}}{i}, \dots, \frac{B_{xmax}}{i}\}$ and $Y = \{\frac{B_{ymin}}{i}, \dots, \frac{B_{ymax}}{i}\}$

15. The method according to any one of claims 1 to 14, wherein the transforming of step (g) is provided by a physics engine solver.
16. The method according to any one of claims 1 to 15, wherein the method is used for tracking a plurality of moving objects and records events based on relative positions of said plurality of moving objects.
17. The method according to any one of claims 1 to 16, wherein the method is applied for tracking said physical object in a substantially real-time mode.

10

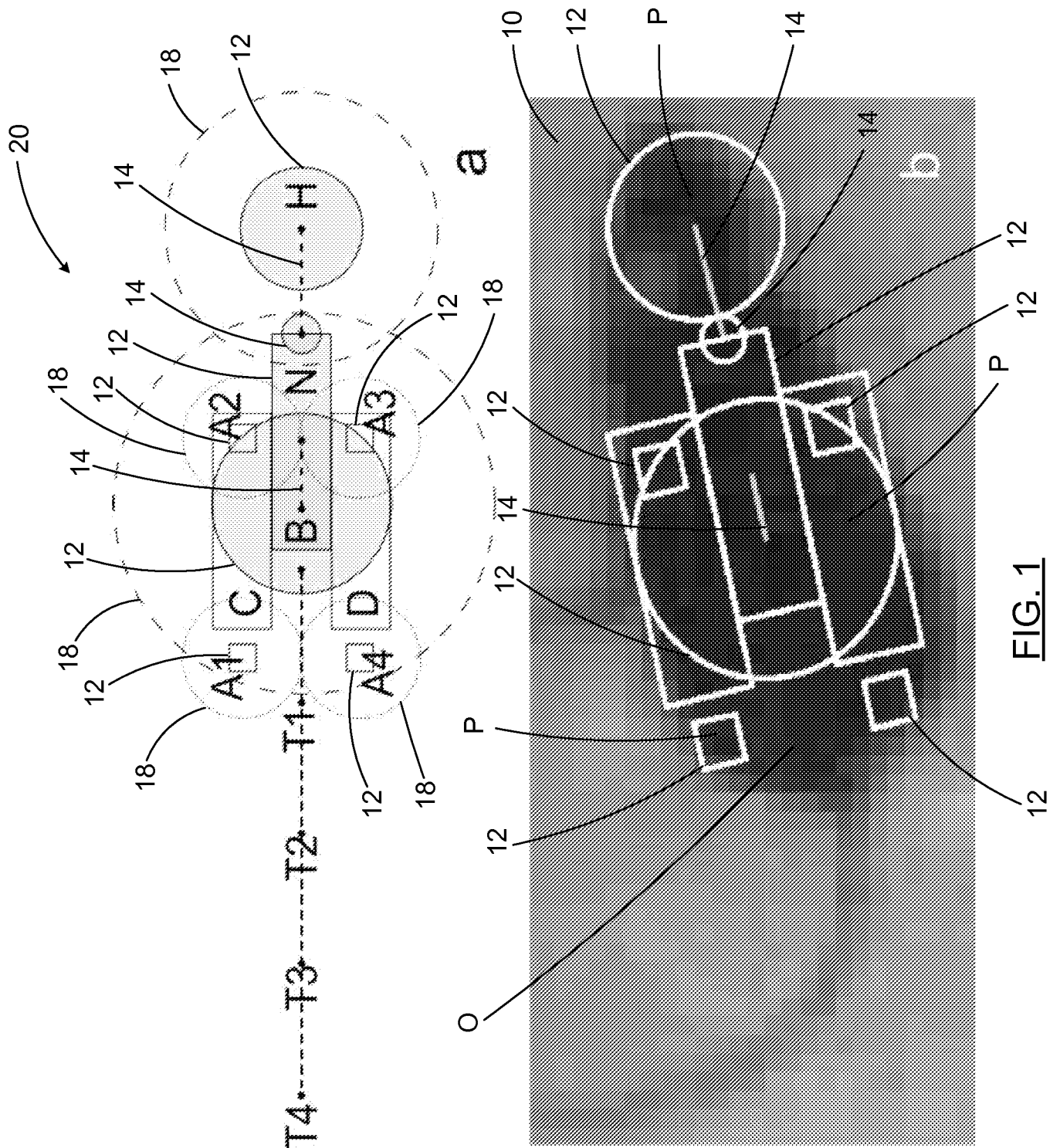


FIG. 1

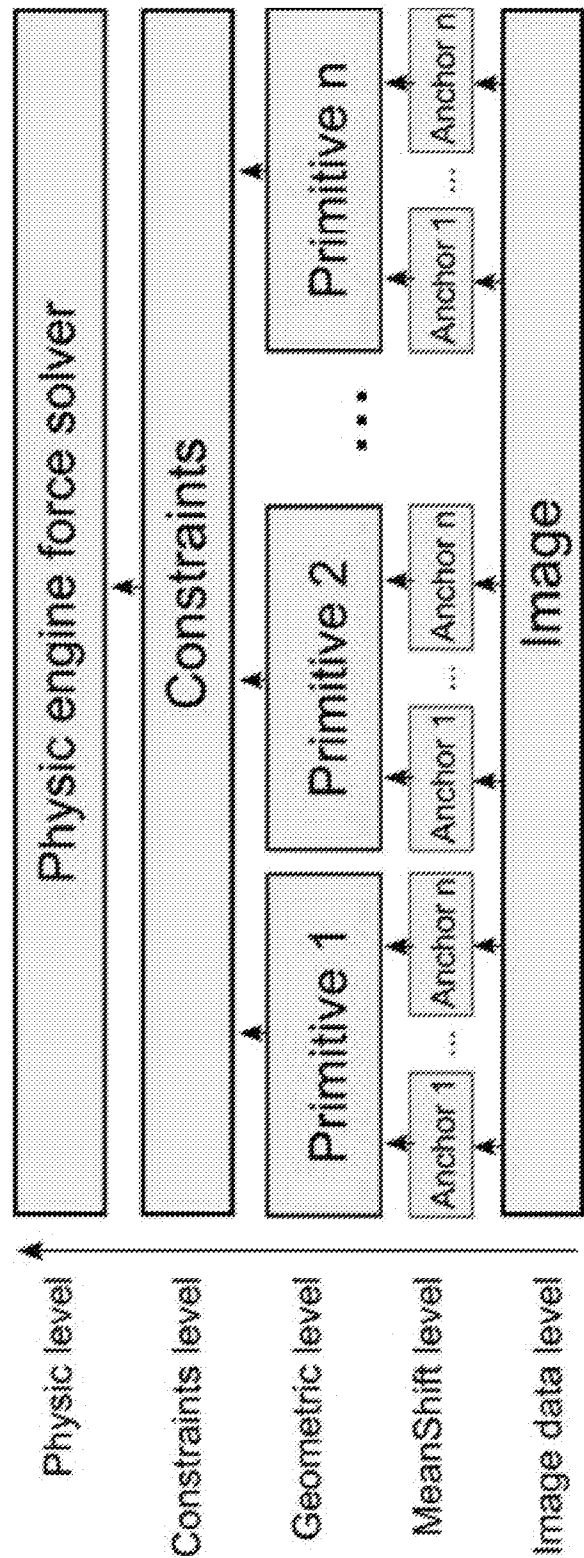


FIG. 2

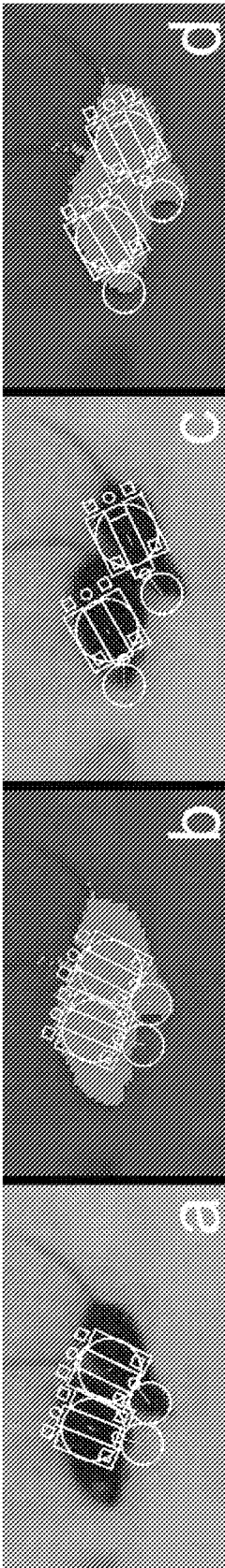


