"You're it!": Role Identification using Pairwise Interactions in Tag Games

Alejandro Moreno and Ronald Poppe
Human Media Interaction, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
{a.m.moreno, r.w.poppe}@utwente.nl

Abstract

We aim at designing interactive playgrounds that automatically analyze the behavior of children while playing games, in order to adapt the gameplay and make the games more engaging. In this paper, we focus on recognizing roles in tag games, where children are taggers or runners. We start by tracking the location and motion of individual players, and subsequently recognize pairwise interactions: approach, chase and avoid. At each moment in time, we inspect the full set of pairwise interactions to determine the role of each player. Our approach is fully probabilistic, deals with any number of players and can easily be extended to include other interactions and roles. We evaluate our algorithm using simulations, which show promising results. We intend to extend our framework to recognize variants of the tag game, and to address actual play interactions.

1. Introduction

Sensing and interpreting human behavior has been an active field of research in computer vision for many years. Many tracking algorithms have been proposed [9], as well as activity recognition methods [19]. These works mainly considered the behavior of individuals. Recently, there has been a shift towards the analysis of behavior of groups, for example to determine group activity [3] or analyze pedestrian movement [18]. The developed models all adhere to unseen forces, social or psychological, that determine the proximity of people in groups [13, 1, 12].

When it comes to games, these forces are less apparent or non-existent. This makes it difficult to identify groups during games since physical proximity is no longer a key cue. For instance, two children far away from each other might be passing a ball between them, and it would make sense to think they belong to the same group. In a social setting, two people in a crowd passing a ball at each other from far away would be frowned upon. Analyzing individual actions such as walking or jumping will not be sufficient to understand the joint activity of the two children. Therefore, we propose to analyze interactions between individuals to understand what is going on in the game.

We propose to recognize two roles during tag games, taggers and runners, using pairwise interactions between the players. We refer to pairwise interactions as events that require and are limited to two children: one which executes the action and one whom is on the receiving end of it. For instance, a child that follows another child. A benefit of using this definition is that any scenario can be broken down into simple pairwise interactions. For instance, even if a child is being chased by two (or more) children, each one has a pairwise relationship with the chased child. As a result, the proposed model can deal with any number of players. It is important to note that not all pairwise interactions are meaningful in recognizing roles during games. Moreover, there are interactions that are difficult to define, due to the chaotic nature of children's play. This makes the recognition of roles in game settings a challenging task. We analyze three interactions that we consider the most relevant for our purposes: approach, chase, and avoid. These interactions are common in children's play, and our approach is general enough to add other interactions. As such, it is possible to cover different games and their variants.

We first introduce tag games and how these could be made more engaging by introducing technology into the playground. Next, we discuss related work on group analysis. We describe our algorithm in Section 3, and evaluate it on simulated data in Section 4. We conclude in Section 5.

2. Interactive Tag Game Playgrounds

Tag games are popular children's games where one or more taggers have to chase the runners. When a tagger touches a runner, the roles of the two switch. As such, tag is a dynamic game since the roles that children play change frequently during the game. The number of chasers and runners can vary. Variations of tag games exist, for instance, those where a tagged runner is "out", "frozen", or becomes a tagger without the tagger assuming the runner role.

We are interested in implementing an interactive tag game playground that is able to recognize the roles of children while they are playing. The playground will be enhanced with sensors and actuators to sense the children's behavior and provide adequate feedback. For instance, taggers and runners could have either red or blue circles projected around them, and arrows could point from a tagger to the closest runner, or to the runner that has not been tagged for a while. Additionally, we also want to study how adapting the game could be used to influence the playing behavior of children. For instance, if changing the color of a child's circle could encourage him to change his role. This could be used to promote positive behavior such as encouraging shy children to adopt a more proactive role in the game, *i.e.* the tagger role [14].

Current interactive playgrounds use computer vision to sense children's behavior. For instance, Tetteroo *et al.* designed an interactive playground that tracks children's movement and recognized some actions using cameras and accelerometers [20]. Depending on the children's positions and their actions, shapes projected on the floor responded in different ways. Tracking children in playgrounds is useful to create interactive systems. However, reaching a higher understanding of play is difficult relying solely on this. Studying social context during play, the interactions that children engage in and with whom, would allow us to understand more clearly what is happening in the playground.

