

# Identifying Malicious Players in GWAP-based Disaster Monitoring Crowdsourcing System\*

Changkun Ou

Institute of Computer Science  
University of Munich  
Munich, Germany  
changkun.ou@ifi.lmu.de

Yifei Zhan

Institute of Computer Science  
University of Munich  
Munich, Germany  
yifei.zhan@campus.lmu.de

Yaxi Chen<sup>1,2</sup>

1. The Key Laboratory for Computer Systems of  
State Ethnic Affairs Commission  
2. School of Computer Science and Technology  
Southwest Minzu University  
Chengdu, China  
yaxichen@swun.cn

**Abstract**—Disaster monitoring is challenging due to the lack of infrastructures in monitoring areas. Based on the theory of Game-With-A-Purpose (GWAP), this paper contributes to a novel large-scale crowdsourcing disaster monitoring system. The system analyzes tagged satellite pictures from anonymous players, and then reports aggregated and evaluated monitoring results to its stakeholders. An algorithm based on directed graph centralities is presented to address the core issues of malicious user detection and disaster level calculation. Our method can be easily applied in other human computation systems. In the end, some issues with possible solutions are discussed for our future work.

**Index Terms**—Human Computation, Network Analysis, Large-scale Crowdsourcing, Game-With-A-Purpose

## I. INTRODUCTION

MANY Non-Profit Organizations (NPOs) such as the United Nations Children’s Fund (UNICEF) provide [1] humanitarian assistance in developing countries. The largest challenges for these organizations are those unreachable zones [2] where the real time war situation or disaster level are extremely difficult to be derived. Lack of sufficient local infrastructures, disasters can only be monitored from the sky level. Satellite sensors are widely deployed in order to report images of monitoring areas [3].

Nowadays automatic disaster monitoring has not achieved satisfying success while highly costly manual methods cannot satisfy real-time requirements. Therefore, the fields of human computation and crowdsourcing are investigating methods to harvest crowd wisdom. GWAP is one representative theory which converts time- and energy-consuming image processing problems into games in which players are motivated to contribute. Inspired by this theory, we present a novel large-scale crowdsourcing disaster monitoring system. The system analyzes tagged satellite pictures from players and then calculates the disaster level automatically. An algorithm based on directed graph centralities is presented to address the core issues of malicious player detection as well as disaster level calculation. Our method can be applied to other human computation systems in general. As justification, the

mathematical correctness of the system is proved. In the end, we also discuss some limitations and relevant solutions for the future work.

## II. RELATED WORKS

Human computation system is a paradigm for utilizing the human processing power to solve problems that neither computers nor humans can solve independently [4], [5]. Most of the human computation systems can be seen [6] as crowdsourcing trade, which rely on the *wisdom of crowds*. Surowiecki claimed [7] four critical properties of wisdom of crowds: diversity of opinion, independence, decentralization and aggregation. Oinas-Kukkonen further concluded [8] the theoretical foundation of wisdom of crowds based on network analysis. For instance, PageRank was first proposed by Larry Page [9] and applied to social network analysis [10]. It is commonly used for expressing the stability of physical systems and the relative importance, so-called centralities, of the nodes of a network. PageRank fulfills the four conditions of a wisdom of crowd mentioned above.

The fundamental theory for this paper is Game-With-A-Purpose (GWAP), which involves game theory [11] in human computation systems [12], [13]. It outsourced within a computational process to humans in an entertaining way, namely gamification, and recently considered as the power of addressing large-scale data labeling costs in machine learning research [14]–[16]. Nevertheless, the data collection mechanisms for a game is variety that should be considered in a proper way [17]. In long-term research, ESP [12], and ARTigo [18] have verified through years of operations that human inputs are valuable and meaningful, and the most important two challenges in GWAP systems are game incentivization and malicious player detection.

Unfortunately, these existing representative GWAP-based human computation systems have the following issues: (1) They require two online players competing with each other, which may harm the degree of playability and even meet troubles when lacking of players. (2) They only use the most commonly appeared tags that cannot prevent massive malicious players attacking the system and providing meaningless tags. However, manually managing the tag database

\* This research has been supported by the Bavarian IUK Program (IUK-1805-0004/IUK577/002) and Sichuan Science and Technology Program (2019YFH0055).

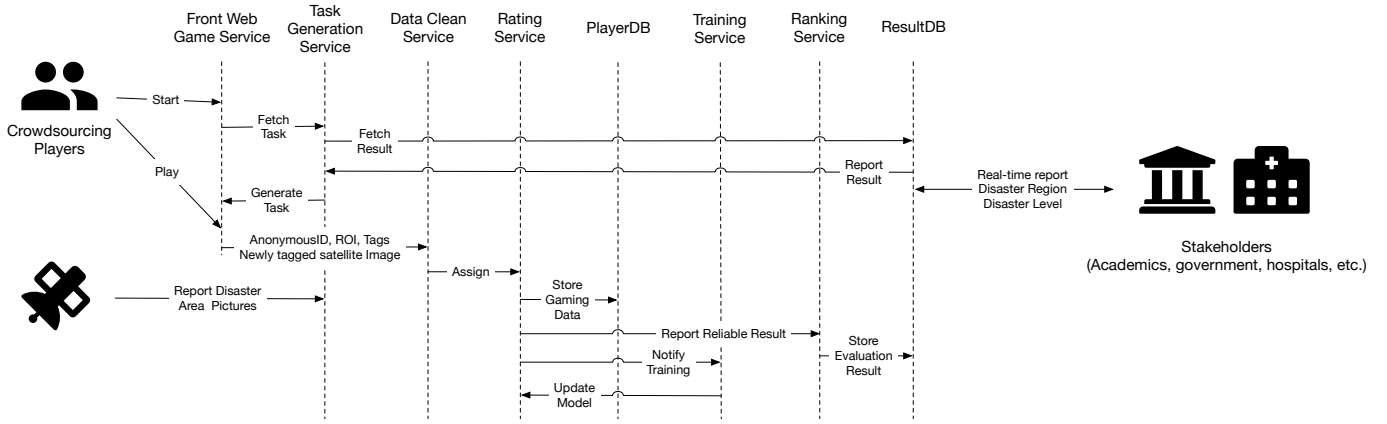


Fig. 1: System architecture: the system is designed to cooperate with multiple microservices, the most critical components are task generating service, the rating service and ranking service. The task generating service mixes reliable resulted images and new reported satellite images to generate player’s tasks. The rating service and ranking service are based on the PRM and DEM model respectively.

is not feasible due to the high cost of human labor and the inevitable issue of system cold start. In order to deal with the lack of players, our system turns multiplayer-required game into game between new players and existing reliable players. Furthermore, a malicious player detection algorithm based on directed graph centralities is proposed which requires only one single reliable players to avoid the issue of cold start.

