MINISTRY OF EDUCATION AND TRAINING

VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

GRADUATE UNIVERSITY SCIENCE AND TECHNOLOGY

Nghiem Van Hung

RESEARCH ON COLLISION DETECTION TECHNIQUES IN DEFORMABLE OBJECTS AND COBOT ARM

SUMMARY OF DISSERTATION ON COMPUTER

Major: Information system

Code: 9 48 01 04

Ha noi - 2024

The dissertation is completed at Graduate University of Science and Technology, Vietnam Academy of Science and Technology

Supervisor: Assoc. Prof. Dr. Dang Van Duc, Institute of Information Technology, Vietnam Academy of Science and Technology

Referee 1: …………………………………………………………..

Referee 2: …………………………………………………………..

Referee 3: …………………………………………………………..

The dissertation is examined by Examination Board of Graduate University of Science and Technology, Vietnam Academy of Science and Technology at……………………….. (time, date…)

The dissertation can be found at:

1. Graduate University of Science and Technology Library
2. National Library of Vietnam

INTRODUCTION

1. The urgency of the topic

The Fourth Industrial Revolution with its breakthrough advances in science and technology has created momentum to promote many fields of research and development, of which Virtual Reality (VR) and Robotics are typical. These are two of the key technologies that are prioritized for research, development and application.

Collision Detection (CD) is one of the basic tasks of VR simulation systems, computer graphics, robotics control, games, etc. Objects in each system have their own movements, which can collide with other objects, or with the environment and obstacles.

There have been many publications on collision detection techniques in the world, most of which are based on the Bounding Volume Hierarchies (BVH) or on the Signed Distance Function (SDF) calculation technique. For deformable object models (such as cloth material), the processes of browsing, restructuring BVH and calculating SDF require a lot of system resources, causing a bottleneck. The problem is to design an algorithm that can work well on different object models.

In Vietnam, VR research is mainly in modeling or simulating virtual displays, but there is not much research on the problem of detecting collisions in deformable objects and cobot arms. Research on collision detection methods for deformable object models such as cloth material models is not yet popular, and research on improved collision detection techniques for solid object models still has issues that need to be further studied.

VR technology and Robotics have a special mutual relationship. The result of this mutual relationship is the emergence of advanced scientific and technological achievements. In the field of robotics, Collaborative robots (Cobots) are gradually replacing traditional robots. Collision detection is one of the mandatory requirements for cobots. Therefore, in recent years, research on robotics has gradually shifted its focus to studying the problem of collision detection of cobot arms. In order to promote the effectiveness of collision detection techniques for VR and robotics applications, I found that the research and development of collision detection techniques is suitable for practical requirements. Therefore, I chose the topic "***Research on collision detection techniques in deformable objects and cobot arm***" as the topic of my dissertation.

2. The aim of the study

The aim of this dissertation is to propose a method to improve the efficiency of collision detection in deformable objects and 6 Degree of Freedom (DoF) cobot arms. The research results are used in VR simulation applications and the field of 6 DoF cobot arms control.

3. Subject and scope of research

To achieve the set objectives, the dissertation conducts the following research contents: VR technology and cobot arms; the problem of detecting collisions of 3D model objects (solid objects and cloth objects) in virtual environments; the problem of detecting collisions of 6 DoF cobot arms; some algorithms and methods of detecting collisions; proposing improvements to collision detection methods, testing and evaluating research results.

CHAPTER 1. OVERVIEW OF COLLISION DETECTION PROBLEM AND SOME BACKGROUND KNOWLEDGE

1.1. Introduction

The collision detection problem in virtual reality environment is stated as follows: Given n objects $\{O_0, O_1, \dots, O_{n-1}\}$, collision detection is the process of determining whether the objects intersect or not, that is, checking $O_i \cap O_j, \forall i \neq j,\ i = 0, \dots, n-1, j = 0, \dots, n-1$ and $n \in \mathbb{N}$. In which, $O_i = \{p_0^i, p_1^i, \dots, p_k^i\}$ is a set of k primitives and $O_j = \{p_0^j, p_1^j, \dots, p_m^j\}$ is a set of m primitives, the collision of O_i and O_j occurs when $p_a^i \cap p_b^j \neq \emptyset$ with $a = 0, \dots, k$, $b = 0, \dots, m$ and $k, m \in \mathbb{N}$.

The result of the collision detection problem is in the binary form Yes/No, if a collision occurs, it is necessary to calculate the contact points and the contact time.