2.1. Vision-Based Analysis of Groups

A substantial amount of research has considered the analysis of groups, specifically proxemics, a term coined by Hall [12]. One of the first studies that hinted how social forces could affect behavior was carried out by Argyle and Dean, who proposed the equilibrium theory [1]. This theory stated that people dynamically adapt their postures, gestures, gaze and proximity to others depending on how intimate their relationship is. Similarly, Hall also proposed that we regulate our distance to others when interacting, and that these distances are dependent on our social intimacy with them [12]. More recently, Helbing and Molnár proposed that the motion of pedestrians is subject to influence from other pedestrians as well as internal motivations in their social force model [13]. All these theories state that an individual's behavior is influenced by other people present in their surroundings, and that modeling these relationships helps to explain the exhibited behavior.

Following this line of research, the computer vision community has begun to automatically analyze grouping behavior because modeling social forces also helps in tracking and understanding behavior [8]. In pedestrian tracking, the consideration that social factors influence people's behavior, helps in the prediction of their movement when alone or in groups. Leal-Taixé *et al.* developed a people tracker that worked in semi-crowded environments and considered social and grouping behavior that helped in predicting movement and handling occlusions[16]. Ge *et al.* developed a tracker of small groups in crowded scenarios [10]. They used knowledge from sociological models of human collective behavior, exhibited by people when in groups. Yamaguchi *et al.* proposed a method to improve pedestrian tracking algorithms by modeling social, personal and environmental factors [22].

The notion of social norms has also helped in the understanding and recognition of individual, pairwise and group activities. Groh *et al.* developed a system that classified when people were engaging in social interactions using interpersonal distances and body orientation [11]. Bazzani *et al.* also used body orientation, represented by the subjective view frustum, along with spatial cues to infer when people were interacting [4]. On a related study, they proposed a decentralized particle filter for joint individual-group tracking to recognize the birth, life, and death of groups in outdoor settings [2]. Kong *et al.* proposed the use of semantic motion relationships between two interacting people to recognize human interactions [15]. They called these descriptors interactive phrases, and allowed them to recognize actions such as boxing and handshake. Several studies have proposed diverse methods to recognize group activities such as fighting, walking in groups or queuing by analyzing individual and group information together [6, 7, 17]. Activities have also been recognized in games, however their chaotic nature means that all the players must be observed. Wang *et al.* presented an algorithm that exploited repetitions in social games to recognize stages during play and pairwise social interactions even in unstructured collections of videos [21]. Their games involved only two participants.

In this study, we leverage the concept of "social context" to a set of pairwise interactions, as a way to deal with the chaotic nature of play and the notion of different roles that govern these interactions. Our approach is similar to that of Chang *et al.*, whom proposed a probabilistic group-level motion analysis for recognizing group behavior in unconstrained surveillance environments [5]. They recognize scenarios such as meeting, following and loitering. The main difference with our approach is that they use proxemics to define groups. They propose a soft-grouping approach, where physical proximity and a path-based connectivity method define to which group an individual belongs to. Afterwards, they use this information together with the analysis of the individuals' behavior to recognize given scenarios. In contrast, we use spatial information to interpret the behavior of the individuals which is used afterwards to define their roles. Thus, the players are grouped based on their roles and not on how close to each other they are. In surveillance settings, the use of proxemics to define groups is common, however our setting is that of games and playgrounds, where proxemic conventions do not hold and the identification of teams is more meaningful.

3. Role Recognition in Tag Games

Our algorithm starts by describing the location and motion of individual players. From these, we determine pairwise interactions between players. For tag games specifically, we consider chasing, avoiding and approaching behavior. Finally, from the full set of pairwise interactions, we determine the role of each player. We discuss these three steps in the following three sections.

3.1. Individual Motion Analysis

The first step in the recognition of the players' interactions is the analysis of their motion. Consider a player i with a speed v_i and direction ϕ_i. Based on the individual movement information, we can classify the player's type of motion (m^t) into one of two states, where $m^t \in \{stand, run\}$. Other motion types could also be recognized, such as walking or jumping, however they are not informative in our current context. We use sigmoid functions, defined as $\frac{1}{1+e^{-a(x-c)}}$, to model the probability of each motion type. The variable a determines how severe the threshold should be (*i.e.* how steep), and c controls the function's displacement from the origin. The value assigned to c is usually based on facts or common knowledge, for instance, the average running speed of children to decide the speed threshold for running. On the other hand, a is assigned based on how quick the probability should change when approaching the threshold and is more subjective. The sigmoid responses of m^t against the speed of the players can be seen in Fig. 1.

