# Object Recognition Process and Its Applications in Face Detection and Augmented Reality


Geraldi Dzakwan 13514065
*Informatics Undergraduate Program*
*School of Electrical Engineering and Informatics*
*Bandung Institute of Technology, Ganesha Avenue Number 10 Bandung 40132, Indonesia*
*geraldzakwan@students.itb.ac.id*



***Abstract*—Object recognition is a branch of computer science that is related to image processing and has been implemented to wide variety of computer science products, especially for artificial intelligence field (e.g. for robots to identify objects around them so they can deal with them). To recognize an object, we first start by modeling an object to a structure in which computer is familiar with so that it can be processed furthermore by a computer. Then, we have to run some algorithms on the computer so that the computer can recognize the object. Finally, the computer will remodel the object to a structure similar with the real things we usually see so that we can interpret the result. This paper will review the process of object recognition using one approach which models an object by Saliency Map Graph and processes it with the SMGAT and SMGBM algorithm. In the last section, there will be as well an explanation about how object recognition has been implemented to support artificial intelligence product such as face detection technology and to a brand new product in computer science world, Augmented Reality.**



***Keywords*—view-based object recognition, saliency map graph, bipartite graph, geometrical similarity, topological similarity, face detection, augmented reality**


## I. Introduction

Object recognition is indeed a new field of computer science compared to another computer science fields. But, it is actually a new expansion of digital image processing field which is a field that has been developed for decades. Although it's new, it has been a vital thing to computer science world since it has so many applications in artificial intelligence, such as Android Eyes, Face Detection, Manufacturing Quality Control, Automated Vehicle Parking Systems, and Augmented Reality.

There are actually many approaches and methods on how to recognize an object. But, as stated before in the abstract, this paper will only review an object model named Saliency Map Graph and then continued by explaining the SMGBM and SMGAT algorithm to process the model. The details explanation of the Saliency Map Graph model will not be included here, but instead in the object recognition domain section later. The details of both algorithms also will not be included here, but instead in processing image representation section later.

Here I will only show you an illustration of the algorithms basic idea. Although SMGAT and SMGBM is a distinct method, they are on one similarity. They are algorithms which match two saliency map graph based on formulating the problem as a maximum cardinality minimum weight matching in a bipartite graph.

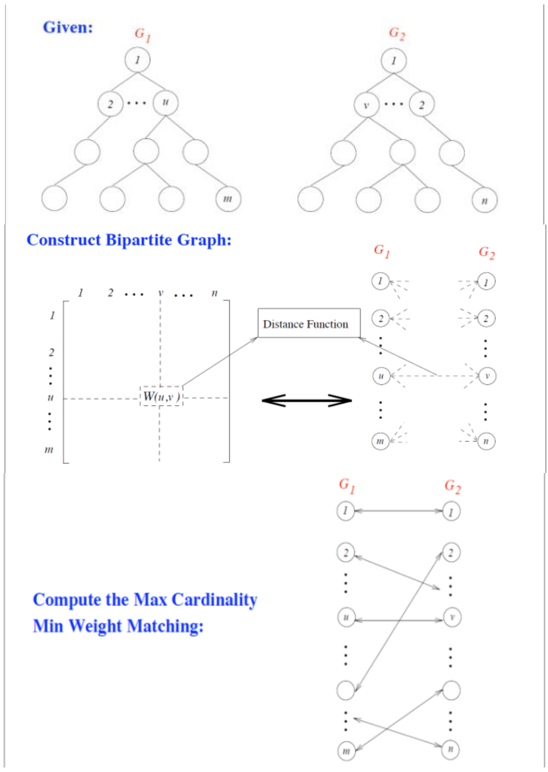



Fig 1.1 maximum cardinality minimum weight matching in a bipartite graph
Source :
http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf

As a complement, in the last section of this paper, I would like to introduce and give insight about Augmented Reality and Face Detection as a new innovation in computer science. Their concept is based on object recognition so I think this paper's last section will correspond to the previous sections preceding.

Hence, I would say that this paper's contains hold the whole thing about object recognition. This paper starts with basic theories of graph which underlies object recognition technology. Then, how an object recognition process can be done will be reviewed in the middle part of the paper. Last but not least, there will be an insight of artificial intelligence product and augmented reality technology as some of the implementations.

