PRIMITIVES PENETRATION DEPTH COMPUTATION USING DYNAMIC PIVOT POINT TECHNIQUE

Hamzah Asyrani Sulaiman[1*], Abdullah Bade[2]

[1]Universiti Teknikal Malaysia Melaka,
Durian Tunggal, Melaka, Malaysia
[2]Faculty of Science and Natural Resources,
Universiti Malaysia Sabah,
Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia.
[*]Email: h.a.sulaiman@ieee.org

ABSTRACT. *Computing penetration depth between two or more polygons commonly described by most researchers as one of the high computational cost process. Major implementation required numbers of pre-processing function just to find the minimum penetrating depth between those penetrated objects or polygons. In this paper, we proposed a technique that manipulates the advantages of Dynamic Pivot Point into computing penetration depth between two or more objects. Comparing our proposed technique (DyOP-PD) with the well-known Lin-Canny technique, the conducted experiments proved that our proposed technique has achieved better efficiency. Overall time for DyOP-PD technique to compute penetration depth was significantly faster than the Lin-Canny PD technique (refer Figure 6.9). Our technique was faster than the prominent technique where the computational time significantly reduced, solved a larger fraction of problems, and produced better paths of penetration depth. The lowest results recorded from our simulation was in average at 10.22 milliseconds for DyOP-PD and 21.33 milliseconds for Lin-Canny PD technique. The findings proved that DyOP-PD technique is robust to handle efficient, nearly accurate, and fast penetration depth detection compared to Lin-Canny-PD technique.*

KEYWORDS. C*ollision detection, penetration depth, virtual environment*

INTRODUCTION

In 2001, Bergen proposed an algorithm that exploited the capability of the Minkowski sum by implementing them to find the penetration depth. By estimating the lower bound between two convex polytopes, their algorithm iteratively expands the polyhedral approximation. Kim in 2002 also used the Minkowski Sums by introducing an incremental algorithm that approximate the penetration using local optimal solution (Kim *et al.*, 2003). They also improved it by implementing hierarchical method by decomposing some non-convex objects into convex type objects and recursively refined the result of penetration. Instead of using the local optimal solution based on Kim *et al*. (2003), Stephane and Lin (2006) developed an algorithm that is capable of finding penetration depth locally using a graphics hardware.

In 2012, a technique that approximate the local contact space and perform iteration based on projection of Gauss Seidel LCP solver using translational method. The technique however does not solve the n-body problems where it involves numbers of polygons into the environment. Recently, Zhang *et al*. (2014) produced an algorithm of continuous PD where they used binary classification technique and then construct a bijective mapping between a spherical spaces with a pre-computation contact space. Bijective mapping is a process that set a function from one element to another elements. It can inverse mapping and at some point, it is regarded as one element for two elements. However, the algorithm depends heavily on pre-sampling during classification of contact point and handles only translational penetration depth (not rotational) and almost use a lot of pre-computation that dominated the overall computation before collision took place (Zhang *et al*., 2014).

Penetration depth (PD) involved the measurement of total penetration occurred between one object into another object which is overlapped (Zhang *et al*., 2007). There are many applications that used PD for their simulation process. Traditionally, PD computation has been limited to the translational PD. Translational PD is defined as penetrating object that actually penetrates deep into another object and find their minimum distance. Among the good algorithm for PD were Bergen (2001), Shengzheng and Jie (2009) and Redon *et al*. (2002).

Hence, in this paper we described our related work on theoretical framework of computing penetration depth between two primitives which is the intersected and penetrated triangles that might happens frequently. Our target for this research is to present our ideas on how Dynamic Origin Point been implemented and can optimize the computation of penetration depth and calculate nearly accurate result (Sulaiman *et al*., 2013).

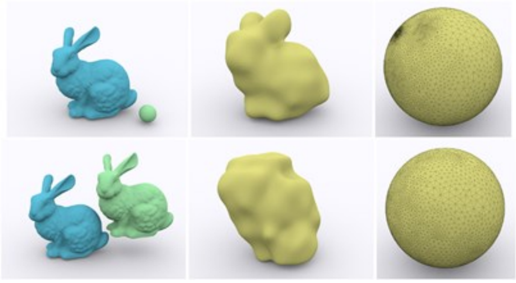


Figure 1: Spherical parameterization method that used binary classification on Zhang *et al.* (2007)

METHODOLOGY

A. Penetration Depth Algorithm for DyOP (DyOP-PD0 for 2D Concept

In this procedure, we need to use line equation of currently penetrated triangle or primitive. Given an example from Figure 2, blue line represents the currently penetrated edges that actually have their own line equation while the blue dash line is a new imaginary edge based on the blue line. Noted that the blue dash line is a parallel line/edge to the blue line. To obtain the corresponding edges of Triangle PQR (which is the intersected triangle), we just need to use DyOP technique that could find which one is the nearest edge to the pivot point. Then, by using that edge, we could find both blue line and blue dash line.