1.2. Related studies

1.2.1. Collision detection of 3D object models

Each 3D object is made up of many faces, so the cost of collision detection is very high. Most virtual reality systems use collision detection methods based on bounding boxes. A bounding box is a closed geometric space that completely encloses an object. Figure 1.1 shows some types of bounding boxes used in collision detection techniques.

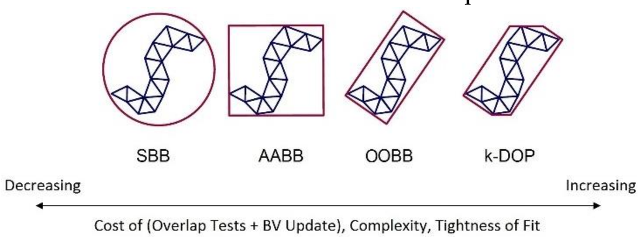

Figure 1.1. The bounding boxes commonly used in collision detection

Figure 1.1 describes some common types of bounding boxes used in collision detection techniques, in which the bounding boxes have increasing tightness from left to right, and decreasing computational complexity from right to left.

1.2.2. Collision detection of cobot arm

Contact with other objects is one of the required tasks of a cobot arm. Collision detection methods based on machine learning and sensors are developed to meet this need.

1.3. Some limitations of traditional methods

1.3.1. Limitations related to collision detection of solid object models

Two objects are considered to be in collision when their envelopes overlap. Collision detection is the process of projecting the envelopes of objects along the coordinate axes and determining whether these projections overlap. Because AABB envelopes are oriented along the axes of the spatial coordinate system, their projection onto the coordinate planes is faster than OOBB envelopes. However, OOBB envelopes give more realistic results because they more accurately simulate the volume of the object.

For two OOBBs, the situations in which they are in contact with each other can only be one of the following 6 cases: {Face - Face; Face - Edge; Face - Vertex; Edge - Edge; Edge - Vertex; Vertex - Vertex}. The SAT algorithm has some limitations in computational speed and is less efficient when there are many collisions to be detected.

1.3.2. Limitations related to collision detection of cloth models

In general, there are two approaches to speed up collision detection: the first is to use the subsystem structure of the dynamic region [6, 8] and the second is to use the filtering algorithm [15-18].

Curtis et al. [9] proposed to use representative triangles to eliminate duplicate checks, although it increases the efficiency of collision detection, but due to the randomness of its allocation algorithm, it is difficult to integrate with other algorithms. In collision events where the models have a high degree of deformation, the method using the filtering algorithm in the publication [15] and [16] is less effective because it cannot handle the case of the triangle mesh being flipped.

1.3.3. Limitations related to collision detection of cobot arms

Collision detection methods on cobot arms are classified into two groups: (1) methods using machine learning and (2) methods without machine learning. The latter group often relies on tactile surface sensors covering the outside of the cobot [30, 31]. However, tactile surface sensors are often expensive, not durable, and can be damaged when subjected to strong or repeated collisions. Machine learning-based methods [25, 27, 29] usually estimate the external joint torques due to collisions, and then compare them with a set of pre-determined threshold values to check whether a collision has occurred. Direct torque measurement would be ideal, but using torque sensors can be expensive, an alternative is to estimate joint torques from current measurements at the joint actuators.

1.4. Background knowledge

1.4.1. Virtual reality

1.4.2. 3D modeling

1.4.3. Deformable model

1.4.4. Descartes' sign rule and Vincent's theorem

Descartes' sign rule [16]: Let $f(t)$ be a polynomial in decreasing power of t. The transformation occurs whenever the non-contiguous groups are of different signs. The notation $VAR\big(f(t)\big)$ is the number of pairs of non-contiguous numbers of opposite signs.

Since we are only interested in the interval [0,1], the following theorem is a result of Descartes' sign rule when mapping t from $(0,\infty)$ to $(0,1)$ with the mapping $t \to \frac{1}{t+1}$.

Vincent's Theorem [17]: For a function $g(t) = (t+1)^n f\left(\frac{1}{t+1}\right)$, if $VAR\big(g(t)\big) = 0$, with n is the degree of $f(t)$ then $f(t)$ has no root on [0,1]. The number of positive real root of $g(t)$ is equal to the number of real root of $f(t)$ on [0,1].

1.4.5. Cobot and the structure of the cobot arm

The main parts of a cobot include: the base, the rotating or translational joints, the links between the joints and the cobot arms, also known as the *End of Effect*.

