Title	Augmented reality-based block piling game with superimposed collapse prediction
Author(s)	Okamoto, Kazuya; Kume, Naoto; Tokunaga, Tatsuya; Tanaka, Yoko; Terasawa, Noriaki; Tsukasa, Takashi; Takemura, Tadamasa; Yoshihara, Hiroyuki
Citation	Virtual Reality (2013), 17(4): 279-292
Issue Date	2013-08-28
URL	http://hdl.handle.net/2433/193297
Right	The final publication is available at Springer via http://dx.doi.org/10.1007/s10055-012-0219-0
Type	Journal Article
Textversion	author

Augmented Reality–based Block Piling Game with Superimposed Collapse Prediction

Kazuya Okamoto[1,2], Naoto Kume[1,2], Tatsuya Tokunaga[2], Yoko Tanaka[2], Noriaki Terasawa[2], Takashi Tsukasa[2], Tadamasa Takemura[1,2], and Hiroyuki Yoshihara[1,2]

[1]*Department of Medical Informatics, Kyoto University Hospital, Japan*

[2]*Graduate School of Informatics, Kyoto University, Japan*

Tel: +81-75-751-3165

Fax: +81-75-751-3077

{kazuya, kume, tokunaga, tanaka, terasawa, ttsukasa, takemura, lob}@kuhp.kyoto-u.ac.jp

Abstract: Understanding what cannot be seen is difficult. Physical behavior can be explained on the basis of physical theories even if the behavior cannot be observed. Explanation of what is physically happening in the real world would become easy, however, if annotations were superimposed on the real objects. Herein, the authors demonstrate how an understanding of a physical event can be facilitated by overlapping a real-world situation with a simulation that predicts a future state. This idea is demonstrated in a game application in which a player stacks blocks into a pile until it collapses. In general, it is easy to estimate whether a block on the edge of a table will fall or not. However, it is more difficult to predict whether a stack of many blocks will collapse, and in what manner the stack will collapse. Even though previous research has demonstrated that the problem of how two-dimensionally stacked blocks collapse can be reduced to solving a sequence of convex quadratic programs, algorithms for convex quadratic programs require massive computational resources. Hence, the authors developed a fast and new algorithm based on a linear program. The proposed algorithm realizes real-time simulation based on physics that superimposes predicted collapse. The block that is predicted to fall is superimposed on the real block with a lit background projection. The system was evaluated in an experiment and superimposed augmented reality (AR) annotation was observed to be efficient. The system was also demonstrated in game contests and received positive feedback and comments.

1. Introduction

Understanding what cannot be seen is difficult. Physical behavior can be explained by physical theories even if the behavior cannot be observed. Explanation of what is physically happening in the real world would become easy if annotations were superimposed on the actual objects. People have a general grasp of physical behavior and can predict what is going to happen in the future. For example, we can predict where an oncoming car will pass in the next few seconds, and we can predict how a thrown ball will fall. However, the precise prediction of consequence is not easy. For instance, we cannot precisely predict the course of an oncoming car with just a glance even if we know the car is oncoming. In this case, for precise prediction, we must know the accurate velocity of the car, the direction of the wheels, etc. If annotation of information that is not able to be perceived in nature is superimposed over the car, it would be quite easy to predict what would happen next in the real world. Eventually, the annotation of predicted future events may enhance human perception to such an extent that it would be like we had a new ability. This idea is the basis of augmented reality.

The development of methods for the registration of augmented virtual contents in the real world has a long history of improving the accuracy of its position. If the represented position is not augmented precisely, the annotation would appear unnatural or, what is worse, the annotation would lose the proper position and not make any sense. Previous research has thus been very focused on how to achieve correct position and context. A typical solution is to employ a head mount display (HMD) and location sensors in the environment. Even if the system achieves accurate registration, the complicated system makes installation difficult, however. Also, the size of the augmented content is restricted to inside the environment. Moreover, user's focus is generally distracted by the equipped gadgets such as HMD when the visual quality is not enough to get immersion. Hence, to the furthest extent possible, AR should be free of any gadgets that are attached to users. An ideal AR could be achieved by real objects that are equipped with the virtual annotation content.

In this context, the authors attempt to design AR that is free from any gadgets attached to users, and to pursue the expression of annotations by the real objects. Needless to say, it cannot be said that all real object is suitable to equip virtual contents in a natural way. The authors suppose that a simple object can be designed to be annotated on itself. This research focuses on a block that is simple enough to simulate physical behavior. There exists a classical physics problem related to blocks, the so-called "Overhang problem." In this paper, the authors present a method that can be used to determine how events happening in the real world can be physically simulated, especially events relating to the collapse behavior of piled blocks. The authors also provide a method to superimpose the result of the simulation on real objects.

2. Related works

General AR provides additional information through text and images that are superimposed on the view of the real world. Conventional AR preliminary prepares the additional information. The information is presented based on the user's location. This works in an environment that provides static information, such as the name of gadgets and the name of buildings. However, the static information is not enough to understand several phenomena in the real world. It is hard to explain dynamics and other theories with the prepared static information because phenomenon is caused by user's behaviors. Therefore, the annotation is required to be generated dynamically related to user's behavior. As Matysczo demonstrated, simple dynamic AR connects user's behavior to the input of a virtually simulated environment (Matysczo et al. 2004). In this case, the input and output interfaces are in the real world. And simulation of physical behaviors is completed in the virtual environment. Particularly in relation to teaching dynamics, the simulation model should be closed in the virtual environment because uncontrolled results should not occur when theories are being learned (Kaufmann and Meyer. 2008). Therefore, from the viewpoint of interaction with the virtual environment, only an AR-based viewer is needed to get an intuitive view.

On the other hand, from the viewpoint of interaction with the real world, a key feature of AR is superimposition on real objects. Dynamic AR would be realized if the additional information were given by the context that reflects user's behaviors in the real world. As such, real-time simulation is a key technology in achieving dynamic AR. Some of the conventional dynamic AR systems are designed for trajectory annotations in billiards by physical simulation (Jebara et al. 1997; Shih et al. 2009). This research achieves accurate prediction of ball behavior through dynamics simulations. The annotation is provided between user's shots. Eventually, measurement and simulation have enough time to give feedback. If the trajectory annotation is provided on the rolling ball, the AR system requires real-time simulation. To simulate physical phenomena, a simulation model that is fast enough to calculate the result in real time for superimposition is needed.

To begin with, the simulation model must also be capable of precise physical simulation. However, almost all physical simulations have been evaluated by judging not whether the result of the simulation is the same as the real situation but whether the result appears sufficiently natural. As a typical example, a simulation model of connected bodies (Bender and Schmitt 2006) and a collision simulation model of rigid and deformable bodies (Shinar et al. 2008) were developed. Although the simulation models treat pair-wise relations of bodies, Peterson and Zwick demonstrated that pair-wise relations are incapable of simulating the movement of more than three things (Paterson and Zwick 2006; Paterson and Zwick 2009). Also, Rosato et al. simulated the fact that when a can containing different sized grains is shaken, the big grains rise—this simulation is well known for demonstrating the "Brazil nut effect" (Rosato et al. 1987). Shinbrot demonstrated that large-grain "Brazil nuts" may sink, however, depending on their size and density. In this case, if these incomplete simulations and the real objects were superimposed, we would recognize that

the simulations do not correspond to reality. Conversely, several simulations have not been developed based on physical principles.

