COMPUTER GRAPHICS I COMP.5460

LITERATURE REVIEW 1

Computational Design of Wind-up Toys

This research paper is a deep study on the traditional wind-up toys and along with that the ways in which they can be improved to function better. Wind-up toys are mechanical assemblies that perform intriguing motions driven by a simple spring motor. Due to the limited motor force and small body size, wind-up toys often employ higher pair joints of less frictional contacts and connector parts of nontrivial shapes to transfer motions. This paper presents a computational system to aid the design of wind-up toys, focusing on constructing a compact internal wind-up mechanism to realize user-requested part motions. Key contributions include an analytical modelling of a wide variety of elemental mechanisms found in common wind-up toys, including their geometry and kinematics, conceptual design of wind-up mechanisms by computing motion transfer trees to realize the requested part motions, automatic construction of wind-up mechanisms by connecting multiple elemental mechanisms, and an optimization on the part and joint geometry with an objective of compacting the mechanism, reducing its weight, and avoiding collision.

Firstly, the paper introduces the present system of wind-up toys. they come as lightweight toys with compact internal mechanical assemblies that are powered by clockwork motors attached to a spring key. Once such a key is rotated to tighten the spring and released, the stored potential energy drives the toy’s internal mechanical parts, which in turn drive the end-effector parts of the toy to perform intriguing motion(s). First, they are driven by a simple spring motor that can only give out limited energy and torque. Second, their parts usually perform simple periodic motions that are only roughly specified. Third, they often have a small lightweight body, particularly for those that perform locomotion. Lastly, they are usually presented as cartoon shapes with the wind-up mechanism and motor enclosed inside.

Subsequently, the paper goes on explain an interactive system to aid the design and fabrication of wind-up toys, motivated by the use of 3D printing for making personalized wind-up toy designs. But to implement this idea, first several challenges have to be tackled. First, unlike the hand-operated cranks and electric motors employed in previous works, common spring motors in wind-up toys provide only small torque. To drive the various toy parts, such torque has to overcome both friction (contacts) and weight (parts). Second, wind-up toys employ a rich variety of higher pair joints for transferring different types of motions; many of them have not been explored.

To implement this idea and meet the challenges, the key idea is to construct and optimize a compact and lightweight wind-up mechanism to progressively transfer motion from the internal spring motor to toy parts through higher pair joints. The idea was implemented in the following steps:

1. By exploring common wind-up toys, we constructed a table of elemental mechanisms that involve higher pair joints, and categorized a wide range of fundamental motion transfer patter.

2. To produce conceptual designs, we developed computational methods to automatically enumerate all possible valid mechanism designs composed from the elemental mechanisms.

3. It was turned each conceptual candidate design into a working mechanism with full geometry by connecting the involved elemental mechanisms from the driving source (usually composed of a round cam) to the toy's end-effector parts recursively.

4. Devised an optimization model to maximize the similarity between the user-requested part motions and resulting toy motions, to compact the wind-up mechanism.

The system takes 3 inputs as follows:

- A 3D mesh model that has been pre-segmented into logical parts.
- User-prescribed motion (an axis and a range) for each segmented toy part.
- The pose of the motor.

The paper has a study on model elemental mechanisms and in that it is stated that there are eleven elemental mechanisms (abbreviated as eleMech) that build up the wind-up mechanisms, each of which delivers a fundamental motion transfer task. Each eleMech has three kinds of parts: (i) a driver, which initializes an eleMech's motion, (ii) a follower, which reacts and moves accordingly, and (iii) supporter(s), which are fixed structures that constrain the motion of driver and follower. Moving on with the inlialization of wind-up mechanism. Candidate conceptual designs that roughly match the user-prescribed motions at the end effectors and connect eleMechs into an initial wind-up mechanism for a candidate design are constructed.

In the first step, candidate conceptual designs are generated. Then later on mechanism geometry are initialized. to initialize a wind-up mechanism for each candidate conceptual design, such that the mechanism is functioning but may not stay inside the toy body, and its end effectors may not follow the prescribed motions. Given a motion transfer tree associated with left or right cam, we first pick eleMechs associated with the tree edges, initialize them with default geometric parameters. After that the next step is to optimize wind-up mechanism. For complete optimization some sub steps like formulation and solving the optimization are required.

