Qualitative Spatial Representation and Reasoning in Angry Bird: the Extended Rectangle Algebra

Peng Zhang and **Jochen Renz**
Research School of Computer Science
The Australian National University
Canberra, Australia
{u4969566, jochen.renz}@anu.edu.au

Abstract

Angry Birds is a popular video game where the task is to kill pigs protected by a structure composed of different building blocks that observe the laws of physics. The structure can be destroyed by shooting the angry birds at it. The fewer birds we use and the more blocks we destroy, the higher the score. One approach to solve the game is by analyzing the structure and identifying its strength and weaknesses. This can then be used to decide where to hit the structure with the birds.

In this paper we use a qualitative spatial reasoning approach for this task. We develop a novel qualitative spatial calculus for representing and analyzing the structure. Our calculus allows us to express and evaluate structural properties and rules, and to infer for each building block which of these properties and rules are satisfied. We use this to compute a heuristic value for each block that corresponds to how useful it is to hit that block. We evaluate our approach by comparing the consequences of the suggested shot with other possible shots.

Introduction

Qualitative spatial representation and reasoning has numerous applications in Artificial Intelligence including robot planning and navigation, interpreting visual inputs and understanding natural language (Cohn and Renz 2008). In recent years, plenty of formalisms for reasoning about space were proposed (Rajagopalan 1994; Liu 1998; Renz and Ligozat 2005). An emblematic example is the RCC8 algebra proposed by Randell *et al*. (1992). It represents topological relations between regions such as ”x is disconnected from y”; however, it is unable to represent direction information such as ”x is on the right of y” (Balbiani, Condotta, and del Cerro 1999). The Rectangle Algebra(RA) (Mukerjee and Joe 1990; Balbiani, Condotta, and del Cerro 1999), which is an extension of the Interval Algebra(IA) (Allen 1983), can express orientation relations and at the same time represent topological relations, but only for rectangles. When we want to reason about multiple aspects of relations between regions, a possible method is to combine several formalisms. For example, when we want to reason about topology and direction relations of regions with arbitrary shapes, we can combine RCC8 and RA. It has been shown that the problem of deciding consistency of a joint basic network of RCC8 and RA constraints is still in polynomial time (Liu, Li, and Renz 2009). However, if we only consider the maximum bounding rectangles(MBR) of regions, RA is expressive enough to represent both direction and topological information.

RA is designed for reasoning about rectangular objects in 2-dimensional space whose sides are parallel to the axes of some orthogonal basis. However, when we consider a 2-D structure of such objects under the influence of gravity, we need to be able to represent information about the stability of the structure. Ideally, we want a representation that allows us to infer whether the structure will remain stable or whether some parts will move under the influence of the gravity or some other forces (e.g. the structure is hit by external objects). Additionally, if the structure is regarded as unstable, we want to be able to infer the consequences of the instability, i.e., what is the impact of movements of the unstable parts of the structure.

The Rectangle Algebra is not expressive enough to reason about the stability or consequences of instability of a structure. For example, in Fig. 1(a) and (b), assume the density of the objects is the same. The RA relation between object 1 and object 2 in these two figures are both (start inverse, meet inverse), but obviously the structure in Fig. 1(a) is stable whereas object 1 in (b) will fall. In order to make such distinctions, we need to extend the granularity of RA and introduce new relations that enable us to represent these differences. In this paper, we introduce an *extended Interval Algebra* (EIA) which contains 27 relations instead of the original 13. We use the new algebra as a basis for an *extended Rectangle Algebra* (ERA), which is obtained in the same way as the original RA. Depending on the needs of an application, we may not need to extend RA to 27 relations in each dimension. Sometimes we only need the extended relations in one axis. Thus, the extended RA may include 13×27, 27×13 or 27×27 relations depending on the requirement of different tasks.

We built an agent based on this model which aims to play the Angry Birds game automatically and rationally. The result shows that the agent based on this model is able to interpret low-level information from the scene or video input as higher level semantic descriptions (Fernyhough, Cohn, and

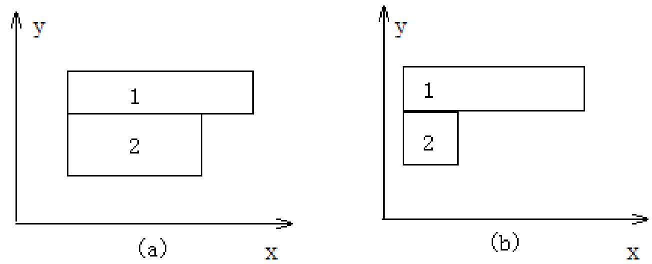

Figure 1: Two configurations with the same RA relation (si,mi)

Hogg 1999).

Interval Algebra and Rectangle Algebra

Allen's Interval Algebra defines a set $\mathcal{B}_{int}$ of 13 basic relations between two intervals (see Fig.2). It is an illustrative model for temporal reasoning. Denote the set of all relations of IA as the power set $2^{\mathcal{B}_{int}}$ of the basic relation set $\mathcal{B}_{int}$. The composition ($\circ$) between basic relations in IA is illustrated in the transitivity table in Allen [1983]. The composition between relations in IA is defined as $R \circ S = \cup\{A \circ B : A \in R, B \in S\}$.

