ISSN:0975-9646

Nur Saadah Mohd Shapri et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015, 619-625

FixDec: Fixed Spherical *n* Points Density-Based Clustering Technique for Cloth Simulation

Nur Saadah Mohd Shapri[#1], Riza Sulaiman[#2], Abdullah Bade[*3]

[#]*Faculty of Information Science and Technology*
Universiti Kebangsaan Malaysia
**School of Engineering & Information Technology*
Universiti Malaysia Sabah

Abstract— **The ability to simulate cloth with crumpled effect is critical in virtual animation. Simulating a cloth model with realistic collision handling is a crucial issue due to the highly complex collision calculations. Researchers have proposed several techniques to solve the above issues; however, the weakness still remains particularly in collision handling computations. In this paper, a technique called Fixed Spherical *n* Points Density-based Clustering Technique (FixDeC) is introduced to reduce the collision handling computations once cloth surfaces have undergone crumpling processes. The collision checking procedure between cloth surfaces can only be performed when cloth surface points contained in the predefined spherical clusters. Any cloth surface points beyond the cluster region is ignored and removed from the collision calculation list. From the experiment, when compared to density-based clustering, a series of brute force collision checking is reduced due to the lesser amount of cloth surface points within the cluster that have to be considered for collision checking. Moreover, FixDeC can also ensure that two adjacent cloth surface points are not penetrating between each other even though the cloth surface is crumpled. The results revealed that the proposed technique has produced significant improvement in terms of collision calculation compared to the traditional approach. The FixDeC technique can guarantee penetration among cloth surfaces would not occur, and self-collision handling for cloth surfaces could be more efficient and robust.**

Keywords—**Cloth simulation, Cloth self-collision handling**

I. INTRODUCTION

Virtual environment application is a complex system which consists of various computer graphics components and interactions. Computer games are an application which requires fast and nearly accurate collision detection system. The application may consist of rigid and deformable models. The interaction between these kinds of models is of course will utilize highly intensive mathematical operation. Based on the previous research finding, to simulate realistic cloth which looks "real as the physical cloth" is not an easy task. During the simulation, the virtual cloth surfaces will entangle with other surfaces that construct the whole cloth surfaces.

As far as the collision detection system is concerned, the self collision detections usually have various issues that must be resolved in order to come out with good collision detection algorithm. The first challenge is of course the complexity of mathematical computation. Self-collision detection is a special case of collision detection where both geometrical primitive or cloth surfaces are located side by side. In other words, these primitives are the collection of geometrical primitives that create the whole deformable surfaces of the cloth model. Series of computational issues are going to be faced such as detecting multiple collisions, collision inconsistency and adjacency in primitives and bounding boxes. Considering these issues, the propose algorithm and technique should be robust and efficient enough to handle various type of collision settings simultaneously.

Back in 1980s, computer graphics community has tried to conduct research in modeling cloth for animation, simulation and computer games application. They tried to utilize physics laws so that the implemented cloth model looks natural. Since then, the studies of cloth modeling in computer graphics community has been grappling almost two decades. Up to this date, they are trying to improve latest issues in cloth simulation such as wrinkles, crack and efficient self-collision handling.

Handling collision detection when cloth undergone wrinkles still requires full attention and improvement in particular on how to reduce highly intensive computation and maintaining the cloth wrinkles realistically and stable. An efficient self-collision technique which can improve self-collision handling that is able to avoid penetration of cloth nodes are considered as the main goal that needs to be achieved in simulating virtual cloth in computer graphics. Current study as stated in [1] suggested that handling self-collision for cloth is very much inefficient and always mess up the performance of cloth simulation system. In [2,3] the authors mentioned that self-collision performance basically involved knowledge like mechanical feedback and movement physics that can give natural effect cloth simulation in virtual world. In [4], the realistic virtual cloth application should behave like "rubbery" and "elastic" motion. The words "rubbery" and "elastic" are the key words to make cloth behave and look as the real physical cloth. Once the realistic movement and scenery of cloth requirements are fulfilled, we can see that people cannot differentiate between the synthetic cloth model and the real cloth. Basically, high quality performance of cloth simulation requires thousand of cloth mass particles and this may end up and lead to multiple, inconsistent and highly intensive collisions handling [4].

To detect collision between cloths is not an easy process and it needs thousand and hundred collision queries. This is because the cloth structure is quite complex and the calculation workload is tremendously complicated. Based on this situation, many researchers try to find out the best solution but the process is very much in progress besides a lot of incoming setback such as memory consumption, surface crack, inconsistent and instable collision checking and updating process growth surprisingly worst as mentioned in [3]. Considering the matters, the entry point to improve and preserve the realistic appearance of cloth which undergone wrinkles is of course to improve the self-collision detection system, makes it fast, robust and efficient to handle thousand cloth mass particles. Besides that we need as well to consider colliding points, new correct position after collision occur and velocities of colliding nodes.