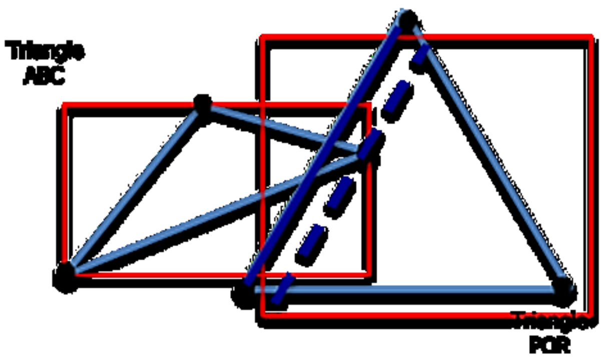


Figure 2: Blue dash line and blue line for penetration depth algorithm base on DyOP (DyOP-DP)

Based on Figure 2, both blue line and blue dash line will be used as penetration depth mechanism in DyOP-PD algorithm. Both triangle will undergo the same process that determine the intersection point based on the line equation information provided by the blue line and blue dash line equation of each triangle (Triangle ABC and Triangle PQR).

In order to calculate both lines depict in Figure 2, we will use the preliminary equation below based on Triangle PQR:

$$Y_{PQ} = M_{PQ}X_{PQ} + C_{PQ} \tag{1}$$

$$Y_{i-PQ} = M_{i-PQ}X_{i-PQ} + C_{i-PQ} \tag{2}$$

where $i - PQ$ is the imaginary form of parallel blue dash line that will be used to calculate the penetration depth of corresponding penetrated triangle of ABC into triangle PQR.

In order to find M_{i-PQ}, we just need to maintain and use the current M_{PQ} as it will be at the same slope. While to find C_{i-PQ}, we need to calculate a new vertex point of Triangle ABC that currently penetrated into the Triangle PQR. The current coordinate of that vertex will be used to calculate our new line equation of $i - PQ$.

Finding M,

$$M_{PQ} = M_{i-PQ}$$

$$= \frac{Y_P - Y_Q}{X_P - X_Q} \tag{3}$$

Obtaining C,

$$C_{PQ} = Y_P - \frac{Y_P - Y_Q}{X_P - X_Q} X_P = Y_Q - \frac{Y_P - Y_Q}{X_P - X_Q} X_Q \tag{4}$$

$$C_{i-PQ} = Y_A - \frac{Y_P - Y_Q}{X_P - X_Q} X_A \tag{5}$$

Put in back into equation (2)

$$Y_{i-PQ} = M_{i-PQ} X_{i-PQ} + C_{i-PQ}$$

$$= \left(\frac{Y_P - Y_Q}{X_P - X_Q}\right) X_A + Y_A - \left(\frac{Y_P - Y_Q}{X_P - X_Q} X_A\right) \tag{6}$$

Then, we just need to calculate whether the slope of corresponding edge for blue line is equal to zero (slope at y-intercept where $M_{PQ} = 0$) or does not equal to zero.

For $M_{PQ} = 0$,

$$\text{Penetration depth} = C_{PQ} - C_{i-PQ} \tag{7}$$

For $M_{PQ} \neq 0$,

$$\text{Penetration depth} = \left(\left| C_{i-PQ} - C_{PQ} \right| \frac{1}{\sqrt{X_P^2 + Y_P^2}} \right) = \left(\left| C_{i-PQ} - C_{PQ} \right| \frac{1}{\sqrt{X_Q^2 + Y_Q^2}} \right) \tag{8}$$

b) where $C_{PQ} - C_{i-PQ}$ and $C_{i-PQ} - C_{PQ}$ depend on the penetrated triangle based on M_{PQ} value of Triangle PQR. For case $M_{PQ} \neq 0$, we used ordinary line-to-line distance calculation/formula where X and Y can be replaced with either P or Q vertex point from Triangle PQR.