## II. RELATED THEORIES

### A. Graph Overview

A graph G = (V, E) consists of a nonempty set V of vertices (or *nodes*) and a set E of *edges*. Each of edge in the graph has either one or two nodes associates with it, called *endpoints*. An edge is a line which connects its endpoints. An edge that connects a vertex to itself is called a *loop*.

Based on the edge types, we can differ graph into three types listed below :

1. *Simple Graph*
   In a simple graph, each edge connects two different vertices and no edges connect the same pair of vertices. Simple graph also doesn't have a loop
2. *Multigraph*
   Multigraph may have multiple edges connecting the same two nodes but multigraph cannot have a loop
3. *Pseudograph*
   Pseudograph is the only graph which can include loops as well as multiple edges connecting the same pair of vertices

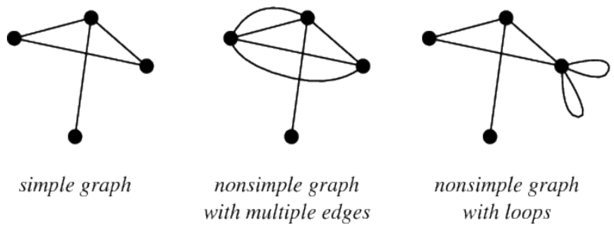



Fig 2.1 Graph illustrations based on its edge types
Source : *http://mathworld.wolfram.com/images/eps-gif/SimpleGraph_950.gif*

A graph can have orientation for its edge. A graph which has edge orientation is called *directed graph* or *digraph* and another one with no orientation is called *undirected graph*.

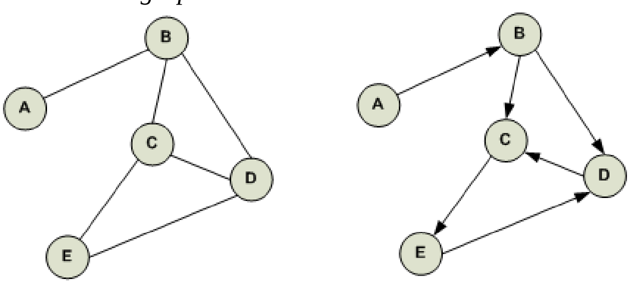



Fig 2.2 Graph illustrations based on the edge orientation, left side is undirected graph and right side is digraph
Source : http://www.codediesel.com/wp-content/uploads/2012/02/d-graph1.gif

### B. Specific Graphs

Some simple graphs have their own characteristics which distinguish them from the others so they are called a *specific graph*. There are four important graphs which are mainly used in computer science :

1. *Complete Graph*
   A complete graph with n vertices, denoted by $K_n$, is a simple graph which contains exactly one edge between each pair of distinct vertices.

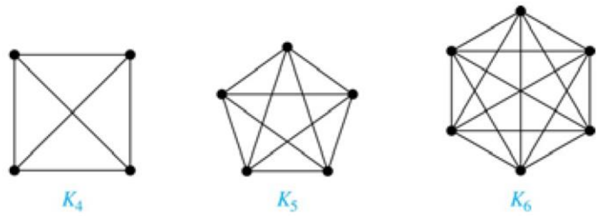



Fig 2.3 Some complete graphs : $K_4$, $K_5$, $K_6$
Source :
http://people.cs.pitt.edu/~milos/courses/cs441/lectures/ Class25.pdf

2. *Cycle Graph*
   A cycle with n vertices, denoted by $C_n$, for n bigger than two, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.

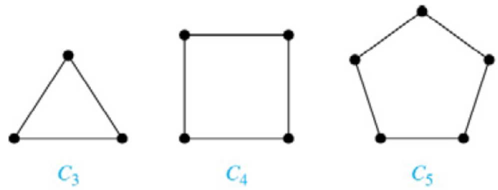



Fig 2.4 Some cycle graphs : $C_3$, $C_4$, $C_5$
Source :
http://people.cs.pitt.edu/~milos/courses/cs441/lectures/ Class25.pdf

3. *Regular Graphs*
   Each vertex of regular graph has the same degree. Note that complete graph and cycle graph are also regular graphs because each vertex's degree in complete graph is n-1 and each vertex's degree in cycle is two.