### III. DESIGN AND MODELS

In this section, we describe the overall design and proposed models in detail. First, we propose the system architecture and specify the most critical components: *player task generator (PTG)*, *player rating model (PRM)* as well as *disaster evaluation model (DEM)*. With these components, the disaster monitoring system can handle the common issues in human computation system, such as system cold start and malicious player detection. It is also expandable, portable and can be easily applied to any other similar human computation systems.

#### A. System Architecture and Functionalities

Figure 1 illustrates the architecture of our disaster monitoring system. The system databases are composed of two different type of databases. The player database (PlayerDB) stores gaming data including the player’s property and raw tagging inputs. The other database is called ResultDB where persists the reliable players’ inputs that rated by our rating service. The overall data flow can be described as following:

*Step 1: Player task generation:* The *PTG* mixes the reliable gaming results from ResultDB and new reported images from satellite, and then assigns them to the future players.

*Step 2: Malicious player detection:* A reliable player requires to pass the malicious detection algorithm (see Algorithm 1) embedded in the *PRM*. Then the system will mark all the results from this player as reliable and then send them to the ranking service.



Fig. 2: System interface. a) Player game panel overview; b) Multi-tags selection for selected areas; c) Disaster level report in stakeholder view. Satellite images are taken from [19].

*Step 3: Disaster level evaluation:* the system reuses the reliable players’ inputs into *DEM* that embedded in ranking service and calculates the disaster level of the monitoring region then persists it in ResultDB.

After these three major steps, a disaster level report can be retrieved from ResultDB.

In our game, a player can execute infinity rounds of tasks, and each single round of task contains  $n$  image tagging tasks. In one task, the player is asked to tag  $n$  images (see Figure 2a). The player needs to draw a rectangle to select an area where a sign of danger or damage (such as fire or explosion) is discovered. System-suggested tags will then pop up and the player can select relevant ones by simply clicking on them (see Figure 2b). The player can also input new tags. The system analyzes the user input and creates a disaster level report (see Figure 2c) for this region which can be used by NGOs and governments.

#### B. Preliminaries

To describe and establish our models, we describe a few basic definitions in this subsection.

**Definition 1.** *The region of interests (ROI) is an indicator that represents player-selected two-dimentional region. The  $i$ -th ROI from player  $p$  in image  $k$  at image creation time  $t$  is denoted by  $ROI_{p,i,k,t}$ .*

Considering image  $k$  implies its creation time  $t$  (an image always contains its creation time), for convenience,  $ROI_{p,i,k,t}$  is simplified as  $ROI_{p,i,k}$ . For instance, Figure 3 shows some examples of ROIs in different images.

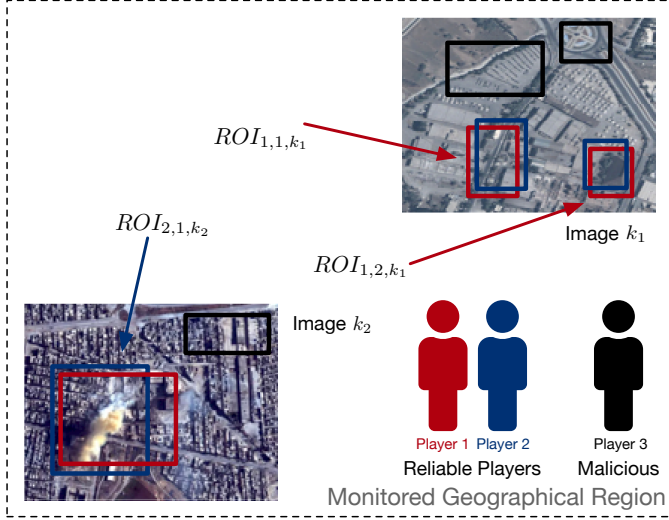


Fig. 3: Examples of ROIs in the disaster monitoring system: In this figure, reliable players (red and blue) draw rectangles to indicate area with disaster, however a player that does not cooperate with the game (black) selects different ROIs. The  $ROI_{p,i,k}$  indicates the  $i$ -th ROI from player  $p$  in image  $k$ .

**Remark 1.** The definition is designed for our database scheme, which includes PlayerDB and ResultDB, and their data schemes are illustrated in Listing A.1 and Listing A.2. See Appendix A for more descriptions.

As we discussed in Section III, *each tag can only be selected once, and players are allowed to input new tags for the selected ROIs*. Then, We define the ROI tag vector for the model:

**Definition 2.** Assuming the database stores  $n$  different tags  $g_1, g_2, \dots, g_n$  for a certain image  $k$ , the tag vector  $\mathbf{T}_{p,i,k}$  of  $ROI_{p,i,k}$  (the  $i$ -th ROI in image  $k$  of player  $p$ ) is a vector that is denoted by the following formula:

$$\mathbf{T}_{p,i,k} = (|g_1|, |g_2|, \dots, |g_n|)^T \quad (1)$$

where  $g_l$  is the  $l$ -th tag where  $l = 1, 2, \dots, n$ ,  $|g_l|$  is the count of  $g_l$  in a player task object, and  $n$  equals to the number of tags.

**Remark 2.** Since each tag can only be selected once, the components of tag vector is either 1 or 0. This definition performs a popular data preprocessing technique, which called One-Hot Encoding trick [20], [21].

**Remark 3.** For instance, for a certain image  $k$ , 5 different tags  $g_1, g_2, g_3, g_4, g_5$  were input by our game player. Assuming player  $p$  selects the first ROI and inputs tags for  $ROI_{p,1,k}$ :  $\{g_1, g_2, g_3, g_4\}$ , and player  $q$  selects the first ROI and inputs

tags for  $ROI_{q,1,k}$ :  $\{g_1, g_3, g_4, g_5\}$ . Then tag vector  $\mathbf{T}_{p,1,k}$  of  $ROI_{p,1,k}$  is  $(1, 1, 1, 1, 0)^T$  and tag vector  $\mathbf{T}_{q,1,k}$  of  $ROI_{q,1,k}$  is  $(1, 0, 1, 1, 1)^T$ .