3. Purpose

The purpose of this study is to provide physics-based annotation explaining the phenomenon in the real world in real time. For the real-time explanation of the phenomenon, AR is apparently suitable. The annotation also should not prevent the users from observing the phenomenon itself. The accuracy of physics-based simulation is crucial as well as the registration accuracy of the presentation. Therefore, the requirements of dynamic annotation AR for interaction with real objects are as follows:

1. AR annotation with superimposition on real objects without gadgets equipped by the user
2. Real-time simulation for prediction of the phenomenon
3. Physical simulation based on physical principals

In this context, the authors aim to provide a dynamic annotation AR that provides annotations on real objects in real time to suggest the prediction of physical phenomena that reflects user behaviors. Especially, the authors select cubical block as primitive object to observe dynamics. This study focuses on the balance visualization of piled blocks. The internal power balance of the piled blocks should be calculated. Also, a superimposing method related to the piled block collapse is required. Generally, the problem of the piled block collapse is known as "Overhang problem" in mathematics domain. Consequently, the authors propose an AR-based block piling game related to the overhang problem. In the game, a player piles cubical blocks vertically along a wall two-dimensionally, and the system indicates which block will lose balance.

In several studies, the overhang problem has been calculated theoretically in mathematics, but not on the real-time simulation. In the overhang problem, unit-length blocks are piled two-dimensionally like a block wall. Recently, Röte and Zwick developed an algorithm that explains how piled blocks collapse (Röte and Zwick 2011). However, the algorithm is not appropriate in terms of execution time to realize superimposition of the simulated result and reality. Hence, a new and fast algorithm to determine which blocks will start to fall is needed to realize AR annotation.

3.1. Overhang problem

The overhang problem has frequently been reported as a classical physical problem in engineering textbooks, even those published in the nineteenth century (Phear 1850; Walton 1855). The problem involves determining how far a stack of unit-length blocks can hang over the edge of a surface. The optimal solution using four blocks is depicted in Figure 1. In this case, the overhang distance of the stack is approximately 1.168 (Ainley 1979). It would appear quite simple to find the optimal solutions for this problem. However, the optimal solutions in the case of a large number of blocks have not yet been found. A classical solution of n blocks achieves half of the nth harmonic number (H_n) by piling n blocks on top of the others, with the ith block from the top

extending by $1/2i$ beyond the block below (see the stack of white blocks in Figure 2, the harmonic stack of 20 blocks). The overhang distance of the harmonic stack is $\frac{1}{2}H_n = \frac{1}{2}\sum_{i=1}^{n}\frac{1}{i} \approx \frac{1}{2}\ln n$.

However, the solution is not optimal because the overhang distance of the harmonic stack of four blocks is 1.042, which is less than 1.168 in Figure 1. Paterson and Zwick demonstrated the optimal solution using 20 blocks, as depicted in Figure 2 (Paterson and Zwick 2006; Paterson and Zwick 2009). The optimal solution with 20 blocks suggests that the optimal solution in the case of any number of blocks cannot be derived from any regularity. Therefore, numerous researchers have been interested in the overhang problem and have studied it for a long time. There is another interesting aspect to this problem. Even in a simple case, we cannot guess which block in a stack will cause the collapse. For instance, can you estimate which of the piled blocks in Figure 2 will collapse or if the stack will not collapse? Since the stack is the optimal solution to the overhang problem, a slight shift to the right will cause collapse. Where do you think a block will start to collapse when the rightmost block shifts slightly to the right? In the same way as in Figure 2, it is difficult to determine the collapse of a given stack of blocks just by looking at it.

Conventional algorithms of the overhang problem give the solution whether the blocks collapse or not. It is not sufficient for the annotation to give the solution which block will start to collapse. Therefore, the development of new algorithm is required as well as the improvement of the conventional algorithm to real-time simulation.

3.2.   AR-based block pile game

The authors aim to develop an AR-based system of a block piling game. Players try to solve the overhang problem in practice by piling blocks along a wall two-dimensionally. Figure 3 depicts how to pile up the blocks. Because the wall stands vertically, a stack of blocks along the wall has a two-dimensional deployment. The goal of the overhang problem is to find out how far the stack of blocks can reach from the edge of a table. A unit of length is defined as the length of a block to be 100 points. The distance from the edge of a table to the far point of the stack is measured by the unit. When the stack protrudes three blocks from the edge of a table, the score is 300 points. The score unit is equal to 100 times the overhang distance written in LP papers (e.g., Ainley 1979).

Even though the overhang problem itself is interesting, players must predict the physical phenomena before they actually experience it. While the players pile blocks, an annotation predicting which block is unbalanced should be superimposed on the actual blocks. Thus, players have time to consider how the next block should be placed on the stack in order to keep the stack stable before they take their hand away from the piled blocks. Players' hands on the stack while piling blocks works as a fictitious force to sustain it. Annotations are needed to be generated based on physical simulation using the new algorithm which supports the detection of the collapse starting block. If a player ignores an annotation of unbalanced blocks and separates his/her hands from the stack, these fictitious forces will be missing and the stack will collapse.

5

4. Methods

The design of the game system structure is depicted in Figure 4. The system consists of three modules: a measurement module, calculation module, and presentation module. The authors employ simple structure on the measurement and the presentation because the key technology of the proposed dynamic annotation AR is the fast prediction algorithm on the calculation for the proof of concept. Consequently, transparent block is employed for back projection because the block should be annotated without preventing user's manipulation. The system components are transparent acrylic blocks, a frosted glass board as a wall, a projector, a video camera, and a simulation computer.

The measurement module sends images captured by a camera to the calculation module. Image processing analyzes the screen image and determines the position of the blocks in the image. The calculation module detects unbalanced blocks by the position of the blocks. The simulation computer calculates the balance of the given stack using the proposed algorithm which is described at section 4.3. Simulation derives whether the stack will collapse or not. Also, the simulation concludes which blocks will topple. The presentation module displays a colored shadow of the detected blocks on the frosted glass screen by overlaying the shape of the blocks via rear projection. Thus, the detected blocks are illuminated from the player's point of view. Therefore, the players can recognize the exact block that is losing balance as soon as they place it on the stack.

First, the overview of the measurement and the presentation module is given by the following sections. Later, the detail of the proposed simulation algorithm on the calculation module is described.

4.1. Measurement module: Image processing

The difficulties in the system include how to measure the location of blocks that are adjacent to each other, and how to eliminate the rear projection effects of the augmented annotation on blocks. The authors employ a simple image processing method to allocate block position in a picture retrieved via a DV camera. The camera captures an approximately 30,000 pixel (640 × 480) DV format image. Captured images of the piled blocks are transported to a simulation computer. Transparent acrylic block size is 7.5 cm by 2.5 cm by 1.5 cm. Every block has two black lines to mark the edges. The markers make it easy to allocate the end of the block by image processing.

The frosted glass board is an input interface of the block measurement as well as an output interface of the rear projection. The board, acting as an input interface, requires appropriate optical transparency. If optical transparency is too low, image processing cannot find the black line markers. Conversely, if optical transparency is too high, image processing frequently errs in regarding the player's shadow, which is a black area in a picture, as a black line marker. Moreover, presentation by rear projection requires a low optical transparency to enable a projector to display the annotation image on the board. Hence, a frosted glass board that has appropriate optical transparency was carefully chosen. Figure 5 depicts a block coming in contact with the frosted glass board. Figure 6 depicts a block that is kept approximately 1 cm away from the board. Figures

6

5 and 6 illustrate that an object which comes in contact with the board can be clearly captured in the picture, and that an object which is kept away from the board cannot be captured clearly. Distanced objects get blurred so that the black area caused by the player's shadow does not appear while the player's hands and body are kept off the board. Then, the player's hands make no noise in regard to block recognition in the image processing.

