Theoretical Fiction or Practical Use

ALGORITHMS COMPLEXTITY AND NON-LINEAR CO-ORDINATES*

Vaclav Skala*

Department of Computer Science and Engineering, University of West Bohemia, Univerzitní 8, Box 314
306 14 Plzen, Czech Republic skala@kiv.zcu.cz http://iason.zcu.cz/~skala

ABSTRACT

There are some applications, where the polar, cylindrical or spherical co-ordinate systems can be used for finding a solution of the given problem. Especially some technical problems [Lio97a], like sonar and radar applications, where the distance from an object is measured under known angles, flow computation, radiation, medical imaging and ultrasound imaging etc. could benefit from their use.

It is a usual practice that all the data from those applications are transformed to the Cartesian orthogonal co-ordinate system, where all data are processed. The data are then displayed directly or transform back to the original co-ordinate system. Nevertheless it is well known that representation of a point in E^2 is different from a line representation and therefore the processing pipeline have to respect this fact. The polar, cylindrical and spherical co-ordinate systems offer some possibilities how to handle graphical information in an unambiguous way and also make an effective processing. On the other hand, it is necessary to say that in usual practical graphical applications the direct use of non-linear co-ordinates can be quite complicated can hopefully lead to new understanding of some fundamental algorithms and developing of new more effective methods.

Keywords: algorithm complexity, computer graphics, geometric transformation, non-linear co-ordinate system, point-in-polygon, line clipping.

1. CARTESIAN CO-ORDINATE SYSTEM

Cartesian co-ordinates and point representations are used nearly exclusively. In the majority of applications the homogeneous co-ordinates are used and a point is represented as $[\,x\,,\,y\,,\,w\,]^T$, where: w is the homogeneous co-ordinate. The problem arises when we need to manipulate with the lines. How can we handle them easily? Let us consider a point **x** and its co-ordinates in the polar co-ordinate system $\mathbf{x} = [\,\rho\,,\,\varphi\,]^T$. This representation enables us to represent a point unambiguously and define operations like 'move' and 'rotate'. There is a problem with concatenation of several geometric transformations, indeed. Also if we want to display geometric primitives, e.g. points, in the Cartesian co-ordinate space, it will lead to expensive *cos* and *sin* function computations.

It can be seen that every point A defines also unambiguously a line p that is orthogonal to the line defined by this point and by the origin of the polar co-ordinate system, see fig.1. It is well known, that the geometric transformations in the homogeneous co-ordinates can be represented by matrices generally, e.g. $\mathbf{x}' = \mathbf{Q}\,(\alpha, a\,)\,\mathbf{x}$, where α stands for rotation, a stands for translation. But what does these geometric transformations mean for applications in the field of radar signal processing where the polar representation is native?

Of course there are unambiguous transformations to the polar and from the polar to the Cartesian co-ordinate systems. Therefore the question whether similar operations for geometric transformations can be defined for the polar co-ordinate system, what would be properties and possible use should be answered.

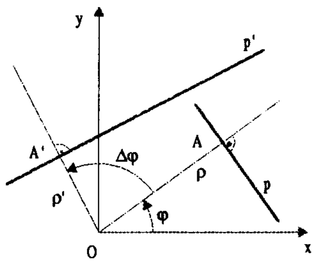

Figure 1

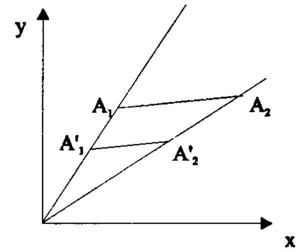

Figure 2

2. POLAR CO-ORDINATE SYSTEM

It can be seen that the point A is unambiguously represented by the vector $[\,\rho\,,\,\cos\varphi\,,\,\sin\varphi\,]^T$ as well as the line p. The line p is defined as a line passing the point A that is orthogonal to the line OA, where O stands for the origin of the polar co-ordinate system. To be able to represent geometric transformations by matrix operations we introduce "polar homogeneous co-ordinates" of the point as $[\,\rho\,,\,1\,,\,\cos\varphi\,,\,\sin\varphi\,]^T$. As the "1" stays for homogeneous co-ordinate. In the polar co-ordinate system the radial displacement, analogue to the translation, resp. scaling and rotation, the matrix is defined as

$$\begin{bmatrix} \rho' \\ 1 \\ \cos\varphi' \\ \sin\varphi' \end{bmatrix} = \begin{bmatrix} \lambda & r & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\Delta\varphi & -\sin\Delta\varphi \\ 0 & 0 & \sin\Delta\varphi & \cos\Delta\varphi \end{bmatrix} \begin{bmatrix} \rho \\ 1 \\ \cos\varphi \\ \sin\varphi \end{bmatrix} \qquad \text{i.e. } \mathbf{x}' = \mathbf{Q}\,(\,r\,,\lambda\,,\,)\,\mathbf{x}$$