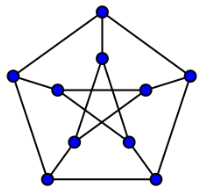


Fig 2.5 Regular graph which degree is three
Source :
https://upload.wikimedia.org/wikipedia/commons/thumb/9/91/Petersen1_tiny.svg/1024px-Petersen1_tiny.svg.png

4. *Bipartite Graph*
   A simple graph G is *bipartite* if G can be partitioned into two disjoint subsets $G_1$ and $G_2$ such that every edge connects a vertex in $G_1$ and a vertex in $G_2$. In other words, there are no edges

which connect two vertices in $G_1$ or in $G_2$. If all vertices in $G_1$ are adjacent to all vertices $G_2$, then the graph is called *complete bipartite graph*. Bipartite graph is denoted by $K_{m,n}$ where m is the number of vertices in $G_1$ and n is the number of vertices in $G_2$ or vice versa.

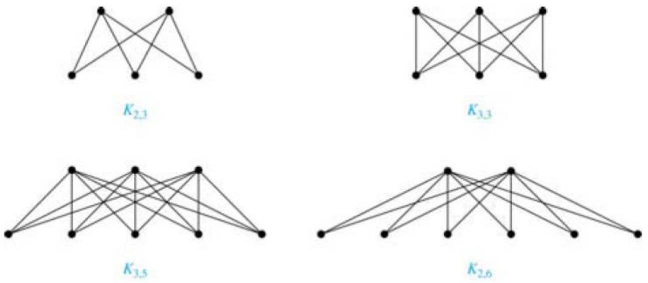


Fig 2.6 Four complete bipartite graphs
Source :
http://people.cs.pitt.edu/~milos/courses/cs441/lectures/Class25.pdf

The last specific graph, bipartite graph, will be mainly used to explain most of the contents in this paper. That is because the bipartite graph is a graph which underlies the process of object recognition.

### C. Bipartite Matching

There are two problems related to bipartite matching that will be covered here, which are maximum cardinality and minimum weight perfect matching problem. They are both used later in either SMGBM or SMGAT algorithm method.

1. Maximum cardinality matching problem

   A maximum matching (also known as maximum-cardinality matching) is a matching that contains the largest possible number of edges. There may be many maximum matchings.

   The matching number v(G) of a graph G is the size of a maximum matching. Note that every maximum matching is maximal, but not every maximal matching is a maximum matching. The following figure shows examples of maximum matchings in the same three graphs.

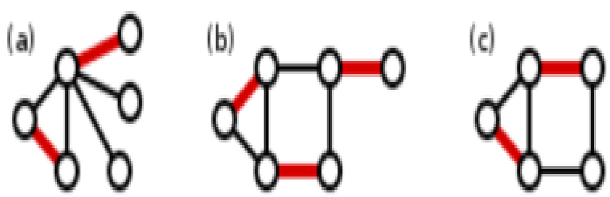



Fig 2.7 Maximum cardinality matching between three graphs
Source :
https://en.wikipedia.org/wiki/Matching_(graph_theory)

2. Minimum weight matching problem

   Formally, an edge cover of a graph *G* is a set of edges *C* such that each vertex in *G* is incident with at least one edge in *C*. The set *C* is said to *cover* the vertices of *G*. The following figure shows examples of edge coverings in two graphs.

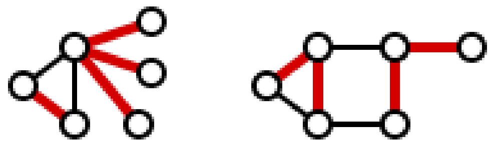


Fig 2.8 Edge covering illustrations between two graphs
Source :
https://en.wikipedia.org/wiki/Edge_cover

A minimum edge covering is an edge covering of smallest possible size. The edge covering number ρ́(G) is the size of a minimum edge covering. The following figure shows examples of minimum edge coverings.

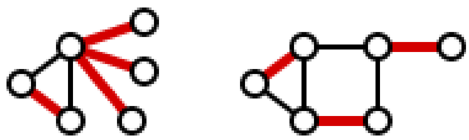


Fig 2.9 Minimum weight matching illustrations between two graphs
Source :
https://en.wikipedia.org/wiki/Edge_cover

Note that the figure on the right is not only an edge cover but also a matching. In particular, it is a perfect matching : a matching *M* in which each vertex is incident with exactly one edge in *M*. A perfect matching (if it exists) is always minimum edge covering.