Because the contrast between the black line markers of the block and the transparent body of the block is high enough to recognize the block position by a single primary color (green), values are chosen among the three primary colors to detect the markers. That is, image processing detects the block position by scanning green values of a screen image. Figure 7 depicts a raw image of green values. Several blocks are only visible at black markers. Blocks can be recognized by parsing the green values and matching the black lines of each block based on already-known block length. Figure 8 depicts an image of the blocks recognized by image processing. Every block is recognized by detecting pairs of markers while moving from the left to the right. As a vulnerable point, if either the left or the right edge is missing, the entire block recognition will fail. To improve the accuracy of the block position recognition, image processing utilizes the already-known size of black line height and width. Although the black line width is very narrow, recognition is improved efficiently. Moreover, red values and blue values are used only by the presentation module. Thus, isolation of the utilizing color aspects can enable interference between measurement and presentation on the same screen to be avoided.

4.2.   Presentation module: Annotation

There are some possible options to annotate collapse phenomenon as whether, which and how the collapse happens. Simplest annotation is just an indication of whether the pile will collapse or not. However, it is not sufficient to suggest users how to prevent. Therefore, the annotation that gives which block will collapse is needed. Finally, to understand the collapse phenomenon, how each of the blocks moves and falls down during collapse should be given. However, it is too much to simulate the entire physical phenomenon. Also, the simulation is basically not necessary to cause an action for preventing the collapse. Therefore, the authors provide the solution to annotate that that whether the pile is balanced or not, and which block starts collapse.

Annotation is needed to be given on time during play. Annotation should provide information how is the state of the internal power balance between blocks before actual phenomena such as collapse happens. Although the exact coordination of unbalanced point on the block losing balance can be located, it is too detail to the players. Therefore, an abstract annotation that suggests the block as the cause should be given.

Figure 9 depicts the scenery for playing the game named "Bricklayer". Players pile blocks that are prepared in front of them. The frosted glass board is an interface against which the players pile blocks as well as a display for the prediction result. Prediction results are superimposed on the actual piled blocks. If a block illuminates blue, the block is stable. Conversely, if a block illuminates red, that block may cause a stack collapse. Figure 10 depicts one of the high scores of the overhang problem (around 300 points). When all blocks are illuminated blue, this suggests a

stable stack. In contrast, Figure 11 depicts a case in which a block is illuminated red. This suggests that the stack is about to collapse starting from the illuminated block when the player lets go of the block on the stack. In addition, the system beeps when a red block appears. The score is indicated at the top right corner of the screen (item 'MAX' is the score calculated from the edge of the surface). The structure of piled blocks and the current prediction results are illustrated at the top left corner.

4.3.   Calculation module: Collapse prediction algorithm

As a result of image processing, the simulation program understands how each block is piled on other blocks. The author proposes a fast collapse prediction algorithm which gives the solution which blocks will start collapse. The proposed algorithm is developed based on conventional algorithms which give the solution whether the pile collapses or not.

Paterson and Zwick demonstrated that the problem of whether two-dimensionally stacked block collapse will occur can be reduced to solving a linear program (LP) (Paterson and Zwick 2006; Paterson and Zwick 2009). Moreover, Röte and Zwick demonstrated that the problem of how two-dimensionally stacked blocks collapse can be reduced to solving a sequence of convex quadratic programs (Röte and Zwick 2011).

Algorithms for convex quadratic programs require an $O(n^3L)$ time, where n is the number of variables and L is the input size (Monteiro and Adler 1989). Using a mainframe, an algorithm developed by Röte and Zwick took approximately one-third of a second to solve a convex quadratic program when the number of blocks was 25 (Röte and Zwick 2011). Since the number of variables and the input size are nearly proportional to the number of blocks, the algorithm requires considerable computational complexity to achieve real-time simulation. For example, when the number of blocks was 50 ($n = 50$), using a quadratic program, the calculation time would theoretically require over 5 seconds. Hence, the authors focus on the development of a new and fast algorithm based on LP to predict the exact blocks that will lose balance.

First, the existing method of the reduction from the problem of detecting collapse to an LP will be described in subsection 4.3.1. Subsequently, an improvement in the method will be explained in subsection 4.3.2.

4.3.1. Prediction of whether collapse will occur in the overhang problem

A collapse detection algorithm is not simple. For example, when a block is piled on another block, the center of gravity of the upper block is equal to the force exerted on the lower block. However, when a block is piled on more than one block, the forces that are assigned to each lower block cannot be uniquely determined. Hence, the forces between blocks must be handled as variables. In the two-dimensional piling problem, since the lengths of all blocks are the same, the maximum number of blocks on top of each block is two and the maximum number of blocks under each block is also two. Figure 12 illustrates the forces corresponding to the block in the middle of

a stack. The black arrows indicate downward gravitational forces of unit magnitude equal to the weight of a block. The red arrows indicate downward forces from upper blocks. The blue arrows indicate upward forces from lower blocks. The magnitude of a downward force and the magnitude of the upward force corresponding to the downward force are the same due to the principle of action and reaction. It is important to denote the forces between the piled two blocks. There are probable combinations of many forces that are applied to the surface of the contact area of each block. Mathematically, forces over a contact surface can be integrated into a resultant force at the edge points that belong to the contacted blocks, as depicted in Figure 12.

When a stack holds, the forces applied to every block satisfy the following two equations. As an example, consider block b. Note that the weight of each block is a unit (1, that is to say). The forces from the lower blocks to block b are u_1, u_2, …, u_m and the forces from the upper blocks to block b are d_1, d_2, …, $d_{m'}$. When the x-coordinate of the center of block b is 0, let $x(u_1)$, $x(u_2)$, …, $x(u_m)$, $x(d_1)$, $x(d_2)$, …, $x(d_{m'})$ be the x-coordinate of u_1, u_2, …, u_m, d_1, d_2, …, $d_{m'}$, respectively.

$$\sum_{i=1}^{m} u_i = \sum_{i=1}^{m'} d_i + 1. \qquad (1)$$

$$\sum_{i=1}^{m} x(u_i)u_i = \sum_{i=1}^{m'} x(d_i)d_i. \qquad (2)$$

The first equation states that the net force applied to block b is zero whereas the second states that the net moment to block b is zero. A stack does not collapse while every force is of non-negative magnitude such that the forces realize the above two equations of each block with the principle of action and reaction held. In contrast, when no value can be assigned to the magnitude of each force realizing all equations, the stack collapses. Value assignment is calculated utilizing an LP with all equations as constraint conditions (Paterson and Zwick 2006; Paterson and Zwick 2009). Namely, if a solution on an LP exists, a stack does not collapse; otherwise, the stack collapses. The algorithm exploiting this reduction, however, never shows what the magnitude of each force is and which blocks will start to fall when a stack collapses. Therefore, blocks losing their balance are not detected when a stack collapses.

4.3.2. Prediction of which block is unbalanced

If the pile is stable, it stands by itself. Otherwise the pile requires additional force to sustain the structure. For the prediction of which block is unbalanced, the author proposes an algorithm to find the minimum external force which is indispensable to sustain the pile. When the structure does not require the external force, the pile is stable. On the other hand, when there is a block which requires the minimum external force, the algorithm can define the block is unbalanced. Also, actual block collapse is strongly affected by friction among block surfaces as well as between block and wall. Moment forces determined by the relation between block edges also should be considered. The authors employ the assumption that the surface is not sticky and enough smooth , to neglect friction. Then, the authors propose the prediction algorithm involving the moment effect to improve the accuracy of the collapse block definition.