B. Algorithm for Computing Penetration Depth

Our simplified algorithm to find the penetration depth using DyOP-PD is depicted in Figure 3 (based on Triangle ABC and PQR):

1. Identify the vertex either Triangle ABC or Triangle PQR that penetrate the corresponding edge (Triangle ABC or Triangle PQR).
2. Check penetrating vertex whether it is inside the penetrated triangle. (in our case, Triangle PQR)
    a. Check for penetrated Triangle area (Triangle PQR)
    b. Calculate area of penetrating vertex by using this vertex as one of the main point to calculate new area
    c. Repeated for all three vertices of the penetrated Triangle.
    d. If the sum of all three vertices with penetrating vertex area is equal to penetrated Triangle area, then the vertex is inside the penetrated Triangle.
3. Check $Y_{edge} = M_{edge}X_{edge} + C_{edge}$ (in our case, $Y_{i-PQ} = M_{i-PQ}X_{i-PQ} + C_{i-PQ}$)
4. Match the parallel line equation between Y_{edge} and Y_{i-PQ} where $M_{edge} = M_{i-PQ}$.
5. Find the penetration depth using the formula of

$$\text{Penetration depth} = -\left(\left|C_{i-PQ} - C_{PQ}\right|\frac{1}{\sqrt{X_P^2 + Y_P^2}}\right) = -\left(\left|C_{i-PQ} - C_{PQ}\right|\frac{1}{\sqrt{X_Q^2 + Y_Q^2}}\right)$$

6. End

Figure 3: DyOP-DP algorithm for finding penetration depth between collided primitives.

From Figure 4, list No. 2 is a normal point inside Triangle testing that used the Triangle area to determine whether the corresponding point is inside other triangle. This can be achieved by measuring their area and then compared the sum of area with the point. By referring to Figure 3, we can extend the following No. 2 procedure.

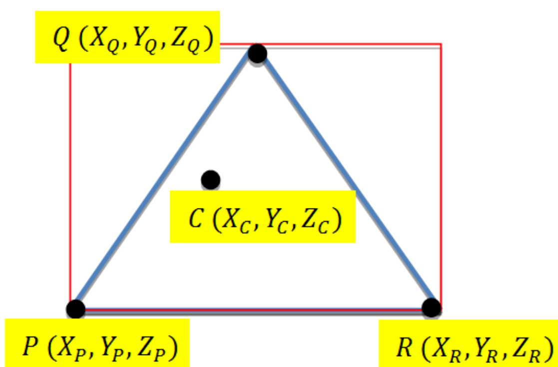


Figure 4: Let vertex C where we need to determine whether the vertex is located inside Triangle PQR

C. Object Setup for Penetration Depth

Our simulation experiment used 10 pre-defined triangle types. In the section, we will briefly explain all the experiment analysis for both DyOP-PD algorithm and Lin-Canny-PD algorithm. Figure 5 shows the corresponding triangles.

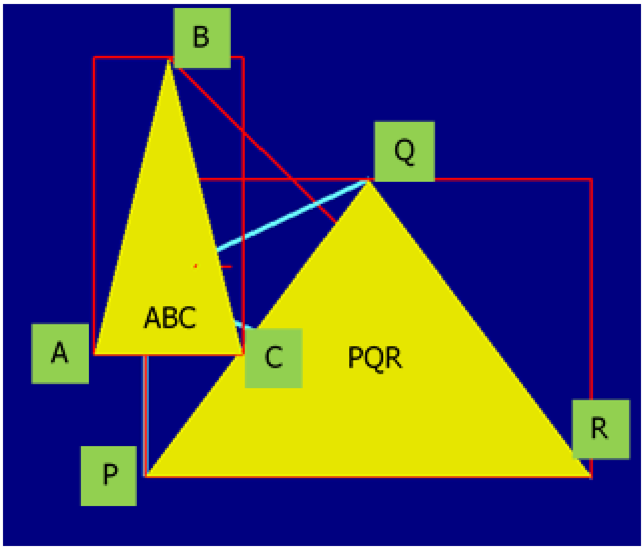


Figure 5: Triangle ABC (Triangle Shape No.2) and Triangle PQR (Triangle Shape No.3)

DyOP-PD detects any potential penetration by calculating a new intercept point by using the equation from the nearest edge of triangle PQR toward the DyOP. At the triangle ABC, the algorithm will choose the nearest vertex to DyOP which means that vertex C and for triangle PQR, edge PQ was the closest one to the DyOP and both will be used in the calculation for new intercept point using equation 9 (based on equation 5):

$$C_{i-PQ} = Y_C - \frac{Y_P - Y_Q}{X_P - X_Q} X_C \tag{9}$$

Then, the algorithm will start to find whether the penetration occurs based on equations 7 and 8. Penetration occurred when the equation 7 (penetration depth) is equal to negative number with the absolute value for the algorithm which is also negative. Positive number of absolute value for the algorithm means that there is no penetration occurs for the corresponding triangle. Equation 7 and 8 rely heavily on each triangle self-testing where each corresponding triangle will check their absolute value before confirming the penetration result. Figure 6 shows the result for DyOP-PD algorithm for this experiment.