### C. Player Task Generator

The *PTG* creates task images by combining images from satellite and ResultDB. A player task contains  $2n$  different images in random order, in which  $n$  images are untagged new satellite images and other  $n$  images are tagged images from ResultDB, PTG thus contains two generating steps:

*Step 1:* PTG splits a monitoring region into small pieces of images, assigning a unique identifier for each piece (The reason is discussed in Section IV-B).

*Step 2:* PTG retrieves tagged images from ResultDB, then combines these two types of images to create a task for a new upcoming player.

### D. Player Rating Model

The PRM is responsible for detecting malicious players. We convey the basic idea of centralities of a network and use eigenvalue as the trust value for each player to identify malicious players among all players.

The model is established from image dependent perspective. For a certain image  $k$ , considering a directed *player rating graph (PRG)* between players who tagged the image  $k$ . Each player is a node of PRG, as illustrated in Figure 4.

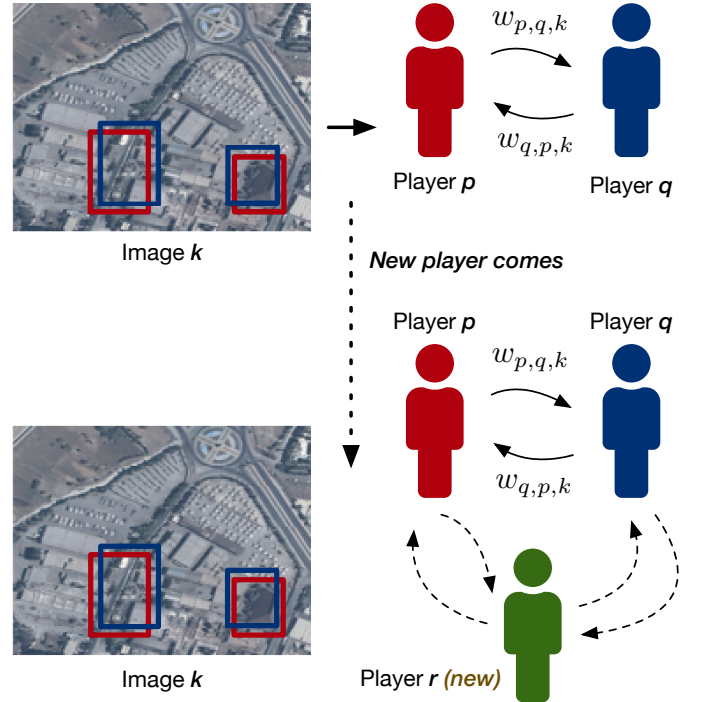


Fig. 4: PRG for certain images: Assume player  $p$  and  $q$  are former players who have been evaluated as reliable players. Consider each of the player as a node in a graph, the  $w_{p,q,k}$  is players' PRMR shown as a directed arc in the graph. A new player is composed with former players in the graph as a game network.

**Definition 3.** Assuming the database stores  $n$  different tags  $g_1, g_2, \dots, g_n$ . The system weight vector  $\mathbf{v} = (p(g_1), p(g_2), \dots, p(g_n))^T$  of all tags can be calculated by the following Equation 2:

$$p(g_i) = \frac{|g_i|}{\sum_{j=1}^n |g_j|} \quad (2)$$

where  $|g_i|$  is the count of  $g_i$  in the system.

**Definition 4.** Assuming different tags  $g_{r_1}, g_{r_2}, \dots, g_{r_s}$  were tagged in a certain image  $k$ , the image weight vector is a vector for image  $k$  that is composed by part of the system weight vector, which is denoted by  $\mathbf{v}_k = (p(g_{r_1}), p(g_{r_2}), \dots, p(g_{r_s}))^T$  with  $r_i (i = 1, 2, \dots, s) \in \{1, 2, \dots, n\}$ ,  $r_i \neq r_j (i \neq j, j = 1, 2, \dots, s)$  and  $s \leq n$ .

**Remark 4.** For instance, the system has 2 different images. The first image is tagged by two players. One is  $g_1, g_2, g_5$  and another is  $g_1, g_2$ ; The second image is tagged by three players, their results are:  $g_1, g_2, g_5$ ;  $g_2, g_4, g_5$ ;  $g_3, g_4, g_5$ . Thus, the system currently has 5 different tags  $g_1, g_2, g_3, g_4, g_5$ . Each tag has corresponding counts: 3, 4, 1, 2, 4; Therefore the system weight vector is  $(\frac{3}{14}, \frac{2}{7}, \frac{1}{14}, \frac{1}{7}, \frac{2}{7})^T$ ; the image weight vector of the first image is  $(\frac{3}{14}, \frac{2}{7}, \frac{2}{7})^T$  since the first image only is tagged by  $g_1, g_2, g_5$ , and the image weight vector of the second image is the same as the system weight vector since the second image is tagged by all exist tags.

**Lemma 1.**  $p(g_i)$  holds the properties: a)  $0 \leq p(g_i) \leq 1$ , b)  $\sum_{i=1}^n p(g_i) = 1$ , and c)  $\sum_{i=1}^s p(g_{r_i}) \leq 1$ .

So far our player has two different type of inputs: the ROI, and its tag vector. To define the PRG edge weight, we introduce two input measurements in the subsequent Definition 5 and 6.

**Definition 5.** The players ROI matching ratio (PRMR) is an importance measurement that measures the proportion of two different ROI intersection surface from player  $p, q$  and the ROI surface from player  $p$  in a certain image  $k$ , which is denoted by the following formula:

$$PRMR(p, q, i, j, k) = \frac{|ROI_{p,i,k} \cap ROI_{q,j,k}|}{|ROI_{p,i,k}|} \quad (3)$$

where  $ROI_{p,i,k}$  is the  $i$ -th selected ROI from player  $p$ , and  $|ROI_{p,i,k}|$  is the surface area of  $ROI_{p,i,k}$ .

**Lemma 2.** The following inequality holds:

$$0 \leq PRMR(p, q, i, j, k) \leq 1 \quad (4)$$

**Definition 6.** The players input tag correlation (PITC) is an importance measurement that measures the proportion of the covariance of two different tag vectors  $\mathbf{T}_{p,i,k}, \mathbf{T}_{q,j,k}$  from player  $p, q$  and the covariance of  $\mathbf{T}_{p,i,k}$  from player  $p$  with itself under the image weight vector  $\mathbf{v}_k$ , which is denoted by the following formula:

$$PITC(p, q, i, j, k) = \frac{Cov(\mathbf{T}_{p,i,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)}{Cov(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k)} \quad (5)$$

where  $Cov(\mathbf{X}, \mathbf{Y}; \mathbf{w})$  is the weighted covariance between  $\mathbf{X}$  and  $\mathbf{Y}$ , which denoted by:

$$Cov(\mathbf{X}, \mathbf{Y}; \mathbf{w}) = \quad (6)$$

$$\frac{\sum_{i=1}^n w_i (x_i - \frac{1}{n} \sum_{i=1}^n w_i x_i) (y_i - \frac{1}{n} \sum_{i=1}^n w_i y_i)}{\sum_{i=1}^n w_i} \quad (7)$$

with  $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$ ,  $\mathbf{Y} = (y_1, y_2, \dots, y_n)^T$ ,  $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$ .