Improvement by applying fictitious forces on a linear program

The authors propose an improvement for the collapse block prediction based on LP. The previously mentioned algorithm predicts *whether* a stack will collapse or not, but it does not indicate *which* blocks will lose balance. In order to detect the unbalanced blocks, the authors utilize fictitious forces to sustain an unstable stack. When the stack is stable, a fictitious force is not required. Otherwise, when the stack loses balance, a positive solution is given to fictitious forces. Therefore, we can see which blocks require assistance based on observation of the required fictitious forces.

When a block is on top of two blocks, the upper block is stable and never collapses. Hence, each fictitious force is provided to a block atop another block (a one-on-one block, in short) and applied to the edge on the other side of the lower block. For example, suppose block b of Figure 13 starts to roll and fall. In this case, an upward fictitious force f' is applied to the left edge (see Figure 14). Fictitious force f' can prevent block b from rolling in a counterclockwise direction.

Since fictitious forces prevent a collapse, feasible solutions of an LP always exist through their application. Consequently, the blocks that we should support to sustain a stack can be determined by setting the summation of the magnitudes of all fictitious forces as an objective function, which should be minimized. Note that the values of fictitious forces are restricted to non-negative values by exploiting constraint conditions and that the objective function value of an optimal solution is at least zero. When the objective function value of an optimal solution equals zero, the stack never collapses. Conversely, when the objective function value of an optimal solution exceeds zero, the stack will collapse. It is evident that there are no feasible solutions when all fictitious forces are zero. Moreover, unbalanced blocks that can be assisted by the minimum force can be derived by checking which fictitious forces are positive. However, the blocks that are given minimum force to sustain the stack are not always the same as the blocks that cause collapse. For instance, there are three patterns for the cause of collapse depicted in Figure 15. First, only the upper block b_1 is unbalanced. Second, both blocks b_1 and b_2 are unbalanced. Third, only the middle block b_2 is unbalanced.

1) The case that only the upper block b_1 is unbalanced: For the first case, a fictitious force is applied to block b_1 because block b_1 is obviously the cause of collapse. Therefore, the AR system indicates block b_1 as a result of prediction.

The case where both blocks b_1 and b_2 are unbalanced: For the second case, there are two probable results of fictitious forces indicating the cause of collapse. One possibility is that fictitious forces to both blocks b_1 and b_2 are applied to sustain the stack. Another possibility is that a fictitious force is only applied to block b_1 even if both blocks lose balance. In other words, there is a case in which only a fictitious force to block b_1 can sustain both blocks. For the former possibility, the result of fictitious forces directly indicates the cause of collapse. Thus, it is natural to say that the AR system should indicate both blocks b_1 and b_2 as the cause of collapse. However, for the latter possibility, it is necessary to distinguish the cause of collapse that originates from block b_1 or from both blocks.

In the latter possibility, even though the result of fictitious force is given only to block b_1, the AR system should somehow display the prediction on block b_1 as well as on block b_2.

2) The case in which only the middle block b_2 is unbalanced: For the third case, there are also two probable results of fictitious forces indicating the cause of collapse. One possibility is that only a fictitious force to block b_1 is applied even if only block b_2 loses balance. In other words, there is a case in which only a fictitious force to block b_1 can sustain block b_2. Another possibility is that the fictitious forces to both blocks b_1 and b_2 are applied. For both possibilities, the result of fictitious forces does not indicate the cause of collapse. It is necessary to determine that the cause of collapse originates from block b_2, so that the AR system can indicate the prediction for block b_2.

Problematically, in the third case, the proposed algorithm cannot indicate block b_2 by applying fictitious force to block b_2 even though block b_2 loses balance. This is because the proposed algorithm optimizes the force for sustaining the stack. This is the same reason why the algorithm is not able to distinguish the collapse of both blocks from the collapse of block b_1. If block b_2 is unbalanced, even if a support at block b_2 can stabilize the stack, a support at block b_1 is prioritized over the support at block b_2 as a result of fictitious forces. That is, the force required by the support at block b_1 is smaller than the force required for block b_2.

Therefore, the algorithm should recognize all possible blocks by modifying the minimum fictitious forces assumption. A prioritization method of collapse block indication is required for precise collapse detection to achieve annotation for *which* block is the cause.

Proposed algorithm improved by moments

The basic idea of our improvement, which is achieved by applying fictitious force on LP, was explained in the top of section 4.3. The proposed algorithm remains inadequate to precisely predict *which* blocks will start to collapse. In the case of the above objective function, an optimal solution indicates the blocks that can be sustained by the minimum fictitious forces. Therefore, when a collapse is prevented by supporting blocks other than blocks that will start collapse, the algorithm may detect unexpected blocks by mistake. Suppose that, in Figure 15, block b_2 starts fall and block b_1 does not; we can prevent block b_2 from collapsing by supporting either block b_2 or block b_1. The more a block is distanced from a supporting point, the smaller fictitious force is required to make torque zero. Hence, the algorithm shows that the block that we should support is block b_1, but the algorithm cannot detect that block b_2 will start to roll.

Thus, the authors take into account the moments of fictitious forces. The objective function is improved by involving the distances between the fictitious forces and the supporting points and coefficients are provided to fictitious forces on the objective function. All coefficients are decided so that the following inequality is satisfied with each combination of a one-on-one block b and a one-on-one block b' higher than block b:

$$cx' < c'x, \tag{3}$$

11

where c and c' are coefficients of fictitious forces corresponding to blocks b and b', respectively, and x and x' are distances along the direction of the x-axis from the supporting point of rotation of block b to edges where the fictitious forces of blocks b and b' are provided, respectively. For instance, the case of Figure 15 is to be considered. When fictitious forces f'_{b1} and f'_{b2} have the same coefficient (i.e.,1) on the objective function, f'_{b1} has an advantage over f'_{b2} to stop the rotation of block b_2. In other words, f'_{b1} requires smaller force than f'_{b2} for stabilization. However, f'_{b2} must have an advantage over f'_{b1} to stop the rotation of block b_2 in order to detect the rotation of block b_2. Let x_1 and x_2 be the distances along the direction of the x-axis from the left edge of block b_3 to the left edges of blocks b_1 and b_2, respectively (see Figure 16). If x_2 and x_1 are set as coefficients of f'_{b1} and f'_{b2} in the objective function, respectively, the magnitudes of f'_{b1} and f'_{b2} to stop the rotation of the block b_2 are the same. Hence, coefficient c_{b1} of f'_{b1} and coefficient c_{b2} of f'_{b2} must be set in order to realize the following inequality:

$$c_{b2}x_2 < c_{b1}x_1. \tag{4}$$

By setting the appropriate coefficients to every fictitious force to realize inequalities on the objective function, the proposed prediction algorithm can calculate which blocks will start to collapse.

5. Results

The authors evaluated the effectiveness of superimposed AR annotations which the proposed algorithm generates. The authors attempted to check whether the annotations are useful for users to understand physical phenomena of piled block collapse, and whether users can learn how to pile blocks efficiently on the overhang game.

As mentioned above, a quadratic program requires huge computational resources. The appropriate number of blocks required to play the overhang problem game on the employed screen size of 65.0 cm by 40.0 cm is over 60 at least. Conventional research on quadratic programs has proved that it takes over 0.3 seconds at $n = 25$, executing on a mainframe. If the quadratic program employs $n = 50$, because computational complexity is theoretically linear n to the fourth power, theory suggests that it takes over 5 seconds. On the other hand, LP executed on a laptop computer (CPU: Intel Core i7 2.4 GHz, Memory: 8 GB) requires 4.7 msec and 79.5 msec at $n = 25$ and $n = 50$, respectively, as the mean of 100 actual measurement values. LP computational complexity is known to increase with n, but total computational complexity depends on the number of repeated calculations implemented on the employed LP solver. The simulation engine employs the Lindo API 5.0 provided by Lindo Systems Inc. to solve the calculations. The Lindo API implements a simplex method for solving LP. As the actual performance depends on the solver performance, the relation of performance between the cases of $n = 25$ and $n = 50$ cannot be discussed solely on the basis of theories.

