Tracking Perceptually Indistinguishable Objects using Spatial Reasoning

Xiaoyu Ge and Jochen Renz

Research School of Computer Science
The Australian National University
Canberra, Australia

Abstract. Intelligent agents perceive the world mainly through images captured at different time points. Being able to track objects from one image to another is fundamental for understanding the changes of the world. Tracking becomes challenging when there are multiple perceptually indistinguishable objects (PIOs), i.e., objects that have the same appearance and cannot be visually distinguished. Then it is necessary to reidentify all PIOs whenever a new observation is made. In this paper we consider the case where changes of the world were caused by a single physical event and where matches between PIOs of subsequent observations must be consistent with the effects of the physical event. We present a solution to this problem based on qualitative spatial representation and reasoning. It can improve tracking accuracy significantly by qualitatively predicting possible motions of objects and discarding matches that violate spatial and physical constraints. We evaluate our solution in a real video gaming scenario.

1 Introduction

Image understanding (Sridhar et al. 2011) and object detection (Papageorgiou et al. 1998) are essential methods for extracting useful information from images. Equally essential is object tracking (Yilmaz et al. 2006), the ability to identify the same object in a series of images or in videos and to track its movement and changes. Existing object tracking methods typically rely on the visual appearance of objects and on their trajectories to successfully track objects (Cutler and Davis 2000; Yilmaz et al. 2004). Data association techniques (Cox and Hingorani 1996; Khan et al. 2005) are broadly used for tracking multiple objects. These methods can handle false and missing observations reasonably. However, they usually have high computational complexity.

In this paper we look at the problem of tracking perceptually indistinguishable objects (PIOs) (Santore and Shapiro 2005), i.e., objects in images or videos that have the same appearance and cannot be visually distinguished. We want to be able to identify which PIO at time t_2 is identical to which PIO at time t_1 without continuously monitoring the changes between t_1 and t_2. The observations made at t_1 and t_2 are not continuous if the time gap between the two time points is not negligible (> 50 ms). Under the assumption of discrete observations

we have to be able to re-identify each PIO whenever we obtain a new observation. While all permutations of identity assignments are theoretically possible, the task is to find an assignment that is consistent with a physical event that is responsible for the changes.

Our interest in this problem is motivated by the Angry Birds AI competition (www.aibirds.org) where the task is to build an AI agent that can play the popular game Angry Birds as well as the best human players. A major problem in this context is to accurately predict the outcome of a shot, i.e., to infer how the game objects move when hit by a bird in a particular way. One way of estimating consequences of shots is to know which object after a shot corresponds to which object before a shot. Then we can use this information to learn consequences of actions by using before and after object locations as input to machine learning algorithms. Once we can estimate consequences of shots, it becomes possible to plan good shot sequences that can solve given game levels. Therefore, matching objects after a shot to objects before a shot is an important step in building a sophisticated Angry Birds AI agent.

The main contribution of the paper is the successful application of qualitative spatial reasoning techniques (QSR, see Cohn and Renz 2008 for a survey) to provide good and efficient solutions to a relevant open problem. We developed an algorithm that allows us to infer matches of PIOs that are consistent with the physical effects of a single impact. We evaluated our proposed solution using the Angry Birds scenario. We took subsequent screenshots of an active Angry Birds game with varying time gaps and applied our method to match the objects between successive screenshots. We measure the accuracy of our method by using the percentage of correct matches out of the total number of possible mismatches. As expected, it turns out that the smaller the time gaps, the higher the accuracy of the matches. But overall the quality of our matches is very high.

2 Related Work

There are many intelligent systems using QSR techniques. For example, SOAR (Laird 2008), a cognitive architecture in pursuit of general intelligence, has a QSR component (Wintermute and Laird 2007, 2008) that performs spatial reasoning with bimodal representations. The component incorporates continuous motion via simulation with motion-specific models (e.g. the Falling Block Model). Qualitative physics (Forbus et al. 1991, 2008; Kuipers 1986) uses symbolic computations to model and analyze physical systems. The modeling processes often require information about system dynamics (e.g. force), object properties (e.g. elasticity), and detailed spatial configurations (e.g. contact points). This information is not usually available in our problem domain and is not necessary to solve the problem. Another weakness of qualitative physics methods is that they lack mechanisms to handle occluded objects. The idea of combining logics with QSR is also relevant here (Aiello et al. 2007; Kreutzmann et al. 2013).

There are also extensive studies on qualitative spatio-temporal reasoning (QSTR). In recent years, the community has developed various mechanisims

(Galton 2000; Cabalar and Santos 2011) intended for commonsense reasoning and reasoning about spatial changes and actions. Some mechanisms are used in real-world applications, such as planning (Westphal et al. 2011), cognitive vision (Dubba et al. 2010) and scene analysis (Xu and Petrou 2011). Another related branch is simulation-based reasoning. (Battaglia et al. 2013) proposed a system of physical reasoning using probabilistic simulations. However, simulation-based approaches are not applicable to our problem mainly because of their inability to deal with incomplete information (unknown physical properties) and lack of well-defined domain models. (Davis and Marcus 2013) provides an in-depth look at the limitations of simulation-based approaches.