1.5. Evaluation index

1.5.1. Collision detection performance evaluation metrics

The metrics for evaluating collision detection performance are *Precision* and *Recall*, which are defined as follows:

$$Precision = \frac{TP}{TP+FP}$$

and

$$Recall = \frac{TP}{TP+FN}$$

where *TP* is the number of true positives, *FP* is the number of false positives and *FN* is the number of false negatives (collisions are positive).

1.5.2. Criteria for evaluating the types of bounding boxes

The choice of bounding boxes type is determined by many factors such as: the cost of constructing the bounding boxes, the cost of updating the bounding boxes when the object moves or changes shape/size, the cost of determining the collision point and the accuracy of the collision detection method, etc.

The PhD student chose to use the k-DOP bounding boxes to compromise between the evaluation criteria and facilitate the representation as well as the efficiency of calculating collision detection of cloth models.

1.6. Conclusion of Chapter 1

In Chapter 1, the PhD student presents a general overview of virtual reality technology and the problem of collision detection in virtual environments; an overview of cobots, collision detection of cobot arms and related studies. Research on techniques to increase the processing speed of virtual reality systems is very necessary, focusing on improving collision detection calculation techniques because it plays an important role in interactive simulation systems.

Based on the Chapter 1, in Chapter 2, the PhD student focuses on presenting solutions and proposing improvements to filtering algorithms in detecting collisions of interactive cloth materials in virtual reality environments.

Developing from the basic knowledge of Chapter 1, the PhD student presents in Chapter 3 solutions and proposing improvements to collision detection techniques of CURA6 cobot arms.

Part of the research results presented in chapter 1 have been published by the PhD student in [CT1, CT2, CT3] in the List of published works related to the dissertation.

CHAPTER 2. METHOD OF COLLISION DETECTION IN DEFORMABLE OBJECTS

2.1. Proposed method to improve the collision detection efficiency of cloth material model

2.1.1. Proposed method

PhD student inherits and develops on the basis of the filtering solution of Tang [10] published in 2009 and Zhang [18] published in 2015 because of its feasibility and suitability for cloth material objects. The collision detection process is divided into two phases: wide phase and narrow phase as illustrated in Figure 2.1.

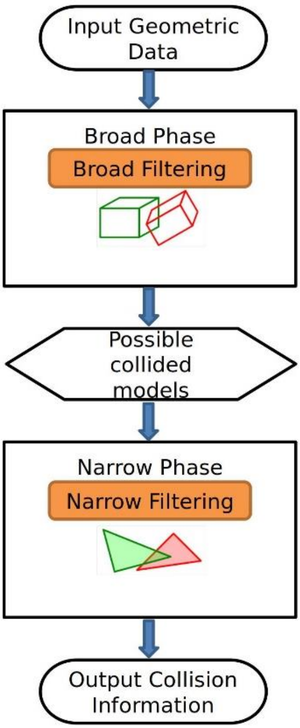

Figure 2.1. General diagram of the proposed algorithm

The wide phase algorithm is described as follows:

Algorithm1

Input: The 3D model is represented as a triangular mesh.

Output: Pairs of objects that are likely to collide.

1: Create BV;

2: Establish a top-down hierarchical structure for BVH;

3: Set up the information protection tree structure based on browsing the protection hierarchy structure;

4: Check for overlap, remove pairs of objects that are unlikely to collide with the leaf node of the information protection tree.

4.1: if there is no overlap, exit;

4.2: else (possible collision) then go to Algorithm2.

The narrow phase algorithm is described as follows:

Algorithm2

Input: Coordinates of the 04 vertices of the VF pair or EE pair that need to be checked for collision.

Output: Conclusion of collision (contact point information, normal vector of contact surface,...) or conclusion of no collision.

1: Check and perform filtering based on geometric characteristics (coplanarity of VF pair or EE pair);

2: Calculate VAR(f(x)) to analyze the existence in [0, 1];

3: Apply Descartes' sign rule and Vincent's theorem

3.1: if no solution exists, then eliminate and conclude that the VF pairs (or EE pairs) do not collide;

3.2: else (there exists a solution) apply Vincent's theorem and solve the equation to conclude the collision (contact point information, normal vector of the contact surface,...).

The narrow phase is calculated to accurately determine whether a collision will occur or not. The first thing is that the Algorithm2 algorithm checks the coplanarity of VF (or EE) pairs. Then apply algebraic methods to determine the existence or non-existence of solutions to the distance equation and draw corresponding conclusions.

2.1.2. Results and discussion

The testing environment is an Intel(R) Core(TM) i7-7820HQ @ 2.90 GHz CPU computer, using the C++ programming language on Windows operating system and test data sets from the UMD GAMMA Research Group of the University of Maryland and Walt Disney Company, USA.