The actual measurement value of the throughput time, including both the image processing and simulation, is 52.25 msec and 148.1 msec at $n = 25$ and $n = 50$, respectively, as a mean value of

100 measurement times. The system response time, which includes the visual presentation of the annotations, is approximately 248.1 msec at maximum because the presentation thread has an interval time to check the simulation result every 100 msec. Therefore, we have demonstrated that the game system achieves real-time interaction. Moreover, the response time can be reduced by shortening the interval to less than 100 msec because the presentation thread interval only depends on the disk access performance.

The theoretical maximum score of the overhang problem at $n = 30$ is 271 points (Paterson and Zwick, 2009). Even though the maximum score at $n = 60$ is not given in the paper, the authors assume that the maximum score on the employed screen is 350 points at $n = 60$. Thus, over 60 blocks are prepared.

5.1. Experiment

5.1.1. Purpose

The proposed dynamic annotation AR implemented as a block piling game was evaluated by an experiment and an exhibition. For the proof of the concept, the experiment was carried out to measure improvement of the phenomenon understanding. The performance of the subjects were measured by non-annotation, annotation with that whether the pile is balanced, and annotation with that which block is unbalanced. The experiment was planned to reveal the difference of the understanding between annotation levels. For the evaluation of the system, the growth curve on several game trials and the comparison between the difference of annotations were adopted. Finally, an exhibition was performed to get feedback from players in play freely.

5.1.2. Methods

In the experiment, we prepared three types of conditions. Under the first condition, all blocks are always illuminated blue. Under the second condition, all blocks are illuminated blue while the stack is stable; all blocks are illuminated red when the stack is unstable. Conversely, instability is only annotated. In the third condition, the proposed annotation is applied; only blocks which will start to collapse are illuminated red and the others are illuminated blue. In other words, not only the instability of the stack but also which block will cause collapse is given. The three conditions are referred to as the blue condition, the red condition, and the proposed condition, respectively. Subjects play under one of the three conditions.

The process of the experiment is as follows: (i) First, the subjects play the overhang game for two minutes. (ii) Next, they solve three problems using the systems. (iii) Finally, they again play the overhang game for two minutes. We prepared three problems to evaluate the understanding of the subjects and to facilitate their learning. Next, the second trial is performed and we measure the performance ratio of the first and second trials. The first and second problems are "consider how to

13

stabilize a given stack using other blocks with the minimum number of blocks." Figures 17 and 18 depict the given stacks in the first and second problem, respectively. Note that, in these problems, the subjects are not allowed to touch the blocks for trial but just answer the problem after considering it in their mind. Under the blue condition, all blocks, including the red blocks in Figures 17 and 18, are illuminated blue. Under the red condition, all the blocks are illuminated red. Under the proposed condition, only the red blocks in Figures 17 and 18 are illuminated red and the others are illuminated blue. Figures 19 and 20 depict the solutions of the first and second problems, respectively. We need at least two blocks and one block to stabilize the unstable stacks in the first and second problems, respectively. The third problem is "score at least 100 points using four blocks within a minute."

We separate thirty subjects into three groups of ten at random. The subjects are randomly chosen men and women ranging in age from 20 years old to 60 years old, all of whom were university students and teachers who had not played the developed system before the experiment. The subjects in each group performed under one of the three conditions. The group performing under the blue condition, the group performing under the red condition, and the group performing under the proposed condition are referred to as the blue group, the red group, and the annotation group respectively.

5.1.3. Results

Table 1 summarizes the results of the experiment. The column 'Excess' of the first and second problems shows the number of subjects whose answer stabilized the stacks of the problems but who use more than the minimum number of blocks. The p-values for F-tests regarding the performance ratio of the first and second trials among the three groups are 0.352 (between the blue and the red groups), 0.497 (between the red and the annotation groups), and 0.114 (between the blue and the annotation groups), which are larger than the conventional significance level of 0.05. Therefore, we can assume the homogeneity of the variances among the three groups and use the t-test for equal variance. The p-values for t-tests regarding the ratio of the first and second trials among the three groups are 0.196 (between the blue and the red groups), 0.457 (between the red and the annotation groups), and 0.063 (between the blue and the annotation groups). Since all of the p-values are larger than the significance levels of 0.05, the p-values are not statistically significant. However, only the last p-value is less than the significance level of 0.1. Therefore, it can be concluded that the result indicates that the proposed annotation is marginally significant for learning.

5.1.4. Discussion

The results of the first and second problems suggest that the annotation system helps users understand the physical phenomena of piled block collapse. Regarding the third problem, the number of subjects who answered correctly decreases in the following order: red group, annotation group, blue group. It seems difficult to evaluate the difference between the three groups because the third problem is an advanced problem that aims to make the subjects consider it deeply.

However, for the third problem, we observed some subjects in the red and annotation groups who noticed slide blocks using annotations. In addition, they seemed to use annotations effectively.

The results of the ratios of the first and second trials suggest that users have an important opportunity to learn how to pile blocks efficiently using annotations. In particular, annotations suggesting which blocks will start to fall tend to be effective. From the viewpoint of the ratio, a group that has initially high score is disadvantageous on the growth curve. For instance, the result in Table 1 includes two subjects who have the highest score 191 and 185 on the first trial in the blue group. It suggests that the blue group is initially disadvantaged on the ratio because the initial average score is relatively high. Now, tentatively, these two subjects are dropped from the statistics. Providing *F*-tests and *t*-tests after the two subjects are excluded, the average score of the first trial for the blue group is 110.4, which makes it closer to that of the other groups. The *p*-value of the *t*-test between the blue and annotation groups is 0.094, which indicates a trend of significance. Note that the other *p*-values, between the blue and the red groups as well as the red and the annotation groups, are larger than the significance level of 0.1. Therefore, it is concluded that the efficiency of the method is proved even when the disadvantage of the blue group is corrected.

5.2. Exhibitions

The overhang game itself is very simple but requires deep thinking. The number of players playing the game is not restricted by game play rules. Collaborative play makes the game much more interesting. In some cases, the prediction result illustrated at the top left is helpful in collaborative play. This game, based on the overhang problem, has the goal of extending the right end of the stack as far as possible from the right edge of the stacking surface (see Figure 4). Grading of the stack allows players to receive feedback on their performance. This makes players more concentrated on the game. Moreover, the authors set a time limit of play to three minutes. If more time is given to allow further consideration, the mean score easily exceeds 200. The time limit makes the players rush, which makes the game more interesting.

The authors exhibited the AR-based block piling game system "Bricklayer" in several contests. The game was also demonstrated on a television program. In September 2008, a preliminary version of "Bricklayer" was exhibited and awarded a special jury prize at the Sixteenth International Collegiate Virtual Reality Contest in Japan (IVRC 2008)[1]. In April 2009, the newest version of "Bricklayer," which had an improved AR interface, was exhibited at the Eleventh Virtual Reality International Conference (VRIC '09) - Laval virtual in France (Okamoto et al. 2009). Finally, the game system was demonstrated on the NHK (Japanese national television) program Digital Stadium and received an award[2] in July 2009.

---

[1] IVRC official web site: http://ivrc.net/. Last accessed 26 July 2012

[2] "Bricklayers" on YouTube: http://www.youtube.com/watch?v=5j5CUpFkjRo, http://www.youtube.com/watch?v=xxHUjTw7Sh8. Last accessed 26 July 2012.