## III. OBJECT RECOGNITION PROCESS

The whole explanations for this section (the whole explanations about object recognition process) are taken from the paper : A. Shokoufandeh and S. Dickinson, Applications of Bipartite Matching to Problems in Object recognition.

The explanations have been modified with my own writings as well as with other references. Some pictures and illustrations are taken from the paper and some other are taken from other references.

## A. Object Recognition Domains

Object recognition domains are used to represent image to a model which can be processed furthermore. That step is needed because a computer is different from a human. Computer cannot compare two object just based on its look. It needs a model that satisfies some terms so it can compare between objects based on their model. The representation or the model of an image that will be reviewed here is the *Saliency Map Graph*.

Saliency Map Graph is an image representation in multiscale view-based description of 3D objects. Each region maps to a node in a directed acyclic graph, in which an arc is directed from a coarser scale region to a finer scale region if the center of the finer scale's region falls within the interior of the coarser scale's region. The resulting hierarchical graph structure, called the saliency map graph (SMG), encodes both the topological and geometrical information found in the saliency map. Details of the representation, including its computation and invariance properties, can be found in **[3, 4, 6]**. Below are the illustrations of saliency map graph which correspond an image of a moneybox and a mug.

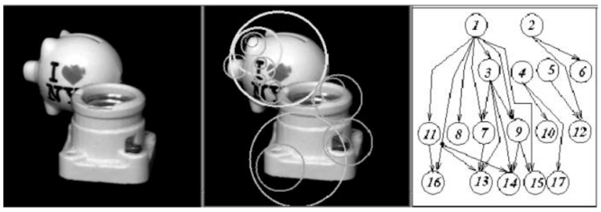


Fig 3.1 From left to right : original image, saliency map, saliency map graph
Source :
http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf

## B. Processing the Image Representation

Because the model which has been reviewed before is the saliency map graph, then this part will explained how an object can be recognized by matching two saliency map graphs. We will start by defining the problem formula. Then, based on the method, we differ two process of matching two SMG, which are matching algorithm based on topological similarity (SMGBM) and based on geometric similarity (SMGAT).

### B.1 Problem Formulation

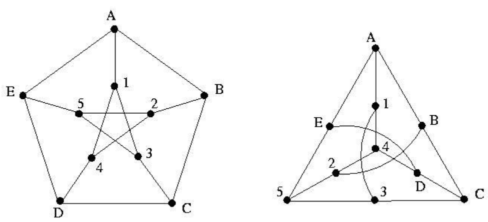


Fig 3.2 Two isomorphic graphs
Source :
https://www.sonoma.edu/users/f/fordb/M416S01/PS1Solutions/img6.jpg

Two graphs G = (V,E) and G' = (V',E') are said to be isomorphic if there exists a bijective mapping f : V -> V' satisfying, for all x, y € V (x, y) € E <-> (f(x), f(y)) € E'. To compute the similarity of two SMG's, we consider a generalization of the graph isomorphism problem, which we will call the SMG *similarity problem*.

### B.2 Saliency Map Graph Bipartite Matching (known as SMGBM)

In this section, we describe an algorithm which finds an approximate solution to the SMG similarity problem. The focus of the algorithm is to find a minimum weight matching between vertices of G1 and G2 which lie in the same level. We compare only the topological or structural similarity of the graphs, a weaker distance measure designed to support limited object deformation invariance.

Our algorithm starts with the vertices at level 1. Let A1 and B1 be the set of vertices at level 1 in G1 and G2, respectively. We construct a complete weighted bipartite graph G(A1,B1,E) with a weight function defined for edge (u, v) (u € A1 and v € B1) as w(u, v) = |s(v) − s(u)|. Next, we find a maximum cardinality, minimum weight matching $M_1$ in G using **[1]**. All the matched vertices are mapped to each other. We define f(x) = y if (x, y) is a matching edge in $M_1$.

The remainder of the algorithm proceeds in phases as follows, as shown in figure 3.3. In phase i, the algorithm considers the vertices of level i. Let $A_i$ and $B_i$ be the set of vertices of level i in G1 and G2, respectively. Construct a weighted bipartite graph G($A_i$, $B_i$, E) as follows : (v, u) is an edge of G if either of the following is true :

1. Both u and v do not have any parent in $G_1$ and $G_2$, respectively, or
2. They have at least one matched parent of depth less than i, that is, there is a parent $p_u$ of u and $p_v$ of v such that ($p_u$, $p_v$) € $M_j$ for some j < i.

