PREVISUALIZATION IN ROBOTICS: AN ATOMIC APPROACH

Mauno Rönkkö
University of Kuopio, Department of Computer Science
P.O.Box 1627, 70211 Kuopio, Finland

Keywords: Robotics, previsualization, modelling, emergent dynamics, simulation, formal methods.

Abstract: In this paper, we discuss use of previsualization in robotics. In previsualization, a computer animates a highly underspecified model; thus, helping the designers to detect missing components. When applied to robotics, previsualization should reveal detailed dynamics that is not explicitly specified in the model. In particular, it should reveal interaction dynamics of the physical components. Such a requirement is non-trivial. In this paper, we investigate an atomic approach that uses emergent dynamics to reveal the interaction dynamics. The approach is based on use of atoms that obey simple, compositional interaction laws. The laws produce emergent interaction dynamics for physical components composed of atoms. As the main contribution, we shall illustrate how the atomic approach, despite its simplicity, captures compactly non-trivial settings, and reveals effortlessly intricate dynamics that are otherwise considered too laborious for modelling.

1 INTRODUCTION

Previsualization is a design technique used originally in film industry (Ferster, 1998). In previsualization, a computer animates an underspecified scene. The animation reveals quickly, if the scene is missing some key components, or if the dynamics is not as desired.

In this paper, we generalize the idea of previsualization to *robotics* (Arkin, 1998). In robotics, it is used as a design tool for improving a model until it is detailed enough for formal analysis and prototyping.

Unlike *simulation* (Mosterman, 1999), previsualization does not deal with a detailed model; rather, the model is highly underspecified and details appear as *emergent dynamics* (Odell, 1998; Arkin, 1998). Consequently, previsualization does not reveal "exact" results, although the results are rich in realistic details.

When using previsualization in robotics, we wish to avoid specifying explicit interactions between the physical components in the model. One reason is that, during previsualization, the specifics of the components are still unknown. Another reason is that the number of required interaction equations is quadratic to the number of components in the system, for instance, 100 components require 10 000 interaction equations.

One way of avoiding specifying the interaction equations is to use emergent dynamics to reveal it. However, this is not trivial in robotics due to inherent compositional diversity. In robotics, there are many different kinds and shapes of components interacting with each other (Arkin, 1998). In addition, the interaction dynamics itself shows various kinds of motions, rotations, collisions, and other kinematics.

Mere encoding of the text-book equations on dynamics, like those found in (Young and Freedman, 2000) and in (Thornton and Marion, 2004), is not enough, as they describe the dynamics in isolation. Figuring out the precise interaction equations, on the other hand, is not reasonable, since the specifics of the components are not known ahead of the previsualization. Nevertheless, previsualization must somehow capture detailed interaction dynamics.

As a solution, we propose in this paper an *atomic approach*, where interaction dynamics of the physical components appear as emergent dynamics. The approach is based on use of *atoms* [1] that are indivisible particles of constant shape and mass. The atomic approach is, thus, a specific variant of a *particle system* (Reeves, 1983; Witkin, 1997).

[1] The notion of atoms, here, differs from the notion of atoms in modern physics (Young and Freedman, 2000), as well as from the notion of atoms put forth by Greek philosophers Democritus and Leucippus around 400 BC.

Rönkkö M. (2004).
PREVISUALIZATION IN ROBOTICS: AN ATOMIC APPROACH.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 110-117
DOI: 10.5220/0001127301100117

Atoms may move, resist movement, be bound to each other, and collide with each other. For these dynamics, there are simple equations that are also compositional. *Moreover, these equations are enough to produce emergent dynamics capturing interactions between physical components composed of atoms, regardless of the shape of the components* (Rönkkö, 2003b).

An advantage of the atomic approach is that it is computationally light. A three dimensional model can be computed and rendered on screen in real-time.

An additional advantage of the atomic approach is that it supports inclusion of software components to the model. For instance, we could add a neural net to the model for controlling some physical components. Then, the neural net interacts with the physical components by manipulating the forces on the corresponding atoms. In this way, we do not need to state explicit interaction dynamics in the model, and we can experiment with different software components to see their effect on the global, *behavioral dynamics* (Arkin, 1998; Reynolds, 1987).

As the main contribution, we shall illustrate in this paper how the atomic approach, despite its simplicity, captures compactly non-trivial settings, and reveals effortlessly intricate dynamics that are otherwise considered too laborious for modelling.