where: r is the translation parameter, resp. λ moves points λ times and $\Delta\varphi$ for a rotation. It should be noted, that the translation operation in polar co-ordinate system is something "strange" as if we move points, let us say the end-points of the line segment A_1A_2, we will get a line segment $A'_1A'_2$, see fig.2. We can imagine the radial displacement as a translation operation in some sense because it is the operation when objects move forward to or backward from an observer in the origin of the polar system. The rotation operation is defined in a similar way as the rotation in the Cartesian co-ordinate system. If we compare the structure of the matrix $\mathbf{Q}$ with the structure of a matrix for a transformation in homogeneous co-ordinate system we can see that the matrix $\mathbf{Q}$ has a simple structure. A user can easily recognize single transformation parameters, that is not so easy in the homogeneous co-ordinate system representation. We can see that each transformation parameter has a 2 x 2 block of values in the matrix $\mathbf{Q}$. In case that we need the translation operation known from the Cartesian homogeneous co-ordinate representation in E^2 we can use also a different representation, see [Ska2000a] for details, where is also extension to E^3 space, too. It is necessary to note, that there is a possibility to use concatenation to express more complex transformations.

3. APPLICATIONS OF THE NON-LINEAR CO-ORDINATE SYSTEMS

There are some possible applications that could use an advantage of the polar, cylindrical or spherical representations in some extent. Nevertheless these models seem to be useful especially for non-traditional approach to the algorithm design. Let us imagine two simple problems like:

- the point-in-convex polygon test, usually solved in $O(N)$ [Ska93a], resp. $O(\lg N)$ steps,
- the line clipping by convex polygon often solved by the Cyrus-Beck's algorithm that has $O(N)$ complexity, see [Ska94b], [Bui97a], or $O(\lg N)$ algorithm can be considered [Ska94a], [Bui99a], see fig.3.

where: N is a number of edges of the given polygon. It can be seen that the both problems are dual in some sense, see [Kol94a]. The principle of duality is well known and used often [Joh96a].

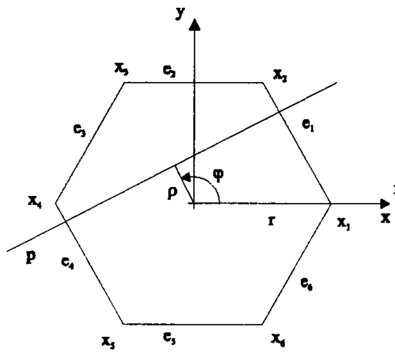

Figure 3

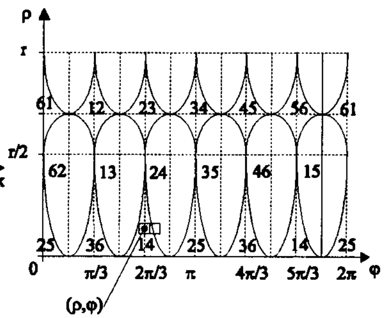

Figure 4

The line clipping by convex polygon in E^2, see fig.3, is a problem when we want to find a part of the line p that is inside of the given polygon if any. If we transform the convex polygon to the polar co-ordinate system representation we get representation in (ρ,φ) co-ordinates, see fig.4, where each region represents the a set of (ρ,φ) values of the clipped line that intersect the same edges, in our case edges e_1 and e_4.

It can be seen that if the space subdivision technique is used we can directly determine edges that are intersected by the given line p regardless to the number of edges of the given polygon for infinite subdivision in (ρ,φ) space. It means that if the subdivision is coarser we will have to test more than two edges, see fig.3. Similar approach has been recently developed in the cartesian co-ordinate space, verified and tested, see [Ska94c], [Ska96b], [Ska96e], and extended to E^3 case [Ska96d], too.

3. CONCLUSION AND FUTURE WORK

In the previous sections we discussed properties of the polar co-ordinate system in E^2, of the cylindrical and spherical co-ordinate systems in E^3, representation of points, geometric transformations, representation of concatenated geometric transformations and some possible applications of the proposed representations. The advantage of the proposed approach is that geometric transformations are handled in a unique way using matrix multiplication. This representation gives also a very simple way how to handle widgets or cones that might be very useful in some cases.

The most important result is that this approach gives a quite different view on some problems to be solved. It is expected that non-linear co-ordinate systems with application of dual representation principles can lead to new, more simple, more robust and faster algorithms.

There are two direct results for N-sided convex polygon:

- The test point-in-convex polygon is of $O(\lg N)$ complexity. We can reach **$O(1)$ run-time complexity**, if we use a pre-processing of $O(N M)$ complexity, where: M defines number of wedges to be pre-computed and **depends on geometric properties of the given polygon** $M \to \infty$. This is in correlation with previously obtained results.
- The line clipping against convex polygon is of $O(\lg N)$ complexity, see [Ska94a], [Ska94b], [Bui99a]. It was shown that it is possible to reach **$O(1)$ run-time complexity**, if we use a pre-processing of $O(N^2 M R)$ complexity where: M, resp. R define number of subdivision for $\Delta\varphi$, resp. $\Delta\rho$ to be pre-computed and **depend on geometric properties of the given polygon** $M \to \infty$, $R \to \infty$. This is in correlation with previously obtained results, see [Ska96b], [Ska96c], [Ska96e] and with extensions to E^3 case, too.