Relation	Illustration	Interpretation
X b Y Y a X	—X— —Y—	X takes place before Y
X m Y Y mi X	—X— —Y—	X meets Y(*i* stands for *inverse*)
X o Y Y oi X	—X— —Y—	X overlaps with Y
X s Y Y si X	—X— —Y—	X starts Y
X d Y Y di X	—X— —Y—	X during Y
X f Y Y fi X	—X— —Y—	X finishes Y
X = Y	—X— —Y—	X is equal to Y

Figure 2: The 13 basic relations of the Interval Algebra

RA is an extension of IA for reasoning about the 2-dimensional space. The basic objects in RA are rectangles whose sides are parallel to the axes of some orthogonal basis in 2-dimensional Euclidean space. The basic relations of RA can be denoted as $\mathcal{B}_{rec} = \{(A, B)|A, B \in \mathcal{B}_{int}\}$ The relations in RA are defined as the power set of $\mathcal{B}_{rec}$ The composition between basic RA relations is defined as $(A, B) \circ (C, D) = (A \circ C) \times (B \circ D)$.

The Extended Rectangle Algebra (ERA)

In order to express the stability of a structure and reason about the consequences of the instability in a situation which observes physical rules, we extend the basic relations of IA from 13 relations to 27 relations denoted as $\mathcal{B}_{eint}$ (see Fig.3).

Relation	Illustration	Interpretation
X b Y Y a X	—X— —Y—	X takes place before Y
X m Y Y mi X	—X— —Y—	X meets Y (*i* stands for *inverse*)
X mom Y Y momi X	—X— —Y—	most part of X overlaps with most part of Y
X lol Y Y loli X	—X— —Y—	less part of X overlaps with less part of Y
X mol Y Y moli X	—X— —Y—	most part of X overlaps with less part of Y
X lom Y Y lomi X	—X— —Y—	less part of X overlaps with most part of Y
X ms Y Y msi X	—X— —Y—	X starts Y and cover most part of Y
X ls Y Y lsi X	—X— —Y—	X starts Y and cover less part of Y
X ld Y Y ldi X	—X— —Y—	X during left part of Y
X rd Y Y rdi X	—X— —Y—	X during right part of Y
X cd Y Y cdi X	—X— —Y—	X during Y and the midperpendicular of Y through X
X mf Y Y mfi X	—X— —Y—	X finishes Y and cover most part of Y
X lf Y Y lfi X	—X— —Y—	X finishes Y and cover less part of Y
X eq Y	—X— —Y—	X is equal to Y

Figure 3: 27 basic relations $\mathcal{B}_{eint}$ for extended IA

Definition 1 (The extended IA relations)**.** *We introduce the centre point of an interval as a new significant point in addition to the the start and end points. For an inteval* a*, denote centre point, start point and end point as* c_a*,* s_a *and* e_a*, respectively.*
1. The 'during' relation has been extended to 'left during', 'centre during' and 'right during' (ld, cd & rd).

- *"x ld y" or "y ldi x" :* $s_x > s_y, e_x \leq c_y$
- *"x cd y" or "y cdi :* $s_x > s_y, s_x < c_y, e_x > c_y, e_x < e_y$
- *"x rd y" or "y rdi x" :* $s_x \geq c_y, e_x < e_y$

2. The 'overlap' relation has been extended to 'most overlap most', 'most overlap less', 'less overlap most' and 'less overlap less'(mom, mol, lom &lol).

- *"x mom y" or "y momi x" :* $s_x < s_y, c_x \geq s_y, e_x \geq c_y, e_x < e_y$
- *"x mol y" or "y moli x" :* $s_x < s_y, c_x \geq s_y, e_x < c_y$
- *"x lom y" or "y lomi x" :* $c_x < s_y, e_x \geq c_y, e_x < e_y$

- *"x lol y" or "y loli x" :* $c_x < s_y, e_x > s_y, e_x < c_y$

3. The 'start' relation has been extended to 'most start' and 'less start' (ms & ls).

- *"x ms y" or "y msi x" :* $s_x = s_y, e_x \geq c_y$
- *"x ls y" or "y lsi x" :* $s_x = s_y, e_x > s_y, e_x < c_y$

4. Similarly, the 'finish' relation has been extended to 'most finish' and 'less finish' (mf & lf).

- *"x mf y" or "y mfi x" :* $s_x > s_y, s_x \leq c_y, e_x = e_y$
- *"x lf y" or "y lfi x" :* $s_x > c_y, s_x < e_y, e_x = e_y$

Denote the set of relations of extended IA as the power set $2^{\mathcal{B}_{eint}}$ of the basic relation set$\mathcal{B}_{eint}$. Denote the set of relations of extended RA as the power set $2^{\mathcal{B}_{erec}}$ of the basic relation set$\mathcal{B}_{erec}$.

Note that EIA can be expressed in terms of INDU relations (Pujari, Kumari, and Sattar 2000) if we split each interval x into two intervals x_1 and x_2 that meet and have equal duration. However, this would make representation of spatial information very awkward and unintuitive. There is also some similarity with Ligozat's general intervals (Ligozat 1991) where intervals are divided into zones. However, the zone division does not consider the centre point.

Application of extended RA in Angry Birds

Representation of rectangular objects in MBRs

A MBR can only sketchy describe the area of a rectangle. Thus, when representing a rectangular object using MBR, some important spatial information of the object, such as direction and size, will lose. Ge and Renz (2013) demonstrate a set of unary relations of rectangles in MBRs which concern the contact points between the object and its MBR. Generally, the rectangular objects are distinguished as two main classes, namely *regular rectangle* which means the rectangle exactly fits its MBR and *angular rectangle* which means the object is different from its MBR. Then angular rectangles are further divided into several subclasses based on different subdivision of the areas of contact points.