**Remark 5.** The definition of PRMR and PITC share the same intent for measuring asymmetric importance between player  $p$  and player  $q$  (namely how  $p$  thinks of  $q$ ).

**Remark 6.** The definition of PRMR is inspired by intersection over union (IoU), a wide-used computer vision criteria also as known as Jaccard index [22], [23]. statistically used to compare the similarity and diversity of sample sets. Differ from IoU, we only divided a single ROI surface area to guarantee the asymmetric property for directed graph weight.

**Remark 7.** The definition of PITC is inspired by the weighted pearson correlation coefficient [24], which is a measure of the linear correlation between two variables. In our case, with the same intent of PRMR, we drop the part of covariance of player  $q$  in denominator to guarantee the asymmetric property for directed graph weight.

**Remark 8.** The PRMR and PITC both are not metrics of distance due to  $PRMR(p, q, i, j, k) \neq PRMR(q, p, i, j, k)$  as well as  $PITC(p, q, i, j, k) \neq PITC(q, p, i, j, k)$ .

**Lemma 3.** The following inequality holds:

$$-1 \leq PITC(p, q, i, j, k) \leq 1. \quad (8)$$

So far, we have enough techniques to define the edge weight of PRG.

**Definition 7.** For a certain image  $k$ , the edge weight of the PRG between player  $p$  and  $q$  is denoted by the formula 9:

$$w_{p,q,k} = \sum_{j=1}^n \sum_{i=1}^m PRMR(p, q, i, j, k) (PITC(p, q, i, j, k) + 2) \quad (9)$$

with player  $p$  selected  $m$  ROIs, player  $q$  selected  $n$  ROIs.

The Perron-Frobenius theorem guarantees our goal can be drifted to the calculation of the adjacency matrix of PRG. In consequence, one can use the normalized adjacency matrix by using formula 10:

$$\mathbf{A}_k = (a_{p,q,k}) = \left( \frac{w_{p,q,k}}{\sum_q w_{p,q,k}} \right) \quad (10)$$

where  $k$  is the image indicator.

**Theorem 1 (Soundness).** The normalized adjacency matrix  $A_k$  of PRG of a certain image  $k$  is irreducible, real, non-negative, and column-stochastic, with positive diagonal element.

**Remark 9.** From the proof (in Appendix F) of property of positive diagonal elements, one can observe that the number “2” is a translation that guarantees  $\text{PITC}(p, q, i, j, k)$  lies on closed interval  $[1, 3]$  which helps us prove this theorem successfully.

According to Perron-Frobenius theorem and Theorem 1, one can infer that there exists a uniqueness eigenvector  $\mathbf{V}_k = (\lambda_{1,k}, \dots, \lambda_{n,k})^\top$  of  $\mathbf{A}_k$  (Perron vector), with a uniqueness eigenvalue  $\rho(\mathbf{A}_k)$  is the spectral radius of  $\mathbf{A}_k$  (Perron root), such that:

$$\mathbf{A}_k \cdot \mathbf{V}_k = \rho(\mathbf{A}_k) \cdot \mathbf{V}_k, \lambda_{i,k} > 0, \sum_{i=1}^n \lambda_{i,k} = 1.$$

Therefore, we define the trust value of a player as following:

**Definition 8.** A trust value  $\lambda_{i,k}$  of player  $i$  on image  $k$  is a score that equals to the  $i$ -th component of the Perron vector of the normalized PRG adjacency matrix  $\mathbf{A}_k$ .

This definition represents the rating score from player  $p$  to player  $q$  for a certain image  $k$ , as same as the centrality of the player  $q$ . With the trust value of players, we propose our classification algorithm:

---

**Algorithm 1:** Malicious Player Detection

---

**input :** New Player  $p$ ,  
Trusted Player  $p_1, p_2, \dots, p_m$ ,  
Task Images  $k_1, k_2, \dots, k_{2n}$ ,  
Acceptance Threshold  $\delta$

**output:** Reliability of Player  $p$

**begin**  
    counter  $\leftarrow 0$   
    reliability  $\leftarrow$  false  
    **for**  $k \in [k_1, k_2, \dots, k_{2n}]$  **do**  
        **if**  $k$  is tagged image **then**  
            calculate  $\lambda_{p,k}, \lambda_{p_1,k}, \dots, \lambda_{p_m,k}$   
            **if**  $\lambda_{p,k} \geq \frac{1}{m} \sum_{i=1}^m \lambda_{p_i,k}$  **then**  
                | counter  $\leftarrow$  counter + 1  
            **end**  
        **end**  
    **end**  
    **if** counter  $\geq \delta$  **then**  
        | reliability  $\leftarrow$  true  
    **end**  
**end**

---

**Remark 10.** The criterion of classifying new players performs the action that the trust value of a new player should not be less than the mean value of overall trust value of players on image  $k$ , which means the tagging performance of new player should not be worse than result performance of former players. The acceptance threshold is a customizable parameter that can be set beforehand. For instance, if  $\delta = 1$ , the new player only needs to pass one singular image of all tagged images; if  $\delta = n$  (half images of the task), the new player has to pass all tagged

images, which makes the system unbreakable if the system is initialized by a trusted group.

Note that sometimes new player carries new tags into the system. It will influence the tag vector calculation and cause the weight not computable due to the unequal dimensions of the tag vector of new player and old player. A solution for this issue is proposed in the following steps:

- If a new player does not provide new tag: Directly perform the calculation with the algorithm;
- If a new player carries new tags only: Directly drop them because they are unreliable;
- If a player carries both selected and new tags: a) Perform the calculation with the algorithm without new tags; b) Merge and update all weight vector  $v$  via formula 9 if the player is reliable; c) Otherwise drop and mark the result as unreliable.

