Tomasz GRZEJSZCZAK
Politechnika Śląska, Instytut Automatyki

Semantic representation of Block World Environment: algorithm of scene reasoning from incomplete information.

***Abstract**. Aim of this paper is to show the way of reasoning basing on the incomplete information about the initial state. The article presents an algorithm created in order to reason the state of scene from block world basing on incomplete information from two cameras observing the scene from top and side. The algorithm is explained using an example. Additionally, possible types of uncertainties are presented.*

***Streszczenie.** Artykuł przedstawia sposób wnioskowania wiedzy bazujący na niepełnych informacjach o stanie początkowym na przykładzie świata klocków. W artykule znajduje się opis algorytmu stworzonego w celu wywnioskowania stanu sceny ze świata klocków w oparciu o niepełne informacje pochodzące z dwóch kamer obserwujących scenę z góry i boku. Algorytm zilustrowany jest przykładem. Dodatkowo opisane są możliwe typy niepewności związane z problemem. (**Semantyczna reprezentacja Świata Klocków: algorytm wnioskowania w oparciu o niepełną informację**).*

Słowa kluczowe: Świat klocków, algorytm wnioskujący, wykrywanie obiektów, semantyczna reprezentacja, wnioskowanie.
Keywords: Block World, reasoning algorithm, objects detection, semantic representation, reasoning.

Introduction

The presented algorithm was created in order to recognize the scene from the Block World. In typical Elementary Block World problem there is a finite number of blocks, each on the working space. Each block can be either on the table or on another block. The whole scene contains towers of blocks placed on the table [1]. In given task there were a number of blocks on the scene that could be on the table in form of a line or on each other. Each block was individual and could be distinguished by color. There were two manipulators that were moving the blocks. They were operating on the blocks from left and right.

In order to recognize the blocks, two cameras were placed from top and side. This was the only possibility of cameras placement, because in that case a white background could be applied, which improves the quality of recognition. Each of those cameras was able to recognize the blocks that was seen either from top or side of blocks piles. In that case some of the blocks was not seen and needed to be concluded.

The main aim of the presented algorithm is to process the knowledge obtained from side and top recognition and conclude the whole scene construction in form of a matrix of data containing the number of block.

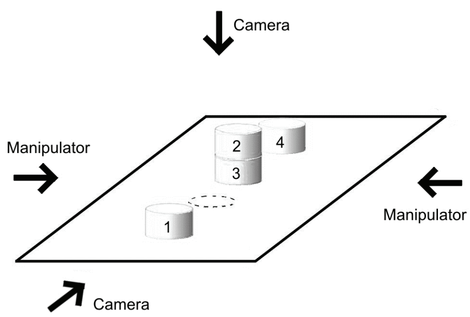

Fig.1. Scene construction

The algorithm is working by means of semantic representation. The term is used when one tries to write down the rules describing a way of obtaining conclusions from input data [2]. In application to Block World and scene recognition, the semantic representation can be defined as the way of transforming the scene into computer data. The input of semantics are two frames from cameras and output is an array of blocks positions. In this case semantic representation is an algorithm in form of rules, that can recognize the scene with a given level of uncertainty.

Preparation of data necessary for proper working of recognition algorithm

The main information that is obtained is a identification number of block. The goal is to create a matrix of all positions and fill it with the number of block that is in a given position.

Two cameras can observe the scene from top and side and can create a vector of seen block. First camera can produce a vector of all blocks seen from top, that is all top blocks from each pile of blocks and also can determine there are those piles. The second can observe the scene from side, calculating the vector of blocks seen from side of some piles. This information can determine the count of blocks from the highest pile.

The scene should be prepared in the simplest way. A block world environment has been chosen because it is a simplified world where a blocks have simple shapes and a background is uniform. During the detection, algorithm searches for circles and rectangles, which are a 2d projections of 3d cylinder blocks [3]. Such simplification results in better recognition and is simpler to construct leaving the opportunity to focus on semantic representation.

Ways of obtaining the top/side vector

In order to obtain the vectors of information a vision system needs to be constructed. In the simplest case the blocks can differ by color and the system can work on HSV color space.