FIG. 3

4/18

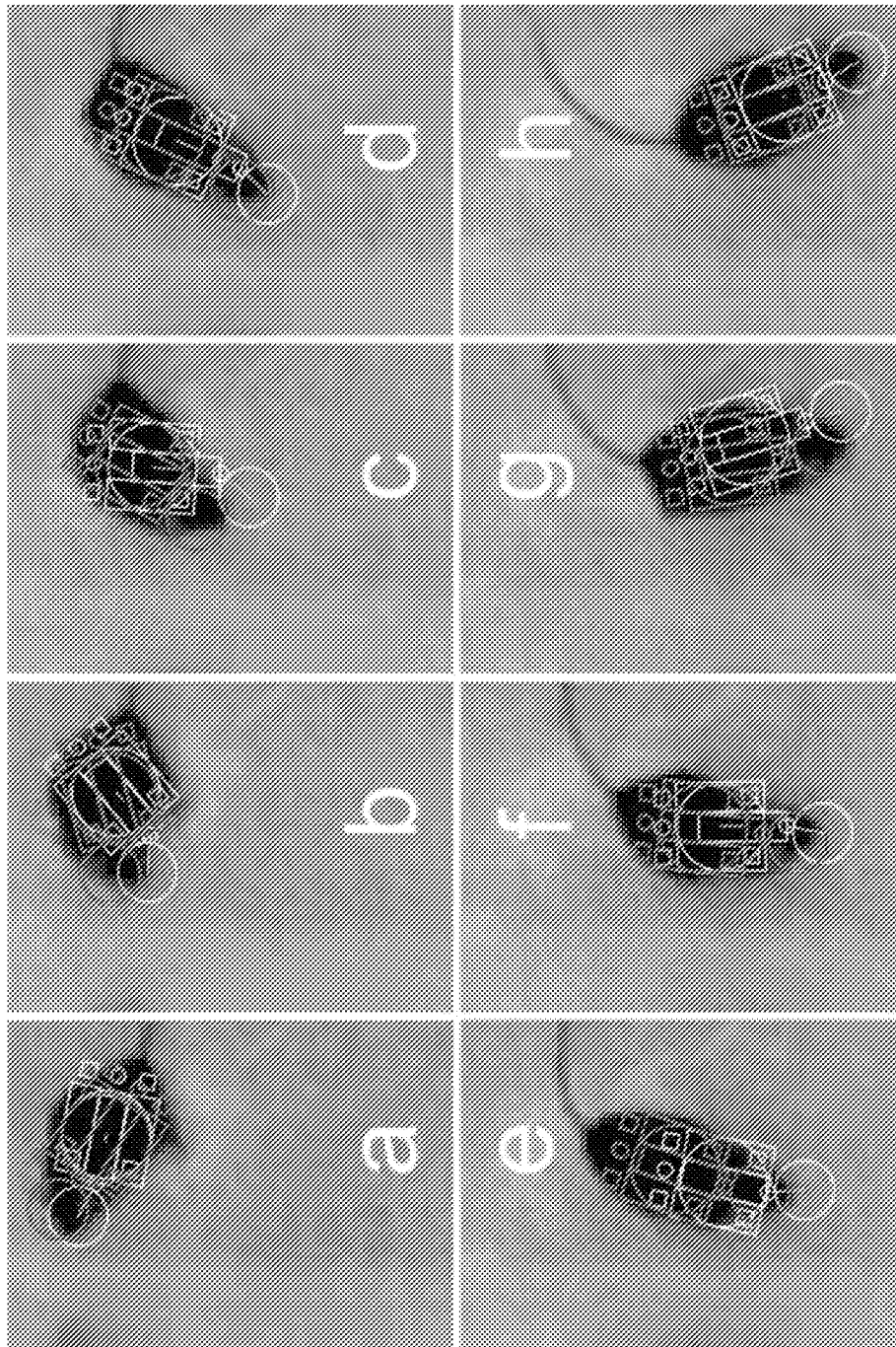


FIG. 4

5/18

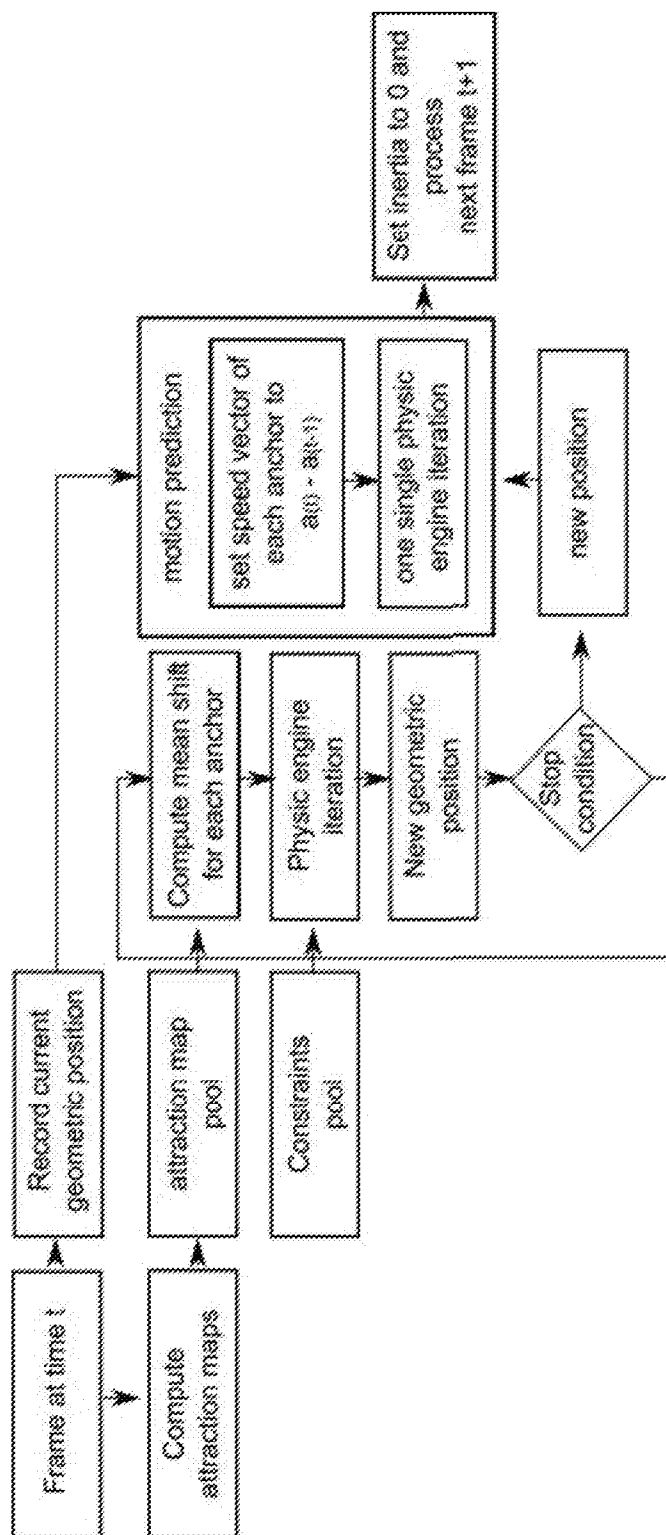
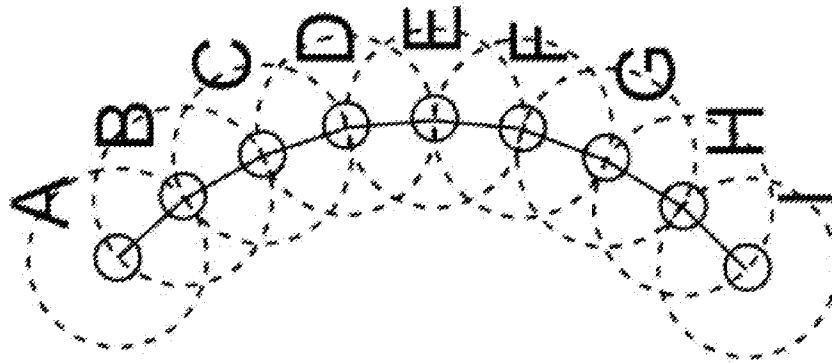


FIG. 5

6/18



b

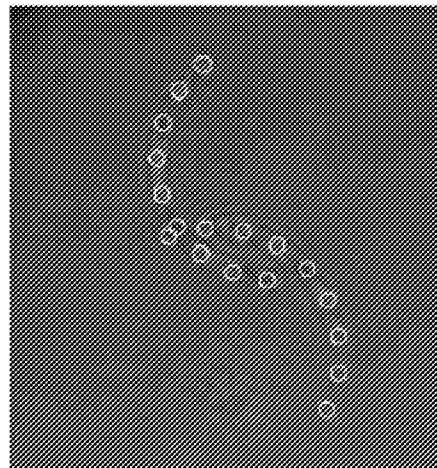
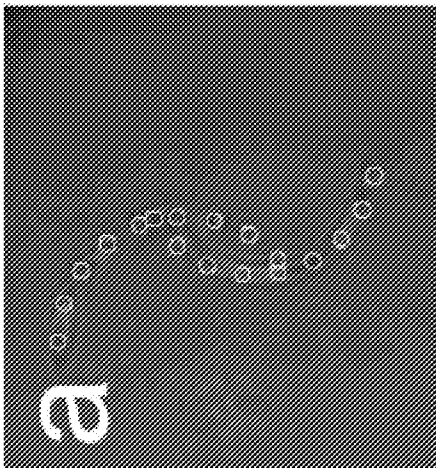
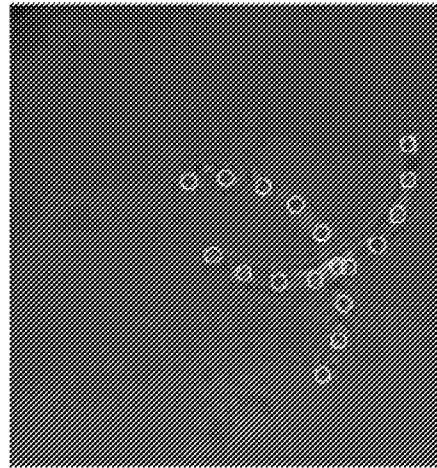
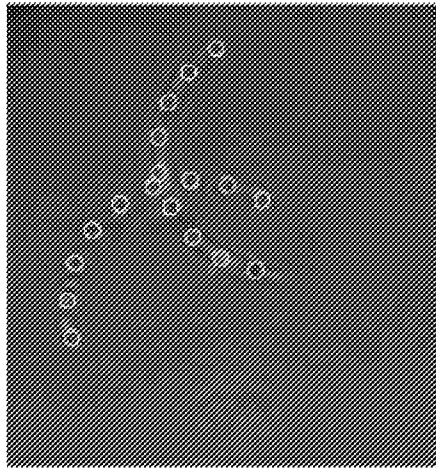