This method is useful to extract qualitative direction and size information, and fortunately, we can use ERA to express the very basic case in the unary relations described above. Specifically, each edge of the MBR is separated into two intervals using the centre point suggested in ERA. Despite the *regular rectangle* case, the angular rectangles fall into 9 classes (see Fig. 4). It is obvious that for an *angular rectangle*, two contact points on adjacent edges of its MBR can determine its class. Here we can arbitrarily pick two orthogonal edges of the MBR and the contact points on them to represent the direction and thickness of the real object. To be specific, we use the ERA relations between the contact points and their corresponding MBR edges to represent real object. The ERA representation of the nine classes and their physical significance are described below:

(a) A relatively thin rectangle that leans to left: The top contact point falls into region *H1* and the right point falls into region *V3*, thus the corresponding ERA relation for this configuration is (ld,ld). The physical features of this class of objects are : this kind of objects seek for a support force from on its left side or a relatively large pressure on its right side to remain stationary (see Fig. 5 (a) and (b)). The reason is that without external forces, this object has a overall moment in the direction of anti-clockwise, both the two kinds of forces mentioned above can give it a clockwise moment to balance out the moment in opposite direction, thereby making the object possible to remain stable. Notably, the case in Fig. 5(b) barely appears in the Angry Birds world; and even it appears, the stability can be easily destroyed. Thus we omit this case in the following analysis.

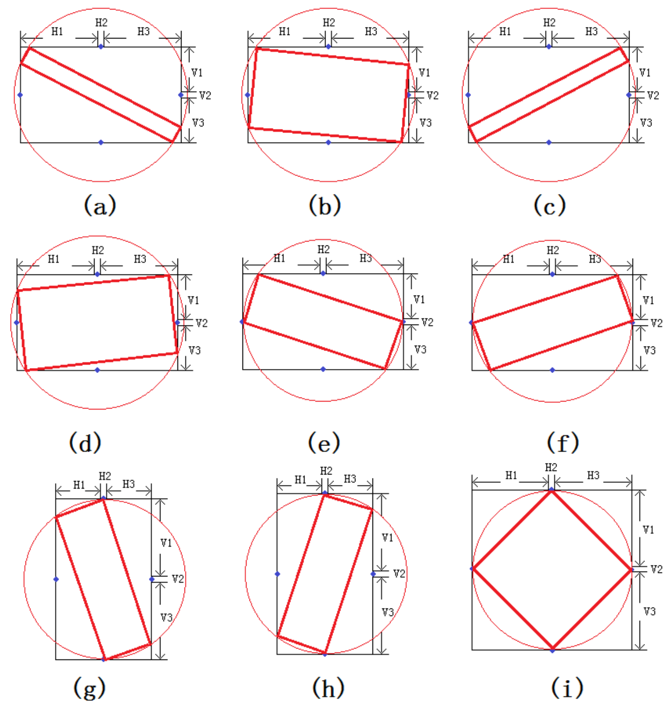

Figure 4: 9 classes of rectangles in MBRs

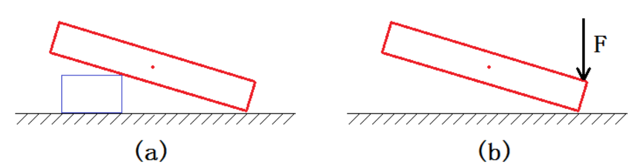

Figure 5: 2 methods to keep an object in Fig. 4(a) stable

(b) A relatively thick rectangle that leans to left: the ERA relation for this configuration is (ld, rd). The physical features are similar with (a), the difference is that this case describes a relatively thick object compared with other cases.

(c) A relatively thin rectangle that leans to right: the ERA relation for this configuration is (rd, ld). This is the opposite case against (a).

(d) A relatively thick rectangle that leans to right: the

ERA relation for this configuration is (rd, rd). This is the opposite case against (b).

(e) A medium rectangle that leans to left: the ERA relation for this configuration is (ld, cd). The physical features are similar with (a), the difference is that this case describes an object with medium thickness compared with other cases.

(f) A medium rectangle that leans to right: the ERA relation for this configuration is (rd, cd). This is the opposite case against (e).

(g) This case seemingly represent a medium rectangle that can stand by itself. However, both in the Angry Birds world and the real world, this configuration rarely holds. A reasonable reasoning about this case is that it has support on both side if it remains stationary. The ERA relation for this case if (cd, rd).

(h) The ERA relation for this case if (cd, ld). The physical analysis for this case is the same as above.

(i) This is a special case of (g) and (h), it represents a square object. The ERA relation for this case if (cd, cd).

Rules based on the extended RA relations for analysing the structure

Before presenting the rules, we first introduce different contact configurations between rectangular objects.

Definition 2 (Contact Relations)**.** *Ideally, two rectangles can contact in 3 ways:*

- *For rectangles R_1 and R_2, if one corner of R_1 contacts with one edge of R_2, then R_1 "point to surface" contacts with R_2. Denote as $CR(R_1, R_2) = ps$*
- *For rectangles R_1 and R_2, if one edge of R_1 contacts with one edge of R_2, then R_1 "surface to surface" contacts with R_2. Denote as $CR(R_1, R_2) = ss$*
- *For rectangles R_1 and R_2, if one corner of R_1 contacts with one corner of R_2, then R_1 "point to point" contacts with R_2. Denote as $CR(R_1, R_2) = pp$*
- *For rectangles R_1 and R_2, if R_1 does not touch R_2, then R_1 "has no contact" with R_2. Denote as $CR_{R_1,R_2} = n$*

Definition 3 (Contact Dimension)**.** *Contact dimension expresses in which dimension(horizontal or vertical in this case) two rectangles contact with each other, specifically, if $CR(R_1, R_2) \in \{ps, ss, pp\}$:*

- *For rectangles R_1 and R_2, if $ERA(R_1, R_2) \in m, mi \times \mathcal{U}_{EIA}$, then R_1 "horizontally" contacts with R_2. Denote as $CD(R_1, R_2) = hc$, where $\mathcal{U}_{EIA}$ is the set of all EIA relations.*
- *For rectangles R_1 and R_2, all other contacts are vertical contact. Denote as $CD(R_1, R_2) = vc$*