Inside virtual cloth there are some parts called particle forces that contribute to create realistic effect to the virtual cloth (cloth illustration on Fig. 1). The particle forces play an important role which can give various effects to self-collision handling during the simulation of the synthetic cloth. Some physic element such as gravity and etc. have been enforced into the particle forces to emulate realistic motion of cloth model as suggested in [3].

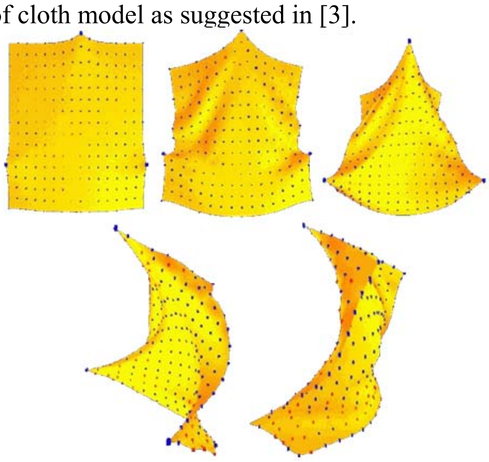

Fig. 1 Virtual cloth representations with collision and particle forces application

Previous researcher such as [5] described that self-collision handling needs dedicated process as compared to general collision detection system. Its mean that a collision handling and technique must be robust to tackle and handle incomplete collision detection information. In [6] also mentioned that stability, robustness and speed for physical simulation of cloth in real time must fulfil various requirements which can seriously limit the development process. Stability is needed because the propose self-collision detection will face various condition and case presented during the simulation and it must indeed response correctly as required. Unnatural cloth appearance and movement must be avoided at all costs.

Robustness on the other hand is even more difficult to handle because our consideration is interactive platform. Various condition and circumstances need to put forward such as abrupt movement of cloth particle, forces and other physical condition such as gravity, crack and wind. In short we can say that robustness issues will dealt with internal and external factors instead of cloth models itself and the process to detect collision. Highly accurate physically based cloth simulation systems are just too slow to be used in real time application. They are computationally heavy and difficult to optimise to a level that would permit them to be significantly accelerated.

In term of collision handling for virtual cloth, time is the most important aspect that researcher must pay attention indeed. This is followed by the memory usage which can be handled with partitioning approach of collision detection algorithm such as space decomposition and hierarchy bounding box [3]. The algorithm will improve the efficiency of collision detection by reducing number of elements to be calculated for collision checking. Several self-collision handling strategies have been taken into account, starting from detecting collision until the procedure to maintain the consistency of both detected collision and appropriate response. Other researchers tried to handle collision for virtual cloth such as suggested by [7] where the authors proposed a new technique that grouped the particles together and considered it as a rigid object. In [8] the authors proposed the same technique as [7] which can divide the particle into collision cluster and able to avoid possible subsequent collisions.

II. Collision Handling between Cloth Surfaces

Fig. 2 represents the research main contribution where it consists of several steps for collision handling construction. In second phase, density-based clustering involves programming and implementation for the proposed technique. This phase starts with modeling cloth simulation with particles and handling the collision between cloth surfaces using clustering method and running the self-collision between cloth surfaces. The improvement computational heuristic method to handle intersection will be used to test between particles intersection. This method is able to avoid penetration of cloth nodes undergone wrinkling and twisting process.

Self-collision detection will handle between the cloth surfaces and the process proceeds to collision testing. Collision testing will be conducted based on user intervention during running time. Graph will be generated based on the data collected by collision report that automatically running with collision testing. In the construction testing, external force will be applied to create wrinkle and the penetration among cloth surfaces would not occur so that self-collision handling for cloth surfaces can reduced the computational time.

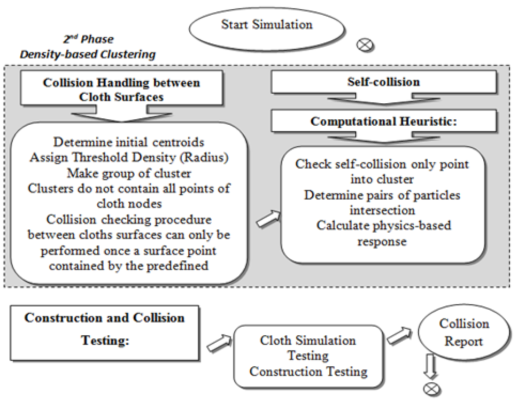

Fig. 2 Collision Handling Frameworks

The computational heuristic method for collision checking procedure between cloths surfaces can only be performed once a surface point contained by the predefined spherical clusters. The improvement heuristic to handle collision for any surface point beyond the cluster region is ignored and removed from the collision calculation list. As it compared with density-based clustering, series of brute force collision checking have been reduced due to a lesser amount of surfaces point within the spherical cluster are considered for collision checking and then a physics-based response is calculated. This contribution also can ensure that two adjacent surface points are not penetrable.