FIG. 6

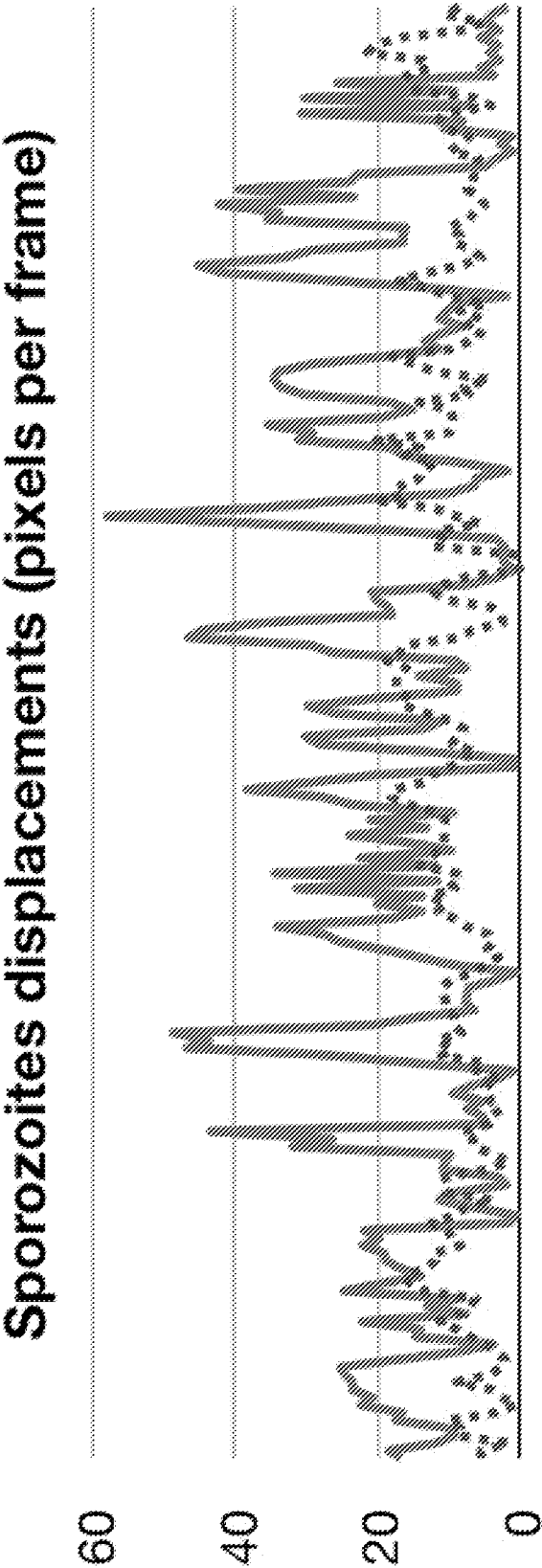


FIG. 7

8/18

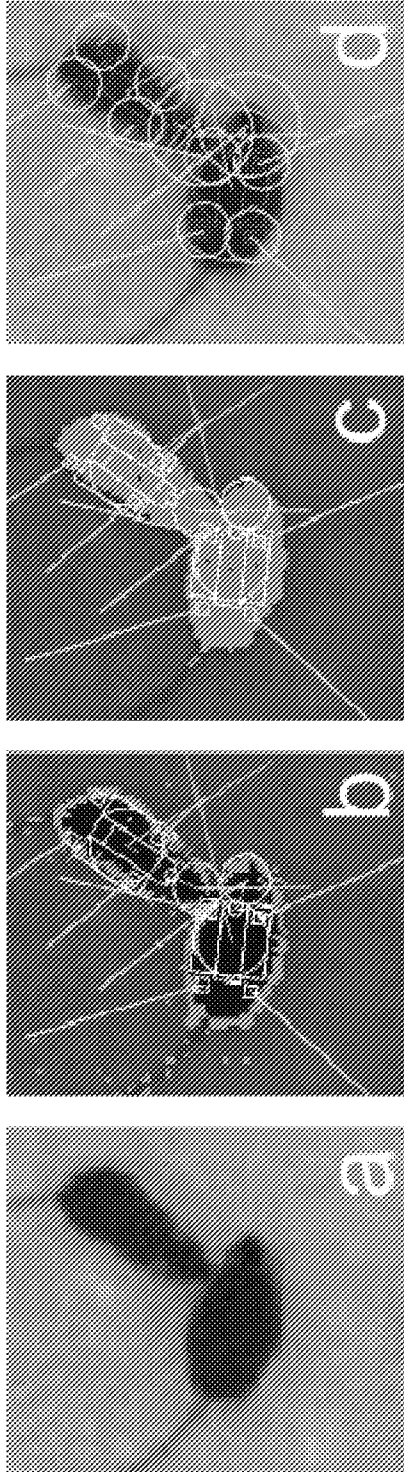


FIG. 8

9/18

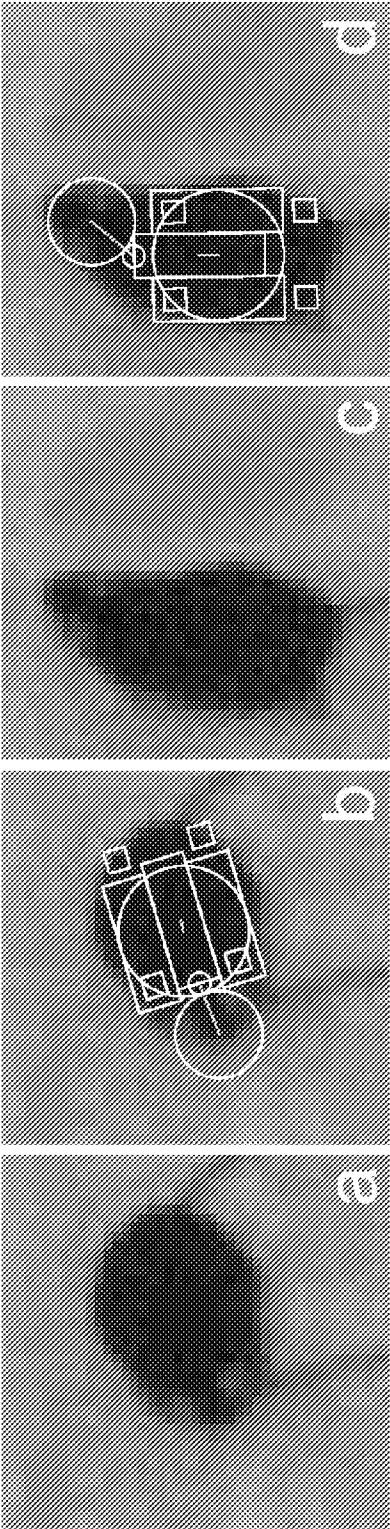


FIG. 9

10/18

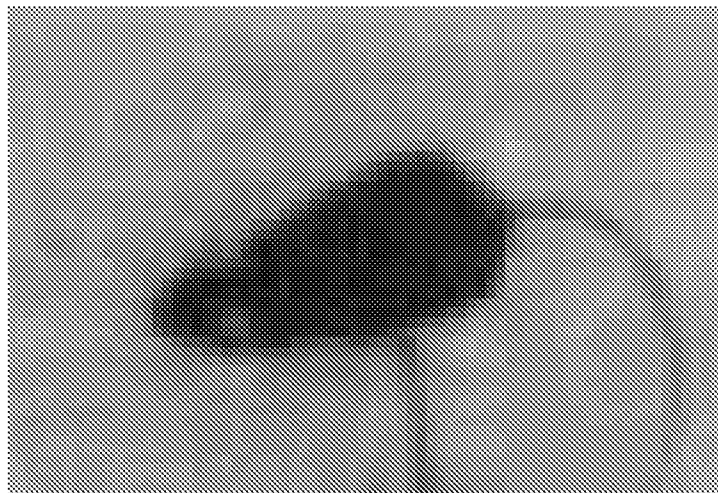


FIG. 10A

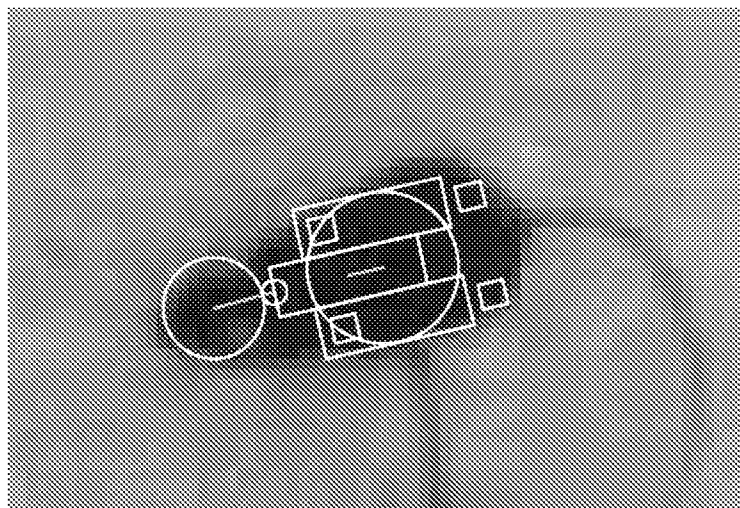


FIG. 10B

11/18

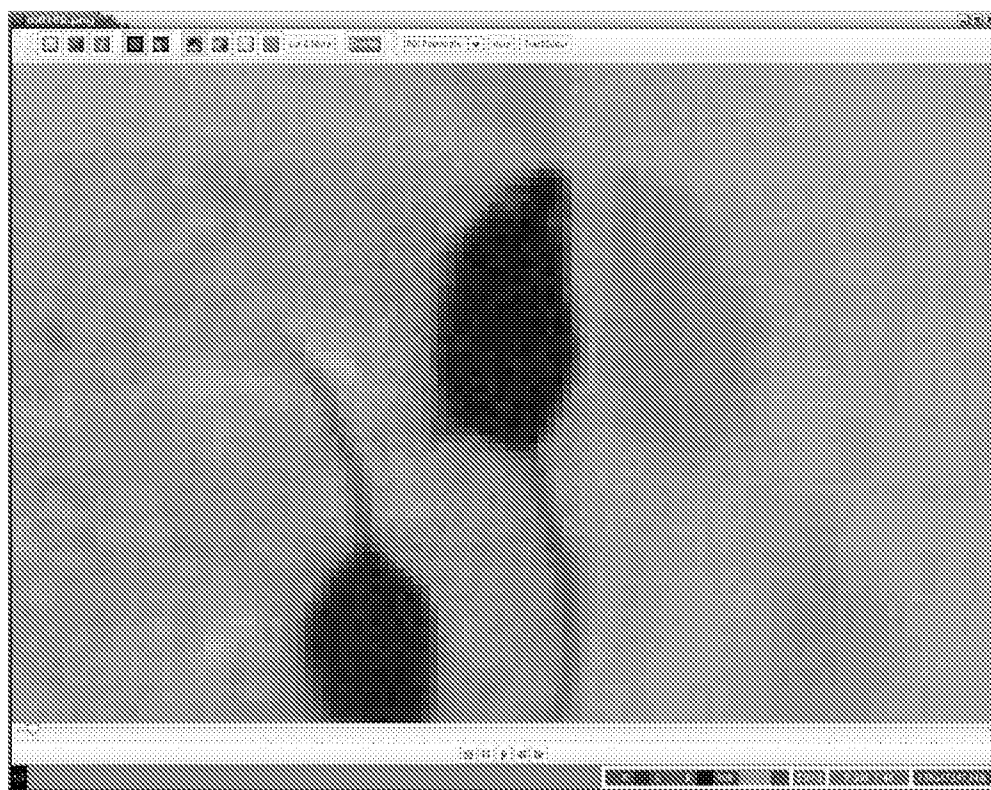


FIG. 11A



FIG. 11B

12/18

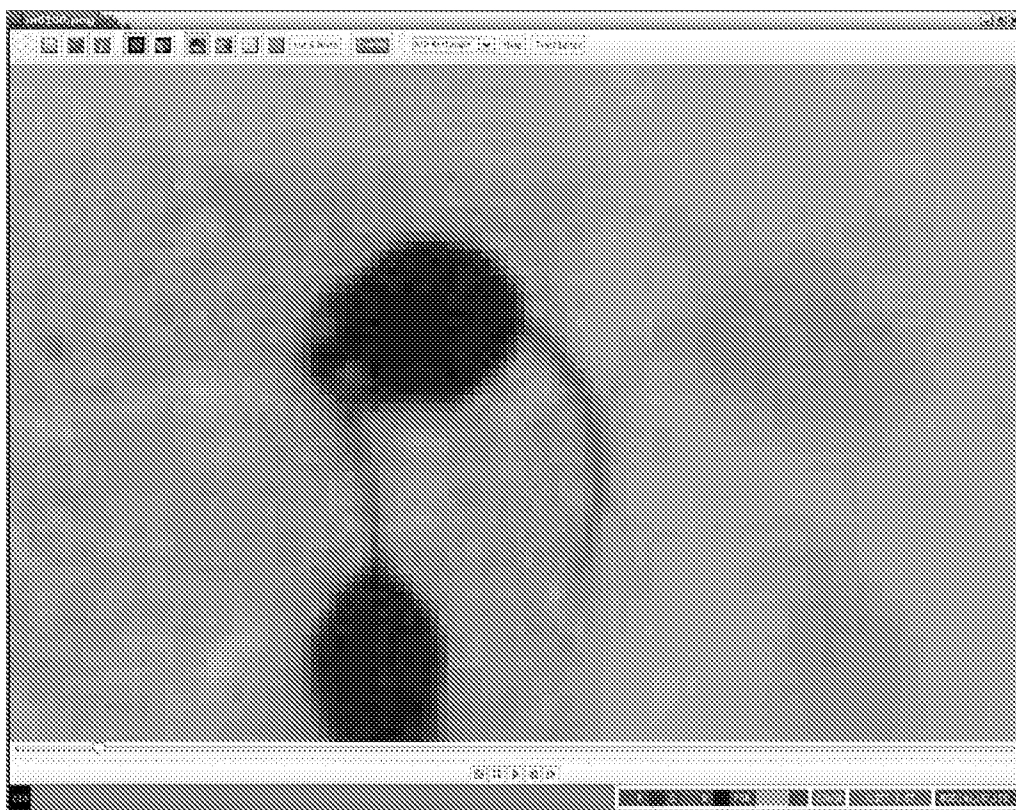


FIG. 12A



FIG. 12B

13/18

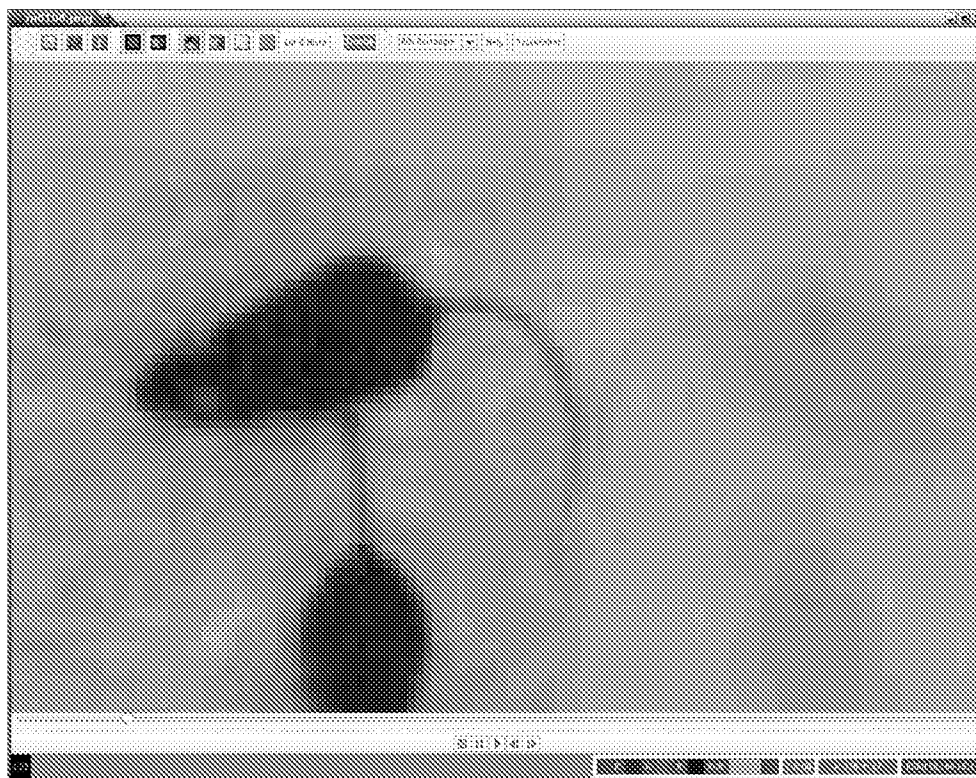