5.2.1. Players' comments

Feedback from several players was received in the above exhibitions. There were two types of comments, the first relates to the game, and the second relates to how to achieve the best score. For instance, the major comments are as follows:

✓ "How much is the game? I would use this for my child's education," said a mother.

✓ "I notice that the blocks must maintain a certain distance from one another to achieve the best score," said a player.

✓ "I cannot imagine how it works; to me, that is more interesting than the game itself," said a researcher.

✓ "An excellent part of this game is that the system simulates physical phenomena in real-time," said a principal of a computer game school.

✓ "Multitasking is very effective in the prevention of dementia. This game requires you to use your fingers while requiring you to think at the same time. It could work for training and rehabilitation," said an orthopedist.

Nearly all of the feedback is positive. Some people try the game several times. This suggests that the overhang problem itself stimulates intellectual curiosity. It can be said that the game enhances one's understanding of the overhang problem. The second comment in particular indicates deep thinking about the problem. Most people are convinced of the idea of the prediction of physical phenomena. A valuable comment from the viewpoint of a medical professional suggests that the game could have applications in rehabilitation for patients with dementia. In addition, there were no negative comments regarding the system response. Therefore, it is concluded that the developed system can annotate based on physics-based prediction in real-time. The system can also extend players' interest in physics.

6. Discussion

From a system setup point of view, several exhibitions proved that the system is sufficiently durable for a long-run demonstration. Recognition of the blocks is one of the most critical parts of the whole system. Even though the transparency of the frosted glass board was carefully chosen, recognition fails sometimes when a player wearing extremely thick clothing stands too close to the screen and casts a shadow on the screen. Also, external light sources prevent accurate recognition (e.g., light position, brightness, or high light). Registration of the camera view and projector requires precise adjustment. If optical axes are on skew lines, the block projection size and the real block size are eminently incongruous at all positions on the screen. Therefore, the positions of both the camera and the projector need to be fixed strictly before the start of the simulation. The restart time depends on the time required for module positioning. If the system employs a bigger screen resolution such as XGA (1024×768) instead of VGA (640×480), the accuracy of block recognition is significantly improved even though the FPS rate is reduced. Because the proposed system requires blocks to contact the screen, the effect of friction to collapse prediction has to be considered.

16

After careful installation, the system requires players to just pile blocks during play. It is a very natural interface for everyone. Also, players can receive annotation feedback in real time so they can focus on the overhang problem. The result shows that the proposed system achieved a dynamic annotation AR environment without negating immersion.

From the viewpoint of game play, because the system provides a trial environment for piling, the player can learn whether the current action will be successful or not before an actual collapse happens. This aids players in avoiding collapse. Beyond annotation, suggestions on how a player should place the next block is a future development step.

There are two aspects to this system. The first is that players can focus on a physics problem through the game. The second is that players can enjoy applied physics. It is concluded that the proposed dynamic AR promotes an understanding of abstract physics through a real-world application.

7. Conclusion

The authors focus on a traditional physics problem, the overhang problem, to make an AR application. The authors developed a new real-time algorithm to predict blocks that are unbalanced on the overhang problem. An algorithm that is based on a linear program is implemented for simulation of the collapse prediction. Also, an AR-based block piling game system that annotates the result of the prediction based on physical simulation by superimposing color illumination on the actual blocks was developed. Players pile blocks two-dimensionally along a vertical screen. The system immediately indicates which blocks will cause a collapse of the stack. The authors evaluated the proposed annotation system by an experiment that provides the comparison between different annotation levels. The game was also demonstrated at several exhibitions. It showed that the system achieved physics-based real-time simulation for the prediction of actual physical phenomena before it happens, thus aiding the player in the game. The game system also promotes the players' understanding of physical theory by enabling trial and error in a simulated environment. It is concluded that the proposed dynamic annotation AR provides an environment accelerating physical phenomenon understandings.

References

Bender J, Schmitt A (2006) Fast Dynamic Simulation of Multi-body Systems Using Impulses. Proc. of the 3rd Virtual Reality Interactions and Physical Simulations (VRIPhys 2006)

Shinar T, Schroeder C, Fedkiw R (2008) Two-way coupling of rigid and deformable bodies. Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2008): 95-103

Paterson M, Zwick U (2006) Overhang. Proc of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms: 231-240

Paterson M, Zwick U (2009) Overhang. Amer. Math. Monthly 116(1): 19-44

Rosato A, Strandburg KJ, Prinz F, Swendsen RH (1987) Why the Brazil Nuts are on Top-size Segregation of Particle Matter by Shaking. Phys. Rev. Lett. 58: 1038-1040

Shinbrot T (2004) Granular Materials: The Brazil Nut Effect – in Reverse. Nature 429: 352-353. doi: 10.1038/429352b

Röte G, Zwick U (2011) Collapse. Proc of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms: 603-613

Phear JB (1850) Elementary Mechanics. MacMillan, Cambridge

Walton W (1855) A Collection of Problems in Illustration of the Principles of Theoretical Mechanics. Deighton, Bell, and Do. Cambridge

Ainley S (1979) Finely Balanced. Mathematical Gazzette 63: 272

Monteiro RDC, Adler I (1989) Interior Path Following Primal-dual Algorithms. Part II: Convex Quadratic Programming. Math. Programming 44: 43-66

Okamoto K, Tanaka Y, Terasawa N, Tokunaga T, Tsukasa T, Kume N (2009) Bricklayer. Proc. of VRIC'09: 375-380

Carsten Matysczok, Rafael Radkowski, Jan Berssenbruegge (2004) AR-bowling: immersive and realistic game play in real environments using augmented reality. In Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '04). ACM, New York, NY, USA, 269-276. DOI=10.1145/1067343.1067379

Hannes Kaufmann, Bernd Meyer (2008) Simulating educational physical experiments in augmented reality. In ACM SIGGRAPH ASIA 2008 educators programme, Article 3; 8 pages; DOI=10.1145/1507713.1507717

Tony Jebara , Cyrus Eyster , Josh Weaver , Thad Starner , Alex Pentland, Stochasticks (1997) Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers. Proceedings of the 1st IEEE International Symposium on Wearable Computers:138

Chihhsiong Shih, Pao-Ann Hsiung, Chieh-Hao Wan, Chorng-Shiuh Koong, Tang-Kun Liu, Yuanfan Yang, Chu-Hsing Lin, William Cheng-Chung Chu (2009) Integration of a vision-based tracking platform, visual instruction, and error analysis models for an efficient billiard training system. Opt. Eng. 48, 027202; doi:10.1117/1.3083379

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

19

$$\frac{15-4\sqrt{2}}{8} \simeq 1.16789$$
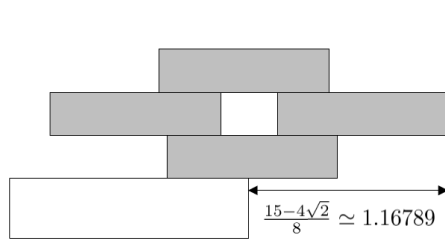
Figure 1: The optimal stack with four blocks (from Ainley 1979).