### E. Disaster Evaluation Model

The idea of stochastic pooling [25]–[27] is applied to define our Disaster Evaluation Model. For a monitoring region at time  $t$ , we address the DEM through disaster level definition as follows:

**Definition 9.** A monitor region is composed by images  $k_1, \dots, k_n$ . Each image exists  $r_{k_i}$  number of ROIs with  $i = 1, \dots, n$ , and each ROI is tagged with tags  $g_1, \dots, g_m$ . The disaster level  $\Delta$  of a monitor region is calculated by the following:

$$\Delta = \sum_{j=1}^m \left( p(g_j) \frac{\sum_{g_j} |ROI|}{\sum_{i=1}^n |k_i|} \right) \quad (11)$$

where  $|ROI|$  is the surface area of a ROI,  $\sum_{g_j} |ROI|$  means accumulated surface area of all ROIs that tagged by  $g_j$ , and  $|k_i|$  is the surface area of image  $k_i$ .

**Remark 11.** The disaster level is defined as a weighted area coverage. The  $\frac{\sum_{g_j} |ROI|}{\sum_{i=1}^n |k_i|}$  is a surface ratio of the ROI over monitoring area, and the  $p(g_i)$  is the corresponding weight of the ratio.

**Theorem 2** (Denseness). The disaster level  $\Delta$  is dense in internal  $[0, 1]$ .

With Theorem 2, one can calculate the disaster level for a monitoring region according to Equation 11. An area gains higher disaster if the disaster level is closer to 1, and vice versa.

### F. Model Initialization

Due to the lack of users in the very beginning, cold start is a common problem in such human computation system. This issue is normally solved by hiring people to create data manually. In this system, we only have to consider one system initialization issue of cold start.

The issue appears in the PTG. To initialize the whole system, we need to address an *initial trusted group* for PTG who shall tag enough initial trusted results as well as a

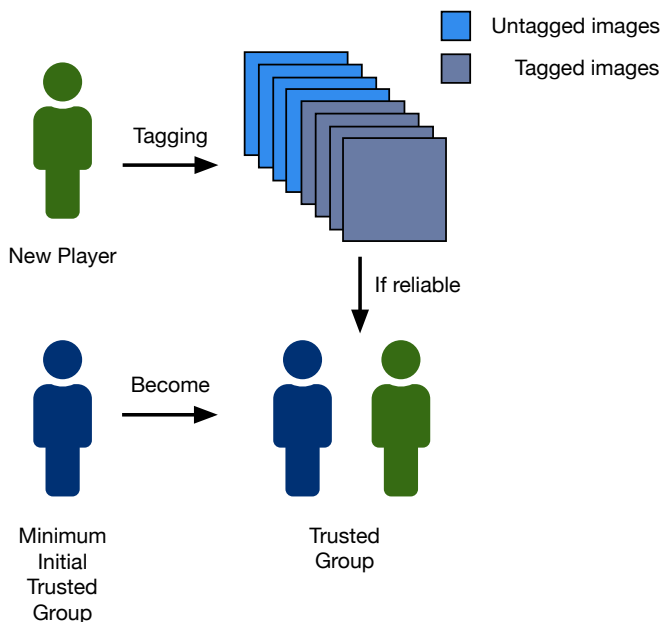


Fig. 5: Initialization of PRM

fixed predefined tag list (containing all of the most important keywords that need to be monitored) for PTG and then assign the tagged images to new upcoming players. Once a new player is included in the trusted group, all the relevant result from this player will be considered as reliable. The trusted group and available dataset grows with gradually growing number of reliable players and their reliable tags, as shown in Figure 5.

Thus, we have only one issue regarding the minimum number of the initial trusted group. Our PRM is based on graph centrality calculation, which means we need a (at least) two dimensional matrix to perform the overall model calculation. Hence, with the new player, *the minimum number of the initial trusted group is 1*. Then the initial trusted group (one person) with the new player form a two dimensional adjacency matrix that makes the model computable. For larger initial trusted groups, the trust value can be simply initialized to  $\frac{1}{n}$  with  $n$  is the number of initial trusted group.

#### IV. DISCUSSION

We have described the system architecture and 3 core models for task generation, malicious player detection and disaster level evaluation. Malicious player detection is essentially a classification problem in which our system determines the reliability of a new player based on the trust value. In this section, we would like to discuss some issues for the future work.

##### A. Simulated evaluation

To evaluate our model, a typical classification model performance evaluation metric is receiver operating characteristic (ROC) curve [28], which plots True Positive (on the y-axis) against False Positive and the ideal *surface under the ROC*

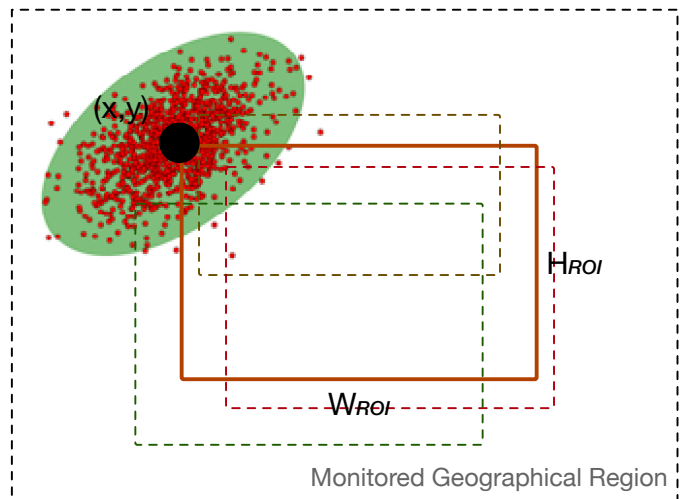


Fig. 6: An example of ROI simulation which can be used in the system evaluation.

*curve* is 1. Nevertheless, before we test the system with real users, one can generate a reasonable random dataset to test the performance of our classification model (PRM).

Our player has two different types of inputs: *the ROI and its tag vector*. For a reasonable player data entity, one has to define the ROI selection and its corresponding tag vector. To generate reasonable ROI for simulating real user behavior, we would like to discuss a desktop target click behavior first.

The target click behavior on a screen has been explored for years [29], [30]. It has been modeled and proved that the distribution of click behavior for a certain point satisfy Gaussian distribution [31]. Thus, from frequency statistic view, the actual ROI(s) certainly exists. No matter where the user starts, according to the Fitts Law and FFitts Law, the starting click point should follows normal distribution around the actual point, as shown in Figure 6. Similarly, the end point of the selection of ROI(s) should also follows a normal distribution.