3 Detection and Representation of Objects in Images

In order to be able to track objects in images, we obviously have to be able to first obtain objects from images by object recognition techniques (Belongie et al. 2002; Lowe 1999). In this paper we assume that objects can be detected and we will use cases where object detection is solved and works. In particular, we use images taken from the Angry Birds game, as this is the main motivation of our work in this paper. The basic software provided by the Angry Birds AI competition organizers includes an object recognition module that detects the exact shape of all known objects with reasonable accuracy (Ge et al. 2014).

We use exact shapes for the general solid rectangles (GSR), i.e. rectangles that can have any angle and are impenetrable, and use minimum bounding rectangles (MBR) to approximate the regions occupied by other shapes. To represent these objects, we use a qualitative spatial representation in addition to the real shape and location of the objects. Many rectangle-based qualitative spatial calculi (Balbiani et al. 1998; Cohn et al. 2012; Sokeh et al. 2013) have been developed in the context of QSR. These calculi typically deal with one or more spatial aspects such as topology, size, or direction, and make a number of qualitative distinctions according to these aspects. There is currently only one spatial calculus that specifically deals with rectangles of arbitrary angles, the GSR-n calculus proposed by (Ge and Renz 2013). It defines eight contact sectors that correspond to the eight edges and corners of the rectangles. As shown in Fig. 1.a, we distinguish eight sectors for regular rectangles and eight sectors for angular rectangles. Given two GSRs o_1 and o_2 that contact via $s_1, s_2 \in \{A_1, ..., A_8, R_1, ..., R_8\}$, the contact relation between o_1 and o_2 can be expressed as the constraint $o_1\,(s_1, s_2)\,o_2$ (Fig. 1.b). With the contact relations, GSR-n allows us to distinguish if and how two objects contact. Since the objects can only physically interact via contacts, we can further infer the possible motions of an object from the GSR relations the object holds with others.

3.1 Objects Representation with the extended GSR-n relations

Given two GSRs, we obtain the contact relation by enumerating all the plausible combinations of the two GSRs' contact sectors and for each combination calculating the distance between the two sectors. The combination with the shortest

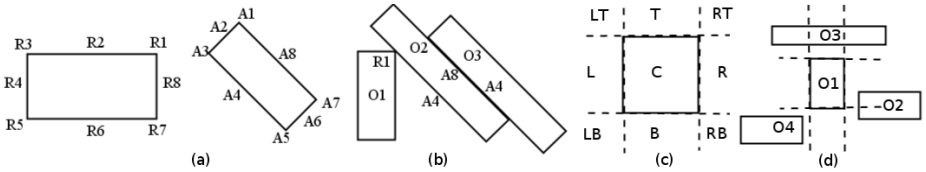

Fig. 1. (a) Contact sectors of a normal rectangle (without rotation) and an angular rectangle. (b) An example scenario where $o_1(R_1, A_4)o_2$, $o_2(A_8, A_4)o_3$. (c) The nine cardinal tiles (d) An example scenario where o_3 *(T)* o_1, o_2 *(R)* o_1, and o_4 *(BL)* o_1

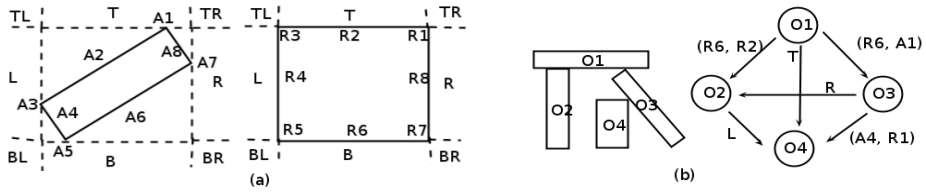

Fig. 2. EGSR Contact sectors and Cardinal directions of an angular rectangle and a normal rectangle. (b) A spatial scenario where the four rectangles form a stable structure under downward gravity and the corresponding QCN

distance constitutes the contact relation. Note, the shortest distance can be non-zero. A non-zero distance means the two GSRs are separate, otherwise touch.

The problem with GSR-n is that it uses $(\emptyset, \emptyset)$ to represent the spatial relation between all non-touching GSRs. Thus, it does not distinguish cases where rectangles are disconnected (not touching). To add this distinction, we extend the original GSR-n by integrating it with the cardinal tiles (Goyal and Egenhofer 1997). We partition the embedding space around a reference object into nine mutually exclusive tiles (Fig. 1.c). The center tile C corresponds to the MBR of the reference object and the other eight tiles correspond to the eight cardinal directions. We call the tiles L, R, B, T the *core tiles*.