Overview. We start by formalizing the atomic approach in Section 2. In Section 3, we discuss and illustrate modelling with strings of atoms. In Section 4, we discuss and illustrate modelling with planes of atoms. In Section 5, we discuss and illustrate some advanced emergent properties, such as friction and rotation. Finally, in Section 6, follows the conclusion.

2 THE ATOMIC APPROACH

In the atomic approach, a model consists of a fixed number of atoms and forces. An atom is an independent and indivisible particle. Based on the defined forces, the atoms in the model may move, resist movement, be bound to each other, and collide with each other.

2.1 An Atom

An atom has only four properties: *shape*, *mass*, *position*, and *velocity*. Each atom has the same constant shape of a *unit ball*, and the same *unit mass*. Only the position and the velocity may vary from atom to atom; however, even these values are bound by the model of motion.

As an atom exists in $\mathbb{R}^3$-space, we use three dimensional *vectors* (Adams, 1999) to capture the properties of an atom. Let $\vec{c}$ denote the position of an atom, i.e., the displacement of the center of the atom from the origin. Then, the *magnitude* of $\vec{c}$, i.e., $|\vec{c}|$, gives the distance of the center of the atom to the origin.

An atom occupies the set P of points in $\mathbb{R}^3$, where $|\vec{p} - \vec{c}| \leq 1$ holds for any $\vec{p}$ in P. Within this space, the unit mass of an atom is evenly distributed. Consequently, according to (Young and Freedman, 2000), the center of an atom is also its *center of mass*.

We do not, however, consider atoms as objects of dense matter; on the contrary, atoms are objects of *sparse matter*. Therefore, several atoms may overlap in space without any interference, if so desired.

2.2 Model of Motion

Atoms follow a trivial model of motion. A change in the position of an atom is determined by its velocity. A change in the velocity of an atom, in turn, is determined by an impulse on it. The impulse is the sum of all forces on the atom.

As shown in (Rönkkö, 2003b), the model of motion is obtained from the trivial model of motion with acceleration. First, Euler's method (Zill and Cullen, 1997) is applied to obtain an approximation equation. Then, as the approximation equation operates in discrete time on atoms with a unit mass, we may substitute the acceleration term in the equation with an *impulse* term (Young and Freedman, 2000).

Formally, the obtained model of motion is a *difference equation* (Zill and Cullen, 1997). Let n and $n+1$ denote two successive iteration rounds, and τ denote a reasonably small, fixed, time step. Also, let $\vec{c}(n)$, $\vec{v}(n)$, and $\vec{j}(n)$ denote the center of an atom, its velocity, and the impulse on the atom in the iteration round n. Then, the model of motion is:

$$\vec{v}(n+1) = \vec{v}(n) + \vec{j}(n)$$
$$\vec{c}(n+1) = \vec{c}(n) + \tau \cdot \vec{v}(n+1)$$

Here, the impulse $\vec{j}(n)$ is the sum of forces on the atom in the iteration round n. The forces are computed based on the position and velocity of atoms in the iteration round n.

The model of motion above is a refinement of that presented in (Rönkkö, 2003b). In particular, the model above provides a more stable motion.

Note that the model of motion for atoms is compositional; in particular, it allows us to consider the position, velocity, and the impulse of an atom independently of each other within each iteration round.

2.3 An Impulse

An impulse is, by definition (Young and Freedman, 2000), the sum of forces affecting the velocity. In our model, there are only four kind of forces: *damping*

forces, *bond forces*, *collision forces*, and other *case-specific external forces*.

For simplicity, all the impulses considered in this paper depend only on the properties of atoms; in particular, we do not consider here any impulses that depend on the number of the iteration round.

In the sequel, we shall refer to a specific atom in the model using subscripts. For instance, we shall denote the center of the i'th atom in the model by $\vec{c}_i$, and its velocity by $\vec{v}_i$.

A damping force. A damping force resists the motion of an atom. Its direction is opposite to the direction of the velocity, and the magnitude is always less or equal to the magnitude of the velocity.

Assuming a coefficient $m \in [0, 1]$, we define a damping force $\vec{fd}(i, m)$ with a magnitude of $m \cdot 100\%$ on the i'th atom in the model (Rönkkö, 2003b):

$$\vec{fd}(i, m) = -m \cdot \vec{v}_i$$

A bond force. A bond force maintains a predefined distance between two specific atoms. Generally speaking, a bond force is modelled as a function over the *desired distance* [2] and the current distance between the two atoms.