Therefore, to generate ROI(s), let  $(x, y)$  as the player ROI starting point,  $(H_{ROI}, W_{ROI})$  as the height and width pair of this ROI, then we generate noise for the ROI starting points and landing points:  $(x + \epsilon, y + \epsilon)$ ,  $(H_{ROI} + \epsilon, W_{ROI} + \epsilon)$  where  $\epsilon \sim N(0, \delta)$ . For the parameter  $\delta$ , one can use maximum likelihood estimator [32] to perform the inference for all manually ROI selection samples from initial trusted group.

The generation of tag vector for a certain image is simpler than ROI's. A randomly pick from initial trusted group is sufficient for the simulation case because these tags are trusted results and a partially randomly selection already introduced the noise in this case.

Eventually, one can apply this random dataset to evaluate *surface under the ROC curve* as an indication of the overall performance (the model may show good performance if the surface approximate to 1).



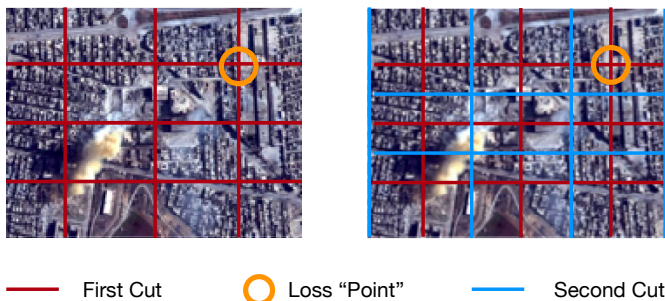


Fig. 7: Information loss may occur on the intersection lines; a possible solution is to perform a “half shifting” cut.

### B. Data leakage and information loss

In order to prevent leakage of data to malicious players, we intentionally cut original satellite images into small segmentations. However, this method may cause information loss if some important ROIs are located at the intersection of two dividing lines. A possible solution is to consider “half shifting” cut, as shown in Figure 7.

### C. Limitations

a) *Outdated Evaluation*: Our PRG network is based on image dependent perspective, that leads, each calculated disaster level may become invalid if the region image is outdated. We assume the satellite takes pictures for the monitoring area between intervals. However, our model only calculates the disaster level at a unique moment, which means the disaster level needs transvaluation when a new image is generated. If none of the new images gets evaluated, then the disaster level will not be updated. The disaster level of a certain region over time is essentially a non-stationary process [33] time series prediction method [34] can be applied on the disaster level time series.

b) *Game Playability*: Considering the fact that most parts of the earth are lake, forest, desert and so on, during the game playing, players may meet the situation that there is no available ROI in several continuous rounds. Obviously, it will decrease the playability and enjoyment of the game. A possible solution is pre-filtering these images from the image database.

## V. CONCLUSION

In this paper, we explored a GWAPs-based disaster monitoring system. We firstly proposed a *player rating model* based on eigenvalue centralities to calculate the trust value of a player. And then we proposed an algorithm for malicious user detection. As justification, we proved the mathematical correctness of this model. We then calculate the regional disaster level in the *disaster evaluation model*. We also deal with the general problem of system cold start by introducing the method of image half shifting cut. Our system design can also applied to other similar human computation systems. Furthermore, we discussed theoretical evaluation criteria for

this system, and then addressed corresponding solutions for the issues of data leakage, information loss and game playability.

## ACKNOWLEDGMENT

The authors would like to thank Prof. François Bry and Prof. Andreas Butz for their valuable input; we also thank colleague Yingding Wang for his inspiration on system design, algorithm rationalizations as well as system evaluations. Finally, we also thank Huimin An for his inspiration on Bayesian perspective that helps us handling human inputs with new tags successfully.

## REFERENCES

- [1] Unicef, *The state of the world's children*. 1998. Unicef, 1994.
- [2] T. Unicef. (2017) Humanitarian Action for Syrian Refugees: Situation Reports. [https://www.unicef.org/appeals/syrianrefugees\\_sitreports.html](https://www.unicef.org/appeals/syrianrefugees_sitreports.html). [Online; accessed 31-July-2017].
- [3] J. Zhang, C. Zhou, K. Xu, and M. Watanabe, “Flood disaster monitoring and evaluation in China,” *Global Environmental Change Part B: Environmental Hazards*, vol. 4, no. 2, pp. 33–43, 2002.
- [4] A. J. Quinn and B. B. Bederson, “A taxonomy of distributed human computation,” *Human-Computer Interaction Lab Tech Report, University of Maryland*, 2009.
- [5] L. von Ahn. (2016) Human computation systems. <http://www.cs.cmu.edu/~biglou/>. [Online; accessed 31-July-2017].
- [6] A. J. Quinn and B. B. Bederson, “Human computation: a survey and taxonomy of a growing field,” in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2011, pp. 1403–1412.
- [7] E. A. Mennis, “The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations,” *Business Economics*, vol. 41, no. 4, pp. 63–65, 2006.
- [8] H. Oinas-Kukkonen, “Network analysis and crowds of people as sources of new organisational knowledge,” *Knowledge Management: Theoretical Foundation*, pp. 173–189, 2008.
- [9] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web,” Stanford InfoLab, Tech. Rep., 1999.
- [10] P. Bonacich and P. Lloyd, “Eigenvector-like measures of centrality for asymmetric relations,” *Social networks*, vol. 23, no. 3, pp. 191–201, 2001.
- [11] S. Jain and D. C. Parkes, “A game-theoretic analysis of games with a purpose,” in *Proceedings of the 4th International Workshop on Internet and Network Economics*, ser. WINE '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 342–350.
- [12] L. Von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 319–326.
- [13] L. Von Ahn, “Games with a purpose,” *Computer*, vol. 39, no. 6, pp. 92–94, 2006.
- [14] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, “Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, p. 7, 2018.
- [15] W. Ertel, *Introduction to artificial intelligence*. Springer, 2018.
- [16] A. Ardalán, *Large-scale Information Extraction Using Rules, Machine Learning, and Human Computation*. The University of Wisconsin-Madison, 2018.
- [17] L. Von Ahn and L. Dabbish, “Designing games with a purpose,” *Communications of the ACM*, vol. 51, no. 8, pp. 58–67, 2008.
- [18] C. Wieser, F. Bry, A. Bérard, and R. Lagrange, “ARTigo: building an artwork search engine with games and higher-order latent semantic analysis,” in *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.
- [19] E. Graham-Harrison. (2016) ‘I couldn’t take anything except dignity’: stories of the leaving of Aleppo. <https://www.theguardian.com/world/2016/dec/23/i-couldnt-take-anything-except-dignity-people-aleppo-syria-on-fleeing-city>. [Online; accessed 31-July-2017].
- [20] J. Wu and S. Coggeshall, *Foundations of predictive analytics*. CRC Press, 2012.