HSV is an model based on human eye perception. It contains of three channels: hue, saturation, and value, also called brightness. Hue is a color indicated as an angle. Colors can have different saturation. Saturation is indicated by the second component. For example pure color have maximal saturation. Lower the saturation, color becomes more faded. If the brightness is changed, denoted as value, the color becomes more dark. Value can be treated as gray scale image recognizable for human [4].

Using this color model and saturation channel, color blocks can be distinguished from low saturated regions, such as background or shadows. Moreover, changing the external lightning has an influence on brightness and saturation leaving small influence on hue channel.

The algorithm for blocks detection can be proposed in such way:

a) Detect blocks from top camera.
1. Change frame1 from camera into HSV image.
2. Threshold image and create mask of high saturation region with specified block color.
3. Perform morphological functions in order to improve quality of mask.
4. Find position of block pointed by mask.
5. If the detected object is too small, that means it is not seen from top, thus it has not been detected.
6. Find the closest column of blocks for this block to assign it and write its number into vector of data.
7. Repeat steps 2-6 for all blocks.
8. For all detected blocks find their positions and create an output vector with detected blocks. Point detected errors.

b) Detect blocks from side camera.
1. Change frame2 from camera into HSV image.
2. Create mask containing highly saturated regions.
3. Calculate color histogram of image with mask and find the most popular color.
4. Create additional mask with this color, and perform morphological functions.
5. Detect this color position.
6. subtract detected color, by subtracting both masks.
7. Using new mask perform steps 3-6 until detected color would have too small area.
8. For all blocks sort them that the lowest placed block is on the bottom, and next blocks are on it. This would create side vector.

It is also assumed that all colors of blocks are known and the algorithm only looks for those blocks that are declared.

Main reasoning algorithm used to recognize the scene.

If the both vectors are calculated properly, the next step is to pass them to reasoning algorithm, recognizing the state of the matrix.

The size of the matrix is determined by the number of blocks. If a scene contains of n blocks, there are two extremal cases. Each block is on the table, creating n long top vector, or each block is in one pile, creating n long side vector. This indicates that the vectors needs to be n long and a matrix needs to be n x n. In other cases simply most of the records in matrix would be empty, however it needs to be prepared to hold the extremal cases.

The algorithm of reasoning the final state matrix from vectors is:
1. Assign the blocks detected on side camera. First block from side vector should be also detected on top camera. If it has been detected, then its position is known. All positions above it and in front of are for sure empty. All positions below it and after are unknown.
2. If next block has not been detected on top camera, it is below previous. If it has been detected, it is assigned to new position. Under the previous block there are some unknown blocks, because they can not be hold in air.
3. Repeat step 2 until the end of side vector.
4. Put rest of blocks from top camera on the scene. Calculate the number of detected blocks, and calculate how many blocks has not been detected.
5. Put the undetected blocks in left positions.

In order to fully understand the algorithm it is advised to read the example.

Example

In order to illustrate the algorithm an example has been created. This example shows the case with 6 blocks.

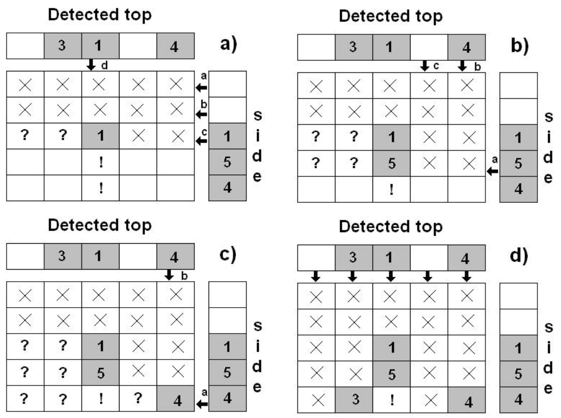

Fig. 2. Example of scene recognition.

First of all it is important to notice that scene consists of 6 blocks, while blocks 2 and 6 has not been detected either on side nor top camera. Those blocks position needs to be concluded.

a) The algorithm analyzes the side vector from top to bottom. The first two empty values indicates that there are no blocks on this level. First found block is number 1. Top vector should also consist the number 1. The position of the highest block has been found. All positions from the left of this blocks are unknown, all positions below for sure consist a block.
b) Next block from side vector is block 5. This time it does not occur in top vector. This means that in needs to be under block 1.
c) Last block is number 4. It occurs in top vector. This means that we have no more information about previous pile, and another lower pile has been detected. In this case the new pile consist of only one block with number 4. All blocks on the left of it are unknown.
d) The bottom of the side vector has been reached. Now the top vector is being analyzed. All empty records means that there is no piles in this position and the used information about block 1 and 4 remains unchanged. The only new information is that there is a pile with block 3 on top. This block is put on the table for a moment.