III. Heuristic Method for Particle Collision

In this paper, concept from [9] which has been aspired from [7] constraint operation is illustrated in Fig. 3. In [9] the authors approach the distance node ($\boldsymbol{D_n}$) at each time step were compared with the rest lengths ($\boldsymbol{R_l}$) of the spring. If the distance ($\boldsymbol{D}$) was larger than the rest length ($\boldsymbol{R_l}$), the distance node ($\boldsymbol{D_n}$) would be reduced accordingly. Likewise, if the distance node ($\boldsymbol{D_n}$) was shorter than the rest length, the distance node ($\boldsymbol{D_n}$) would be lengthened. The code was applied to every spring present in the cloth model. The end of this process, two adjacent surface points are still penetrate because this approach used basic way to collide with distance node ($\boldsymbol{D_n}$) plane. Based on this problem, we need a heuristic method as a rule of thumb that often helps in solving certain problems to reduce penetrable between particles collision.

Particles Collision Algorithm

Input Parameters : ***D, D_n***
- ***D*** Distance
- ***D_n*** Distance Node
- ***R_l*** Rest length

Output : Two adjacent surface points are penetrable

1. Compute a repulsion force

2. Basic way to collide with a y = D_n plane.

```
If (D.y < Dn) {D.y= Dn - D.y;...}
Dn = y0 - Dn;
v.y= - Dn * v.y;
v.x= (1- Rl) * v.x;
v.z= (1- Rl) * v.z;
```

3. New correction position nodes

Fig. 3 Pseudo-code of DetectCollidingParticles() [9]

In our approach Fig. 4, the distance nodes ($\boldsymbol{D_n}$) are compared with Rest length ($\boldsymbol{r_l}$) of the spring. If the $\boldsymbol{dist(p_1, q_1)}$ is larger than the Rest length ($\boldsymbol{r_l}$), the particle will be return back to the original length. However, if the $\boldsymbol{dist(p_1, q_1)}$ shorter than the original length, the particle will be push back to the original distance. This process was taking a several time step in order to meet to the required solution. The pseudo-code is illustrated in Fig. 5. It is because this approached need calculate position for particles using point direction and maintain the particle position along simulation. The following, more detailed about collision particles concepts for the linear spring cloth nodes will be discussed. In a simulation step, all vertices to the new positions in the end of the frame are moved. Here, we list the main concepts of heuristic method for particles collision. In the following, we formalize the heuristics method for cloth nodes:

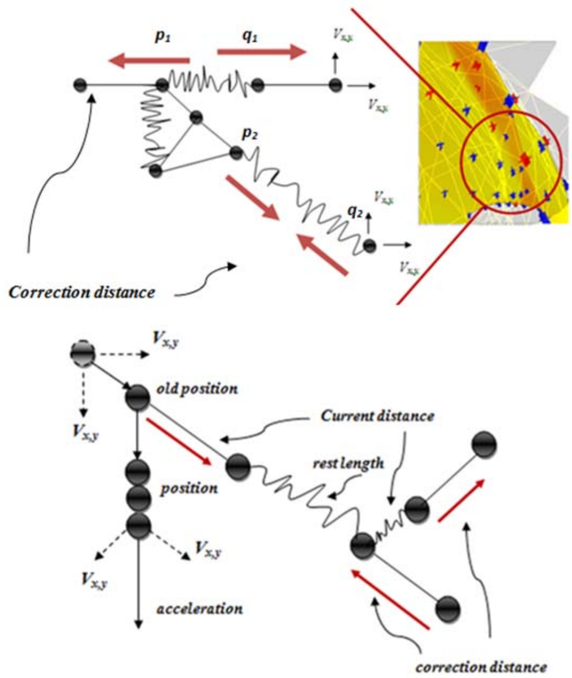

Fig. 4 Particle movements for correction position nodes

1) Collision checking procedure will be started: Let two points be $\boldsymbol{p_n}\,(\boldsymbol{x,y,z})$ *and* $\boldsymbol{q_n}(\boldsymbol{x,y,z})$, where $\boldsymbol{n} =$ 1,2,3 ... $+\,\boldsymbol{n}$. Assume that $\boldsymbol{p_1}$ *and* $\boldsymbol{q_1}$ are the two points from cloth nodes $\boldsymbol{Np} = \{\boldsymbol{p_1} \in \boldsymbol{Np}\ |\boldsymbol{dist}(\boldsymbol{p_1}, \boldsymbol{q_1}) \ \le\ \boldsymbol{r}\}$, where $\boldsymbol{q_1} \in \boldsymbol{Np}$.