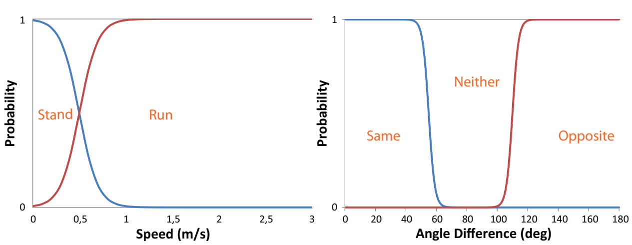

Figure 1. Probabilities of m^t and m^ϕ based on sigmoid functions.

3.2. Probabilistic Recognition of Pairwise Interactions

Next, we recognize important pairwise interactions present in tag games. Given players i and j, we denote a pairwise interaction executed by player i towards player j as $a_{ij} \in \{chase, approach, avoid, none\}$. Each interaction is defined by several terms which can include a relative direction term, a motion type term and a distance term. These definitions are based on the observation of game sessions and common knowledge of tag games. The advantage of defining the pairwise interactions this way is that restrictions, limitations and other terms can be added easily without the need to retrain the models. The modular representation of the pairwise actions also makes the system easily extendable in the future. Ad hoc actions or rules could be defined for specific types of tag games. Moreover, these relationships are also appropriate for games other than tag.

To recognize the pairwise interactions, we first need to compute the distance vector (d_{ij}) between i and j, and its angle (ϕ_{ij}). Afterwards, we calculate the angle difference between the movement direction of i and vector d_{ij}. This value informs whether i is moving towards j and is defined as $\phi_{i,ij} = |(\phi_i - \phi_{ij})|$. We also calculate the angle difference between the movement direction of both j and i, defined as $\phi_{j,i} = |(\phi_j - \phi_i)|$. A graphical description of these variables can be seen in Fig. 2. Based on the relative movement information, we classify relative directions, defined as $m^\phi \in \{same, opposite, neither\}$. We use sigmoid functions to model the probability of the relative directions. The probability distribution for m^ϕ can be seen in Fig. 1.

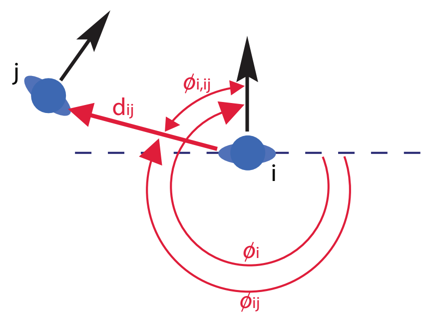

Figure 2. Graphical description of the motion analysis variables.

The informal definition for each pairwise interaction is the following. Player i approaches player j when i runs in the same direction of j, and j is standing. When i is running in the direction of j and j is running away from him, i is chasing j. Player i avoids player j when i runs in the opposite direction of where j is, and j is running in the direction of i. Formally, the interactions are defined as:

$$P(a_{ij} = approach) = P(m^\phi_{i,ij} = same) \cdot P(m^t_j = stand) \cdot P(m^t_i = run) \cdot prx(i,j). \quad (1)$$

$$P(a_{ij} = chase) = P(m^\phi_{i,ij} = same) \cdot P(m^\phi_{j,i} = same) \cdot P(m^t_i = run) \cdot P(m^t_j = run) \cdot prx(i,j). \quad (2)$$

$$P(a_{ij} = avoid) = P(m^\phi_{i,ij} = opposite) \cdot P(m^\phi_{j,ji} = same) \cdot P(m^t_i = run) \cdot P(m^t_j = run) \cdot prx(i,j). \quad (3)$$

where $prx(i,j)$ is a 2-D Gaussian function. This function gives more importance to the interactions of players that are nearby. Given the location of two players $[x_i, y_i]$ and $[x_j, y_j]$, and the standard deviation σ (function's distance falloff rate), $prx(i, j)$ is defined as:

$$prx(i,j) = exp(-(\frac{(x_i - x_j)^2}{2\sigma_x^2} + \frac{(y_i - y_j)^2}{2\sigma_y^2})). \quad (4)$$