Until this moment all known information has been processed. 4 blocks has been detected and there are two more. There is one sure position under block 5. This remains the only possibility that one of unknown blocks is under block 3, causing the unknown pile to be two blocks high.

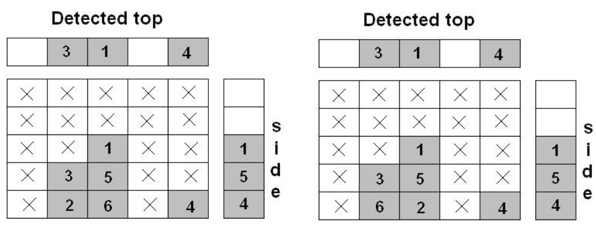

Fig. 3. Two possible results of the recognition.

It is impossible to determine which block from those unknown are on which position. This cause the two possible results. In order to determine the correct result, one block needs to be removed and the algorithm needs to be done once again searching for 5 blocks.

2: Types of uncertainty during scene recognition process.

The problem of the scene construction is that vision system does not have full knowledge about the environment. In this case system need to work under uncertainty. This is the case where uncertainty is caused by the incompleteness of reasoning. On the other hand uncertainty can be also caused by the errors, while obtaining facts. If the knowledge is mistaken, the conclusions would also be wrong.

Uncertainty is a widely handled problem in artificial intelligence. The uncertainty occurs when a obtained information differs in some percentage from the actual real state or can not be concluded at all [5].

Uncertainty caused by the recognition errors in vision system

Dependent on construction of vision system, the errors in recognition process can vary. Changing the external lightning of the scene can influence the detected color of the blocks, causing wrong block identification.

Let us assume that vision system would like to find a blue block denoted as 100 hue value. During the tests, it has been observed that different external lightning can slightly change this value. Moreover, each blocks produces a shadow which can also disturb the results. All those parameters cause that this block contain pixels in range about 96-104. This leads to following problem. Let us assume that we have two blocks: blue (110) and cyan (90). Due to influence of previously described phenomenons, a block with dominant value 100 has been detected. The problem is: is it blue or is it cyan.

This problem can not be solved. The only possibility is to reduce the error influences or calibrate the vision system. However, in less extremal cases, the detected pixels should be assigned to the closest defined color.

The kind of errors described in this chapter are very dangerous and unacceptable. They can cause an invalid recognition. If the blocks would be detected incorrectly, the vector with top and side blocks would be also wrong, causing recognition algorithm to operate on wrong input data. This would cause the wrong interpretation of the final scene.

Uncertainty caused by the incompleteness of information.

As it was described in previous chapters, the scene observed from side and top leaves some blocks that can not be seen. This type of uncertainty occurs only in some special cases. If there are only up to two blocks, there is no uncertainty. If there are more blocks, the uncertainty is more and more often and complicated. There are two types of this uncertainty. The case when, it is sure that block is in certain position, but system do not know which color, and a case when system knows that there is one block that was not detected and it do not know where this block is.

Following examples shows the types of uncertainty for growing number of blocks.

The situation with one block is too simple to consider. If there are two blocks, no matter of combination, they would always be detected with no uncertainty (See Fig.4).

If there are three blocks, there can be one situation where an uncertainty occurs. This is the case when it is sure that a block is in this certain place, but system do not know which block. If system has all information about blocks, knowing that two blocks has been placed, the missing third block can be placed only in one position. All other cases with three blocks are recognized without uncertainty.

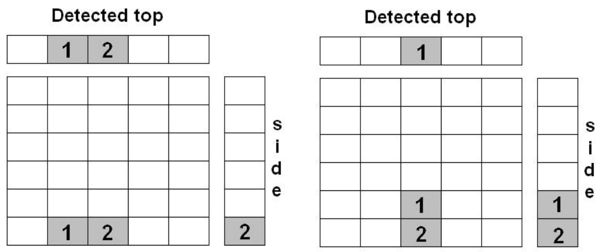

Fig.4. Simple cases of two blocks. No way of uncertainty.