Definition 4 (Support)**.** *An simple interpretation of "Object R_1 supports R_2" is R_1 gives R_2 an upward force. The vertical support relations can also be extracted from vision input, where "vertical support" requires $CD(R_1, R_2) = v$. Denote "R_1 vertically supports R_2" as $Supp(R_1, R_2) = vs$.*

With ERA and contact relations, it is possible to build a set of rules to determine some properties of a structure such as stability of a simple structure or consequences after some external influences act on the structure. Then, integrating all the proposed rules, we are able to do some further inferences to predict the consequences of a shot and calculate a a heuristic value. This value will suggest which object is a proper target to hit to maximize the damage. Notably, the "point to point" contact method scarcely appears in the game, thus it is not considered in the following analysis. Also, when mentioning the ERA relation between two contacting rectangles R_1 and R_2, $ERA(R_1, R_2)$ in the dimension other than contact dimension refers to EIA between the contact point/surface and the object that is supported. For example, in Fig. 6, $ERA(R_1, R_2) = (rd, ls)$, not (mom,ls), then from the ERA relation, we know that R_2 does not have sufficient support and it will either rotates or slides. However the original relation will show that R_2 is stable. Thus by using ERA regarding contact points/surfaces, we can clearly decide if the contact provides sufficient support rather than being confused with the original ERA relation.

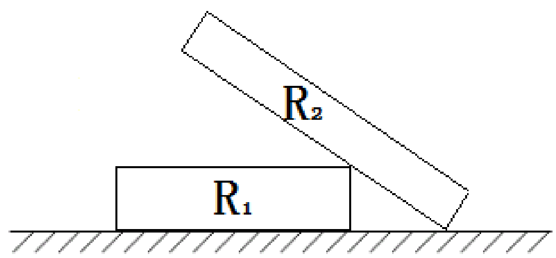

Figure 6: ERA between two contacting objects

Rule 1. Rules for determining stability

We will now specify rules that determine for each target object whether it is stable. First of all, we extract all contact points(surfaces) between each two touched objects from computer vision input. To determine the stability of an object, a useful physical fact is that if the vertical projection of the centre of mass of an object falls into the area of support base, it will not topple. Based on this law, if we do not consider the impacts of the supportees of an object, in the following situations an object will remain stable.

Rule1.1

The target object is a regular rectangle which just lie on the ground => object is stable

Rule1.2

For a *regular rectangle* x $\in O$(O is the set of all objects in the structure), $\exists$ y, z$\in O$:

$Supp(x, y) \in \{vs\}$

$\wedge\ Supp(x, z) \in \{vs\}$

$\wedge\ ERA_x(x, y) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}$

$\wedge\ ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}$

=> x is stable

This rule illustrates that if the vertical projection of the mass centre of an object falls into the support region between two supporters, it is stable.

Rule1.3
For a *regular rectangle* x, $\exists y$:
$Supp(x, y) \in \{vs\}$
$\wedge ERA_x(x, y) \in \{ms, mf, msi, ls, mfi, lf, cd, cdi, ld, rd, mom, momi, lomi, mol\}$
$=>$ x is stable

This rule illustrates that if the vertical projection of the mass centre of an object falls into the region of a single supporter, it is stable.

Rule1.4
For a *regular rectangle* x $\in O(O$ is the set of all objects in the structure), $\exists$ y, z$\in O$:
$CD(x, y) \in \{hc\} \wedge Supp(x, z) \in \{vs\}$
$\wedge ((ERA_x(x, y) \in \{mi\}$
$\wedge ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\})$
$\vee (ERA_x(x, y) \in \{m\}$
$\wedge ERA_x(x, z) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}))$
$=>$ x is stable

This rule shows an object is stable if it is horizontally contacting with another object on one side and has a supporter on the other side. Fig. 7 shows examples of the four stable configurations for a regular rectangle.

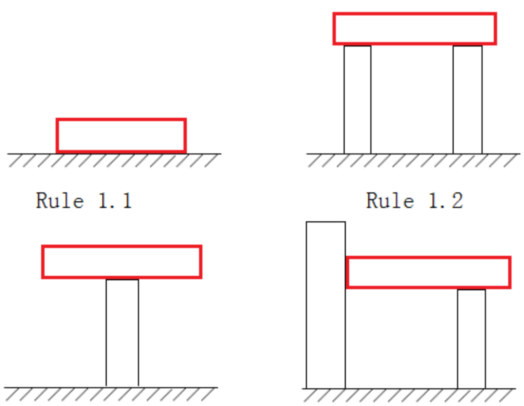

Figure 7: Examples for stable configurations of regular rectangles

The stability test for angular rectangles are more complex. Nevertheless the basic idea is still "vertical projection of centre of mass falls into support area". The rules for determining the stability of an angular object is shown below.

Rule1.5
For a *angular rectangle* x $\in O(O$ is the set of all objects in the structure), $\exists$ y, z$\in O$:
$Supp(x, y) \in \{vs\}$
$\wedge Supp(x, z) \in \{vs\}$
$\wedge ERA_x(x, y) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}$
$\wedge ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\}$

The principle of this rule is similar as Rule 1.2 which tests if the centre of mass falls into the region of the two vertical supporters.

Rule1.6
For a *angular rectangle* x $\in O(O$ is the set of all objects in the structure), $\exists y$:
$Supp(x, y) \in \{vs\}$
$\wedge CR(R_1, R_2) = ss$
$\wedge ERA_x(x, y) \in \{ms, mf, msi, ls, mfi, lf, cd, cdi, ld, rd, mom, momi, lomi, mol\}$
$=>$ x is stable

This rule is similar with Rule 1.3, the difference is that it requires the two objects to be surface-to-surface contact, otherwise the angular rectangle will either topple. Remarkably, the supporter in this case also needs sufficient support to remain stationary.