We define the weight of the edge (u, v) to be |s(u)−s(v)|. The algorithm finds a maximum cardinality, minimum weight matching in G and proceeds to the next phase.

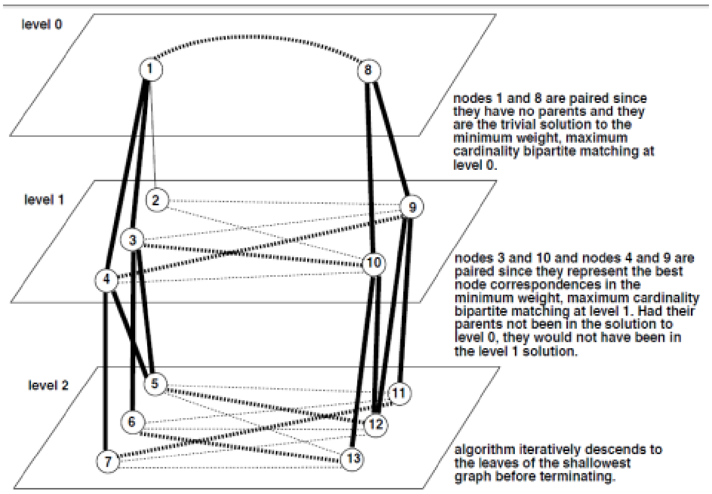



Fig 3.3 SMGBM algorithm illustrations
Source :
http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf

B.3 Saliency Map Graph Affine Transformation (known as SMGAT)

In this section, we describe a second similarity measure, called SMG Similarity using an Affine Transformation (SMGAT), which includes the geometric properties (e.g., relative position and orientation) of the saliency regions. The SMGBM similarity measure captured the structural similarity between two SMG's in terms of branching factor and node saliency similarity. We take advantage of the geometrical information encoded in an SMG and strengthen the similarity measure to ensure geometric consistency, a stronger distance measure designed to support subclass or instance matching.

The affine transformation (A, b) will be applied to all regions in $G_1$ to form a new graph G'. Next, a procedure similar to the minimum weight matching, used in the SMGBM is applied to the regions in graphs G' and $G_2$. Instead of matching regions which have maximum similarity in terms of saliency, we match regions which have minimum Euclidean distance from each other. Given two regions u and v, the distance between them can be defined as the $L_2$ norm of the distance between their centers, denoted by

$$d(u, v) = ( (x_u - x_v)^2 + (y_u - y_v)^2 )^{1/2}$$

In a series of steps, SMGAT constructs weighted bipartite graphs $G_i = (R_i, R'_i, E_i)$ for each level i of the two SMG's, where $R_i$ and $R'_i$ represent the set of vertices of G' and $G_2$ at the i-th level, respectively. The constraints for having an edge in $E_i$ are the same as SMGBM : (u, v) is an edge in $G_i$ if either of the followings holds:

1. Both u and v do not have any parents in G' and $G_2$, respectively.
2. They have at least one matched parent of depth less than i.

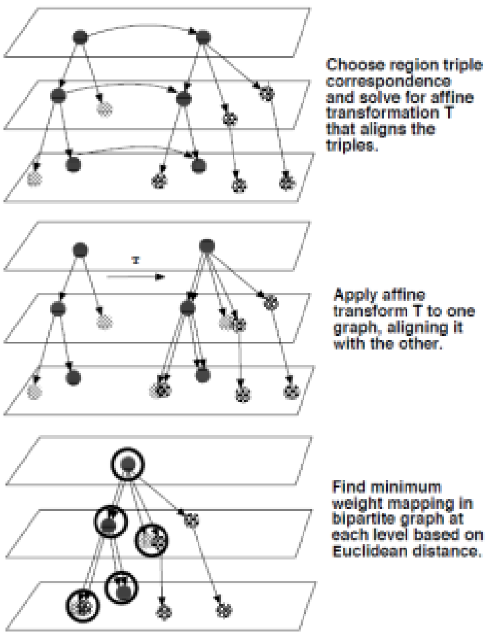



Fig 3.4 SMGAT algorithm illustrations
Source :
http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf

# IV. OBJECT RECOGNITION IMPLEMENTATIONS

## A. Implementations for Face Detection

### A.1 Face Detection in Brief

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process.