Three datasets are used to evaluate the performance of the included algorithms:

(a) (b) (c)

Figure 2.2. Illustrations on the datasets in the GAMMA open library: (a) Princess dataset, (b) Flamenco dataset, (c) Cloth-ball dataset

The results show the effectiveness of the proposed method. The comparison shows that the Flamenco model has a high level of removal efficiency. Two-phase filtering algorithms can exert good filtering properties when the cloth model has a very large degree of deformation. The experimental results of the proposed algorithm are compared with the ICCD algorithm of Tang et al. [10], the DNF algorithm of Tang et al. [16], the SCD algorithm of Zhang et al. [17] and the VCD by Zhang et al [18] with collision detection time unit of ms.

The proposed method has two outstanding advantages:

The first, the time to perform the bounding overlap check and BVH traversal is significantly reduced for cloth models in the test dataset because of early removal of objects that are unlikely to collide.

The second is an effective two-phase filtering strategy that reduces false positives and speeds up collision detection.

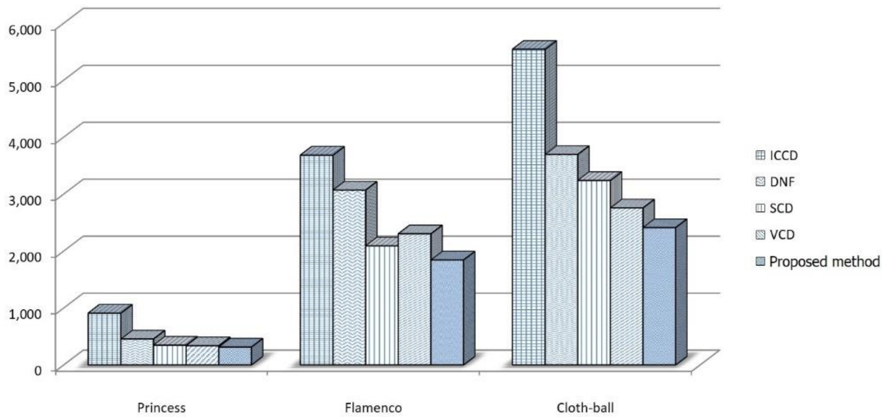

Figure 2.3. Results of executing the algorithms

The comparison chart in Figure 2.3 shows that the proposed method runs faster. The reason is that PhD student use a combination of both geometric and algebraic factors to eliminate VF pairs and EE pairs that do not collide, evaluate the existence of solutions to the distance equation, and save time. for equation solving operations, reducing the number of complex mathematical operations.

2.2. Proposed method to improve the collision detection efficiency of solid objects

2.2.1. Proposed method

The problem of checking whether two convex polygons intersect can be formulated as a linear programming problem (LP). Two convex polygons do not intersect if and only if there exists a separating plane between them. The coefficients of the separating plane equation are considered unknown. The linear constraints are formed by imposing that all vertices of the first polygon lie in one half of the space of this plane and the vertices of the second polygon lie in the other half of the space. Linear programming algorithms are used to check whether there is a feasible solution for the given set of constraints.

The description of the proposed algorithm is as follows:

```
Algorithm3
-----------------------------------------------------------------
Input: 03 parameters (P, Cl are the bounding block space
regions, T is the root node of the tree).

Output: List L contains information about collisions, if there
is no collision then L is empty.
-----------------------------------------------------------------
1: Traveral with the first left child node in the tree T;

2: Repeat with other nodes (call M the node under consideration)

2.1: Perform a quick check to see if the node under
consideration is in the space region Cl or not

2.2: if the node under consideration is not in Cl then return
false;

2.3: else (P and M are likely to collide)

     Perform the Test() procedure to return true if there is a
collision and store, return false if there is no collision;

     If a collision is detected, continue to check between the
child nodes of P and M;
```

```
While the nodes under consideration are leaf nodes, the algorithm has scanned to the two deepest nodes in the tree that collide with each other.
```

The collision test procedure Test() presented below will return true if there is a collision and return false if there is no collision.

The collision test procedure is described as follows:

Test

Input: 02 parameters (P, Q are the objects to be tested for collision).