FIG. 13A

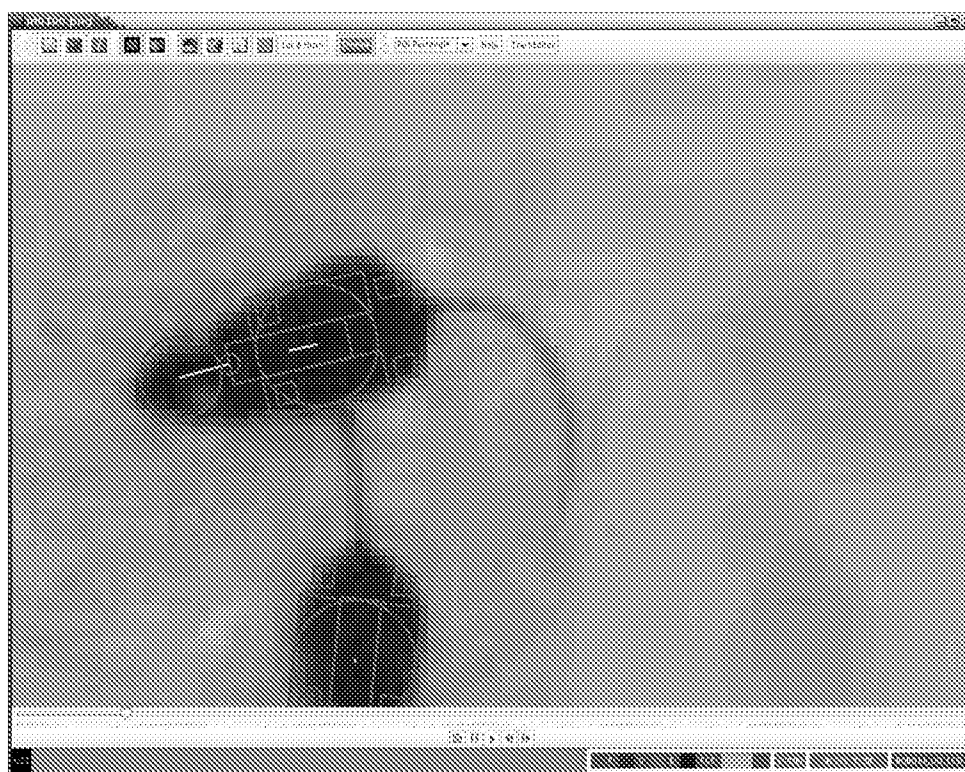


FIG. 13B

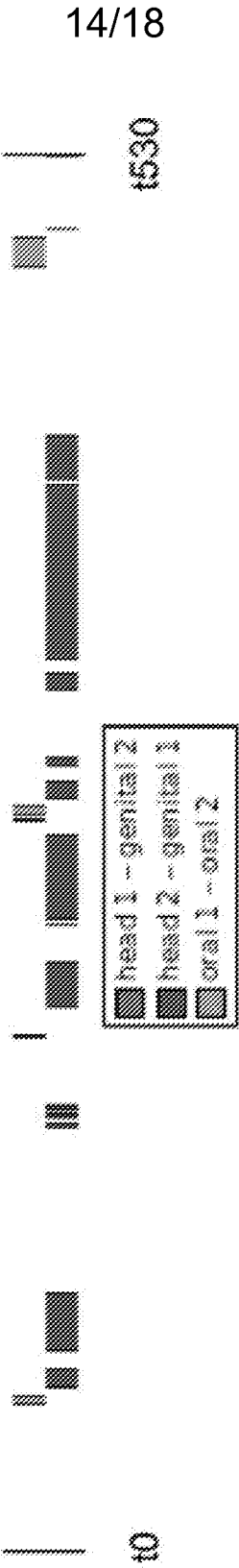


FIG. 14

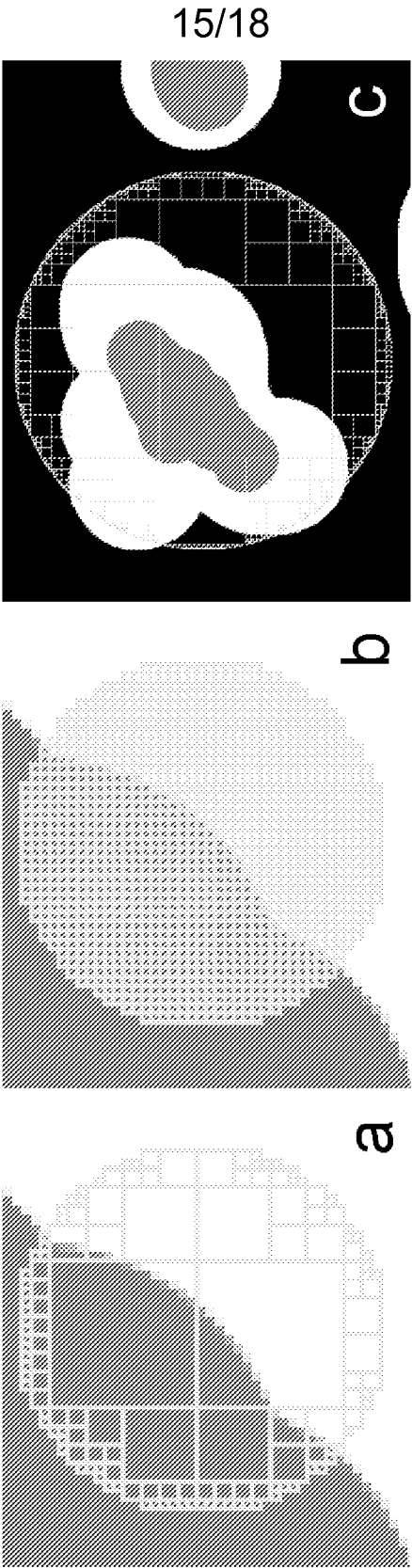


FIG. 15

16/18

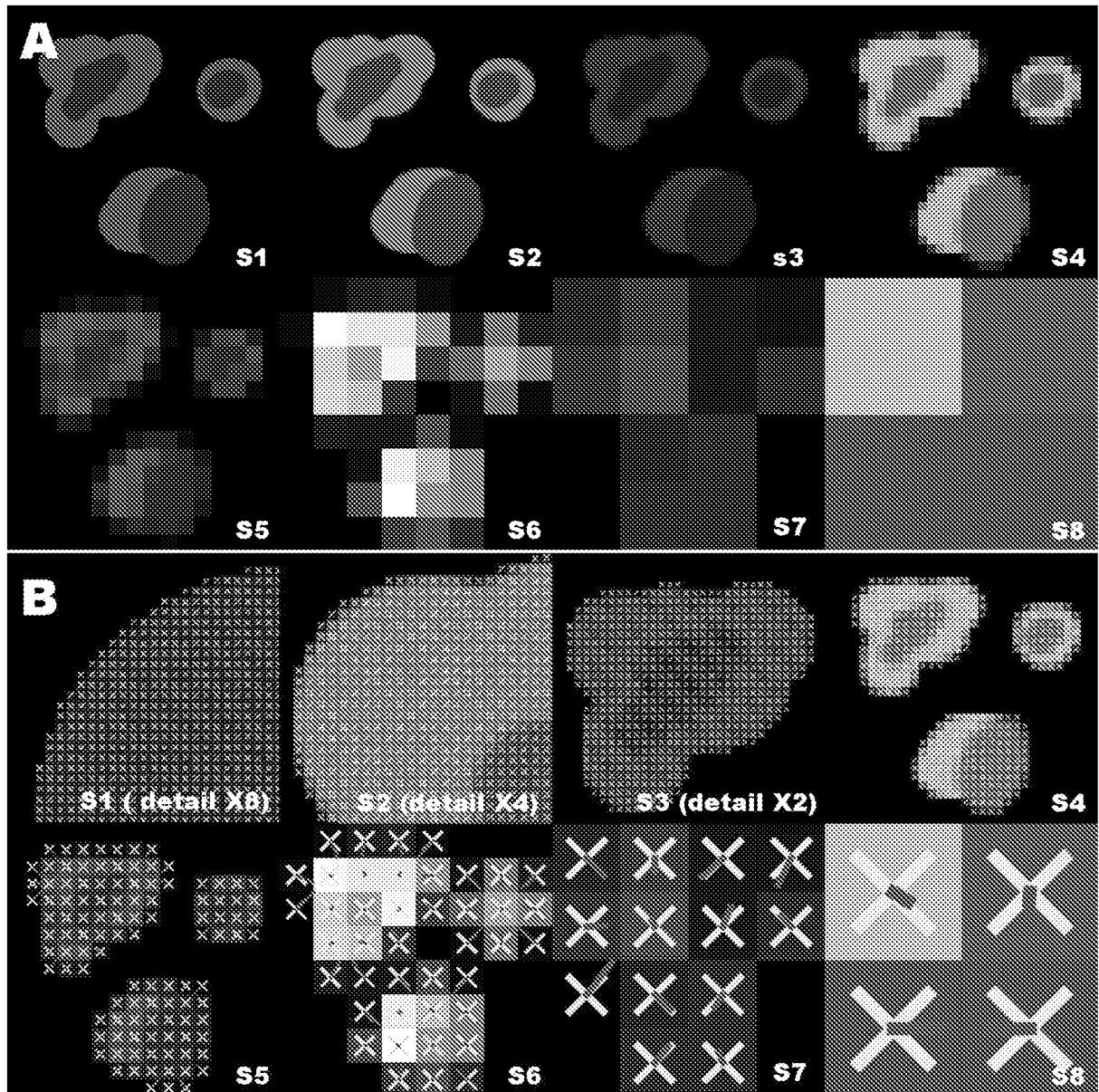


FIG. 16

17/18

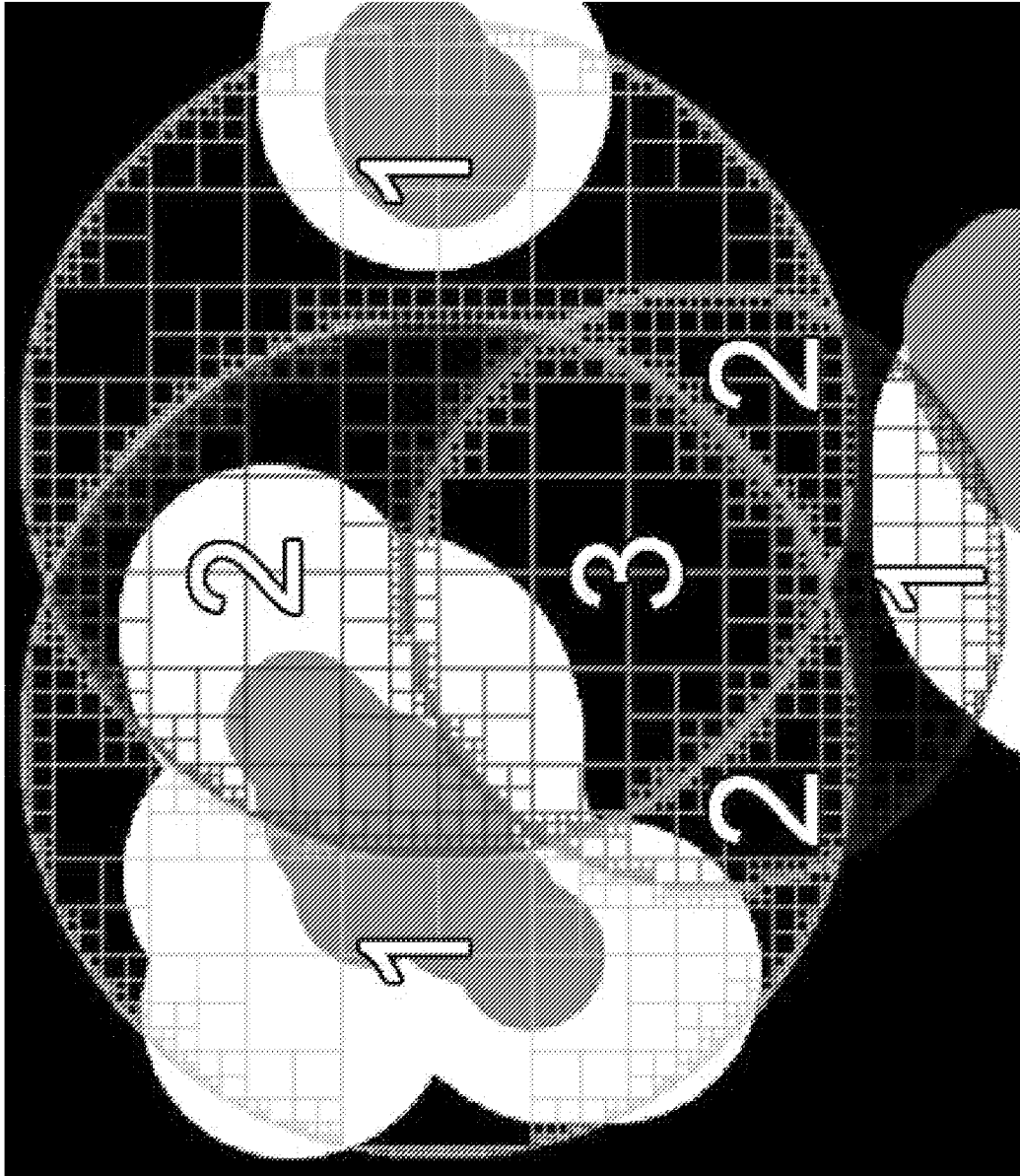


FIG. 17

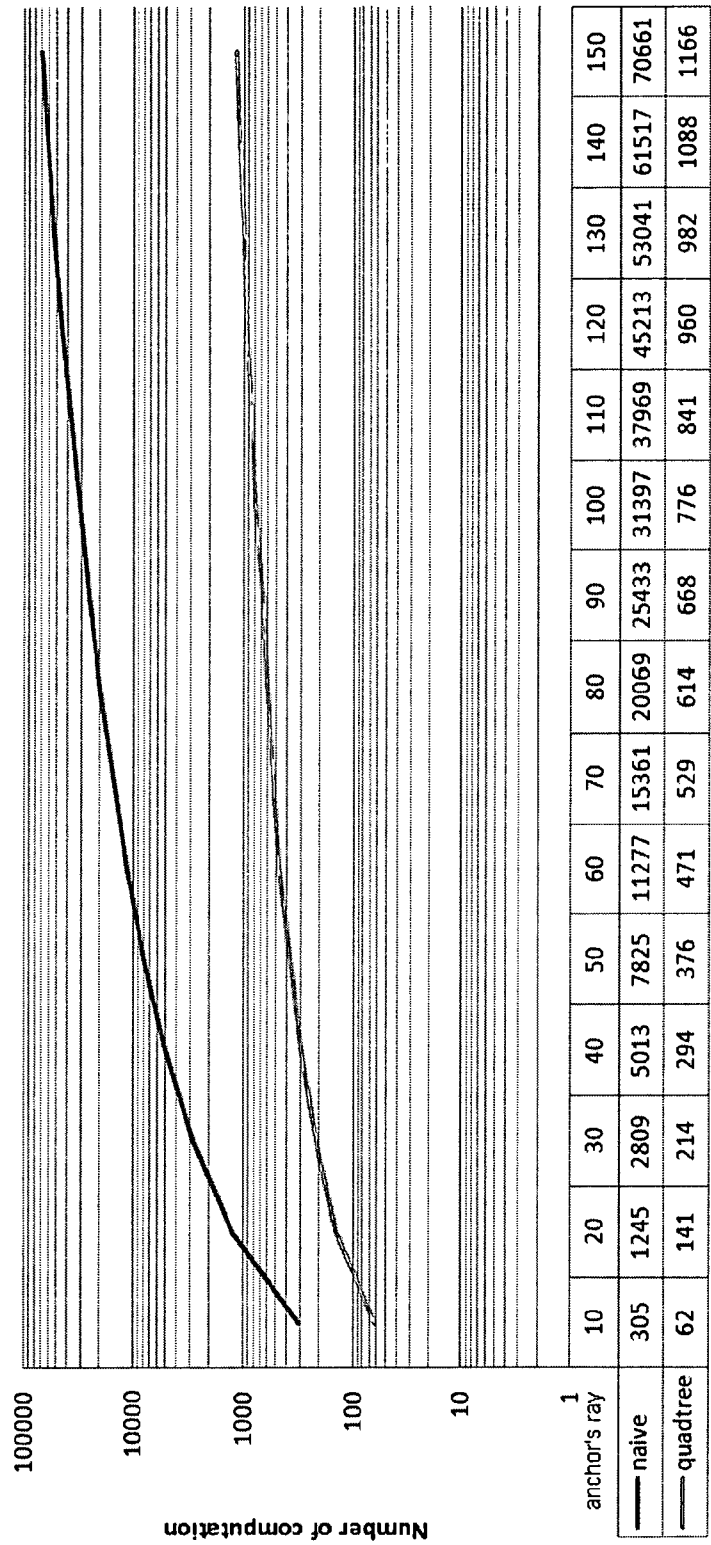


FIG. 18

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/059222

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06T7/20