We model a bond force with a non-linear function, because it has better approximation properties than a linear function (Rönkkö, 2003b). Assuming a desired distance $r \in [0, \infty)$, we define a bond force $\vec{fb}(i, r, k)$ on the i'th atom in the model with respect to the k'th atom in the model (Rönkkö, 2003b):

$$\vec{fb}(i, r, k) = (\vec{c}_k - \vec{c}_i) \cdot (1 - r^2 |\vec{c}_k - \vec{c}_i|^{-2})$$

Note that a bond force is *symmetric*, i.e., $\vec{fb}(i, r, k) = -\vec{fb}(k, r, i)$. Thus, a pair of bond forces preserves *Newton's third law* [3] (Young and Freedman, 2000).

A bond force, as defined above, results in accelerating motion. Consequently, an additional force, such as a damping force, may be needed for the motion to *converge*.

Note that an atom may well be bound to many atoms at the same time. Then, we may consider each bond independently, and the sum of the bond forces is the total bond force on the atom. Due to the model of motion, the bond forces can be evaluated in any order without affecting the result.

[2] We consider only the distance between the centers of the atoms, as the center of an atom is also its center of mass.

[3] The significance of Newton's third law is that it preserves *symmetry* in the model. Then, from the model, we may freely choose one atom, and use it as the center of events. Due to symmetry, the result is independent of our choice.

A collision force A collision force affects only overlapping atoms by pushing them apart. Atoms are considered to be apart, when their distance exceeds the sum of their radii, 2.

We model a collision force as a *semi-bond force*. Thus, we define a collision force $\vec{fc}(i, k)$ on the i'th atom in the model with respect to the k'th atom in the model (Rönkkö, 2003b):

$$\vec{fc}(i, k) = (\vec{c}_k - \vec{c}_i) \cdot \min\{0, (1 - 4 \cdot |\vec{c}_k - \vec{c}_i|^{-2})\}$$

The collision force is also symmetric; hence, a pair of collision forces preserves Newton's third law.

Note that an atom has no materialistic properties, such as elasticity. Consequently, the collision force above defines exhaustively a collision between two atoms. More specifically, it captures an *elastic collision* (Rönkkö, 2003b). An inelastic collision can, thus, never occur between two atoms without any additional forces. This does not, however, preclude an inelastic collision between components composed of atoms, as we shall see later in the examples.

Similarly to the bond forces, there can be many collision forces on an atom at the same time. Then, we may consider each collision independently, and the sum of the collision forces is the total collision force on the atom. Due to the model of motion, the collision forces can be evaluated in any order without affecting the result.

Case-specific external forces. We do not attempt to formalize other case-specific external forces. The reason for this is simply the overwhelming variety of such forces. Examples of case-specific external forces are, among others, gravity, impulses originating from various power sources, and impulses used for visualization purposes, i.e., forcing the atoms to move into desired direction to reveal the emergent dynamics of that motion.

2.4 Modelling with Atoms

Modelling with atoms is simple. In a model, we define for each atom

1. its initial position and velocity
2. the forces on it

The model of motion is then used to compute the position and the velocity of the atom in the next iteration round.

In a model, bond forces are used to compose components from atoms. The number of bond forces per atom defines also the rigidness of the component (Rönkkö, 2003b). Collision forces, in turn, are used for defining which atoms (of the components) may collide with each other.