2) Surface point contained inside the cluster region: If the point is outside the spherical cluster, the intersection test can be skipped as no collision is possible. Only points that come from regions borders will check the collision.

3) Compute a repulsion force: The intersection test between a point $\boldsymbol{p_1} = (\boldsymbol{x,y,z})$ and $\boldsymbol{q_1} = (\boldsymbol{x,y,z})$ is performed by calculating repulsion forces of $\boldsymbol{p_1}$ *and* $\boldsymbol{q_1}$. If distance $\boldsymbol{D} \le \boldsymbol{r}$ = positive, the collision test is aborted. If distance $\boldsymbol{D} \le \boldsymbol{r}$ = negative, then Correction vector $\boldsymbol{Cr}$ are computed:

$$\boldsymbol{Np} = \{\boldsymbol{p} \in \boldsymbol{Np}\ |\boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q})\ \le\ \boldsymbol{r}\} \begin{cases} \text{If } \boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q}) \le \boldsymbol{r} \\ \quad = \boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q})\ \boldsymbol{positive} \\ \text{Collision = false} \\ \text{If } \boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q}) \le \boldsymbol{r} \\ \quad = \boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q})\ \boldsymbol{negative} \\ \text{Collision = true} \\ \text{Correction vector } \boldsymbol{C_r}\text{:} \\ \{\boldsymbol{C_r y} = \boldsymbol{1 - r_l / D_n * distance\ half - 1}\} \end{cases}$$

4) New correction position nodes: The correction position of the position nodes may change immediately after the positions of the particles are updated.

Improvement Pairs of Particles Algorithm

Input Parameters : N_p, D_m, rl, D_n

N_p Cloth nodes
D_m Distance Magnitude
rl Rest length
D_n Distance Node

Output : Two adjacent surface points are NOT penetrable

1. Collision checking procedure will be started
2. Surface point contained inside the cluster region
3. Compute a repulsion force
   I. If Distance Magnitude D_m (Np_1, Np_2) < Distance Node D_n
      - Collision true
      - Correction vector C_r;
        Find point direction between $x_2 - x_1$
        $\{C_r.y = 1 - r_l / D_n$ * **distance half – 1**$\}$:
        Calculate Position for particles using point direction*((**1 - rest length / distance node)* distance half - 1**)
        $x_1 = x_1$ + Position for particles
        $x_2 = x_2$ + Position for particles
4. New correction position nodes

Fig. 5 Pseudo-code of DetectCollidingParticles()

IV FIXED SPHERICAL N POINTS DENSITY-BASED CLUSTERING (FIXDEC)

This technique achieves faster construction time while tracking self-interactions in highly cloth surfaces consisting of a large number of contact points. By conducting the process, the collision checking procedure between cloth surfaces can only be perform once a surface point contained by the predefined spherical clusters. Moreover, we also need to make sure that there is no self-collision in each cluster. Generally, FixDeC technique is based on density-based clustering concept, which is the density in the neighbour, has to exceed some predefined threshold. This technique can be used to reduce the time consumption and also can ensure that two adjacent surface points are not penetrable even though the cloth surface being twisted and clumped.

Here, we list the main concepts FixDeC. This concept then can be applied on cloth surfaces. According to the pseudo-code in Fig. 6, we formalize the FixDeC concepts for cloth nodes:

FixDeC Algorithm

Input Parameters: c, r

c - Center of clusters / initial centroids
r – Radius that delimitate the neighbour area of a points

Output : Set of Spherical Clusters

1. Select c points initial centroids.
2. Assign the r threshold density (Radius).
3. Make groups of cluster Cl_n.
4. Clusters do not contain all points of cloth nodes N_p.
5. Collision checking procedure between cloths surfaces can only be performed once a surface point contained by the border clusters b.
   I. If Distance $(Np_1, Np_2) \leq$ Radius
      If Distance positive
      - Surface point contained outside the cluster region
      If Distance negative
      - Surface point contained inside the cluster region
      - Collision checking procedure will be started
6. Points not classified into any border clusters are classified as noise points S_p. ***noise*** = $\{Sp \in Np \,|\forall i: Sp \notin Cl_n\}$.
7. Update spherical Cluster
8. Repeat step 5 until step 7