There are some very important questions from the algorithm complexity field.

- What is the relation between the complexity of optimal algorithm and pre-processing and run-time complexities?
- What is the lowest bound for the run-time complexity for a given problem and what will be the lowest pre-processing complexity for this?
- Can be the above-mentioned approach extended to E^3 case, for details see [Ska2000a]?

There is a hope that the above shown principles can be used especially for problems in E^3 [Ska96a], [Ska97a] and problems connected to line clipping, intersection computation, ray tracing methods and others.

As the memory capacity and speed of hardware graphics accelerator grow fast it can be reasonable to reconsider fundamental functionality of geometric engines, too. Formulation of new functionality for the graphics and visualization pipeline might be useful, as it is limited generally to the dot vector multiplication nowadays.

ACKNOWLEDEMENTS

The author would like to thanks to all who contributed to this work, especially to MSc. and PhD. students at the University of West Bohemia in Plzen that have stimulated this thoughts and development of new algorithms based on it. This paper benefited from several discussions with them a lot. Special thanks belong to Dr.A.Kolcun, to anonymous reviewers of this paper and recent reports as they shared some valuable insights on this problem solution. Theirs invaluable critical comments and suggestions improved the manuscript significantly.

REFERENCES

[Bui97a] Bui,D.H., Skala,V.: Fast Algorithms for Line Segment and Line Clipping in E^2, The Visual Computer, Springer Verlag, 1997.

[Bui99a] Bui,D.H., Skala,V.: New Fast Line Clipping Algorithm in E^2 with O(lg N) Complexity, Int.Conf. SCCG'99, Budmerice, Slovak Republic, pp.221-228, 1999.

[Joh96a] Johnson,M.: Proof by Duality: or the Discovery of "New" Theorems, Mathematics Today, November/December 1996.

[Kol94a] Kolingerova, I.: Dual Representation and its Use in Computer Graphics, PhD Thesis, University of West Bohemia, Plzen, 1994.

[Lio97a] Lionello,R.: Three Dimensional Imaging for Magnetohydrodynamic Computations, WSCG'97 Int.Conf.Proceedings, Univ.of West Bohemia, Plzen pp.282-289, 1997

~~[Rek69a] Rektorys,K. & others: Survey of Applicable Mathematics, Illife Books Ltd., Dorset House, Stamford Street, London SE1, ISBN 592 03927 7, U.K.~~

[Ska93a] Skala,V.: An Efficient Algorithm for Line Clipping by Convex polygon, Computers & Graphics, Vol.17, No.4, pp.417-421, 1993.

[Ska94a] Skala,V.: O(lg N) Line Clipping Algorithm in E^2, WSCG'94 Int.Conf. Proceedings, Univ.of West Bohemia, Plzen, pp.174-191, 1994.

[Ska94b] Skala,V.: O(lg N) Line Clipping Algorithm in E^2, Computers & Graphics, Pergamon Press, Vol.18, No.4., 1994.

[Ska94c] Skala,V., Kolingerova,I., Blaha,P.: A Comparison of 2D Line Clipping Algorithms, Machine Graphics & Vision, Vol.3, No.4, pp.625-633, 1994.

[Ska94d] Skala,V.: Point-in-Polygon with O(1) Complexity, TR 68/94, University of West Bohemia, Plzen, 1996.

[Ska96a] Skala,V.: An Efficient Algorithm for Line Clipping by Convex and Non-Convex Polyhedra in E^3, Computer Graphics Forum, Vol.15, No.1, pp.61-68, 1996.

[Ska96b] Skala,V.: Line Clipping in E^2 with O(1) Processing Complexity, Computers & Graphics, Vol.20, No.4, pp.523-530, 1996.

[Ska96c] Skala,V., Lederbuch,P. ,Sup,B.: A Comparison of O(1) and Cyrus-Beck Line Clipping Algorithm in E^2 and E^3, SCCG96 Conference proceedings, Comenius Univ. Bratislava, Slovak Republic, pp.27-44, 1996.

[Ska96d] Skala,V.: Line Clipping in E^3 with Expected Complexity O(1), Machine Graphics and Vision, Poland Academy of Sciences, Vol.5, No.4, pp.551-562, 1996.

[Ska96e] Skala,V., Lederbuch,P.: A Comparison of a New O(1) and the Cyrus-Beck Line Clipping Algorithms in E^2, in COMPUGRAPHICS'96 Int.Conf., Paris, 1996.

[Ska97a] Skala,V.: A Fast Algorithm for Line Clipping by Convex polyhedron in E^3, Computers & Graphics, No.2, Vol.21, pp.209-214, 1997.

~~[Ska99a] Skala,V.: Non-orthogonal Co-ordinates and Their Application in Computer Graphics, CE&I'99 International Conference proceedings, Kosice, Slovakia, s. 212-217, 1999.~~

[Ska2000a] Skala,V.: Non-orthogonal Co-ordinates - Theoretical Fiction?, University of West Bohemia Technical Report , 2000.

---

[i] This work was supported by the Ministry of Education of the Czech Republic - projects A2030801 and VS 97155