The new spatial representation is called Extended-GSR (EGSR) (Fig. 2.a). Given a set $\mathcal{B}_{GSR}$ of GSR contact relations and a set $\mathcal{B}_{card}$ of cardinal tiles, we add $\perp$ to both sets to indicate an unassigned relation. An EGSR relation is then written as $(r_1, r_2), r_1 \in \mathcal{B}_{GSR} \cup \{\perp\}, r_2 \in \mathcal{B}_{card} \cup \{\perp\}$. We abbreviate (r_1, r_2) by the cardinal tile r_2 or by the contact relation r_1 if it is clear which one is meant.

We compute the EGSR relation between two spatial objects by first checking whether their MBRs intersect or boundary touch. If not, one of the eight cardinal tiles will be used; and if one object's MBR occupies multiple tiles of the referred object, we will assign the core tile occupied by the MBR (Fig. 1.d). If their MBRs boundary touch, a GSR-n relation will used. When their MBRs intersect, a GSR-n relation will be assigned if both the objects are GSR, otherwise the center tile will be assigned. All EGSR relations are obviously converse, e.g. the converse of *TL* is *BR*, the converse of *(R_4, S_7)* is *(S_7,R_4)*. A scenario containing multiple spatial objects can be qualitatively interpreted by EGSR via a qualitative constraint network (QCN)(Wallgrün 2010). QCN is a labelled graph where each node corresponds to an object and directed edges represents

relational constraints that have to hold between the two objects. Fig. 2.b shows an example of a QCN based on EGSR relations.

4 Efficient Matching by Approximating Movement

Now that we have obtained the relevant objects in images and their qualitative representation, we formally define the problem we solve in this paper. We call it the *PIO-matching problem with single impact (PIO-1)*:

PIO-1 Given a set O of object types where objects of the same type are PIOs, a set O_{t_1} of objects of given type and their locations at time t_1, and a set O_{t_2} of objects of given type and their locations at later time t_2. We assume that a single physical impact P between t_1 and t_2 caused the changes from O_{t_1} to O_{t_2}. The task is to match objects in O_{t_2} to objects in O_{t1} such that the changes in location of the matched objects is consistent with the consequences of P.

We refer to objects in an initial scene as initial objects and objects in a subsequent scene as subsequent objects. The search space of the problem is large: let a be the number of initial objects and let b be the number of subsequent objects, the number of matches of all objects is $\frac{max(a,b)!}{(max(a,b)-min(a,b))!}$.

However, it is not just the size of the search space that makes this problem hard, but the potential unavailability of the exact physical properties of the objects and the physical impact involved. This is a consequence of the fact that the problem is essentially a visual perception problem that involves processing and "understanding" the information contained in the visual observation, and agents typically do not know the exact physical properties of entities and events they perceive.

The search space can be reduced by searching through corresponding objects only in a limited area that depends on the estimated force of the impact. The search area of each initial object o_i should cover only the objects o_j in the subsequent scene that can be potentially matched to the initial object. We use a circular region to represent this area. The circle's center is located at the centroid of o_i and the radius of the circle is the maximum shift of the centroid. The radius is calculated as $v \times \Delta t$ where v is the maximum estimated velocity of o_1 and Δt is the time gap between the initial and subsequent scene. This calculation ensures that the circle can adapt to different time gaps. We call this circle the *movement bounding circle* (MBC). The relative distance between two objects in O_{t_1} and O_{t_2} can then be allocated to two meaningful classes, namely *reachable* and *non-reachable*. An object $o' \in O_{t_2}$ is reachable by $o \in O_{t_1}$ if the center of the MBR of o' is within the MBC of o, otherwise non-reachable. The MBC can be divided into four quadrants to further restrict the search area. A quadrant of an object is said to be *active* if the object is likely to be in that quadrant at the next time point after an impact, otherwise the quadrant is *inactive*. The search space can be reduced by first searching matching objects in the active quadrants. If there are no matches, then other quadrants will be considered. Given a MBC

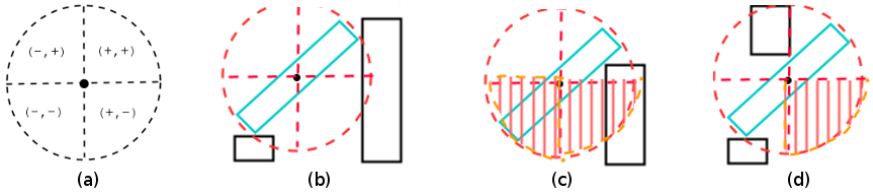

Fig. 3. (a) The four quadrants of a MBC. The active quadrants (shaded area) of (b) a stable object (no active quadrants) (c) an unstable object (d) a right leaning object

C, the active quadrants are one or more of $C^{(i,j)}, i, j \in \{-, +, *\}$ where (+,+), (+,−), (−,−), (−,+) correspond to the right-top, right-bottom, left-bottom, and left-top quadrants, respectively (Fig. 3.a). (*, *) refers to an arbitrary quadrant.