Output: Return true if there is a collision, return false if there is no collision.

```
1: Initialize a variable to store the number of inequalities;
2: Get the basis of Q expressed in the basis of P;
3: Initialize the coefficients of the matrix representing the inequality system;
4: Remove the first variable in the original system;
5: Recurse with the next variables;
6: Convert the limit values in the basis of P and return the corresponding results.
```

2.2.2. Results and discussion

The first experiment

Experiment Scenario: The PhD student generated 1,000,000 random pairs of tetrahedrons/cubes, then performed collision detection on each pair using the SAT algorithm and the proposed algorithm. Finally, the PhD student calculated the ratio (execution time of the SAT algorithm)/(execution time of the proposed algorithm) as shown in Table 2.1.

Table 2.1. Comparison table of collision detection time ratios of algorithms in the first experiment

Context	**Cube – Cube**		**Cube – Tetrahedron**		**Tetrahedron – Tetrahedron**	
	Collision	**Non-collison**	**Collision**	**Non-collison**	**Collision**	**Non-collison**
Static	1.39	2.01	2.44	1.52	4.17	1.12
Dynamic	2.58	1.41	1.37	1.23	1.85	1.05

The ratios in Table 2.1 are always greater than 1, which means that the proposed algorithm is faster than the SAT algorithm.

The second experiment

Experimental Scenario: STUDENTS experimentally detected collisions of the rabbit model provided by the Computer Graphics Laboratory of Stanford University. This model consists of 69,451 triangles and has a file format of .ply, the type of bounding block used is OOBB.

PhD student performed collision detection on the pair of rabbit models using the SAT algorithm and the proposed algorithm. Then PhD student calculated the ratio (execution time of the SAT algorithm)/(execution time of the proposed algorithm) as shown in Table 2.2.

Table 2.2. Comparison table of collision detection time ratios of algorithms in the second experiment

Context	**Collision**	**Non-collision**
Static	1.32	1.84
Dynamic	2.45	1.34

These ratios are greater than 1, which proves the effectiveness of the proposed algorithm in speeding up collision detection.

Advantages

From the experimental results in both experimental scenarios, it is shown that the proposed algorithm is faster than the SAT algorithm, so the proposed algorithm will be a better choice for 3D simulation applications. The SAT algorithm performs the projection of all vertices onto an axis and then checks the result as an intersection condition, it is not possible to predict which axis will result in the checked condition for a non-intersecting pair. In the proposed algorithm, the conditions depend on the way the linear equation system is constructed.

The proposed algorithm has the advantages of being simple, easy to implement, suitable for parallel computing techniques, and can adapt to larger spatial dimensions. In particular, the proposed algorithm uses a data structure compatible with the SAT algorithm, so they can be combined in the same system to take advantage of both algorithms.

Limitations

First, the proposed algorithm can achieve even better performance by rearranging the order of inequalities in the system; however, the cost of rearranging is more expensive than the benefit it brings.

Second, although it is feasible, the parallelization strategy has not been applied. These will be the directions for improvement for the research of the PhD student in the future.

2.3. Conclusion of Chapter 2

In Chapter 2, the PhD student presents the research results on improving the fast collision detection of cloth models represented as triangular surfaces. The proposed method using wide-phase filtering and narrow-phase filtering algorithms is tested and gives good results on three different cloth model datasets in the GAMMA library, the average collision detection speed is about 2.3 times faster than the method of Tang et al., about 1.34 times faster than the method of Zhang et al. In the future, the PhD student plans to study the collision handling with model mesh structures that can be cut or torn, broken, etc. and extend this approach with GPU to further improve and accelerate the algorithm.

Also in chapter 2, the PhD student presents the results of the research on applying linear programming to accelerate collision detection of 3D models. First, the PhD student represents the problem of collision detection between objects as a system of linear inequalities, then presents an improved technique for solving this system of inequalities to improve the speed of collision detection calculation. From the experimental results, it is shown that the proposed algorithm is faster than the SAT algorithm and is proven to be a better choice for 3D simulation applications. The PhD student's research can be extended in the direction of proposing more efficient ways to deploy on different hardware platforms. In addition, the PhD student plans to improve this approach by parallel processing, which can also be hybridized with other algorithms.

Part of the research results presented in Chapter 2 are published in [CT4, CT5] in the List of published works related to the dissertation.

CHAPTER 3. COLLISION DETECTION METHOD OF 6-DoF COBOT ARM

3.1. Proposed method

3.1.1. Collision detection method

Based on the SVR method [62] as a foundation, the PhD student researched and developed a collision detection model for the CURA6 cobot arm. Convention: A time-varying *f*-dimensional signal $s(t) \in \mathbb{R}^f$ is sampled at sampling time t_I on a time window of size t_W; the number of samples is $N + 1 = {}^{t_W}/_{t_I} + 1 \in \mathbb{N}$.