Rule1.7
For a *angular rectangle* x $\in O(O$ is the set of all objects in the structure), $\exists$ y, z$\in O$:
$CD(x, y) \in \{hc\} \wedge Supp(x, z) \in \{vs\}$
$\wedge ((ERA_x(x, y) \in \{mi\}$
$\wedge ERA_x(x, z) \in \{mom, mol, lom, lol, mfi, lfi, rdi\})$
$\vee (ERA_x(x, y) \in \{m\}$
$\wedge ERA_x(x, z) \in \{momi, moli, lomi, loli, msi, lsi, ldi\}))$
$=>$ x is stable

This is a similar rule with Rule 1.4. And this rule describes horizontally supporting relation for angular rectangle.

Rule1.8
For a *angular rectangle* x $\in O(O$ is the set of all objects in the structure), $\exists$ y, z$\in O$:
$CD(x, y) \in \{hc\} \wedge CD(x, z) \in \{hc\}$
$\wedge ERA_x(x, y) \in \{m\}$
$\wedge ERA_x(x, z) \in \{mi\}$
This rule describes a different property of angular rectangle, which is an angular rectangle can remain stable with the support of two horizontally contacting objects on both left and right sides. This is because the two objects on the sides can provide upward frictions to support the angular rectangle.

Fig. 8 shows the above four configurations that can keep angular rectangles stable.

Rule 2. Rules for detecting support and sheltering structures
The entire structure in Angry Birds game is often large and even in some levels all the objects in the world are constructed into only one structure. As can be found in most levels, many pigs are set on support structures sometimes with multi-level supporters. Then a good idea to kill the pig (if not directly reachable) is to destroy the support structure

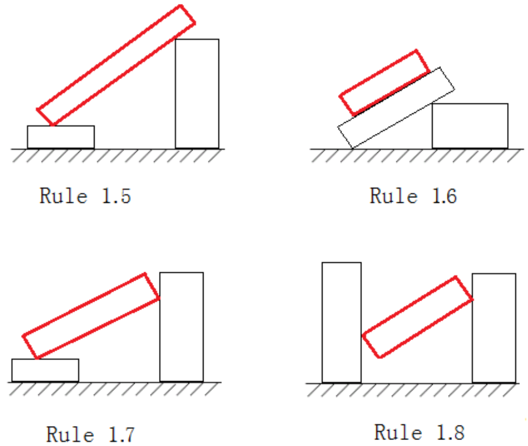

Figure 8: Examples for stable configurations of angular rectangles

and the pig will probably die. Another useful substructure is the shelter of the pigs. The reason is straightforward, if a pig is not reachable, there must be some objects that protect it; these objects form the sheltering structure of the pigs. Similarly, destroying the sheltering structures can either kill the pig or make the pig directly reachable to the bird.

Specifically, in order to separate the support structure of a pig from the larger structure, it is necessary to include the depth information of the supporters(see Fig. 9 the illustration of support structure with depth). This is helpful when only considering the most essential supporters or only several layers of supporters are required.

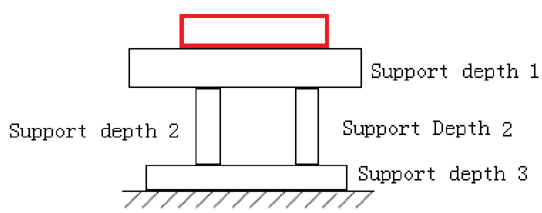

Figure 9: Illustration of support structure

As mentioned in Rule 1, the vertical supporters can be extracted from vision input. And the horizontal supporters can be obtained using rule 1.4, 1.7 and 1.8. Then we can further get the supporters of the supporters in order to collect all direct or indirect supporters of a certain object.

Similarly, a sheltering structure consists of the closest protection objects of the pig that can avoid the pig from a directly hit from each direction including the hit from backward. Specifically, a sheltering structure of a pig could consist of left, right and roof sheltering objects. In order to get the sheltering structure of a certain object (usually a pig), the first step is to get the closest object from the left side of the queried object; then, get the supportee list of the object (similar process as getting the supporter list); after that, get the right closest object with its supportee list. The next step is to check if the two supportee lists have objects in common, if so, pick the one with smallest depth as the roof object of the sheltering structure; if not, there is no sheltering structure for the queried object. If a roof object is found, also put the supportees of both the left and right closest objects with smaller depth than the roof object into the sheltering structure. Finally, put the supporters of both left and right closest objects which are not below the queried object into the sheltering structure.

The rules expressed in extended RA relations for determining sheltering objects consists of three parts (These set of rules can also be expressed in original RA):

Rule 2.1 The rules for getting potential left and right sheltering objects(take left side as an example)
For an object x $\in O$, denote S_l as the set of potential left sheltering objects of x.
$\forall y \in O$,
$RA(x, y) \in \{b, d, di, o, m, fi\}$
$\times \{d, di, o, oi, s, si, f, fi, eq\}$
=> put y into S_l

Rule 2.2 The rules for choosing closest sheltering objects
$\forall y, z \in S_l$,
$RA(y, z) \in \{b, d, o, m, s\} \times \{A, A \in U_{IA}\}$
=> delete y from S_l, otherwise delete z
Finally, the closest objects will remain.

The integration of the rules to evaluate a shot

With the rules described above, we are able to dynamically analyse the possible consequences after a shot has been made. In order to predict the final consequence of an external influence on the structure, the direct consequence and its following subsequences should be analysed in detail. Funt suggested a similar method to simulate the consequence of a structure with a changed object which assumes that the changed object disappears and chooses the most significant unstable object to simulate the consequence (Funt 1987). We separately analyse *regular rectangular* and *angular rectangle*, they can each be affected in four configurations.