Fig. 6 Pseudo-code of FixDeC algorithms

1) *Select **c** points initial centroids:* The initial centroids of cloth nodes c, denoted by $Np(c)$, is defined by $Np(c) = \{p \in Np \,|c \in p\}$, where Np, $p_{i,j}$.....$p_{m,n}$ is the all points of cloth nodes.
2) *Assign the **r** threshold density (Radius):* The radius of the clusters is predefined value of r and assigns it in one variable to generate cluster regions $r = k * SL_{max}$, where k is a positive integer.
3) *Make groups of cluster Cl_n:* To make a group of clusters, user can set the number of clusters depend on the total of cloth nodes.
4) *Border Cluster:* A border spherical cluster b is defined as a cluster region that has multiple collision particles within its centroids of cloth nodes. $Np = \{p \in Np \,|dist(p,q) \leq r\}$, where$\in Np$. Only points that come from regions borders will check the collision. Any surface point beyond the cluster region is ignored and removed from the collision calculation list.

   Use c and b to get the cluster regions.

   If $dist(p,q) \leq r$ $= dist(p,q)$ ***positive***

   Collision = false

   If $dist(p,q) \leq r$ $= dist(p,q)$ ***negative***

   Collision = true

5) *Noise Points of the Cloth Nodes:* The noise points S_p is define as points not classified into any border clusters b. Let $Cl_1, \ldots, Cl_n$ be the clusters of the cloth nodes N_p, then we define the noise as the set of points not belonging to any cluster Cl_n. noise = $\{Sp \in Np \,|\forall i: Sp \notin Cl_n\}$.

This process stops when there is no collision between cloths surfaces occur and then the algorithm will begin and update a process when the cloth nodes around the border cluster and collide each other into the cluster. As we discussed before, the particle connections are not longer than SL_{max} at any time, which mean that the distance between any two given particles, $Np\ (i,j)$ and $Np\ (i + k, j + k)$ is limited in $k * SL_{max}$, where k is a positive integer. That is, the spherical cluster b $(Np\ (i,j), k * SL_{max})$, whose center is particle $Np\ (i,j)$ and radius is $k * SL_{max}$ can bound particles $Np\ (i + n, j + n)$ completely, where n = 0,1,2,3,….+k. See Fig. 7 this kind of sphere is called the spherical cluster.

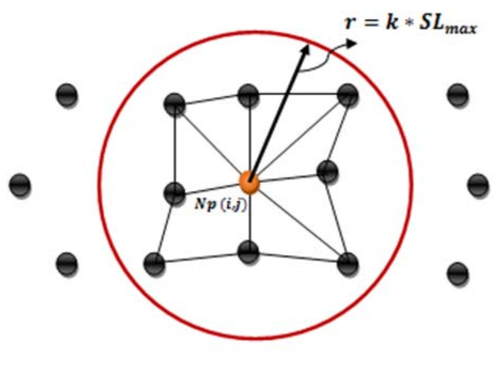

Fig. 7 2D bounding spherical cluster

Depend on pseudo-code of FixDeC algorithm, construction of spherical cluster starts by first state the centers cluster from cloth nodes and assign the threshold density also known as radius for each cluster $\boldsymbol{r} = \boldsymbol{k} * \boldsymbol{SL_{max}}$. To make a group of clusters, user can set the number of clusters depend on the total of cloth nodes. The collision checking procedure only applied once a surface point contained in the cluster region $\boldsymbol{Np} = \{\boldsymbol{p} \in \boldsymbol{Np}\ |\boldsymbol{dist}(\boldsymbol{p},\boldsymbol{q}) \leq \boldsymbol{r}\}$, while points not classified into any clusters are classified as noise points $\boldsymbol{Cl_n}$. noise = $\{\boldsymbol{Sp} \in \boldsymbol{Np}\ |\forall \boldsymbol{i}: \boldsymbol{Sp} \notin \boldsymbol{Cl_n}\}$. Fig. 8 illustrates an example of 2D spherical cluster construction for FixDeC technique with their surface intersection problems and Fig. 9 shows enable FixDeC spherical cluster in 3D that applied on our cloth object.