We can infer the active quadrants for an object by approximating the movement direction of the object, i.e. by estimating which of the quadrants the object is most likely to be in at the next time point after an impact. Object movement can be inferred from the direction of the impact. By estimating the direction and force of an impact, one can approximate the subsequent movements of the objects affected by the impact, directly or indirectly. When impact information is not available, we can still approximate the movement by analyzing structural properties, e.g. the stability of an object or a group of objects. An object is stable when it is supported and remains static (Fig. 3.b). The active quadrants of unstable objects is $C^{(*,-)}$ (Fig. 3.c). In the Angry Birds scenario, a stable object may become unstable if it loses a support due to a bird hit. From the bird's trajectory, we can determine which object will be hit by the bird, and approximate the stability of the resulting scenario with the removal of that object.

We can get a more restricted area by analyzing the direction in which the object is falling. For example, a right leaning rectangle will fall to the right if there is no support at the right side, and the corresponding active quadrant is $C^{(+,-)}$ (Fig. 3.d). (Ge and Renz 2013) defined four kinds of supports that can make a GSR stable and provided the corresponding spatial configurations. Here we illustrate how can we express the rules described in the example using EGSR relations. We denote the left leaning and right leaning objects as o^L, o^R respectively. The active quadrants, e.g. $C^{(+,+)}$, of an object o is written as $C_o^{(+,+)}$. The right leaning (RL) and left leaning (LL) rules can be expressed as:

1. RL: $\forall o_1^R : \exists o_2^* : o_1^R(A_5, *)o_2^* \wedge \neg\exists o_3^* : o_1^R(A_6, *)o_3^* \wedge \neg\exists o_4^* : o_1^R(A_7, *)o_4^* \Rightarrow C_{o_1^R}^{(+,-)}$
2. LL: $\forall o_1^L : \exists o_2^* : o_1^L(A_5, *)o_2^* \wedge \neg\exists o_3^* : o_1^L(A_3, *)o_3^* \wedge \neg\exists o_4^* : o_1^L(A_4, *)o_4^* \Rightarrow C_{o_1^L}^{(-,-)}$

5 Handling Common Movement by Spatial Reasoning

A further challenge is to determine a match between PIOs that are close to each other and have similar trajectories, as these are typically all equally reachable.

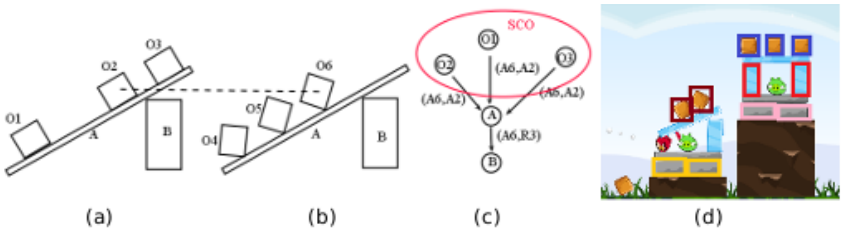

Fig. 4. (a)(b) The initial and subsequent scene (c) The EGSR constraint network of the initial scene (only retain the edges indicating contacts) and the SCO (d) Objects of the same SCO are highlighted by the same color

Fig. 4.a shows a scene where objects A and B form a slope and three indistinguishable squares, o_1, o_2 and o_3, are lying on the slope. Fig. 4.b is a subsequent image where the three squares have rolled down slightly. There are 6 ways in total to match the squares but only $\{o_1 \sim o_4, o_2 \sim o_5, o_3 \sim o_6\}$ makes sense ($\sim$ is an operator that matches one object to another). If we were to find a match by minimizing centroid shift, we would tend to match o_2 with o_6.

Humans can solve this case efficiently using spatial reasoning. Since we know the objects are moving at a similar velocity, the relative spatial changes among them are subtle. Hence the spatial relations between those objects are unlikely to become converse while they are moving. When matching, humans try to keep the original spatial relations among the subsequent objects. We emulate this commonsense reasoning in testing a match by first identifying those objects that are following a similar trajectory and then determining whether any relation has become converse at the next time point.

Objects are likely to follow a common trajectory if they are all in contact with the same other objects and the contact relations are the same. The objects may be influenced in the same way since their interactions are through the contacts with the same other objects. We say that such objects form a *spatially correlated objects set* (SCO). Fig. 4.d shows an example of SCOs in an Angry Birds scenario.