The probabilities of the pairwise interactions are stored in a matrix $A \in \mathbb{R} : T + 2N + C$, where C is the number of possible interactions, N is the number of players, and T is the number of frames in the video. In addition to A, we use matrix Ac with the same dimensions to store accumulated and filtered probabilities over time. If an interaction is performed consecutively, but the recognition probability is low, the confidence that the recognition is correct should increase. This leads to better recognition when an interaction is not recognized accurately, for instance, as a result of tracking inaccuracies. At each frame *f*, if no interaction was recognized with a probability above the recognition threshold δ, we compare the interaction with the highest recognition probability $\hat{a}_{i,j}^{f} = argmax_a(A_{i,j,a}^{f})$, with the one in the previous frame $\hat{a}_{i,j}^{f-1}$. If the interaction is the same, we re-evaluate the probability of that interaction in frame *f*. The new value is calculated as follows:

$$\begin{aligned} maxP &= max(\hat{a}_{i,j}^{f}, \hat{a}_{i,j}^{f-1}) \\ minP &= min(\hat{a}_{i,j}^{f}, \hat{a}_{i,j}^{f-1}) \\ newP &= maxP \cdot 0.75 + minP \cdot 0.25 \end{aligned} \quad (5)$$

This calculation is biased towards the highest recognition probability between adjacent values, but only when the same interaction is recognized in subsequent frames. If the subsequent interactions are not the same, no new estimation is performed since there is no temporal correlation, which means no interaction is classified confidently at frame f.

3.3. Role Estimation

After the pairwise interaction values have been calculated, we estimate the roles of the players. In tag games, a player can only be assigned one of two roles, thus we define a player's role as $r \in \{Tagger, Runner\}$. Because the interactions are closely related to the roles, the assignment becomes trivial. We estimate each player's role as:

$$r_i = \begin{cases} Tagger & \text{if } \exists j | A_{i,j} = Approach \vee Chase, \\ Runner & \text{Otherwise.} \end{cases}$$

This rule represents each role's specific type of behavior: a tagger's goal is to chase and tag runners, whereas runners have to avoid being tagged. In the particular case of tag games, if a player is not chasing anyone, he has to be either avoiding someone who is chasing him, or just moving away from the taggers to maximize his escape possibilities. Approaching is seen mostly in the corners of the playing area, where players have no room to run away.

4. Evaluation and Simulated Data

We evaluated the algorithm on simulated data, which allowed us to analyze the influence of different factors such as the number of players while avoiding potential errors made due to tracking inaccuracies.

4.1. Setup

We used two sets of 50 simulations: the *tag* set and the *run* set. The tag set simulates different tag games, whereas the second set consists of players running inside the play area. While the former set has defined roles, and consequently interactions between players, the run set is used to look at incidental recognition of roles. In this set, we did not specify interactions between players. Both sets are simulated with varying numbers of players, ranging from two to six. Each simulation lasted 200 frames, which gave us a total of 20,000 frames. We assumed a 10 frames per second capture rate, which means each play session lasted 20 seconds. The players moved at a maximum speed of 3 meters per second, the average running speed for a 10-year old. The playing area was set to 10×10 meter. The starting positions of the players were random, making each simulation different. We also varied the number of taggers and chasers in the simulations to simulate diverse playing conditions.

4.2. Simulation

The tag set represents realistic but simple tag game behavior. The taggers' behavior consisted of chasing the closest runner, whereas every runner tried to maximize his distance to every tagger (Fig. 3). Both taggers and runners ran at the same speed, but to prevent the runners from running away in straight lines, their movement direction was modified by adding a random angle ($\pm$ 0°-45°) to it in each frame. This led to a more natural running away behavior, while maintaining the same speed. In the case a runner hit the playing area boundary, he moved parallel to the wall. However, sometimes, the runners would hit a corner and be unable to get out when being chased. While this is somewhat unrealistic, we did not want to make the simulation behavior too complicated to allow for understanding and reproduction. In the case a tagger managed to tag a runner, their roles would switch. The new runner would immediately start running away, whereas the new tagger had to wait 10 frames before he could start chasing. In the run set, the players had no implemented chasing or avoiding behavior, only slightly randomized movement deviation similar to the tag set. When the players reached one of the walls, they would bounce in a random direction.

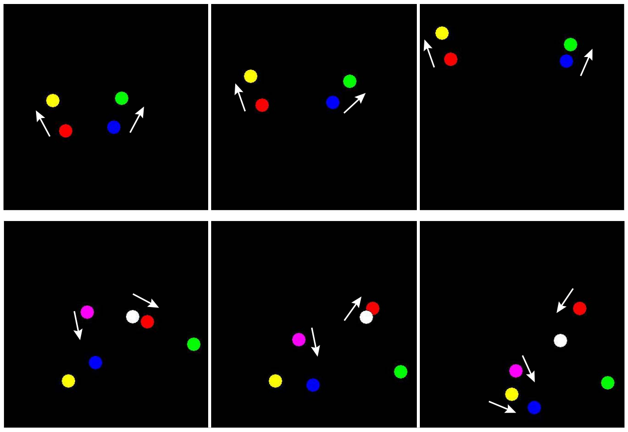

Figure 3. Example frames from two simulations (top and bottom rows) with different number of players. The white arrow indicates the direction the taggers are moving.