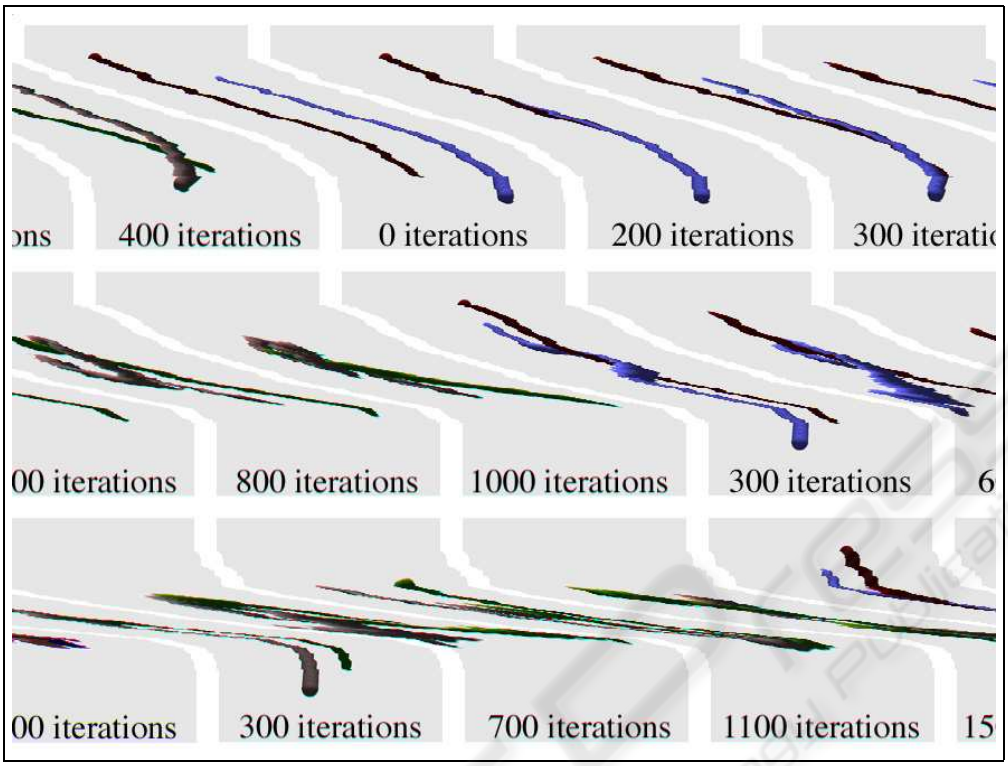

Figure 1: Visualization of colliding rods and threads. The top row shows a collision of two rods, the middle row shows a collision of a rod and a thread, and the bottom row shows a collision of two threads. The computed iteration rounds are indicated in each image. The initial configuration, shown in the leftmost image in the top row, is the same for each collision.

3 MODELLING WITH STRINGS OF ATOMS

For modelling components like rods, threads, triangles, and rings, we use strings of atoms. In a string, bond forces are used for defining the shape of the component and the rigidness of the component (Rönkkö, 2003b). For brevity, we shall consider only threads and rods in this section.

We exemplify use of strings of atoms by visualizing non-trivial collisions of rods and threads. In these examples, we do not consider any damping forces; thus, the rods and threads behave as if they were in space. For clarity, we limit the investigation to collisions of two components at a time.

Both a rod and a thread are modelled using a string of 70 atoms. The colliding strings are aligned diagonally with respect to the y and z-axes. Also, the alignment of the strings with respect to each other is diagonal and non-symmetric, as shown in the leftmost image in the top row in Figure 1.

The atoms in a string are initially positioned so that five neighboring atoms overlap, to give an impression of dense material (Rönkkö, 2003b). The two strings are then set to collide by giving them an opposite initial velocity along the x-axis.

The impulse on an atom in a string is the sum of bond forces and collision forces. In a rod, each atom is bound to every other atom, whereas, in a thread, an atom is bound only to its neighboring atoms. Formally, in a rod, the total bond force on the i'th atom is $\sum_{j=0}^{69} \vec{fb}(i, \sqrt{0.125}, j)$ for all $j \neq i$. Correspondingly, the total bond force on the i'th atom in a thread is $\sum_{j=0}^{69} \vec{fb}(i, \sqrt{0.125}, j)$ for all $|j - i| = 1$.

The total collision forces prevent the atoms in the colliding strings from slipping through each other. Thus, there is a collision force on the i'th atom in a string is $\sum_{j=0}^{69} \vec{fc}(i, j)$, where j refers to atoms in the other string. In addition, we also wish to prevent the non-overlapping atoms within a string from slipping through themselves. Therefore, there is an additional collision force on the i'th atom in a string: $\sum_{k=0}^{69} \vec{fc}(i, k)$ for all $|k - i| \geq 8$, where k refers to atoms in the same string. The total collision force on an atom in a string is then the sum of these two collision forces.