Given a set of initial objects, we obtain the SCOs by checking node equivalence in the corresponding EGSR network. A node is equivalent to another if the two nodes have the same contact relations with other nodes. Thus the slope example has only one SCO: $\{o_1, o_2, o_3\}$ (Fig. 4.c). Having identified a SCO, we then check the spatial relations between the matched objects in the subsequent scene. Formally, let R be a set of EGSR relations. The converse of a relation $r \in R$ is written as $r' \in R$. Given a SCO in the initial scene $O = \{o_1, o_2, ..., o_k\}$ and a set of subsequent objects $O' = \{o'_1, o'_2, ..., o'_k\}$ with a match, $\forall i \leq k, o_i \sim o'_i$, between them, the spatial constraints can be written as $\forall o_i, o_j \in O, \exists r \in R$ such that $o_i(r)o_j \Rightarrow o'_i(r')o'_j$ *does not hold*, for $i, j \leq k$. If a match violates the constraints, we will try all the other possible matches for the SCO until the violation is resolved. If all matches violate the constraints, we keep the original match. In the slope example, the match $\{o_1 \sim o_4, o_3 \sim o_5, o_2 \sim o_6\}$ violates the constraint because $o_2(L)o_3$ and $o_6(R)o_5$ where R is the converse of L.

Algorithm 1 The Object Tracking Algorithm

```
1: procedure MatchObjects
2:     sol ← {}, iniobjs ← PIOs in the initial image
3:     for iniobj ∈ iniobjs do
4:         pobjs←{}, subobjs ← PIOs in the subsequent image
5:         Compute the active quadrants of iniobj
6:         Add obj ∈ subobjs to pobjs if obj is within the quadrants and of the same
   type with iniobj
7:         pmatches←pmatches ∪ {(iniobj, pobjs)}
8:     end for
9:     CreatePreference(iniobjs,pmatches), freeobjs←iniobjs
10:    while freeobjs is not empty do
11:        iniobj ← dequeue(freeobjs)
12:        Get the next preferred obj from iniobj's preference list
13:        if obj is not assigned yet then sol ← sol ∪ {(iniobj,obj)}
14:        else obj has been assigned to some object iniobj′
15:            if obj prefers iniobj to iniobj′ then
16:                sol← sol ∪ {(iniobj, obj)}, freeobj ← freeobj ∪ {iniobj′}
17:                else freeobjs← freeobjs ∪ {iniobj}
18:            end if
19:        end if
20:    end while
21:    Build the QCN on iniobjs, get SCOs by node equivalence
22:    for sco∈SCOs do
23:        Check sco for the violation of spatial constraints, resolve conflicts if any
24:    end for
25: end procedure
```

6 A Method for Tracking PIOs

We propose a method (sketched in Alg.1) for solving PIO-1 that uses all the above mentioned techniques. It first estimates the active quadrants of initial objects according to their spatial relations (Alg.1 line 5). The list of possible matches for each initial object is set so that it contains only the subsequent objects that are of the same type and within the quadrants (Alg.1 line 6). The method then creates a preference list from the possible matches of each of the initial objects: the subsequent objects in the preference list are sorted by the size of the centroid shift from the initial object in ascending order. The method matches using a stable marriage algorithm (Gale and Shapley 1962) with the pre-computed preference lists (Alg.1 line 9–20). The algorithm ensures that no pair of objects would prefer each other over their matched partners. Then, it finds all SCOs from the initial objects and gets their corresponding objects from the match (Alg.1 line 21). The method checks to see whether the spatial constraint has been violated. If it has, it resolves this accordingly (Alg.1 line 23). Our inference rules to estimate stability of objects and to predict their quadrants is application specific, all others can be generalized to other domains, provided that objects do not move independently but are subject to physical forces.

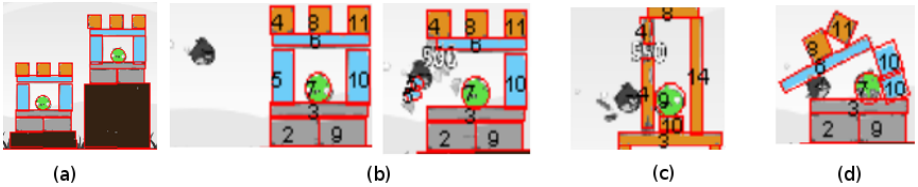

Fig. 5. (a) The vision detects the real shapes of the objects (b) The object with ID 5 is broken into pieces by a hit (c) The object with ID 4 is partially occluded (d) The object with ID 10 is damaged and detected as two separate blocks

7 Implementation

We implemented our method and applied it to Angry Birds where the vision system (Ge et al. 2014) can detect the exact shapes of the objects (Fig. 5.a). The objects' visual appearance are restricted to a finite number of types. However, occlusion or fragmentation are not accounted for in the vision system. It has the following limitations: (1) Debris is not recognized so that it cannot determine whether an object, say a stone, is a real stone or just a piece of debris from a previously destroyed stone (Fig. 5.b). (2) Damaged objects may be detected as several separate smaller pieces (Fig. 5.c). (3) Objects can be occluded by debris or other game effects e.g. scores (Fig. 5.d). All these cases generate object fragments that can severely affect the matching accuracy. There are several techniques for tackling fragmentation and occlusion (Adam et al. 2006; Bose et al. 2007). Most of them largely depend on their underlying tracking algorithms and their own ad-hoc occlusion reasoning models e.g. inference graph, Bayesian network. We present an approach that can effectively deal with this problem in the Angry Birds domain. As a side effect, our approach can also identify which objects have been destroyed.