	GT Tagger	GT Runner
Guessed Tagger	**12,782**	1,319
Guessed Runner	1,253	**19,576**

Table 1. Confusion matrix for the tag set over all sequences.

4.3. Results

The ground truth for the roles and interactions of the tag set was known beforehand because the simulation behavior is based on it. We compared our algorithm's estimated roles against the ground truth for every frame (except the first one where there is no speed information available yet), for each player, in every simulation. We used a recognition threshold $\delta = 0.8$. It must be noted that because of the tag cooldown experienced by recently tagged players, we disregarded the algorithm's classification for that particular player, for those 10 frames. The confusion matrix of the role classification can be seen in Table 1 (GT stands for ground truth).

The algorithm estimates the roles with an overall accuracy of 92.6%. The precision scores for the tagger and runner roles are 90.6% and 94%, respectively, and their recall 91.1% and 93.7%. These differences are partly due to different ratios of taggers and runners. Most of the misclassifications occurred when runners hit the boundaries or the corners of the playing area. Instead of running away from the tagger, runners had to run alongside the wall. When in a corner, runners moved back and forth trying to get away from the corner, which resulted in the chase and approach interactions not being classified correctly. In the run set, none of the players should be classified as taggers. However, some of the pairwise interactions could be such that the emergent behavior appears as tagging. In 12.1% of the cases, a player was falsely classified as a tagger.

These results were obtained over all the sequences. However, it is likely that there are differences between settings. Using the tag set, we first analyze the effect of the

N		GT Tagger	GT Runner
2	**Guessed Tagger**	**36.86%**	0%
	Guessed Runner	6.87%	**56.29%**
3	**Guessed Tagger**	**39.90%**	2.10%
	Guessed Runner	3.76%	**54.25%**
4	**Guessed Tagger**	**38.96%**	1.07%
	Guessed Runner	4.09%	**55.87%**
5	**Guessed Tagger**	**39.59%**	3.63%
	Guessed Runner	2.70%	**54.08%**
6	**Guessed Tagger**	**30.80%**	7.82%
	Guessed Runner	2.78%	**58.60%**

Table 2. Confusion matrices of the tag set for different number of players.

Players per role		GT Tagger	GT Runner
1	**Guessed Tagger**	**25.21%**	3.85%
2	**Guessed Runner**	3.13%	**67.81%**
2	**Guessed Tagger**	**56.09%**	0.16%
1	**Guessed Runner**	4.44%	**39.31%**
2	**Guessed Tagger**	**30.37%**	5.79%
3	**Guessed Runner**	2.63%	**61.21%**
3	**Guessed Tagger**	**49.45%**	1.32%
2	**Guessed Runner**	2.78%	**46.45%**
3	**Guessed Tagger**	**38.22%**	6.62%
3	**Guessed Runner**	3.14%	**52.01%**
2	**Guessed Tagger**	**23.80%**	8.95%
4	**Guessed Runner**	2.45%	**64.81%**

Table 3. Confusion matrices of the tag set for different tagger/runner distributions

number of players. The breakdown appears in Table 2. As the number of players increases, there are more misclassifications. This is especially true for runners being classified as taggers. This is expected since runners that are not being chased try to maximize their distance from the taggers, which can cause two or more runners to run in the same direction, one behind the other. We also analyzed how role estimation varied with the number of taggers and runners. Table 3 shows the confusion matrices for different distributions of taggers and runners. We omitted the sessions where the number of taggers and runners was equal. When there are more runners (3 or 4), the number of false positives for the tagger role increases. This is the same case as before, where a high number of players increased the chance of runners running away in the same direction.

Finally, we varied the recognition threshold δ. We tested the algorithm using 0.7 and 0.9 as thresholds. The results are summarized in Table 4. As expected, as the threshold increases, the precision of the tagger role increases in detriment of its recall. Because the runner role is classified as the complement of the tagger role, it is understandable that its precision and recall have the opposite behavior. Regardless, the accuracy of the algorithm does not change significantly.