First we consider the objects which are regular rectangles. From configuration 1 to configuration 4, the target is assumed to be a regular rectangle.
Configuration 1 The target object in the structure is hit directly by another object. The direct consequence will be in three types which are destroyed, toppling and remaining stationary. Empirically, the way to determine the consequence of the hit depends on the height and width ratio of the target. For example, if an object hits a target with the height and width ratio larger than a certain number (such as 2), the target will fall down. And this ratio can be changed to determine the conservative degree of the system. In other words, if the ratio is high, the system tend to be conservative

because many hits will be determined as no influence on the target. Moreover, if the external object hits a target with the height and width ratio less than one, the target itself will remain stable temporarily because the system should also evaluate its supporter to determine the final status of the target. In some situations, we may also be concerned with the destruction of the target, such as in the Angry Birds game. After deciding the direct consequence of the hit, the system should be able to suggest further consequences of the status change of the direct target. Specifically, if the target is destroyed, only its supportees will be affected. If the target falls down, the configuration will be more complex because it may influence its supporters due to the friction, supportees and neighbours. If the target remains stable temporarily, it will also influence its supporters and its supporters may again affect it from the further simulation.

Configuration 2 The supportee of the target object topples down. Similar to the process that set the height and width ratio to determine the stability of an object, this target object's stability is also represented by the ratio but the number should be larger (about 5) because the influence from supportee is much weaker than it from direct hit. If the target is considered as unstable, it will fall down and affect is neighbours and supporters; otherwise, it will only influence its supporters (see Fig. 10).

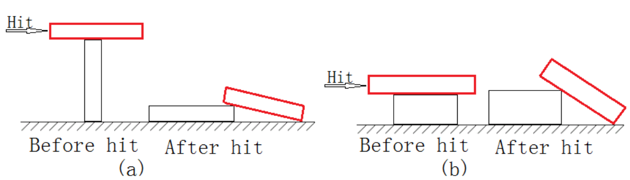

Figure 10: Configuration 2

Configuration 3 The supporter of the target object topples down. Here a structure stability check process (applying Rule 1.1 - 1.4) is necessary because after a supporter falls, the target may have some other supporters and if the projection of its mass centre falls into the areas of the other supporters, it also can stay stable. Then, if the target remains stable, it again will only affect its supporters due to the friction; otherwise, it may fall and affect its supporters, supportees and neighbours (see Fig. 11(a)).

Configuration 4 The supporter of the target is destroyed. This is more like a sub configuration of the previous one. If the target cannot remain stable after its supporter destroyed, it may fall and affect its supporters, supportees and neighbours (see Fig. 11(b)).

Then we consider the objects which are angular rectangles. From configuration 5 to configuration 8, the target is assumed to be an angular rectangle.

Configuration 5 The target object is hit directly. As analysed in Section *Representation of rectangular objects in MBRs*, angular rectangles can be classified into 9 classes. If we only consider the direction of the objects, there only exists 3 classes, which are *"lean to left"*,*lean to right* and

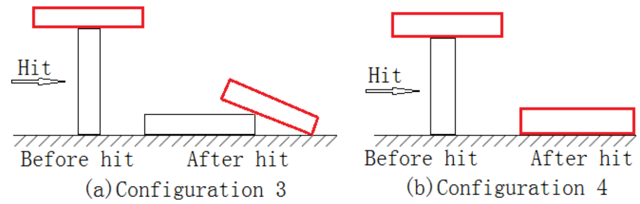

Figure 11: Configuration 3&4

stand neutrally. Assume the hit is from left to right. Then if a *"lean-to-left"* object is hit, ideally if the force is strong enough, the object will rotate around the lowest pivot, i.e. the lowest corner of the object. However, in the Angry Birds game, before the force becomes sufficient to make the object rotate over, the object will be destroyed first. Thus the two possibilities here are either the target object affects its supporters or it is destroyed and affected its supportees. The case of *"lean-to-right"* object is the same as above. For a *"stand-neutrally"* object, it cannot stand by itself and at least one *point-to-surface* touched supporter on each side is necessary. If it is hit from left, there will be no friction applied to its left supporter, thus it will either be destroyed or affect its right supporter.

Configuration 6 The supportee of the target object topples down. Angular objects will not topple due to the fall of their supportees. Because their supporters restrict their spatial form. And therefore the supporters will be affected. Specially, for *"stand-neutrally"* objects, if the friction given by their supportees are from left to right, the left supporters will not be affected.

Configuration 7 The supporter of the target object topples down. First check the stability using Rule 1.5 - 1.8. If the object still has sufficient support, it only affect its supporters, otherwise it will rotate and affects its supportees, other supporters and neighbours.

Configuration 8 The supporter of the target object destroyed. Similarly, a stability check is applied first. However if it remain stable, it will not affect its supporters because no friction applying to the target from the destroyed supporter. If it is not stable, than it will affects its supportees, other supporters and neighbours.

Calculation of the heuristic value

Then, with all the affected objects in a list, the quality of the shot can be evaluated by calculating a total score of the affected objects. The scoring method is defined as: if an object belongs to the support structure or the sheltering structure of a pig, 1 point will be added to this shot; and if the affected is itself a pig, 10 points will be added to the shot. After assigning scores to shots at the objects, the target with highest score is expected to have the largest influence on the structures containing pigs when it is destroyed. Then, based on different strategies, the agent can choose either to hit the reachable object with highest heuristic value or generate a sequence of shot in order to hit the essential support object of the structure.

Algorithm 1 illustrates the whole process for evaluating a shot at all possible targets.