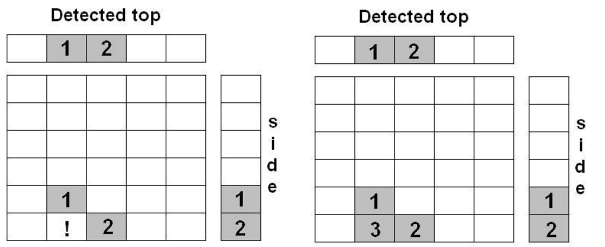

Fig.5. The case of two blocks detected in three blocks scene.

When the recognition state is as shown on Fig. 6, it can be recognized in two ways. The first recognition is correct if scene contains three blocks. This is the case without uncertainty. However if system knows that it should look for four blocks, it finds the place where the block can be placed. In case of four blocks, all detected uncertainty can be recognized completely. This example shows that it is crucial to know how many blocks there are on the scene in order to properly recognize it.

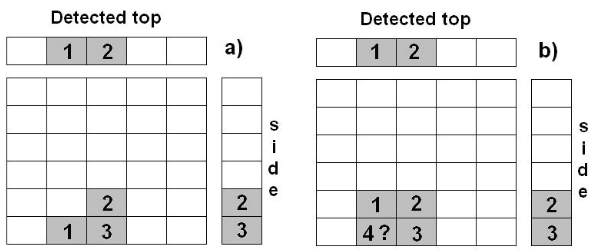

Fig.6. Recognition of scene (a)with three blocks, (b) with four blocks.

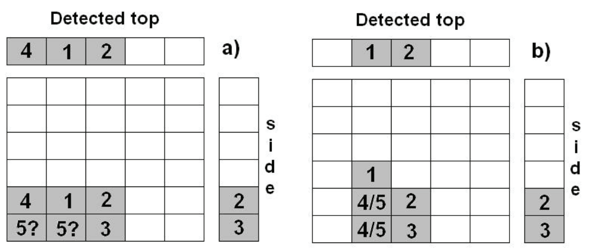

Fig.7. Recognition with unknown (a) where to place fifth block, (b) which blocks put on detected positions.

In case of five blocks the uncertainty level becomes more complicated. Two more uncertainties are introduced on Fig. 7. First case shows the situation when system do not know where to put remaining fifth block. There are two possibilities, and no way of determining the true solution. In second situation, it has been detected that two positions contains unseen blocks. In this case both blocks can be on either positions. Those cases shows the incomplete recognition. In both cases all recognition possibilities are passed to final recognition state.

For more blocks, uncertainties became more complicated. They can mix and grow.

Conclusions

Presented algorithm works correctly if all information from vision system is with no errors. For small number of blocks the output data can be useful, however if the number of blocks grow, the uncertainty in detection becomes more frequent, giving more uncertainty then facts on the output.

The system would work if a number of assumptions are fulfilled leaving the small universality and small appliance to real world. However the algorithm found the appliance in the laboratory stand illustrating the simple artificial intelligence planing problem, cooperating with created planing and learning program that controls manipulators.

REFERENCES

[1] J. Slaney, S. Thiébaux, Blocks World Revisited, *Artificial Intelligence,* 125 (2001) 119–153, 2001.

[2] J. Groenedijk, T. Janssen, M. Stokhof, *Truth, Interpretation and Information*, Foris Publications Holand, 1984.

[3] J. L. Mundy, *Object Recognition in the Geometric Era: a Retrospective*, Division of Engineering, Brown University Providence, Rhode Island, 2006

[4] A. Bovic, *Handbook of Image and Video Processing Second Edition*, USA 2005.

[5] S.J. Russell, P. Norvig, *Artificial Intelligence, A Modern Approach*, Prentice-Hall, New Jersey, USA, 1995.

***Autor**: mgr inż. Tomasz Grzejszczak, Politechnika Śląska, Instytut Automatyki, ul. Akademicka 16, 44-100 Gliwice, E-mail: grzejszczak.tomasz@gmail.com.*