7.1 Handling Fragmentation and Complete Occlusion

We classify the initial and subsequent objects according to their type. For each type T, there is a set T_{ini} of initial objects and a set T_{sub} of subsequent objects with the same type. We treat all objects in T_{sub} as potential fragments if T_{sub} contains more objects than T_{ini}. Fragments are arranged into groups where all the fragments in a group can form one of the types. The shape formed by the fragments is an oriented minimum bounding rectangle (OMBR) containing all the fragments (Fig. 6.a). We treat the OMBR as one object in the subsequent image, so that it can be matched with an initial object. Once the OMBR is matched, the fragments from the corresponding group are also matched. A fragment is not allowed to be in more than one OMBR.

We label the unmatched fragments as debris. Destruction of an object will create debris around the object's location. The debris can be of any shape and will diffuse until it disappear after 1–3 seconds. Given an object o in the initial scenario, we search for its debris if no subsequent objects can be matched with o

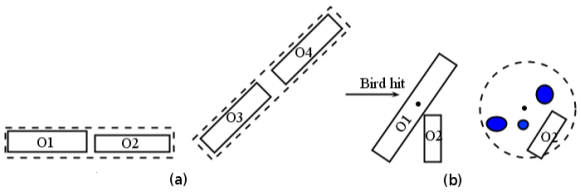

Fig. 6. (a) OMBRs are indicated by the dotted rectangles (b) o_1 has been destroyed by a bird hit, and the blue dots are recognized as debris

(including the OMBRs created from the fragments). We first draw the MBC of o. The set of potential fragments are those fragments within the MBC excluding those that have been matched. The set of objects is labelled as debris of o and o is marked as destroyed (Fig. 6.b). An object can also be completely occluded. Before the matching, we cache the spatial configurations of the initial objects. At the end of the matching, we update the cache by replacing each initial object's configuration with that of the matched subsequent object so that the cache always maintains the latest configurations. If an occluded object recurs in a subsequent image, we match the object by searching through the cache for an unmatched initial object. The occluded object will be matched if it lies in the MBC of that initial object. The method determines an object as destroyed if it detects the debris of the object, or when the object has been occluded for n second(s). n is tunable and we set n to 1 in the evaluation.

8 Evaluation

Matching an object can be trivial if the object has a unique appearance or stays stationary across images. We measure the accuracy of the method by the percentage of correct matches out of the possible mismatches. Given a set n of objects in an initial scenario and assume m of them are either of unique type or stationary across images, we count the correct matches c of the n - m possible mismatches. The accuracy is $\frac{c}{n-m}$. We also show the percentage ($TPercent$) of the number of correctly matched objects out of the total number of objects. The evaluation has been done in two steps. We first collected samples from active angry birds scenarios using the maximum sampling rate (20 screenshots per second) ; for each sample, we obtained the ground truth by manually labelling initial objects and their correspondence in the end screenshot. Then we evaluated the method by varying the time gaps and obtaining the accuracy by comparing against the ground truth.

We collected samples by running an angry-birds agent that always aims at a random pig on poached-eggs levels (chrome.angrybirds.com). The agent starts to capture screenshots once a shot is made, and stops after 10 seconds. For each level, the agent records the screenshots of at most four shots. We obtained 72 non-trivial samples. Each sample contains 50–200 screenshots and around 30 objects. We apply our method to the whole sequence so that the method will

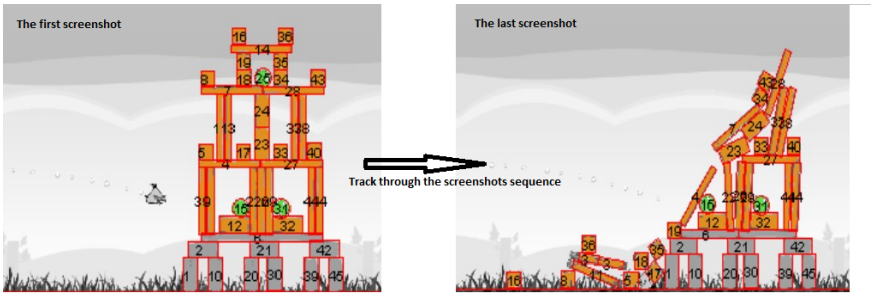

Fig. 7. The method tracks through the images and labels the matched objects with the same ID.