Algorithm 1 process of evaluating a shot

```
for all Objects o in the structure do
    init ongoing list 'ol' and affected list 'al'
    add o into al
    applying rule 1 and 3 (integrating in the 4
    configurations) to get affected objects 'ao'
    add all ao into ongoing list
    for all ongoing objects 'oo' in ol do
        add oo into al and delete oo from ol
        for all objects ao' affected by oo' do
            if ao' ∉ al then
                add ao into ol
            end if
        end for
        if ol = ∅ then
            break
        end if
    end for
    calculate heuristic value of o
    get stabilityofeachobject
end for
output a list of heuristic values for shots at all
    target objects in descending order with reachability
```

We first extract the ERA relations between all objects and then match the rules for all relevant combinations of objects. Thus the process of evaluating the significance of the targets is straightforward and fast.

Planning in Angry Birds

The previous analysis only looks one step forward. Sometimes the shot with highest score is not reachable due to some blocks in the path; as we have the formula of the trajectory parabola, it is possible to recognise the blocks in the path. Then a planner could be used to generate a sequence of shots to clear the blocks and hit the primary target. Specifically, the planner will first check if the highest-ranked shot is reachable. If not, it will retrieve the block objects between the bird and the target in the trajectory. Then it will search for a shot that can affect most of the objects in the block list. Finally, the planner will suggest using the first shot to clear the blocks if applicable. However, as we cannot exactly simulate the consequence of a shot, sometimes the first shot may lead a even worse situation. Fig.12 shows the process of the planner.

Evaluation

We use levels from the to evaluate our approach. In particular, we first compare the predictions given by our system with the real consequence in order to test the quality of of the analysis. Then we compare the result of the suggested shot with other possible good shots to test if the system is able to find optimal shots. Notably, the agent already have the trajectories of the birds and the properties of the birds in its knowledge base, for example, a yellow bird can hit through 2 layers of wood, a blue bird can go through 2 layers of ice, etc. Then the agent can use this knowledge to reason about the reachability of a bird and roughly estimate the damage of the direct consequence of a shot.

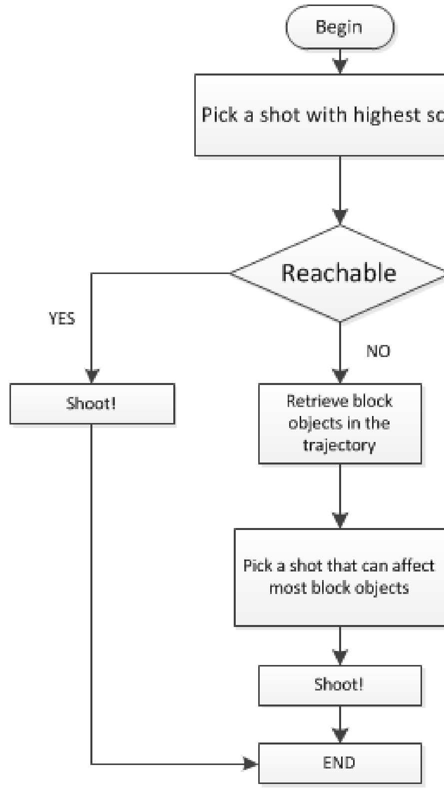

Figure 12: Process of the planner

Here we pick a level with complex structure to illustrate the evaluation process, Fig.13 shows the level and output from the agent. From the output we can find the agent suggests that shoot at object 30 will result in affecting on the left two pigs and the toppling of the left half structure.

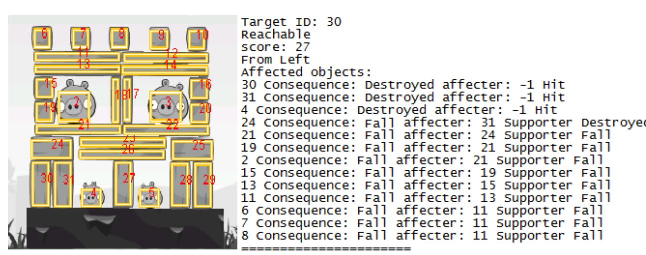

Figure 13: Level 1-16 in Chrome Angry Birds and agent output for this level

Fig.14 is the real configuration after the suggested shot has been made. We can see that the prediction of the system is very accurate in this case. The left two pigs are killed and the left part of the structure toppled. Then we show another two shots that is possible to be good (Fig. 15), which aims at object 19 and 24, respectively. From Fig. 15, we can see that although they are good shots, the suggested shot which killed two pigs is optimal for the first shot. We evaluate our approach in many other levels, for the levels with a complex structure, the agent performs rationally.

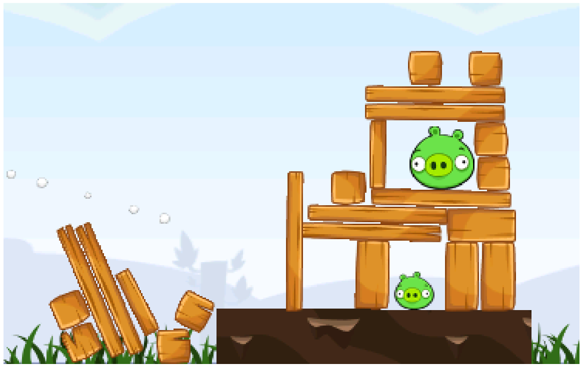

Figure 14: Actual result for the suggested shot

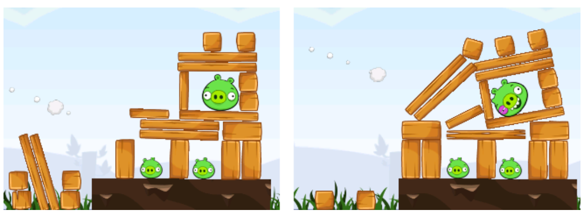

Figure 15: Actual result for another two shots

Related Work

Qualitative spatial representation and reasoning has been applied to some physical systems to do common sense reasoning (Klenk et al. 2005). Physical reasoning has two important theories, namely kinematics and dynamics. Kinematics mainly concerns about the position of objects which may change continuously over time and the shape which is strictly rigid and static over time (Davis 2008). However it considers less about the forces between objects and the type of motions. These features make kinematics easy to be formulated, on the other hand models using kinematics is usually limited by the context and appears to be less expressive. The CLOCK Project (Forbus, Nielsen, and Faltings 1991) is a typical success which uses a variety of techniques but mainly kinematics to represent and reason about the inner motions of a mechanical clock. Under some given environments and constraints, this approach can successfully analyse the mechanisms; however, as this system requires restrict constraints such as the exact shape of the parts and the range of the motion, it may not be applicable in the configurations with high freedom such as the world of Angry Birds.