Experimental results:

- Mechanism finishing: after the optimization, still have a few post-processing steps to complete the wind-up toy, among which the last three steps require certain amount of manual effort.
- Results: implemented our system in C++ and tested it on desktop PC with a 3.4GHz CPU and 8GB memory.
- User evaluation: tested on 6 participants.
- Statistics: the average time taken to generate a set of candidate conceptual designs, to coarsely optimize a single mechanism, and to finely optimize a single mechanism are ∼16, ∼35, and ∼190 sec., respectively.
- Fabrication: fabricated six of our wind-up toy results using a high-resolution desktop SLA printer (printing volume: 130×130×180 mm3, printing resolution: 0.1mm) with photosensitive resin material (density: 1.10-1.15 g/cm3).

Computational Design of Walking Automata

This paper is all about creating mechanical automata that can walk in stable and pleasing manners. It is proposed to use computational design to offset the technical difficulties of this process. A simple drag-and-drop interface allows casual users to create personalized walking toys from a library of pre-defined template mechanisms. Provided with this input, our method leverages physical simulation and evolutionary optimization to refine the mechanical designs such that the resulting toys are able to walk. The optimization process is guided by an intuitive set of objectives that measure the quality of the walking motions. It demonstrates the approach on a set of simulated mechanical toys with different numbers of legs and various distinct gaits.

The method allows users to intuitively create unique automata designs that walk stably once fabricated. First, a space of linkage configurations which are likely to lead to stable walking is learnt. Then, the user designs the automata by placing 2–4 linkage templates onto a body at arbitrary positions. From this initialization, the overall mechanical structures of the automata is optimized, allowing to automatically discover how to walk with the same intuitive set of objective functions. This approach integrates physics-based simulation of mechanical assemblies with an evolutionary optimization algorithm that is able to explore the complex design space of these structures.

The inspiration for this came from character animation which is translating virtual walk simulations into the real world. This is a very difficult task. Also, along with this the field of computation design and fabrication is used for more knowledge. This field reduces the difficulty of design and manufacturing problems by creating tools which forego or reduce the need for expert domain knowledge. Some methods aim to bring virtual characters to the real world, and it is now possible to create 3D printable representations of virtual characters with joints, to design mechanical toys capable of interesting (non-walking) motions.

Working of the system:

Each of automatons has a body with 2–4 linkages attached. Each linkage belongs to a set of linkage classes, with each parameterizable by pin joints and timings. The 12-16 kinematic parameters pk, encode the location of each pin joint, and so define the mechanical configuration of the automata. A simple drag-and-drop design tool to the user is provided, who chooses and places linkage classes at arbitrary positions on the body. Following this, a stochastic genetic algorithm (CMA) and a walking objective function is deployed to optimize the linkage instance parameters with respect to the body in a rigid-body physical simulation. The simulation measures the quality of walking motions as the parameters of the automata — the mechanical linkage structures — change. It also previews the physical prototype output.

Getting the system actually walking is a difficult task after the designing part is over. During optimization, for each explored set of parameters, i.e., each automaton configuration, we compute the mass and moment of inertia of each rigid body part, and then physically simulate the automaton. The simulation proceeds until a failure mode is encountered, or until a fixed amount of simulation time T = 30s has elapsed. For a perfect walk many aspects such as distance, upright, smoothness, effort, regularizer are taken into consideration.

Then optimization comes into the picture. It was noted that linkages can be analysed for validity before optimization to learn valid configurations for walking. It is useful to learn parametric function and data pre-processing for linkage.

Before giving out the final results, the whole process was experimented on different kinds of shapes and robots like dog, grandpa bot, lobster, gorilla and giraffe. Physical prototypes were created for dog and the lobster.

Similarity and differences between the two papers

The first paper mainly focuses improving the design of wind-up toys. The traditional wind-up toys have many issues with fixed design and limited use and wastage of energy. Thus the paper suggests many ways to overcome the issues like automatically construct conceptual designs and initialize mechanisms by connecting elemental mechanisms. While the second paper focus mainly on the walking part of an automata or a robot. Walking is not a easy task to be done for non-living objects. This paper has in-depth study of how to successfully make an object walk without making it jump or wobble. The paper suggests ways to optimize this.

The similarity between both the papers is that they both deal with the motion or movement portion of toys. They want to make user defined objects which can have a perfect walk or the required movement.

References:

Computational Design of Wind-up Toys:

ACM Transactions on Graphics, Vol. 36, No. 6, Article 238. Publication date: November 2017

Computational Design of Walking Automata:

Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation August 2015