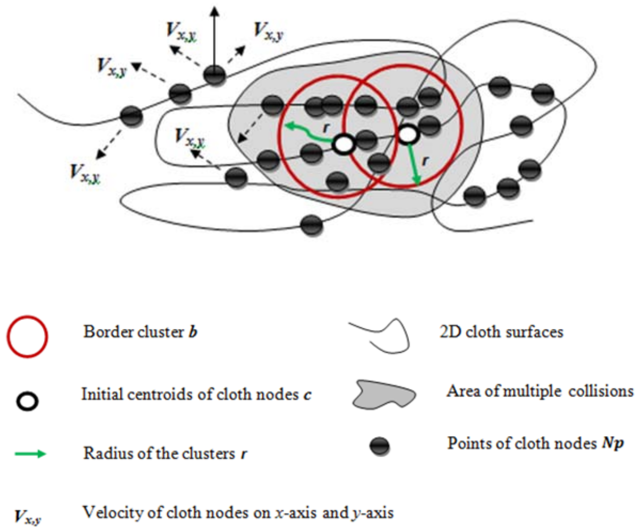

Fig. 8 FixDeC representation

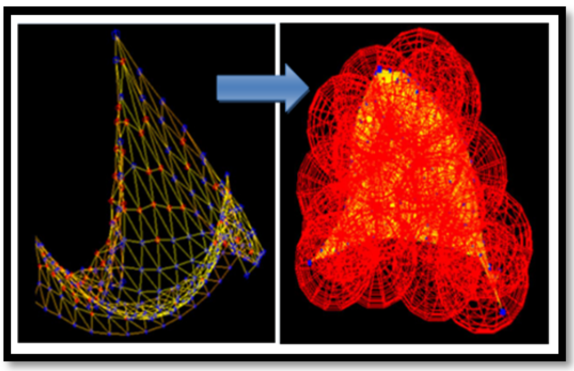

Fig. 9 FixDeC technique display

V EXPERIMENTAL RESULT

B. Self-collision Construction Time Test of Our Improvement Method

This testing is to test self-collision time for our improvement method (IM) compared to the particles collision method (PCM). Although our method consume much time but it can be used to solve the penetration problem compared to particles collision method. The graph of this test is illustrated in Fig. 10.

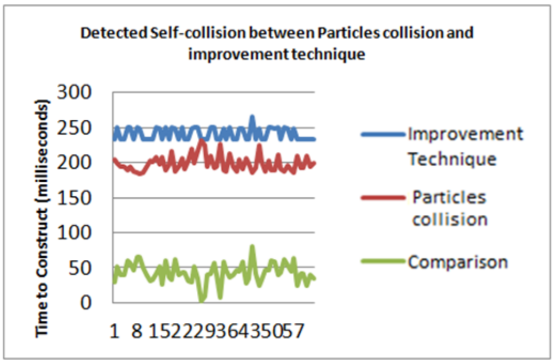

Fig. 10 Average times to construct 500 times for detected self-collision between particles collision method and our improvement particles method.

Graph in Fig. 11 showed that particles collision test is a suitable method to detect self-collision in cloth nodes simulation. By constructing 500 times for particles method, the average times is obtained. Even though our method testing is highly time consumes with an average of 260 milliseconds (52%) but our method is the best way to solve the penetration collision problem. Based on this figure, we can summaries that by using our method, 15 milliseconds (7%) achieved high construction time compared to particles collision method.

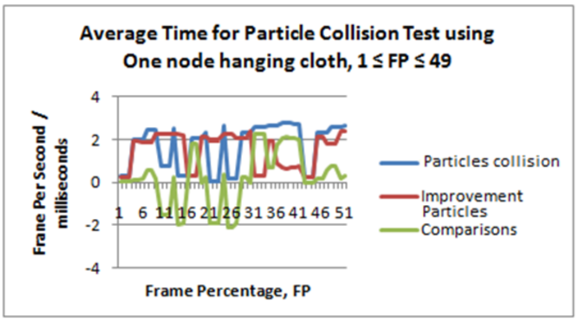

Fig. 11 Average Time for Particles Collision Test using one node hanging cloth, 1 ≤ FP ≤ 49

Fig. 11 shows the average time for particles collision test using one node hanging cloth, 1 ≤ FP ≤ 49. The testing uses PCM and our method for frame percentage collision time, FP, from 1 to 49. At FP=37 to FP=40, our method uses more time to calculate the collision testing for each frame between these approaches by 11 ms. While at FP=4, only 5 ms differences is seen between the two approaches.