In contrast, dynamics takes force analysis into concern which allows reasoning about more complex motions and transfer of motions. However, the limitation is that precisely predicting a consequence of a motion in an arbitrary environment is almost impossible due to the uncertainty. However, it is possible reasoning about some simpler physical features of a structure, such as stability. Fahlman (1973) implemented a stability verification system dealing with the stability of blocks with simple shapes (e.g. cuboid). This system can produce qualitative output to determine whether a structure is stable without external forces, but it requires some quantitative inputs (e.g. magnitude of forces and moments). The stability check in our work is partially inspired by this system, however without quantitative inputs about forces, we use a purely qualitative way to analyse the probability of the balance of forces and moments in a structure. Although our approach may produce a few false predictions in very complex configurations, it is a good estimation of humans' reasoning approach. Our work also focuses on reasoning about the impacts of external forces (a hit from an external object) on a structure, which has not been discussed a lot in the previous works.

Discussion and Future Work

Discussion

In this paper we have introduced an extended rectangle algebra useful for representing and reasoning about stability and other properties of 2-dimensional structures. By splitting some basic interval relations into more detailed ones, we obtained 27 interval relations in each dimension that can express the physical relations between rectangular objects more precisely. We used the new algebra for defining some useful structural rules regarding properties such as stability, reachability, support, and shelter. We tested the usefulness of our rules by designing an agent that performs a structural analysis of Angry Birds levels. Based on these rules, we predict for each block the consequences if it gets hit and calculate a heuristic value that determines the usefulness to hit the block. We then shoot at the block with the highest value that is reachable with the current bird.

Future Work

Some methods in this work is still at a preliminary stage. For example, the motion analysis can only casually predict which objects may be affected by a shot , i.e. we do not have the predictions about a possible state in this stage. And the relations are pairwise, which means we do not analyse the interactions between substructures. In the future, we can try to design a feedback loop between the dynamic analysis and ERA, specifically, we can use the results of dynamic analysis to restrict a range of possible subsequent positions of an objects and use ERA to suggest several possible configurations. Then use dynamic analysis again to check the stability of these configurations and finally decide the reasonable consequential states. Moreover, we will find a method to determine useful substructures to increase the reliability of dynamic analysis.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26:832–843.

Balbiani, P.; Condotta, J.; and del Cerro, L. 1999. A new tractable subclass of the rectangle algebra. In *IJCAI-1999*, 442–447.

Cohn, A. G., and Renz, J. 2008. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*, 551–584. Oxford, England: Elsevier.

Davis, E. 2008. Physical reasoning. In *Handbook of Knowledge Representation*, 597–620. Oxford, England: Elsevier.

Fahlman, S. E. 1973. A planning system for robot construction tasks. Technical report, Cambridge, MA, USA.

Fernyhough, J.; Cohn, A. G.; and Hogg, D. C. 1999. Constructing qualitative event models automatically from video input. *Image and Vision Computing* 18:81–103.

Forbus, K. D.; Nielsen, P.; and Faltings, B. 1991. Qualitative spatial reasoning: the clock project. *Artif. Intell.* 51(1-3):417–471.

Funt, B. V. 1987. Problem-solving with diagrammatic representations. In *Readings in computer vision: issues, problems, principles, and paradigms*, 456–470. Morgan Kaufmann.

Ge, X., and Renz, J. 2013. Representation and reasoning about general solid rectangles.

Klenk, M.; Forbus, K.; Tomai, E.; Kim, H.; and Kyckelhahn, B. 2005. Solving everyday physical reasoning problems by analogy using sketches. In *AAAI-2005 Proceedings*, 209–215.

Ligozat, G. 1991. Gerard ligozat: On generalized interval calculi. In *AAAI-1991*, 234–240.

Liu, W.; Li, S.; and Renz, J. 2009. Combining rcc-8 with qualitative direction calculi: Algorithms and complexity. In *IJCAI-2009*, 854–859.

Liu, J. 1998. A method of spatial reasoning based on qualitative trigonometry. *Artificial Intelligence* 98:137–168.

Mukerjee, A., and Joe, G. 1990. A qualitative model for space. In *AAAI-1990 Proceedings*, 721–727. AAAI Press and MIT Press.

Pujari, A. K.; Kumari, G. V.; and Sattar, A. 2000. Indu: An interval duration network. In *In Proceedings of Sixteenth Australian joint conference on AI*, 291–303. SpringerVerlag.

Rajagopalan, R. 1994. A model for integrated qualitative spatial and dynamic reasoning about physical systems. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, 1411–1417.

Randell; Cui, Z.; and Cohn, T. 1992. An interval logic for space based on connection. In *Proceedings of the 10th European Conference on Artificial Intelligence*, 394–398. Wiley.

Renz, J., and Ligozat, G. 2005. Weak composition for qualitative spatial and temporal reasoning. In *Principles and Practice of Constraint ProgrammingCP 2005,11th International Conference*, 534–548.