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06T G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DE CHAUMONT F ET AL: "Using physics engines to track objects in images" BIOMEDICAL IMAGING: FROM NANO TO MACRO, 2009. ISBI '09. IEEE INTERNATIONAL SYMPOSIUM ON, IEEE, PISCATAWAY, NJ, USA, 28 June 2009 (2009-06-28), pages 1342-1345, XP031502305 ISBN: 978-1-4244-3931-7 the whole document ----- -/--	1-17

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

4 October 2010

Date of mailing of the international search report

13/10/2010

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Borotschnig, Hermann

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/059222

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	DE CHAUMONT F ET AL: "Tracking articulated objects with physics engines" IMAGE PROCESSING (ICIP), 2009 16TH IEEE INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 7 November 2009 (2009-11-07), pages 885-888, XP031628922 ISBN: 978-1-4244-5653-6 the whole document	1-17
X	----- AGGARWAL J K ET AL: "Articulated and elastic non-rigid motion: a review" MOTION OF NON-RIGID AND ARTICULATED OBJECTS, 1994., PROCEEDINGS OF THE 1994 IEEE WORKSHOP ON AUSTIN, TX, USA 11-12 NOV. 1994, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC LNKD- DOI:10.1109/MNRAO.1994.346261, 11 November 1994 (1994-11-11), pages 2-14, XP010100232 ISBN: 978-0-8186-6435-9 the whole document	1-3, 15-17
A	-----	4-14