The experimental programming process is performed in Python language. The proposed methods overcome the obstacles of the uncertainty of parameters in the dynamic model and the unmodeled effects (friction, sensor measurement noise).

Collision detection by SVR

Feature vector design: Feature vector design includes converting the signal into a vector form and assigning labels to the features. For temporal sampling, the PhD student sets t_W = 80 ms and t_I = 8 ms. After sampling the output of the momentum observer and transforming it into a matrix $R \in \mathbb{R}^{n\times(N+1)}$, the values of rows 1, 2, 3, and 4 of *R* form a single vector, resulting in the following feature vector $x(t)$ as:

$$x = [rj_1 \quad rj_2 \quad rj_3 \quad rj_4]^T \in \mathbb{R}^{4(N+1)} \qquad (3.1)$$

where $rj_i \in \mathbb{R}^{1\times(N+1)}$ denotes the value of the row with order *i* of *R*. Note that the row *i* of *R* denotes the sampled momentum observation output of joint *i*. Compared with using the signals of all joints, the size of this feature vector x is smaller, reducing the amount of computation for SVR.

SVR Training: Since exponential functions require more computation than polynomial functions, the PhD student uses the SRQ function [65] instead of the RBF function:

$$K_{SRQ}(x,z) = \frac{1}{\left(1+\frac{\|x-z\|^2}{\sigma^2}\right)^2} \tag{3.2}$$

with parameters $C = 1$, $\varepsilon = 0.02$ (based on [62]) and $\sigma = 3$.

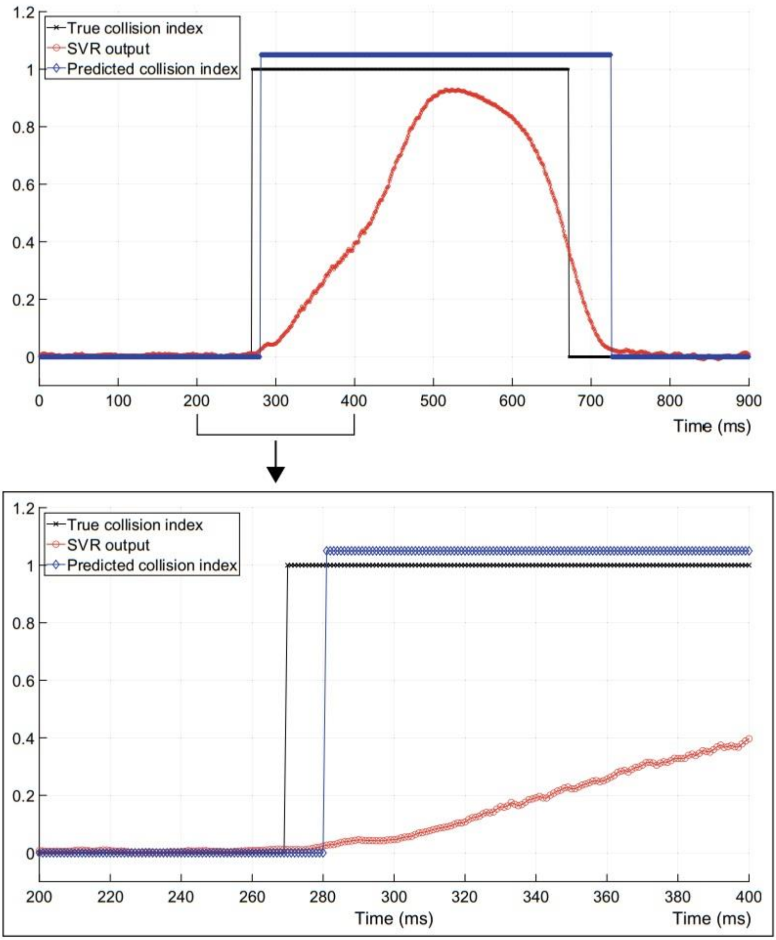

Figure 3.1. SVR output and corresponding collision index

The output value is close to 0 when there is no external force applied, and increases as the estimated external torque of joints 1–4 increases, and decreases as the external torque decreases. The larger the estimated external torque and the more joints affected by the external force, the larger the SVR output value.

Output filtering

Proceed to filter the output to reduce false positives (false alarms). Collisions are only warned if the output of the continuous collision detection method is *True* for a period of t_c ms. The output filter eliminates errors (false alarms with a duration in the range $1 - t_c$ ms) to detect collisions more effectively, the PhD student sets t_c=3 ms.