Table 1. Results with different gaps (*QSR*: the proposed method, *BASIC*: the basic optimization strategy). TPercent, Accuracy, Mismatch are average of all the samples

Time gap (ms)	TPercent	Accuracy		Mismatch	
	QSR	QSR	BASIC	QSR	BASIC
50	0.95	0.89	0.57	1.81	6.32
100	0.91	0.83	0.48	2.67	7.41
200	0.87	0.78	0.44	3.41	8.14
300	0.84	0.74	0.42	4.02	8.72
500	0.81	0.68	0.38	4.89	9.12
1000	0.79	0.66	0.36	5.34	9.44

keep tracking the objects through all of the screenshots, from the first until the last (Fig. 7). We determine the accuracy by comparing the matching between the first and the end screenshots with the ground truth. This accuracy approximates the lower-bound accuracy of matching between a pair of screenshots with the specified time gap, because any incorrect matches made in the intermediate stage may yield mismatches between the first and end screenshots.

We evaluate our method with varying time gaps, namely 50 ms (the time taken to request a screenshot), 100 ms, 200 ms (the maximum delay in getting screenshots in the competition), 300 ms (the time taken by requesting a screenshot plus the vision segmentation), 500 ms, and 1000 ms. For a particular time gap, say 200 ms, the method will start from the first screenshot, go through every $200/50 = 4$ screenshots of the original sequence, until the last. To illustrate the significant improvements achieved by the reasoning techniques, we compare our method with a basic optimization strategy (*BASIC*) that matches PIOs by minimizing the centroid shift between initial and subsequent objects, i.e. without spatial reasoning. *BASIC* is a modified version of our method with the movement approximation, and common movement handling disabled. The results are summarized in Table 1. Using the smallest time gap, the method can match most of the objects with less than 2 mismatches per sample. The method achieves real-time performance with 7–10 ms per pair of images for all the time gaps. As expected, the accuracy drops down when applying larger time gaps.

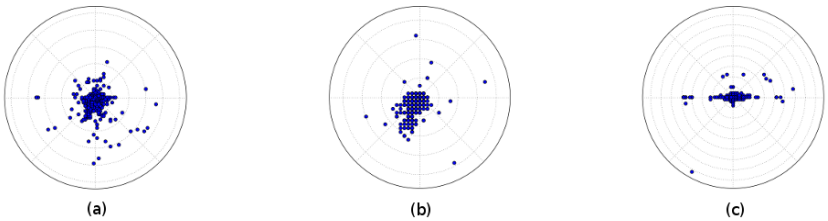

Fig. 8. The SOD of (a) the unstable objects that have no contacts, (b) the objects that have the conact relation (A_5, R_2), (c) the locally stable objects that have the contact relations (R_6, R_2) and (R_2, R_6).

We qualitatively evaluate the rules (see Sect.4) used for predicting active quadrants. We group the initial objects in the samples by their contact relations. For each group, we show the subsequent objects' distribution (SOD) by drawing their centroids in one MBC. Fig. 8.a depicts the SOD of the initial objects that have no contacts. Having no contacts implies the objects are unstable and are most likely free-falling. Therefore most of the subsequent objects appear in $C^{(*,-)}$. Fig. 8.b shows the SOD of a sub-configuration of the left leaning rule. Fig. 8.c shows the SOD of the initial objects that are locally stable. Most of the subsequent objects are close to the center of the MBC. As there are some dots spreading horizontally, it suggests that the locally stable objects are likely to move horizontally instead of vertically.

9 Conclusion and Future Work

We analyzed the problem of tracking PIOs in discrete observations. We developed a method for solving this problem based on a qualitative spatial representation of different object properties. We tested our method in the Angry Birds domain and showed that it is very accurate in identifying which PIOs before a shot correspond to which PIOs after a shot. Our method is useful for the long term goal of building an AI agent that can play Angry Birds better than the best human players. It allows researchers to automatically identify how objects are affected by a shot, which is essential information for learning consequences of shots and for planning successful shot sequences. It also allows us to test the success and predicted outcome of Angry Birds game playing strategies, such as the structural analysis developed by Zhang and Renz (2014). Apart from Angry Birds there are a number of domains where tracking PIOs is useful. This includes areas such as traffic monitoring, surveillance, the study of animal movement patterns, crime scene investigation, or analyzing the impact of projectiles or explosions. One of our goal is to further increase the time gap between observations, but with large time gaps the problem of matching objects is getting extremely hard as there are fewer and fewer cues. We did some initial cognitive studies and asked people to match objects in before and after Angry Birds images. When the time gaps were greater than 2 seconds, people were mostly unable to find or to explain a correct match.