A reliable face-detection approach is based on the genetic algorithm and the eigen-face technique. Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

Some possible applications from face detection technology are as below :

1. *Facial recognition*, face detection is used in biometrics, often as a part of (or together with) a facial recognition system. It is also used in video surveillance, human computer interface and image database management.
2. *Photography*, some recents digital cameras use face detection for autofocus. Face detection is also useful for selecting regions of interest in photo slideshows that use a pan-and-scale Ken Burns effect.
3. *Marketing*, face detection is gaining the interest of marketers. A webcam can be integrated into a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements can be played that is specific toward the detected race/gender/age.

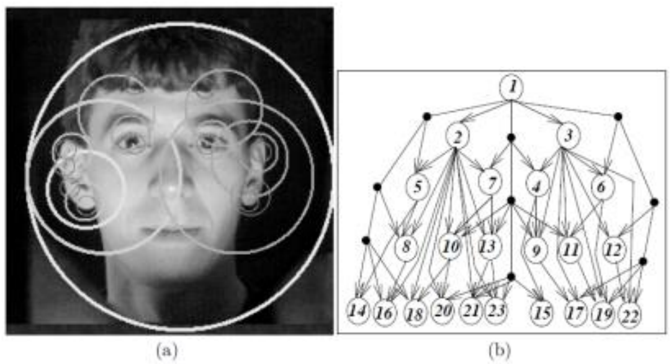



Fig 4.1 Face detection process : modeling someone's face to its corresponding saliency map graph
Source :

http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf

### A.2 Face Detection Product Example

One of my favorite products which implements face detection technology is coming from Intel Corp. That would be their new product, Intel RealSense.

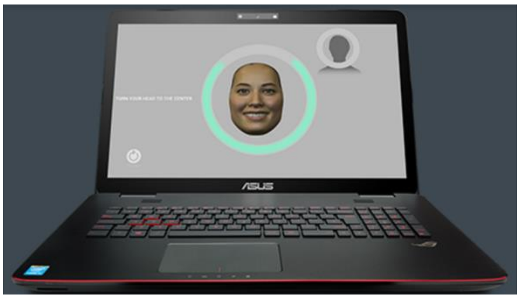



Fig 4.2 A laptop equipped with Intel RealSense.
You can even make your face as a password !
Source :
http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-shortrange.html

Your World in 3D. The revolutionary Intel RealSense Camera uses depth-sensing technology so your PC sees more like you do. Use the added dimension to scan 3D objects, control your PC with gestures, or create a more lifelike video chat environment. Reinvented video chat lets you customize your background, and share content with friends as if you're in the same room together. Below I listed cool features mentioned before from Intel RealSense :

1. Make Your Chat Space Whatever It Needs To Be. Intel RealSense Camera senses depth, so you're able to remove your chat background altogether, or swap in a replacement.
2. Control Your Device With a Wave. Intel RealSense Camera lets you interact with people-with natural movements. There are twenty two tracking points per hand.
3. 3D Scanning at Home. Intel RealSense technology makes 3D scanning a reality right from your device, so you can save a piece of art a flower, a toy-even your own face. You're able to save your scan as-is, or manipulate it into something new. Then, share it digitally, or print a version with the use of a 3D printer.

To explore more about Intel RealSense, just visit this link below : http://www.intel.com/content/www/us/en/architecture

## B. Implementations for Augmented Reality

### B.1 Augmented Reality in Brief

Augmented reality (AR for short) is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or even GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even be diminished rather than be augmented) by a computer. As a result, the technology functions by enhancing one's current perception of reality. By contrast, virtual reality replaces the real world with a simulated one.

Augmentation is conventionally in real-time and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and can digitally be manipulated. Artificial information about the environment and its objects can be overlaid on the real world.

### B.2 The Difference between Augmented Reality and Virtual Reality

How does augmented reality work? Is it similar to virtual reality? It is similar in that the user views a series of images via a pair of 3D glasses or a head mounted display (HMD). Augmented reality glasses are worn in the same way as virtual reality glasses in that they enable the wearer to interact with these images as part of the overall experience.

The difference between augmented reality and virtual reality is the level of immersion. Virtual reality is based upon a complete simulation of a real world environment which the user can explore and interact with by means of a head mounted display (HMD) and input device, e.g. data glove. The user loses or immerses themselves in this environment.