Figure 1 visualizes three different non-trivial collisions involving rods and threads. The top row shows a collision of two rods, the middle row shows a colli-

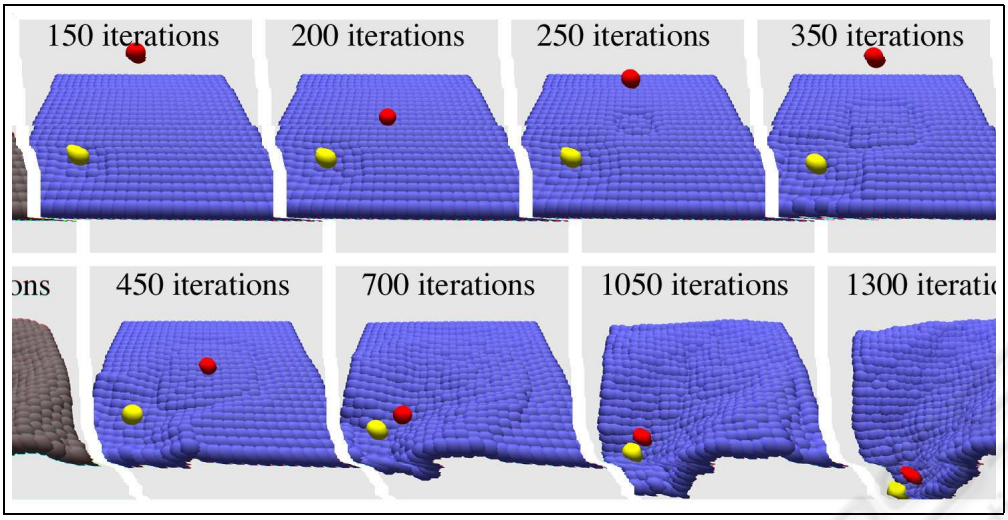

Figure 2: Visualization of two bouncing balls on an elastic surface. Initially both balls are in the air, falling down. The computed iteration rounds are indicated in each image.

sion of a rod and a thread, and the bottom row shows a collision of two threads. The initial configuration is the same in each case, shown in the leftmost image in the top row.

It is important to notice that we do not explicitly specify in the model how rods or threads collide. We only state which atoms of the two strings may collide with each other. Still, the collision dynamics seem very natural in each case, although they are entirely emergent.

Figure 1 reveals also other intricacies that we do not explicitly address in the model. Namely, the top row shows how the impact causes minute *vibrations* to the colliding rigid bodies. Also, the middle row shows how a non-rigid body *wraps* around a rigid body in a collision. The collision is, thus, *inelastic*. Still, as explained earlier, a collision force between two atoms as such captures only an elastic collision. An extreme of an inelastic collision is shown in the bottom row of the figure, where the collision of two non-rigid bodies lead to a severe *entanglement*.

The model of colliding threads and rods is computationally light; it can be computed and rendered in real-time. For instance, a modest 500MHz PowerPC G3-processor computes 4000 iteration rounds in about 3.2 seconds.

When considering robotics, this example shows that the atomic approach supports modelling and previsualization of materialistic properties. For instance, we could model and previsualize a robot lifting non-rigid components and then transporting them to some other location. Such a previsualization would reveal in real-time how gravity affects the non-rigid components during the process.

4 MODELLING WITH PLANES OF ATOMS

For modelling surfaces we use planes of atoms. In a plane, bond forces are used for defining the shape and the rigidness of the surface.

We exemplify use of planes of atoms by visualizing a non-trivial model, where two balls fall and bounce on an elastic surface. The gravitational force affects only the balls; thus, the surface acts as if it was a cloth floating on water.

The balls are modelled using single atoms. The surface is modelled using a square with a side of 25 atoms; thus, the index $n(x, y)$ of an atom at coordinate (x, y) in the surface is $25y + x$.

The two balls reside initially above the surface as indicated in Figure 2. The impulse on a falling ball is the sum of a constant gravity force, and collision forces. The collision forces cover collisions between the two balls, and between all the atoms in the surface. For instance, the total collision force on one of the falling balls is $\vec{fc}(i, j) + \sum_{x=0}^{24} \sum_{y=0}^{24} \vec{fc}(i, n(x, y))$, where i refers to the falling ball, j refers to the other falling ball, and $n(x, y)$ refers to an atom in the surface.

The atoms in the surface are initially overlapping and aligned along x and y-axes around the origin as shown in Figure 2. The initial distance, $d(i, j, x, y)$, between two atoms at coordinates (i, j) and (x, y) in the surface is given by $\sqrt{(i-x)^2+(j-y)^2}$. We consider this also the desired distance between the two atoms.

The impulse on an atom in the surface is the sum of bond forces and collision forces. Since the sur-

face is considered elastic, an atom is bound only to its neighboring atoms. Hence, the total bond force on an atom at coordinate (i, j) in the surface is $\sum_{x=0}^{24} \sum_{y=0}^{24} \vec{fb}(n(i,j), d(i,j,x,y), n(x,y))$ for all $0 < d(i,j,x,y) < 2$. Since we do not consider collisions between the atoms in the surface, the total collision force on an atom at coordinate (i, j) in the surface is simply $\vec{fc}(n(i,j),k) + \vec{fc}(n(i,j),l)$, where k and l refer to the falling balls.