References

Adam, A., Rivlin, E., Shimshoni, I.: Robust Fragments-based Tracking using the Integral Histogram. In: Computer Vision and Pattern Recognition. vol. 1, pp. 798–805 (2006)

Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.): Handbook of Spatial Logics. Springer (2007)

Balbiani, P., Condotta, J.F., Del Cerro, L.F.: A model for reasoning about bidimensional temporal relations. In: Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning. pp. 124–130.(1998)

Battaglia, P.W., Hamrick, J.B., Tenenbaum, J.B.: Simulation as an engine of physical scene understanding. Proceedings of the National Academy of Sciences 110(45), 18327–18332 (2013)

Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, 24(4), 509–522 (2002)

Bose, B., Wang, X., Grimson, E.: Multi-class object tracking algorithm that handles fragmentation and grouping. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. pp. 1–8. IEEE Computer Society (2007)

Cabalar, P., Santos, P.E.: Formalising the fisherman's folly puzzle. Artificial Intelligence 175(1), 346–377 (2011)

Cohn, A.G., Renz, J.: Qualitative spatial representation and reasoning. Handbook of knowledge representation, 551–596 (2008)

Cohn, A.G., Renz, J., Sridhar, M.: Thinking inside the box: A comprehensive spatial representation for video analysis. In: Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning. pp. 588–592. (2012)

Cox, I., Hingorani, S.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)

Cutler, R., Davis, L.S.: Robust real-time periodic motion detection, analysis, and applications. Pattern Analysis and Machine Intelligence 22(8), 781–796 (2000)

Davis, E., Marcus, G.: The scope and limits of simulation in cognition and automated reasoning. Under submission (2013)

Dubba, K.S., Cohn, A.G., Hogg, D.C.: Event model learning from complex videos using ilp. In: Proceedings of 19th European Conference on Artificial Intelligence. pp. 93–98 (2010)

Forbus, K.D., Nielsen, P., Faltings, B.: Qualitative spatial reasoning: the clock project. Artificial Intelligence 51(1), 417–471 (1991)

Forbus, K.D., Usher, J.M., Lovett, A., Lockwood, K., Wetzel, J.: Cogsketch: Open-domain sketch understanding for cognitive science research and for education. In: SBM. pp. 159–166 (2008)

Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly 69(1), 9–15 (1962)

Galton, A.: Qualitative spatial change. Oxford University Press, USA (2000)

Ge, X., Gould, S., Renz, J.: Angry Birds Game Playing Software Version 1.3: Basic Game Playing Software. http://www.aibirds.org (2014)

Ge, X., Renz, J.: Representation and reasoning about general solid rectangles. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. pp. 905–911. IJCAI'13, (2013)

Goyal, R., Egenhofer, M.: The direction-relation matrix: A representation of direction relations for extended spatial objects. UCGIS annual assembly and summer retreat, Bar Harbor, ME pp. 65–74 (1997)

Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. Pattern Analysis and Machine Intelligence 27(11), 1805–1819 (2005)

Kreutzmann, A., Colonius, I., Wolter, D., Dylla, F., Frommberger, L., Freksa, C.: Temporal logic for process specification and recognition. Intelligent Service Robotics 6(1), 5–18 (2013)

Kuipers, B.: Qualitative simulation. Artificial intelligence 29(3), 289–338 (1986)

Laird, J.E.: Extending the soar cognitive architecture. Frontiers in Artificial Intelligence and Applications 171, 224 (2008)

Lowe, D.G.: Object recognition from local scale-invariant features. In: 7th International Conference on Computer vision. vol. 2, pp. 1150–1157. ICCV (1999)

Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: 6th International Conference on Computer Vision. pp. 555–562. ICCV (1998)

Santore, J.F., Shapiro, S.C.: Identifying perceptually indistinguishable objects. Ph.D. thesis, State University of New York at Buffalo (2005)

Sokeh, H.S., Gould, S., Renz, J.: Efficient extraction and representation of spatial information from video data. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. pp. 1076–1082 (2013)

Sridhar, M., Cohn, A.G., Hogg, D.C.: From video to RCC8: exploiting a distance based semantics to stabilise the interpretation of mereotopological relations. In: Egenhofer, M.J., Giudice, N.A., Moratz, R., Worboys, M.F. (eds.) 10th International Conference on Spatial Information Theory. pp. 110–125. Springer (2011)

Wallgrün, J.O., Wolter, D., Richter, K.F.: Qualitative matching of spatial information. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 300–309 (2010)

Westphal, M., Dornhege, C., Wölfl, S., Gissler, M., Nebel, B.: Guiding the generation of manipulation plans by qualitative spatial reasoning. Spatial Cognition & Computation 11(1), 75–102 (2011)

Wintermute, S., Laird, J.E.: Predicate Projection in a Bimodal Spatial Reasoning System. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence. pp. 1572–1577 (2007)

Wintermute, S., Laird, J.E.: Bimodal spatial reasoning with continuous motion. In: In Proceedings of the 23rd AAAI Conference on Artificial Intelligence. pp. 1331–1337 (2008)

Xu, M., Petrou, M.: 3d scene interpretation by combining probability theory and logic: The tower of knowledge. Computer Vision and Image Understanding 115(11), 1581–1596 (2011)

Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys (CSUR) 38(4), 13 (2006)

Yilmaz, A., Li, X., Shah, M.: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. Pattern Analysis and Machine Intelligence, 1531–1536 (2004)

Zhang, P., Renz, J.: Qualitative spatial representation and reasoning in Angry Birds: The extended rectangle algebra. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning. (2014)