But with augmented reality the user sees the real world but with the addition of computer generated images which are overlaid on various objects within the real world. They are still aware that they are in the real world as compared to the full immersion in a virtual world. They use a device such as a smartphone or a wearable device - complete with a webcam - which contains software that recognizes an image and helps displays this onto an object.

Virtual reality replaces the real with the artificial whereas augmented reality enhances real life with artificial images.

Augmented reality is available via mobile phones in particular smartphones such as the iPhone. These phones have GPRS which obtains information about a particular geographical location which can be overlaid with tags etc. Images, videos etc. can be superimposed onto this location.

Handheld devices such as smartphones and the iPad are other ways to use augmented reality. They contain software, sensors, a compass and small digital projectors which display images onto real world objects. Another option is a head mounted display (HMD) which is often used in virtual reality applications.

An important issue is that of successfully integrating computer generated images within the real world. They need to be realistic and useful to be of real benefit to the user.

## B.3 Augmented Reality Product Example

To make a clear explanation, it's best to show you a product which takes benefit of augmented reality technology. That product would be a product which comes from my own institution, Bandung Institute of Technology. The product's name is "Mata-mata Pasar Seni".

Basicly, this product is created to support a big event held in ITB (short for Bandung Institute of Technology) and organized by ITB's art students. The event's name is "Pasar Seni ITB 2014" which was held last year in 23 November 2014. It is about showing all the masterpieces from many artists in Bandung as well showing the students' own work and people who attend the event could buy their own choice of art.

To make it even more rousing, some informatics students from HMIF ITB (Informatics Student Association) initiate to help them by creating a mobile phone application which is based on augmented reality that can run both on Android and iOS platform.

The application is designed to synchronize with mobile phone camera that makes it acts like the 'eye' of the devices. Before the event starts, hundreds of special marks were spread all over the campus. Each mark has its own meaning and located in a unique spot.

The idea is whenever people scan that mark with the 'eye', they would see some creatures on their phone moving around the spot which create augmented reality. Each spot has different creature which characterize them.

So, this is the explanation of how it works. First, the 'eye' or the camera scans the object in front of it. Then, the application would recognize the object as a specific mark. Then, based on the mark identity, the application would bring out the user an animation that is blended with the real image of the spot (blended with surroundings).

You can check out the application in Google Play Store for Android or App Store for iOS. Just search for this query : "Pasar Seni". You can also see the tutorial video for this application on this link : https://www.youtube.com/watch?v=asLaE2dat6w.

## B.4 Product Documentation

Here, in this section I would give you some image documentations of the use of the applications "Mata-mata Pasar Seni" in brief.

There would be description for every image so you can easily follow each step. Every image is taken from the tutorial video so you can alternatively see the tutorial videos in the link provided above.

I am hoping that this application could be developed furthermore for more uses, such as for education. Keep up the good work HMIF ITB!


Fig 4.1 Application interface with description when the apps first launch
Source :
https://www.youtube.com/watch?v=asLaE2dat6w

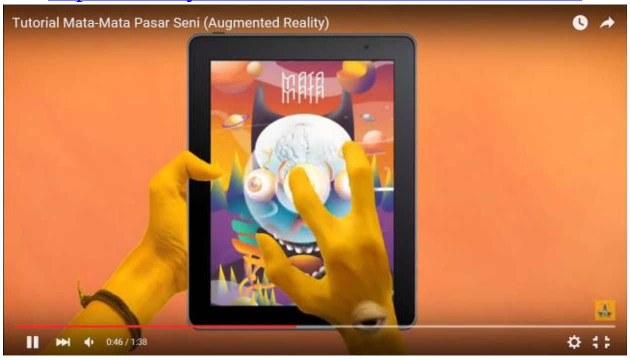



Fig 4.2 Tap the eye in the middle to launch the camera
Source :
https://www.youtube.com/watch?v=asLaE2dat6w


Fig 4.3 While in the camera mode, scan the specific mark. Make sure it is clear and big enough
Source :
https://www.youtube.com/watch?v=asLaE2dat6w