Figure 2 visualizes the bouncing of the balls on the elastic surface. The images in the figure clearly show how the emergent dynamics reveals realistic details that we do not explicitly state in the model. For instance, both of the balls *bounce* from the surface; yet, we only define which atoms may collide in the model. Also, when the balls bounce from the surface, they generate *ripples* that gradually progress throughout the entire surface. This is clearly visible in images after 250 and 350 iterations. Also, once a ball is at rest on the surface, the surface *caves in* due to the gravity force on the ball. This can be observed from the images after 450 and 700 iterations. Lastly, as the surface becomes partially tilted, the balls start *sliding* on it. As there is nothing to stop the surface from tilting, the balls eventually fall out of the surface. This can be observed from the last image in the figure.

This model is also computationally light; it can be computed in and rendered in real-time. For instance, a modest 500MHz PowerPC G3-processor computes 4000 iteration rounds in about 8.3 seconds.

When considering robotics, this example shows that the atomic approach supports modelling of soft and elastic surfaces. For instance, we could model and previsualize a robot moving on an elastic surface. Such a model could be used for analysing how various cruising algorithms function, when the landscape is no longer even, and when the landscape caves in underneath the robot.

5 EMERGENT PROPERTIES: FRICTION AND ROTATION

The advantage of using emergent dynamics in previsualization is that it captures many physical phenomena on its own. Thus, we may omit them in the model entirely. Some of such phenomena are vibration, bouncing, friction, and rotation.

We exemplify emergent friction and rotation by visualizing a non-trivial model, where a ring is dropped on a surface. The surface is slightly tilted, so that when the ring bounces on the surface, it gains momentum and starts rolling. Rolling is due to emerging friction between the ring and the surface. We show also two other variants of this example; one, where the ring is elastic, and another, where the surface soft like a wet ground.

The ring is modelled as a specific string of atoms, whereas the surface is modelled as a plane of atoms. Initially the ring is above the surface and falls down due to gravity. The initial configuration is shown in the leftmost image in the top row in Figure 3.

The ring consists of 20 atoms. The atoms are partially overlapping, and form a perfectly aligned ring. More specifically, the i'th atom of the ring resides at $(4\cos(\frac{i\pi}{10}) - 4.3,\ 6 + 4\sin(\frac{i\pi}{10}),\ 0)$. In such a ring, the distance, $d(i,j)$, between the i'th and j'th atom is given by $\sqrt{(4 - 4\cos(\frac{|i-j|\pi}{10}))^2 + (4\sin(\frac{|i-j|\pi}{10}))^2}$. We consider this also the desired distance between the two atoms.

The impulse on an atom in the ring is the sum of a constant gravity force, and a number of bond forces and collision forces. The number of the bond forces in the ring depends on the elasticity of the ring. For an elastic ring, one atom is bound only to 8 neighboring atoms in the ring. Let $n(i)$ denote the index of the i'th atom in the ring, i.e., $n(i) = (20 + i) \mod 20$. Then, the total bond force on the i'th atom in the ring is $\sum_{j=i-4}^{i+4} \vec{fb}(i, d(i,n(j)), n(j))$ for all $i \neq j$. For an inelastic ring, the total bond force is simply $\sum_{j=i-9}^{i+9} \vec{fb}(i, d(i,n(j)), n(j))$ for all $i \neq j$. The collision forces, on the other hand, capture the collision of an atom in the ring with an atom in the surface. Thus, the total collision force on the i'th atom in the ring is $\sum_{j=0}^{288} \vec{fc}(i,j)$, where j refers to atoms in the surface.

The tilted surface is modelled using a square with a side of 17 atoms; thus, the index $n(x,y)$ of an atom at coordinate (x, y) is $17y + x$. The atoms are positioned sparsely, so the initial distance $d(i,j,x,y)$ between two atoms at coordinates (i, j) and (x, y) is given by $\sqrt{6.76(x-i)^2 + 0.01(x-i)^2 + 4(y-j)^2}$. We consider this also the desired distance between the two atoms.