3.1.2. Test dataset

This dataset was downloaded from Gitlab: *https://gitlab.com/intema-gdansk/cura6-dataset/-/tree/main* [25]. The dataset consists of two sub-datasets, the 1.6 [A] threshold dataset and the 2.0 [A] threshold dataset.

15 random motion slices of the cobot were prepared as training data, each slice containing 10,000 samples (7 minutes of motion). The speed of each joint is designed by the solver to be safe within 25% of the maximum speed of the motor. 03 slices were collected with no payload on the cobot and the remaining 12 slices with payload (in grams) from the following list {782, 1016, 1282, 2298, 2757, 4039}, with 02 slices for each payload (load range from 500 g to 4000 g) [25]. Of which, training data (90%) and test data (10%).

3.2. Results and discussion

3.2.1. Results with the threshold 1,6 [A]

All datasets of this case have 65 actual collisions. The number of non-collision cases is different.

Table 3.1. Confusion matrix, Max speed V 10% - Threshold 1.6 [A]

Max speed V 10% Threshold 1.6 [A]		Reality	
		Collision	Non-collision
Predict	Collision	60	29
	Non-collision	5	113

In total 207 cases, the proposed method gave the following specific results: Correctly detected 60/65 collision cases, correctly detected 113/142 non-collision cases; only 34 cases were confused (5 collision cases were mistakenly judged as non-collision, and 29 non-collision cases were mistakenly judged as collision).

It can be seen that with the Threshold 1.6 [A] case, the ability to correctly detect collisions is good, but the ability to correctly detect non-collision is not as good.

3.2.2. Results with the threshold 2.0 [A]

All datasets of this case have 65 actual collisions. The number of non-collision cases is different.

Table 3.2. Confusion matrix, Max speed V 10% - Threshold 2.0 [A]

Max speed V 10% Threshold 2.0 [A]		Reality	
		Collision	Non-collision
Predict	Collision	51	3
	Non-collision	14	113

In a total of 181 cases, the proposed method gave the following specific results: Correctly detected 51/65 collision cases, correctly detected 113/116 non-collision cases; only 17 cases were mistaken (14 collision cases were mistakenly judged as non-collision, and 3 non-collision cases were mistakenly judged as collision)..

3.2.3. Discussion

The results obtained by the PhD student, compared with M. Czubenko and the research team [25] (using the same test data), gave equivalent results. The advantage of the SVR-based method proposed by the PhD student is that it is suitable for the context of limited training data.

Table 3.5. Evaluation of results and comparison with M. Czubenko [25]

max V	Threshold	Sensitivity		PPV		Specificity		Accuracy	
		Ours	[25]	Ours	[25]	Ours	[25]	Ours	[25]
10%	1.6 [A]	0.923	0.908	0.674	0.670	0.796	0.796	0.836	0.831
20%		0.908	0.923	0.596	0.606	0.771	0.777	0.808	0.817
30%		0.877	0.862	0.429	0.427	0.642	0.646	0.697	0.697
40%		0.923	0.954	0.480	0.492	0.709	0.713	0.757	0.767
50%		0.908	0.908	0.396	0.399	0.667	0.670	0.713	0.716
60%		0.908	0.908	0.404	0.404	0.657	0.657	0.708	0.708
10%	2.0 [A]	0.785	0.800	0.944	0.945	0.974	0.974	0.906	0.912
20%		0.769	0.754	0.926	0.942	0.971	0.978	0.907	0.907
30%		0.815	0.800	0.841	0.852	0.932	0.938	0.896	0.896
40%		0.862	0.877	0.848	0.851	0.941	0.941	0.919	0.923
50%		0.862	0.892	0.778	0.795	0.918	0.923	0.904	0.916
60%		0.908	0.908	0.831	0.831	0.933	0.933	0.926	0.926

In summary, we can see that with the Threshold 2.0 [A] case, the ability to correctly detect non-collision is good, but the correct detection of collision is not as good (this is in contrast to the Threshold 1.6 [A] case).

From the obtained results, it can be seen that the current intensity threshold affects the detection of collision/non-collision. With this experiment, the Threshold 2.0 [A] detects the current change worse (so the collision is detected worse), the analysis results tend to assume that there is no change in current, but the number of non-collision cases in the data is very large, so it leads to more correct non-collision judgment results.

3.3. Conclusion of Chapter 3

The proposed SVR-based method detected collisions of the 6-DoF CURA6 cobot arm quite effectively. The outstanding advantage of the proposed algorithm is that it only requires measurements of motor current sensors along with the cobot's dynamic model; there is no need to model or determine the frictional moments in the joints. Through experiments on the CURA6 cobot dataset, the SVR-based method only requires tuning a constant parameter. The comparison results show that the SVR-based method requires less collision data for training, which is suitable for the context of limited data.