Fig 4.4 There you go. The hidden object comes out and flies around the spot. Tap the screen to adjust focus
Source :

https://www.youtube.com/watch?v=asLaE2dat6w

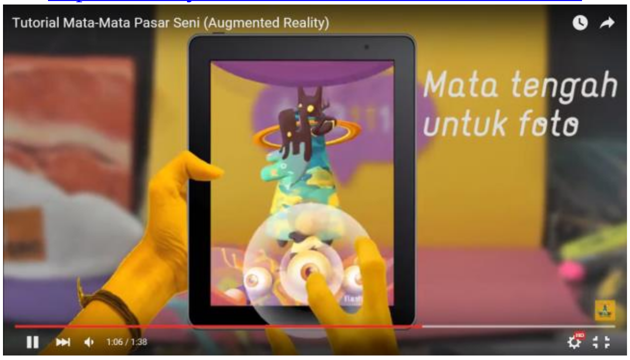



Fig 4.5 If you want to capture the moment as a photo, just tap the eye in the bottom of the screen
Source :
https://www.youtube.com/watch?v=asLaE2dat6w


Fig 4.6 You can go back to home screen and tap the cranium's nose in the middle to check the map to find another mark all over the place in ITB
Source :
https://www.youtube.com/watch?v=asLaE2dat6w

## V. CONCLUSION

In conclusion, graph is a beneficial data structure to use for object recognition as it models an object to a structure that is recognized by a computer. That is what makes graph a versatile tool for object recognition process as every different method of object recognition needs a sophisticated model to deal with.

On the other hand, graph still have a weakness in modeling a big object. Modeling big object means there would be big amount of nodes and edges which will consume memories because of large inputs. Furthermore, the process of matching two graphs would be taking a long time since the algorithm checks every node and edge for testing similarity of two saliency map graph. So, there should be an improvement or new method to handle big object in term of object recognition.

I think that in the near future there would be big and mass use of object recognition for artificial intelligence, especially for robots. To support its function, a robot needs an ability of object recognition so it can deal with its surroundings. I believe it would be realized soon as there are many experts working at this field now.

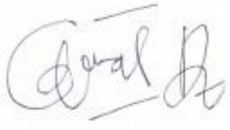
## REFERENCES


[1] E. Edmonds. Paths, trees, and flowers. Canadian Journal of Mathematics, 17:449–467, 1965.
[2] A. Shokoufandeh, I. Marsic, and S. Dickinson. Saliency regions as a basis for object recognition. In Third International Workshop on Visual Form, Capri, Italy, May 1997.
[3] A. Shokoufandeh, I. Marsic, and S. Dickinson. View-based object matching. In *Proceedings, IEEE International Conference on Computer Vision*, pages 588–595, Bombay, January 1998.
[4] A. Shokoufandeh, I. Marsic, and S. Dickinson. View-based object recognition using saliency maps. *Image and Vision Computing*, 17:445–460, 1999.
[5] Munir, R. *Matematika Diskrit*. Bandung: Informatika Bandung,
[6] http://people.cs.pitt.edu/~milos/courses/cs441/lectures/Class25.pdf Accessed on December 5th 2015 at 14.00
[7] https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/matching.pdf Accessed on December 5th 2015 at 19.00
[8] http://www.academia.edu/2797986/View-based_object_matching Accessed on December 5th 2015 at 19.00
[9] http://math.mit.edu/~goemans/18433S07/matching-notes.pdf Accessed on December 5th 2015 at 19.00
[10] www.cs.toronto.edu/~sven/Papers/ivc99.pdf Accessed on December 6th 2015 at 09.00
[11] http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf Accessed on December 6th 2015 at 09.00
[12] http://www.mashable.com/category/augmented-reality Accessed on December 6th 2015 at 11.00
[13] http://www.vrs.org.uk/augmented-reality/how-it-works.html Accessed on December 6th 2015 at 11.00
[14] http://computer.howstuffworks.com/augmented-reality1.htm Accessed on December 6th 2015 at 11.00
[15] http://www.augmentedev.com/how-augmented-reality-works/ Accessed on December 6th 2015 at 11.00
[16] https://www.youtube.com/watch?v=asLaE2dat6w Accessed on December 8th 2015 at 10.00
[17] http://www.intel.com/content/www/us/en/architecture-and technology/realsense-shortrange.html Accessed on December 10th 2015 at 12.00
[18] http://www.techradar.com/news/software/applications/how-face-detection-works-703173 Accessed on December 10th 2015 at 12.00
[19] http://www.pcmag.com/encyclopedia/term/42969/face-recognition Accessed on December 10th 2015 at 14.00
[20] https://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group02.pdf Accessed on December 10th 2015 at 14.00


## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2015



Geraldi Dzakwan 13514065