blocks = 20
overhang = 2.32014
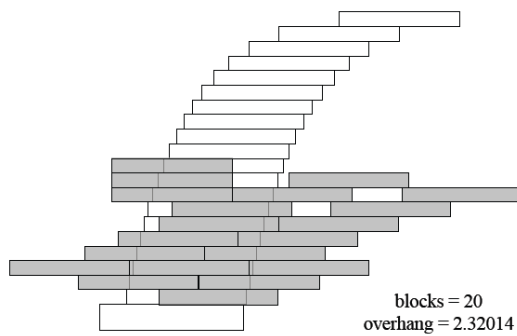
Figure 2: Optimal stack using 20 blocks with the corresponding harmonic stack in the background (from Paterson and Zwick 2006; Paterson and Zwick 2009).
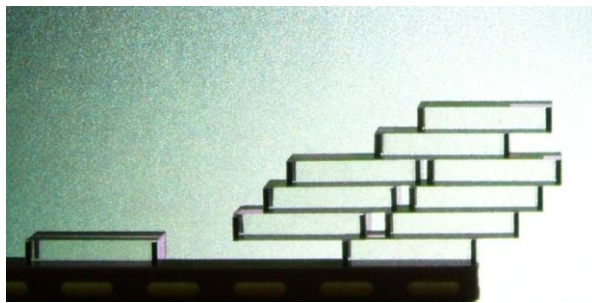
Figure 3: An actual stack of transparent acrylic blocks.
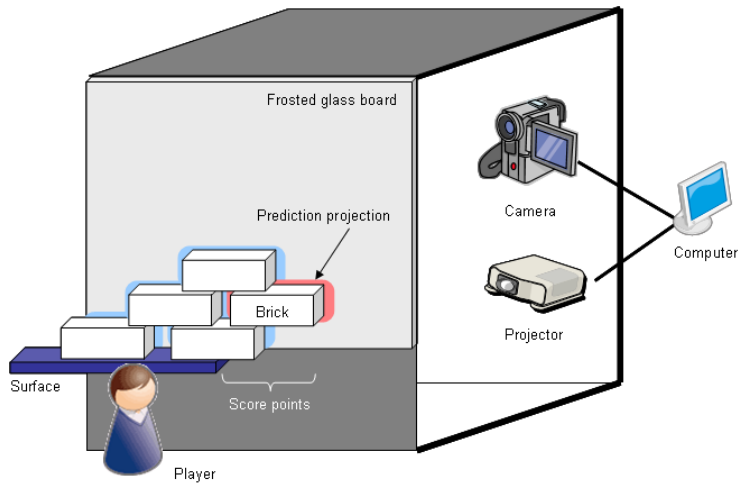
Figure 4: System structure.
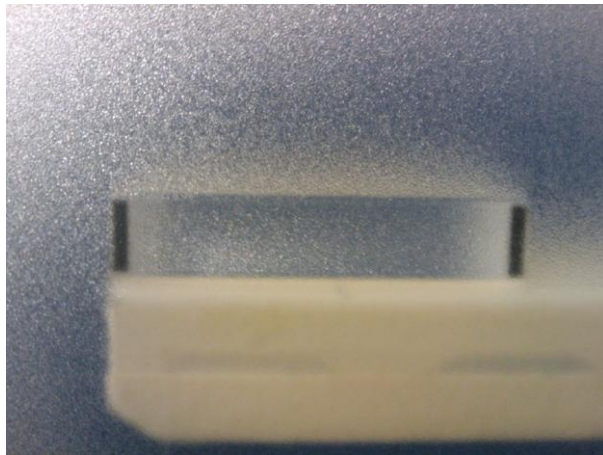
Figure 5: A picture of a block close to a frosted glass board from a camera view.
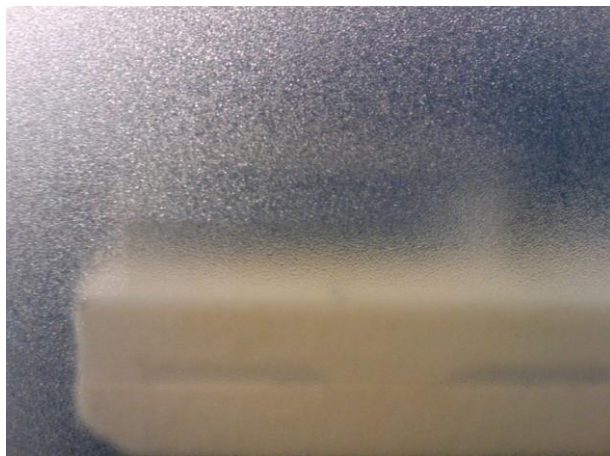
Figure 6: A picture of a block 1 cm away from a frosted glass board from a camera view.

21

Figure 7: An image of green value for image processing ($n = 6$).
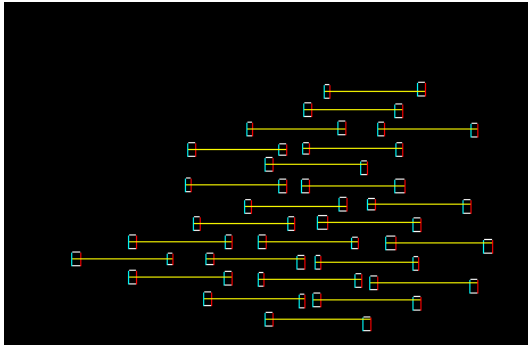
Figure 8: An image of recognized block lines by green value scan on image processing ($n = 25$).