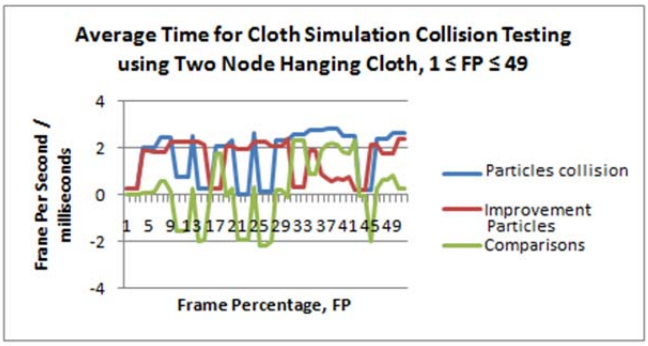

Fig. 12 Average Time for Particles Collision Test using two node hanging cloth, 1 ≤ FP ≤ 49

For average time particles collision test using two node hanging cloth, as shown in Fig. 12, the average for frame percentage using our IM is more similar with one node hanging cloth collision testing. In Fig. 13, the total average time collision testing for PCM is 16 ms while that for our method is 18 ms. From here, we noticed that our method consume more time for checking collision between particles cloth nodes compared to the PCM but the penetration problem between surface points are not penetrable even though the cloth surface is being twisted.

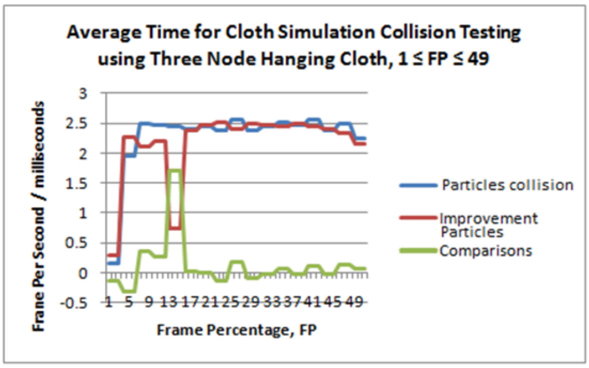

Fig. 13 Average Time for Particles Collision Test using three node hanging cloth, 1 ≤ FP ≤ 49

Table 1 shows in detail about the particles collision experiment and result tests depend on graph in previous discussion.

TABLE I.
EXPERIMENT AND RESULT

	Spherical Cluster	Average Time	Summary
PCM (one hanging cloth)	No	6 ms	
IM (one hanging cloth)	No	11 ms	**
PCM (two hanging cloth)	No	9 ms	
IM (two hanging cloth)	No	13 ms	**
PCM (three hanging cloth)	No	16 ms	
IM (three hanging cloth)	No	18 ms	**

Test the performance of heuristic method
**used more time for checking but can solve the Penetration collision problem

C. Self-collision Handling Construction Time Test of FixDeC

Fig. 14 clearly showed that FixDeC is a right way of solving the self-collision problem compared to particles collision from particles collision method. By constructing 500 times for cloth FixDeC and PCM, the average times is achieved. Even though a PCM is highly accurate in most cases, the cost collisions for cloth surfaces still slow with an average of 245 milliseconds (45%) and can be reduced through the use of FixDeC technique.

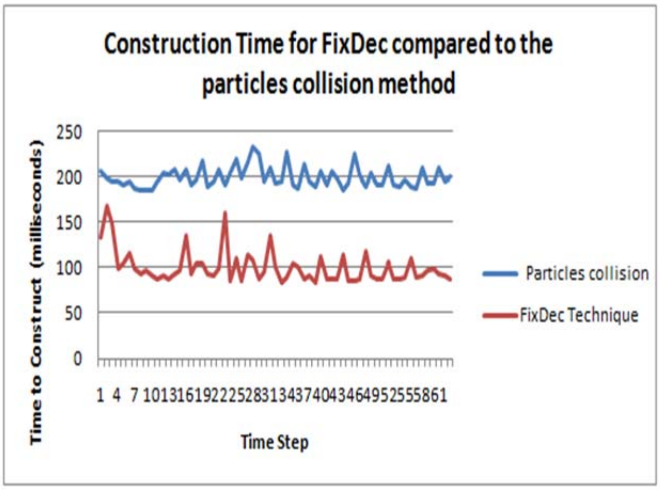

Fig. 14 Average times to construct 500 times for cloth using FixDeC and particles collision method.

From the graph in Fig. 15, the different of time construction show that the average time with FixDeC technique is better compared to without FixDeC technique. By constructing 500 times for with and without FixDeC technique in cloth, the average times is achieved. FixDeC technique produce much faster times an average of 160 milliseconds (32%) from a percentage of total frames compare to the first approach an average of 245 milliseconds (45%). Based on this figure, we can summaries that by using FixDeC technique, 85 milliseconds (13%) of calculation achieved better construction time compared to without using FixDeC technique. Table 2 shows in detail about the FixDeC experiment and result tests.