- [21] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An Enabling Technique," *Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 393–423, Oct 2002. [Online]. Available: <https://doi.org/10.1023/A:1016304305535>
- [22] R. Real and J. M. Vargas, "The probabilistic basis of Jaccard's index of similarity," *Systematic biology*, vol. 45, no. 3, pp. 380–385, 1996.
- [23] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.
- [24] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.
- [25] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1. Barcelona, Spain, 2011, p. 1237.
- [26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
- [27] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012.
- [28] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [29] I. S. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction," *Human-computer interaction*, vol. 7, no. 1, pp. 91–139, 1992.
- [30] X. Bi, Y. Li, and S. Zhai, "FFitts law: modeling finger touch with fitts' law," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 1363–1372.
- [31] N. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)," *The Annals of mathematical statistics*, vol. 34, no. 1, pp. 152–177, 1963.
- [32] S. Johansen and K. Juselius, "Maximum likelihood estimation and inference on cointegration—with applications to the demand for money," *Oxford Bulletin of Economics and statistics*, vol. 52, no. 2, pp. 169–210, 1990.
- [33] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 2013.
- [34] V. Kuznetsov and M. Mohri, "Learning theory and algorithms for forecasting non-stationary time series," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 541–549.
- [35] O. Perron, "Zur theorie der matrices," *Mathematische Annalen*, vol. 64, no. 2, pp. 248–263, 1907.
- [36] F. G. Frobenius, "Über Matrizen aus nicht negativen Elementen," 1912.
- [37] J. R. Marden and J. S. Shamma, "Game theory and control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 105–134, 2018.

## APPENDIX

### A. Examples of database scheme

With Definition 1, the players of our system are able to select ROIs for each image as well as capable of select tags for each ROI. Thus, the *tasks* field in PlayerDB is an array object, stores each player image result with an assigned identifier.

```
{
  "player_id": "E3A6F124-4A6C-4C6E-B7F1-F8BC9A7381CC",
  "tasks": [
    {
      "image_id": "3A21E99E-F074-454B-A590-8D8C5ABD8E77",
      "image_at": "2017-07-31 11:28:40",
      "reliable": true,
      "ROIs": [
        {
          "x": 103, "y": 121, "height": 56, "width": 78,
          "tags": ["burning building", "explosion"]
        }
      ]
    }
  ]
}, ...]
```

Listing A.1: An example of PlayerDB data scheme

Each object in the *tasks* array has a field *reliable*, which indicates the reliability for this object task; Each object also contains a *ROIs* field, which is an array object that contains the player inputs for this object image; Each ROI object in the *ROIs* field has four properties that describes the ROI geometric location: *x*, *y*, *height*, *width*, and also a *tags* array field that describes the input tags for this image from this player. For *tags* field, game players can select the related tags for each ROI, and stores in this array.

```
{
  "region_id": "FBEB6204-0B94-4811-94F0-9DDC5FBBE6D8",
  "history": [
    {
      "image_id": "3A21E99E-F074-454B-A590-8D8C5ABD8E77",
      "image_at": "2017-07-31 11:28:40",
      "ROIs": [
        {
          "x": 103, "y": 121, "height": 56, "width": 78,
          "tags": ["burning building", "explosion"]
        }
      ]
    }
  ]
}, ...]
```

Listing A.2: An example of ResultDB data scheme

### B. Proof of Lemma 1

*Proof:* a) According to the Definition 3,  $|g_i|$  is non-negative, then  $0 \leq |g_i| \leq \sum_{j=1}^n |g_j|$ . Thus, we have  $0 \leq p(g_i) \leq 1$ . b)  $\sum_{i=1}^n p(g_i) = \sum_{i=1}^n \frac{|g_i|}{\sum_{j=1}^n |g_j|} = 1$ . c)  $\sum_{i=1}^s p(g_{r_i}) \leq \sum_{i=1}^n p(g_{r_i}) \leq 1$ . ■

### C. Proof of Lemma 2

*Proof:* According to the Definition of ROI,  $|ROI_{p,i,k} \cap ROI_{q,j,k}|$  can archive its maximum value only and only if  $ROI_{p,i,k} = ROI_{q,j,k}$  as well as its minimum value only and only if  $ROI_{p,i,k}$  has no intersection with  $ROI_{q,j,k}$ . Thus:

$$0 = \frac{0}{|ROI_{p,i,k}|} \leq \text{PRMR}(p, q, i, j, k) \leq \frac{|ROI_{p,i,k} \cap ROI_{q,j,k}|}{|ROI_{p,i,k}|} = \frac{|ROI_{p,i,k}|}{|ROI_{p,i,k}|} = 1. \quad (12)$$

### D. Proof of Lemma 3

*Proof:* We know that the weighted Pearson Correlation Coefficient [24] lies on  $[-1, 1]$ , i.e.

$$-1 \leq \frac{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)}{\sqrt{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) \text{Cov}(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)}} \leq 1$$

To prove Equation 8, we have to show:

$$\frac{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)}{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k)} \leq |\text{Cov}(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)| \sqrt{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) \text{Cov}(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)} \leq 1 \quad (13)$$

and

$$\frac{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)}{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k)} \geq -|\text{Cov}(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)| \sqrt{\text{Cov}(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) \text{Cov}(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)} \geq -1 \quad (14)$$



Then we need to show:

$$0 \leq Cov(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) Cov(\mathbf{T}_{q,j,k}, \mathbf{T}_{q,j,k}; \mathbf{v}_k)^3 \leq 1 \quad (15)$$

Considering  $\mathbf{T}_{p,i,k}, \mathbf{T}_{q,i,k}$  are described in general, with Equation 6, we only need to show ( $s$  is an vector components index instead of exponential):

$$0 \leq Cov(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) = \frac{\sum_{s=1}^n \mathbf{v}_k^s \left( \mathbf{T}_{p,i,k}^s - \frac{1}{n} \sum_{s=1}^n \mathbf{v}_k^s \mathbf{T}_{p,i,k}^s \right)^2}{\sum_{s=1}^n \mathbf{v}_k^s} \leq 1 \quad (16)$$