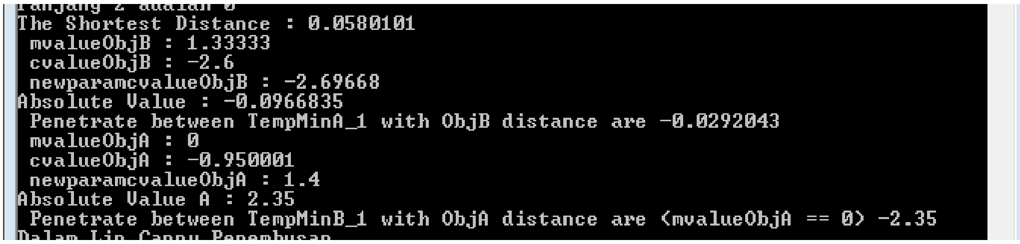


Figure 6: DyOP-PD output for penetration depth experiment

Based on Figure 6, it shows that the *mvalueObjB* was 1.33333 and *cvalueObjB* was -2.6 for the edge of PQ from triangle PQR. *newparamcvalueObjB* represents the C_{i-PQ} where we have obtained -2.69668 (which is a Y intercept point) and the absolute value calculated using $C_{i-PQ} - C_{PQ}$ was a negative value. Thus, the penetration occurred and the penetration depth was recorded at -0.0292043 and confirmed that triangle ABC penetrated the triangle PQR. Meanwhile for the opposite triangle, the absolute value calculated was 2.35 which means no penetration occurred between triangle PQR penetrate into triangle ABC.

RESULTS AND ANALYSIS

This experiment was conducted in closed virtual environment where each variables was assigned in pre-defined movement for the sake of making the experiment simple to analyze and provide only important information. The experiment only handles irregular 2D triangles with variety of sizes. We have implemented two type of techniques DyOP-PD and Lin-Canny-PD for penetration depth check. Figure 7 showed the corresponding result based on this experiment.

Overall time recorded for DyOP-PD technique to compute penetration depth was significantly faster than the Lin-Canny PD technique (refer Figure 7). Our technique was faster than the prominent technique where the computational time significantly reduced, solved a larger fraction of problems, and produced better paths of penetration depth. The lowest result recorded from simulation was 10.22 milliseconds for DyOP-PD and 21.33 milliseconds for Lin-Canny PD technique. The longest time for Lin-Canny technique was 26.89 milliseconds and 11.44 milliseconds for DyOP-PD. The findings proved that DyOP-PD technique was robust to handle efficient, nearly accurate, and fast penetration depth detection compared to Lin-Canny-PD technique. It also showed that our technique is robust solution as compared to Lin-Canny-PD technique.

For every triangle, both techniques are applied together and the system collected the data. Based on the graph, the lowest time only consume 10 milliseconds to all triangles and the highest was recorded at 12 milliseconds for Triangle 4, 5, 6, 9 and 10 by DyOP-PD technique. Meanwhile, Lin-Canny technique recorded the lowest time at 19 milliseconds for Triangle 2 and the highest recorded at 29 milliseconds for Triangle 5. The differences between the lowest time consumed to check penetration between DyOP-PD and Lin-Canny technique are about 62.07% and the highest time consumed differences are 82.93%. Hence, based on this findings, it is confirmed that DyOP-PD shown a significant performance increase in term of checking the penetration with the same accuracy obtained. Although the experiment only conducted in 2D setting, each individual result showed that DyOP-PD technique experienced a significant improvement over the Lin-Canny technique by overall improvement from an average 23.259 milliseconds decrease to 10.765 milliseconds for all 90 simulation steps above. It is almost 53.715% time decreased over the previous technique or 116.06% increased of speed.