The impulse on an atom in the surface is the sum of bond forces, collision forces, and a damping force. The total bond force on an atom at coordinate (i, j) is $\sum_{x=0}^{16} \sum_{y=0}^{16} \vec{fb}(n(i,j), d(i,j,x,y), n(x,y))$ for all $0 < d(i,j,x,y) < 2$. The collision forces capture the collision between an atom in the ring and an atom in the surface. Thus, the total collision force on an atom at coordinate (i, j) in the surface is $\sum_{k=0}^{19} \vec{fc}(n(i,j), k)$, where k refers to atoms in the ring. The damping force makes the surface to appear soft. The damping force on an atom at coordinate (i, j) in the surface is $\vec{fd}(n(i,j), 0.04)$. However, when the surface is considered totally inelastic, there is no damping force. Instead, we use a case specific external force. It is precisely opposite to the current impulse at each iteration round; thus, preventing the

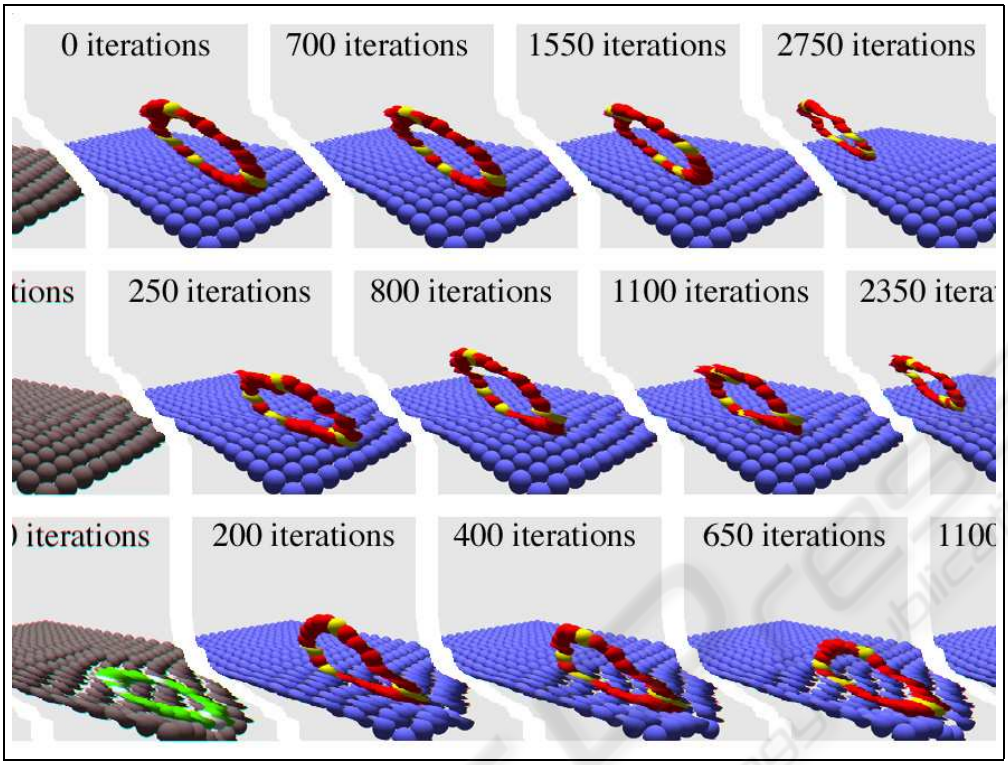

Figure 3: Visualization of what happens when a ring is dropped on a tilted surface. The top row shows an inelastic ring rolling on an inelastic surface. The middle row shows an elastic ring rolling on an inelastic surface. The bottom row shows an inelastic ring falling sideways on an soft surface. The computed iteration rounds are indicated in each image. The initial configuration, shown in the leftmost image in the top row, is the same for each of the three cases.

atoms in the surface from moving altogether.

Figure 3 visualizes three different variants of the example, where the ring is dropped on the surface. The top row shows the case with an inelastic ring and an inelastic surface. The middle row shows the case with an elastic ring and an inelastic surface. The bottom row shows the case with an elastic ring and a soft surface. The initial configuration is the same in each case, as shown in the leftmost image in the top row.

The dynamics is different in each case; however, in the first two cases, as the gravity pulls the ring along the tilted inelastic surface, the dropped ring gains slowly momentum and starts rolling on the surface. The rolling is due to emergent friction between the ring and the surface. Despite the emergent friction, in the second case, the ring preserves its elasticity, as shown by the images in the middle row in Figure 3.