Figure 9: A scene from playing "Bricklayer"
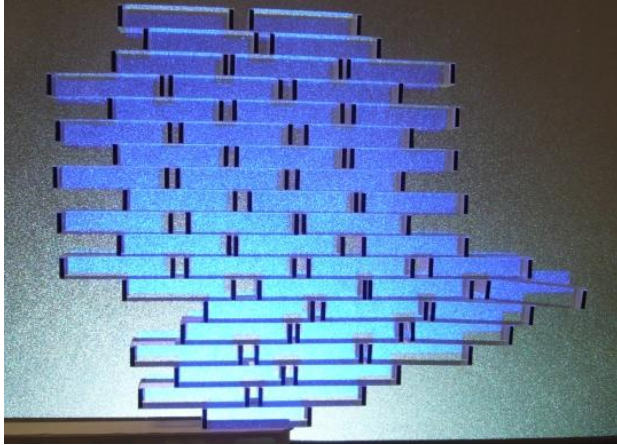
Figure 10: A case of overhang marking 285 points. (All blocks illuminated blue are stable.)
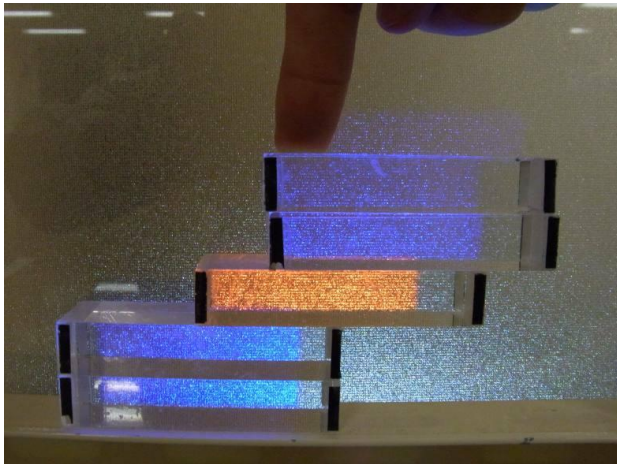
Figure 11: A case of collapse prediction. If the player's finger that sustains the stack as a kind of fictitious force is missing, the block illuminated red will start to fall.
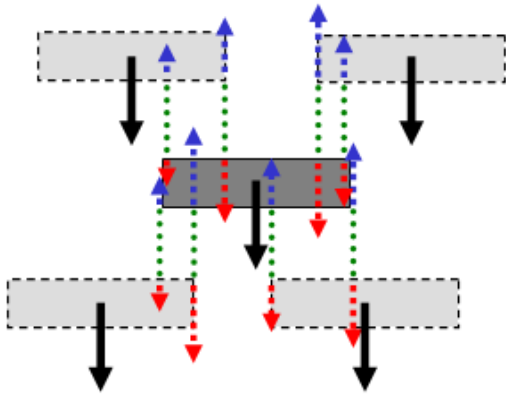
Figure 12: The forces applied to a block.
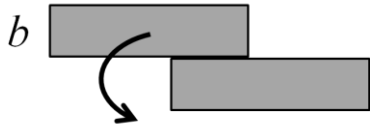
Figure 13: Unbalanced one-on-one block b.
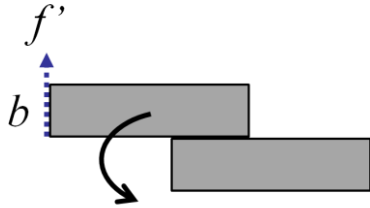
Figure 14: Upward fictitious force f' applied to block b.
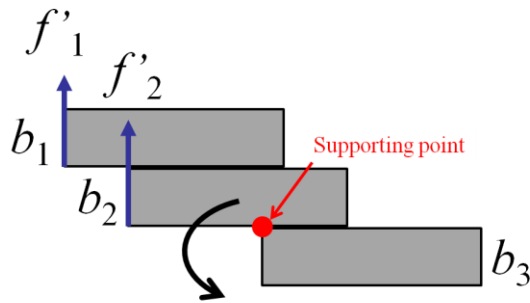
Figure 15: A case where block b'' tumbles.
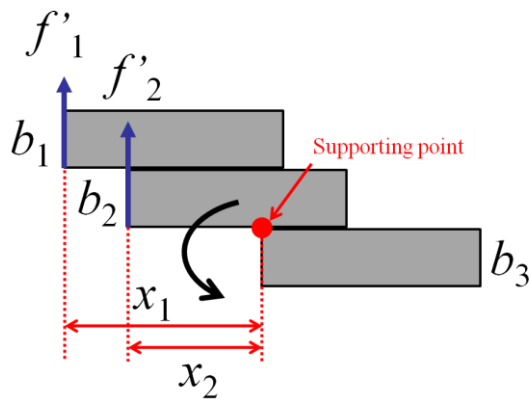
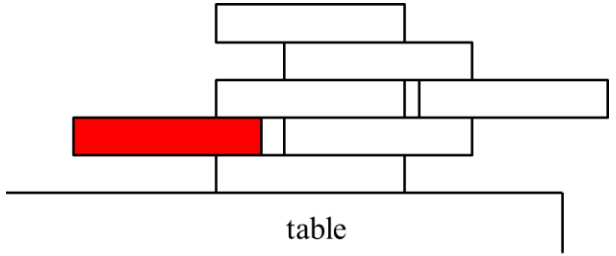Figure 16: Distances of fictitious forces from a supporting point.
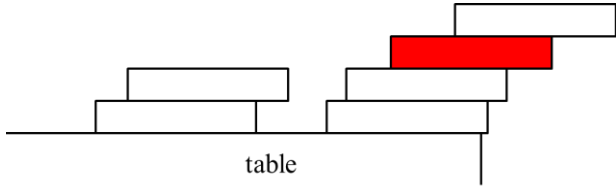
Figure 17: The given stack of the first problem in the experiment. The red block will start to collapse.

Figure 18: The given stack of the first problem in the experiment. The red block will start to collapse.
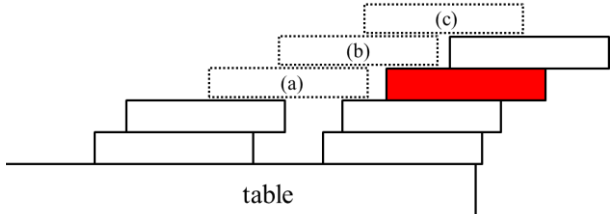

Figure 19: The solutions of the first problem in the experiment. The solutions are putting blocks (a) and (b) or putting blocks (b) and (c).
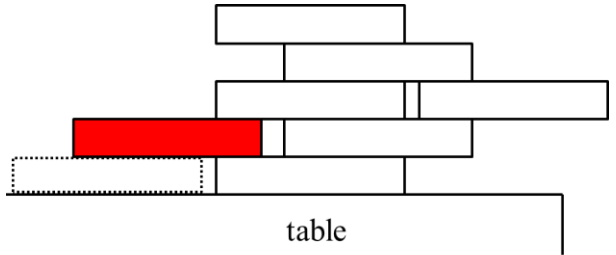

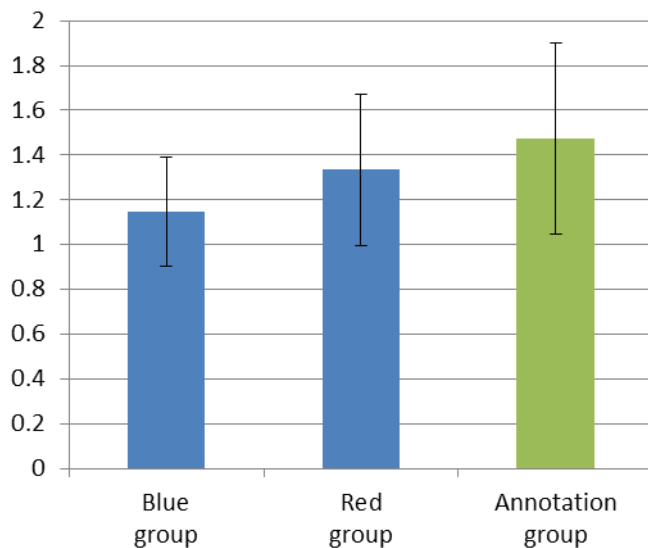Figure 20: The solution of the first problem in the experiment. The solution is placing the dotted block.

Figure 21: The average and the standard deviation of the score growth rate (between two trials in each group)
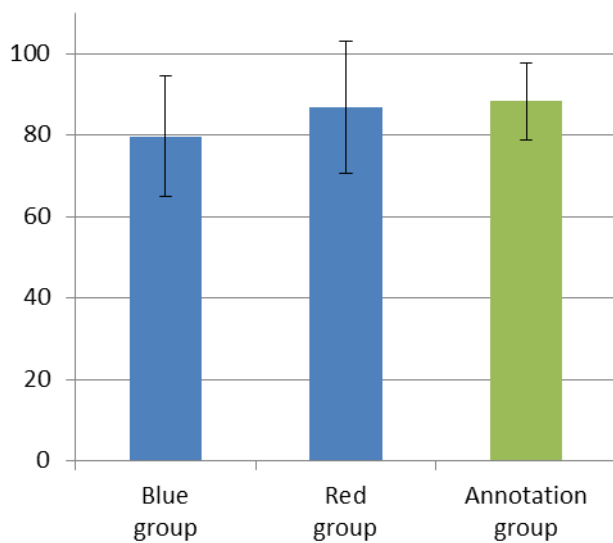

Figure 22: The average and the standard deviation of the score on the problem #3 at the moment 60 seconds passed. The subject who solves the problem within 60 seconds is given 100 points.

Table 1: Experimental results

| | Average score of the 1st trial (points) | 1st problem | | 2nd problem | | 3rd problem | Average score of the 2nd trial (points) | Average of the ratio of the 1st and 2nd trials | Variance of the ratio of the 1st and 2nd trials |
		Correct	Excess	Correct	Excess	Correct			
Blue group	125.9	4	3	1	5	2	141.7	1.149	0.0599
Red group	98.0	2	2	1	6	5	127.0	1.335	0.1129
Annotation group	98.0	6	1	8	1	3	136.6	1.473	0.1818

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65