CONCLUSION

We have successfully implemented both DyOP-PD and Lin-Canny-PD techniques in order to find the penetration depth case. This superior performance for DyOP technique is achieved by avoiding the process of solving any high degree equation into a relatively simple procedure with simple equation, while Lin-Canny needs to solve a system of multiple testing and variables. Our tests showed that this technique significantly faster as compared to the prominent technique. Moreover, instead of performing narrow phase collision detection, DyOP could help by eliminating the requirement of performing two-phase collision detection processes by performing only a single phase. With this result, the penetration depth support for DyOP technique has been proved.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka and Ministry of Higher Education Malaysia for the financial support given through the research grant RACE/2014/UTEM (A Novel Interference Detection Algorithm Using Planar Subdivision of Axis-Aligned Bounding-Box for Virtual Reality Application).

REFERENCES

Bergen, G. V D. 2001. Proximity Queries and Penetration Depth Computation on 3D Game Objects, in Game Developers Conference.

Kim,Y. J., Otaduy, M. A., M. C. Lin & D. Manocha. 2003. Fast Penetration Depth Estimation Using Rasterization Hardware And Hierarchical Refinement. Presented at the Proceedings of The Nineteenth Annual Symposium On Computational Geometry, San Diego, California, USA.

Redon, S., Kheddar, A. & Coquillart. S., 2002. Fast Continuous Collision Detection between Rigid Bodies. Computer Graphics Forum, 21: 279-287.

Shengzheng, W. & Jie, Y. 2009. Efficient Collision Detection for Soft Tissue Simulation In A Surgical Planning System. Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09. 11th IEEE International Conference on, 49-53.

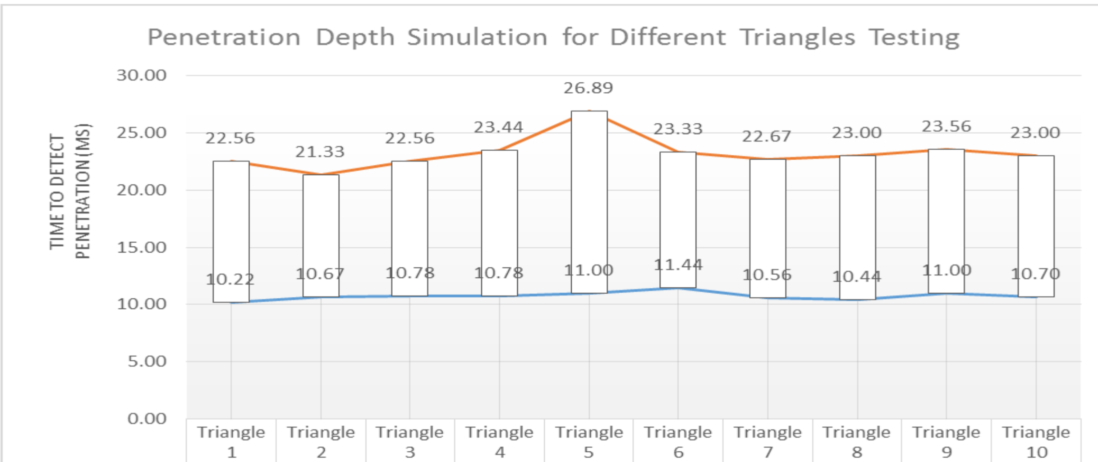
Stephane, R. & Lin, M. C. 2006. A Fast Method for Local Penetration Depth Computation. Journal of Graphics Tools.

Sulaiman, H. A., Othman, M. A., Ismail, M. M, Misran, M. H., Said, M. A., B. M., Ramlee, R. A. 2013. Quad Separation Algorithm for Bounding-Volume Hierarchies Construction In Virtual Environment Application. Journal of Next Generation Information Technology, 4: 63-73

Zhang, L., Kim Y. J., Varadhan, G. & Manocha, D. 2007. Generalized Penetration Depth Computation. Computer-Aided Design, 39(8): 625-638.

Zhang, X., Kim, Y. J. & Manocha, D. 2014. Continuous Penetration Depth. Computer-Aided Design, 46: 3-13

APPENDICES



	Triangle 1	Triangle 2	Triangle 3	Triangle 4	Triangle 5	Triangle 6	Triangle 7	Triangle 8	Triangle 9	Triangle 10
DyOP	10.22	10.67	10.78	10.78	11.00	11.44	10.56	10.44	11.00	10.70
Lin Canny	22.56	21.33	22.56	23.44	26.89	23.33	22.67	23.00	23.56	23.00

Figure 7: Averaged simulation graph of n-body triangles for nine random triangles that were tested against each other in time to detect penetration depth for Lin-Canny technique and DyOP technique.