According to the definition of tag vector and image weight vector, the components of  $\mathbf{T}_{p,i,k}$  are either 1 or 0, the components of  $\mathbf{v}_k$  lies on  $[0, 1]$ , with Lemma 1, we have:

$$0 \leq \left( \mathbf{T}_{p,i,k}^s - \frac{1}{n} \sum_{s=1}^n \mathbf{v}_k^s \mathbf{T}_{p,i,k}^s \right)^2 \leq 1 \quad (17)$$

Therefore,

$$0 = \frac{\sum_{s=1}^n \mathbf{v}_k^s \cdot 0}{\sum_{s=1}^n \mathbf{v}_k^s} \leq Cov(\mathbf{T}_{p,i,k}, \mathbf{T}_{p,i,k}; \mathbf{v}_k) \leq \frac{\sum_{s=1}^n \mathbf{v}_k^s \cdot 1}{\sum_{s=1}^n \mathbf{v}_k^s} = 1 \quad (18)$$

which proves Equation 16. ■

### E. Proof of Theorem 1

*Proof:*

*Irreducibility* As shown in Figure 4, for a certain image  $k$ , the PRG is strong connected because the player who selected ROIs in image  $k$  has a direct connection to any other player who also selected ROIs in image  $k$  (the edge weight is well defined according to Equation 9). Thus, since  $A_k$  is an normalized strong connected PRG adjacency matrix, which proves  $A_k$  is irreducible.

*Real elements* With Lemma 2 and 3, each part of the Equation 9 are real number. Thus, of course, the matrix  $A_k$  elements are calculated by Equation 10 that are real elements.

*Non-negative elements* With Lemma 2 and 3, we have:

$$w_{p,q,k} \quad (19)$$

$$= \sum_{j=1}^n \sum_{i=1}^m (\text{PRMR}(p, q, i, j, k) (\text{PITC}(p, q, i, j, k) + 2)) \quad (20)$$

$$\geq \sum_{j=1}^n \sum_{i=1}^m (0 \cdot (-1 + 2)) = 0 \quad (21)$$

Thus,  $w_{p,q,k}$  has its lower bound when  $\text{PRMR}(p, q, i, j, k) = 0$  (for all  $i = 1, \dots, m; j = 1, \dots, n$ ) and  $\text{PITC}(p, q, i, j, k) = -1$  (for all  $i = 1, \dots, m; j = 1, \dots, n$ ). Meanwhile,

$$w_{p,q,k} \quad (22)$$

$$= \sum_{j=1}^n \sum_{i=1}^m (\text{PRMR}(p, q, i, j, k) (\text{PITC}(p, q, i, j, k) + 2)) \quad (23)$$

$$\leq \sum_{j=1}^n \sum_{i=1}^m (1 \cdot (1 + 2)) = 3mn \quad (24)$$

and  $w_{p,q,k}$  has its upper bound when  $\text{PRMR}(p, q, i, j, k) = 1$  (for all  $i = 1, \dots, m; j = 1, \dots, n$ ) and  $\text{PITC}(p, q, i, j, k) = 1$  (for all  $i = 1, \dots, m; j = 1, \dots, n$ ).

*Positive diagonal elements* According to Lemma 3, the diagonal elements can be formalized by follows:

$$w_{p,p,k} \quad (25)$$

$$= \sum_{j=1}^m \sum_{i=1}^m (\text{PRMR}(p, p, i, j, k) (\text{PITC}(p, p, i, j, k) + 2)) \quad (26)$$

$$\geq \sum_{j=1}^m \sum_{i=1}^m \left( \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,j,k}|}{|\text{ROI}_{p,i,k}|} (-1 + 2) \right) \quad (27)$$

$$= \sum_{j=1}^m \sum_{i=1}^m \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,j,k}|}{|\text{ROI}_{p,i,k}|} \quad (28)$$

$$= \sum_{i=j} \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,j,k}|}{|\text{ROI}_{p,i,k}|} \quad (29)$$

$$+ \sum_{i \neq j} \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,j,k}|}{|\text{ROI}_{p,i,k}|} \quad (30)$$

$$\geq \sum_{i=j} \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,j,k}|}{|\text{ROI}_{p,i,k}|} \quad (31)$$

$$= \sum_{i=1}^m \frac{|\text{ROI}_{p,i,k} \cap \text{ROI}_{p,i,k}|}{|\text{ROI}_{p,i,k}|} \quad (32)$$

$$= \sum_{i=1}^m \frac{|\text{ROI}_{p,i,k}|}{|\text{ROI}_{p,i,k}|} = m > 0 \quad (33)$$

*Column stochastic* according to the definition of matrix  $A$ , the sum of the column elements are:

$$\sum_q a_{p,q,k} = \sum_q \frac{w_{p,q,k}}{\sum_q w_{p,q,k}} = \frac{\sum_q w_{p,q,k}}{\sum_q w_{p,q,k}} = 1 \quad (34)$$

■

### F. Proof of Theorem 2

*Proof:*

According to the Definition 9 and Lemma 1, it is trivial to show  $\sup \Delta = 1$  and  $\inf \Delta = 0$ , since  $p(g_i)$  lies in  $[0, 1]$  and  $\sum_{g_j} |\text{ROI}| \leq \sum_{i=1}^n |k_i|$ .

The rest of the proof will prove  $\forall \Delta_p < \Delta_q$ , there exist  $\Delta_r$  such that  $\Delta_p < \Delta_r < \Delta_q$ .

We may assume two monitored region  $\Delta_p$  has  $m_q$  tags and  $\Delta_q$  has  $m_q$  tags where  $m_p + 2 = m_q$ , which indicates that there exists two tags  $g_\alpha$  and  $g_\beta$  are not appeared in  $\Delta_p$  but in  $\Delta_q$ . Let  $m_r = m_p + 1$ , i.e. one of  $g_\alpha$  and  $g_\beta$  appear in region  $\Delta_r$ , thus we have:

$$\Delta_p = \sum_{j=1}^{m_p} \left( p(g_j) \frac{\sum_{g_j} |ROI|}{\sum_{i=1}^n |k_i|} \right) \quad (35)$$

$$< \sum_{j=1}^{m_p+1} \left( p(g_j) \frac{\sum_{g_j} |ROI|}{\sum_{i=1}^n |k_i|} \right) = \Delta_r \quad (36)$$

$$< \sum_{j=1}^{m_p+2} \left( p(g_j) \frac{\sum_{g_j} |ROI|}{\sum_{i=1}^n |k_i|} \right) = \Delta_q \quad (37)$$

■