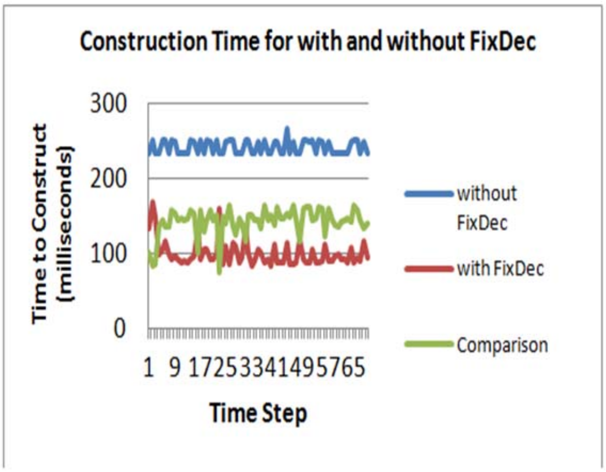

Fig. 15 Average times to construct 500 times for with and without FixDeC technique in cloth.

TABLE II.
EXPERIMENT AND RESULT

	Spherical Cluster	Average Time	Summary
FixDeC	Yes	160 ms (32%)	85 ms (13%) achieved better construction time
Particles collision method (PCM)	No	245 ms (45%)	
Improvement method (IM)	No	260 ms (52%)	15 ms (7%) achieved high construction time - can solve the penetration problem
FixDeC with PCM (one hanging cloth)	Yes	31 ms	*
FixDeC with IM (one hanging cloth)	Yes	23 ms	
FixDeC with PCM (two hanging cloth)	Yes	33 ms	*
FixDeC with IM (two hanging cloth)	Yes	30 ms	
FixDeC with PCM (three hanging cloth)	Yes	49 ms	*
FixDeC with IM (three hanging cloth)	Yes	37 ms	

The focus of this experiment lies on testing the FixDeC performance of a highly cloth model that experiences self-collision.
* used more time for checking.

IV. CONCLUSION

FixDeC technique helps in reducing the problem of handling computation self-collision detection once cloth surfaces undergone wrinkle and twist process. For the use in our cloth simulation system, we showed that clustering is the best choice which does not state the minimum number of particles that must exist in the cluster region. The collision checking procedure between cloths surfaces can only be performed once a surface point contained by the predefined spherical clusters. The implementation of FixDeC technique is successfully done in cloth simulation. This paper evaluates the performances of two techniques FixDeC with improvement particles method and FixDeC with particles collision method for difference node hanging cloth. The series of evaluation has been completed for FP in range 1 to 49. From the results, the average detection time was examined and the FixDeC with improvement particles method could produce fast and efficient self-collision handling compared to the particles collision approach. In conclusion, our technique can be used perfectly in order to handle the self-collision between cloth surfaces with better performance.

ACKNOWLEDGMENT

This work was supported by Center for Artificial Intelligence Technology (CAIT) at Faculty of Information Science and Technology. The authors are thankful to all contributors that willingness to share their ideas and hard-earned experience. Thank you all.

REFERENCES

[1] Cheng, H.-H. Interactive cloth simulation, Master's thesis, Computer Science Department, University of California, Los Angeles, USA. 2009. URL: http: // www. scribd. com/ doc/ 24601463/

[2] Baraff, D., and Witkin, A.. Dynamic simulation of nonpenetrating flexible bodies. In Proc. of SIGGRAPH, ACM Press / ACM SIGGRAPH, Computer. Graphics Proc.,1992, 303–308.

[3] He, Bing, and Liu Cheng. "A Quad Tree Based Self-collision Detection Method for Cloth Simulation." Journal of Computers 5.7 2010: 1070-1077.

[4] Lv, M. Y., et al. "A fast self-collision detection method for cloth animation based on constrained particle-based model." Digital Media and its Application in Museum & Heritages, Second Workshop on. IEEE, 2007.140-145

[5] Gissler, Marc, Ruediger Schmedding, and Matthias Teschner. "Time critical collision handling for deformable modeling." Computer Animation and Virtual Worlds 20.2-3 , 2009: 355-364.

[6] Volino, P., & Thalmann, N. M., Accurate collision response on polygonal meshes. In Computer Animation 2000. Proceedings (pp. 154-163). IEEE.

[7] Provot, X. , Collision and self-collision handling in cloth model dedicated to design garments (pp. 177-189). Springer Vienna 1997.

[8] Huh, S., Metaxas, D. N., & Badler, N. I.. Collision resolutions in cloth simulation. In Computer Animation, 2001. The Fourteenth Conference on Computer Animation. Proceedings (pp. 122-127). IEEE.

[9] Fuhrmann, A., Gross, C., and Luckas, V.. Interactive animation of cloth including self collision detection. Journal of WSCG, Vol 11, 2003. No 1 pages 141- 148.