In the last case, the dynamics differs considerably from the first two cases. Then, the soft surface caves in underneath the ring. This, in turn, causes the ring to fall sideways.

It is important to notice that, in all three cases, bouncing, friction, rolling, and falling are emergent dynamics. We do not explicitly specify them in the model; yet, the visualization clearly displays all these dynamics.

Again, the model is computationally light; and can be computed and rendered in real-time. For instance, a modest 500MHz PowerPC G3-processor computes 4000 iteration rounds in about 2.7 seconds.

When considering robotics, this examples shows that the atomic approach supports modelling of components with varying rigidness. For instance, we could model and previsualize what happens to a robot, when it suddenly has a flat tire.

6 CONCLUSION

In this paper, we discussed use of previsualization in robotics. In previsualization, a computer animates a highly underspecified model, to reveal if the model misses some key components, or if the dynamics is not as desired.

We investigated in this paper an atomic approach to previsualization. Its main advantage is that it uses emergent dynamics to reveal detailed interaction dynamics from a model. Thus, in the model, we only

need to specify the shape of the interacting components using atoms, and determine which of the atoms may collide with each other. In the atomic approach, this information is enough for computing the interaction dynamics between the components.

The interaction dynamics revealed by emergent dynamics is rich in realistic details. We illustrated how effortlessly the emergent dynamics captures vibration, entanglement, bouncing, caving in, friction, and rotation. All of these dynamics are usually considered too laborious when modelling free-form objects.

The atomic approach supports experimentation. In a model, we can replace one component with another without having to consider the changes in the interaction dynamics. The interaction dynamics is emergent. Also, we can easily attach software components, such as neural nets, to the model. They can be used, for instance, for controlling the dynamics on the physical components. This is done by inflicting desired forces on some specific atoms in the physical components.

Another advantage of the atomic approach is that the forces on the atoms are captured by simple and compositional mathematical formulae. Consequently, models become computationally light, and they can be computed and rendered in real-time. An object-oriented implementation of the atomic approach can be found, for instance, in (Rönkkö, 2003a).

Clearly, the use of the atomic approach is not limit to previsualization of solid objects in robotics. It can also be applied to other fields, such as study of *fluid dynamics*. Then, fluid as material is modelled as a collection of individual atoms. The *denseness* of the fluid depends on the number of atoms, whereas the *stiffness* of the fluid depends on the damping forces on the atoms. The *granularity* of the fluid is controlled by introducing pairs, or triples, of atoms with bond forces in to the fluid.

The atomic approach is our first attempt to harness emergent dynamics to help design and development in robotics. In our future studies, we wish to find out how far we can push this technology. In particular, our near future goal is to polish the approach so that it is easily applicable by scientists and researchers studying complex interaction dynamics.

REFERENCES

Adams, R. A. (1999). *Calculus: A Complete Course*. Addison-Wesley, Don Mills, 4th edition.

Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press, Cambridge.

Ferster, B. (1998). Idea editing: Previsualization for feature films. *POST Magazine*.

Mosterman, P. (1999). An overview of hybrid simulation phenomena and their support by simulation packages. In Vaandrager, F. W. and van Schuppen, J. H., editors, *Proceedings of Hybrid Systems: Computation and Control*, number 1569 in LNCS, pages 165–177. Springer-Verlag, Berlin.

Odell, J. (1998). Agents and emergence. *Distributed Computing*.

Reeves, W. T. (1983). Particle systems - a technique for modeling a class of fuzzy objects. *Computer Graphics*, 17(3):359–376.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34.

Rönkkö, M. (2003a). Colliding threads and rods: An atomic approach. Online illustrative document, Department of Computer Science, University of Kuopio.

Rönkkö, M. (2003b). Previsualization in robotics: An atomic approach. Technical Report A/2003/4, Department of Computer Science, University of Kuopio.

Thornton, S. T. and Marion, J. B. (2004). *Classical Dynamics of Particles and Systems*. Brooks/Cole – Thomson Learning, Belmont, 5th edition.

Witkin, A. (1997). Physically based modeling: Principles and practice. Online Siggraph '97 Course notes.

Young, H. D. and Freedman, R. A. (2000). *University Physics with Modern Physics*. Addison-Wesley, San Francisco, 10th edition.

Zill, D. G. and Cullen, M. R. (1997). *Differential Equations with Boundary-value problems*. Brooks/Cole Publishing Company, Pacific Grove, 4th edition.