Thresh.		GT Tagger	GT Runner	Acc.
0.7	**Tagger**	**37.47%**	4.83%	92.4%
	Runner	2.73%	**54.97%**	
0.8	**Tagger**	**36.59%**	3.78%	92.6%
	Runner	3.59%	**56.04%**	
0.9	**Tagger**	**33.34%**	2.92%	90.2%
	Runner	6.84%	**56.90%**	

Table 4. Confusion matrices and accuracy of the tag set for different thresholds.

The highest accuracy was obtained when $\delta = 0.8$.

5. Conclusion

We introduced a probabilistic algorithm that estimates the roles of players in tag game simulations by determining their pairwise interactions. The algorithm is designed to identify two roles: tagger and runner. Given that social information can improve the understanding and classification of human behavior, we detect three pairwise interactions: approach, chase and avoid. These are inferred from the players' individual and relative motion and location. The algorithm was tested on simulations and recognized roles with a 92.6% accuracy. Additionally, we looked at the effect of changing the number of taggers and runners and the recognition threshold δ.

Since the recognized interactions are very common in children's play, role estimation could be attempted in a multitude of different games and their variants. Moreover, other interactions could be included in the model, or different modalities, to further improve the possibilities of recognition and enrich the play experience. The next step in our research agenda is to test the algorithm with real tag game play sessions. Afterwards, an interactive tag game playground could be built where the system recognizes roles and adapts the feedback to create engaging game experiences. We believe this would be an important step towards designing socially aware interactive playgrounds.

Acknowledgment

This publication was supported by the Dutch national program COMMIT.

References

[1] M. Argyle and J. Dean. Eye contact, distance, and affiliation. *Sociometry*, 28(3):289–304, 1965.

[2] L. Bazzani, M. Cristani, and V. Murino. Decentralized particle filter for joint individual-group tracking. In *Proceedings of CVPR*, pages 1886–1893, 2012.

[3] L. Bazzani, M. Cristani, G. Paggetti, D. Tosato, G. Menegaz, and V. Murino. *Video Analytics for Business Intelligence*, chapter Analyzing Groups: A Social Signaling Perspective, pages 271–305. Springer, 2012.

[4] L. Bazzani, M. Cristani, D. Tosato, M. Farenzena, G. Paggetti, G. Menegaz, and V. Murino. Social interactions by visual focus of attention in a three-dimensional environment. *Expert Systems*, 2012.

[5] M.-C. Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *Proceedings of ICCV*, pages 747–754, 2011.

[6] Z. Cheng, L. Qin, Q. Huang, S. Jiang, and Q. Tian. Group activity recognition by gaussian processes estimation. In *Proceedings of ICPR*, pages 3228–3231, 2010.

[7] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Proceedings of CVPR*, pages 3273–3280, 2011.

[8] M. Cristani, R. Raghavendra, A. Del Bue, and V. Murino. Human behavior analysis in video surveillance: A social signal processing perspective. *Neurocomputing*, 100:86–97, 2013.

[9] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, 2009.

[10] W. Ge, R. Collins, and R. Ruback. Vision-based analysis of small groups in pedestrian crowds. *Pattern Analysis and Machine Intelligence*, 34(5):1003–1016, 2012.

[11] G. Groh, A. Lehmann, J. Reimers, M. R. Friess, and L. Schwarz. Detecting social situations from interaction geometry. In *Proceedings of SocialCom*, pages 1–8, 2010.

[12] E. Hall. *The Hidden Dimension*. Doubleday, 1966.

[13] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, 1995.

[14] K. Hendrix, R. van Herk, J. Verhaegh, and P. Markopoulos. Increasing children's social competence through games, an exploratory study. In *Proceedings of IDC*, pages 182–185, 2009.

[15] Y. Kong, Y. Jia, and Y. Fu. Learning human interaction by interactive phrases. In *Proceedings of ECCV*, pages 300–313, 2012.

[16] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Proceedings of ICCV Workshops*, pages 120–127, 2011.

[17] B. Ni, S. Yan, and A. Kassim. Recognizing human group activities with localized causalities. In *Proceedings of CVPR*, pages 1470–1477, 2009.

[18] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Proceedings of ECCV*, pages 452–465, 2010.

[19] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[20] D. Tetteroo, D. Reidsma, E. van Dijk, and A. Nijholt. Design of an interactive playground based on traditional children's play. In *Proceedings of Intetain*, pages 129–138, 2011.

[21] P. Wang, G. D. Abowd, and J. M. Rehg. Quasi-periodic event analysis for social game retrieval. In *Proceedings of ICCV*, pages 112–119, 2009.

[22] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Proceedings of CVPR*, pages 1345–1352, 2011.