Compared with the research results in [25] and [36], the proposed method is superior in the context of handling the impact of different loads. However, the validation for unknown random loads is still a difficult problem. Furthermore, the problem of collision detection of two or more cobot arms is complex and requires further research. For mass-produced cobots, another practical issue is that the processes required for effective collision detection must be replicated on a large scale. These will be open topics for future research of the PhD student.

Part of the research results presented in chapter 3 are published in [CT6] in the List of Publications related to the dissertation.

CONCLUSIONS AND RECOMMENDATIONS

Conclusion

The main results of the dissertation are as follows:

- Proposing the AB2PF collision detection algorithm that gives good results on three different cloth model datasets in the GAMMA library, the average collision detection speed is faster than traditional methods.

- Proposing the ABFME algorithm to detect collisions of pairs of cubes and tetrahedrons. The problem of collision detection between objects is represented as a system of linear inequalities, the dissertation presents an improved technique to solve this system of inequalities to improve the speed of collision detection calculation.

- Proposing the application of SVR supervised learning methods to detect collisions of CURA6 cobot arms based on current measurements along with the cobot's dynamic model.

Recommendations

The collision detection problem has high applicability, if exploited and applied with appropriate algorithms, it will obtain very valuable information and be used for collision detection in deformable objects and 6-DoF cobot arms.

The results of the dissertation contribute to improving the speed and accuracy in the collision detection problem of 3D model objects (solid objects and cloth objects) and the collision detection problem of 6-DoF cobot arms CURA6. The dissertation has expanded the usability of the collision detection problem to suit practical requirements.

LIST OF THE PUBLICATIONS RELATED TO THE DISSERTATION

1. **Van Hung Nghiem**, Van Duc Dang, Hien Anh Trinh, Van Can Nguyen (2018), "Improving automatic bounding volume hierarchy to detect collision of rigid bodies in virtual environments", *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XI về nghiên cứu cơ bản và ứng dụng CNTT*, tr. 209-214, DOI: 10.15625/vap.2018.00027.
2. **Nghiêm Văn Hưng**, Đặng Văn Đức, Trịnh Hiền Anh, Hoàng Việt Long, Nguyễn Văn Căn (2019), "Kỹ thuật phát hiện nhanh va chạm của vải trong thực tại ảo sử dụng kỹ thuật tính toán song song", *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XII về nghiên cứu cơ bản và ứng dụng CNTT*, tr. 239-246, DOI: 10.15625/vap.2019.00031.
3. **Nghiêm Văn Hưng**, Đặng Văn Đức, Trịnh Hiền Anh, Nguyễn Văn Căn, Hoàng Việt Long (2019), "Kỹ thuật phát hiện va chạm của vải trong thực tại ảo sử dụng khối bao k-DOP trên hệ thống CPU đa nhân", *Kỷ yếu Hội thảo khoa học quốc gia lần thứ XXII "Một số vấn đề chọn lọc của công nghệ thông tin và truyền thông"*, tr. 144-150.
4. **Nghiêm Văn Hưng**, Đặng Văn Đức, Nguyễn Văn Căn, Hoàng Việt Long, Trịnh Hiền Anh (2022), "Một cách tiếp cận ứng dụng quy hoạch tuyến tính trong bài toán phát hiện va chạm của các mô hình 3D", *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XV về nghiên cứu cơ bản và ứng dụng công nghệ thông tin*, tr. 510-516, DOI: 10.15625/vap.2022.0262.
5. **Van Hung Nghiem**, Van Duc Dang, Van Can Nguyen, Viet Long Hoang, Hien Anh Trinh (2022), "Fast Collision Detection Technique of Interactive Cloths in Virtual Reality Environment", *Res. Dev. Inf. Commun. Technol.*, vol. 12/2022, no. 2, pp. 49-56, 2022, ISSN: 1859-3526, DOI: 10.32913/mic-ict-research-vn.v2022.n2.1128.
6. **Van Hung Nghiem**, Van Duc Dang, Van Can Nguyen, Hien Anh Trinh (2023), "Collision Detection for 6-DoF Collaborative Robot Arm", *Res. Dev. Inf. Commun. Technol.*, vol. 12/2023, no. 2, pp. 73-79, 2023, ISSN: 1859-3526, DOI: 10.32913/mic-ict-research-vn.v2023.n2.1236.