

Referências Bibliográficas

- Abraham, F., Celes, W., Cerqueira, R., Campos, J.L.. 2004. A load-balancing strategy for sort-first distributed rendering. Brazilian Symposium on Comp. Graphics and Image Processing, pp. 292-299.
- Alembic, 2010. Open framework for storing and sharing scene data. Disponível em: <http://www.alembic.io/> [Acessado em 20/05/2013].
- Allard, J.; Raffin, B., 2005. A shader-based parallel rendering framework, Visualization, 2005. VIS 05. IEEE , vol., no., pp.127,134, 23-28 Oct. 2005 doi: 10.1109/VISUAL.2005.1532787
- Akenine-Möller, T., Haines, E. and Hoffman, N., 2008. Real-time rendering. 3rd Ed, A K Peters.
- Autodesk, 2003. BackBurner: Render Farm Manager. Disponível em: <http://knowledge.autodesk.com/support/3ds-max/downloads/caas/downloads/content/backburner-2015-windows-version-for-3ds-max3ds-max-design-2015.html> [Acessado 10/05/2012].
- Barringer, R. and Akenine-Möller, T.. 2014. Dynamic ray stream traversal. ACM Trans. Graph. 33, 4, Article 151 (July 2014), 9 pages. DOI=10.1145/2601097.2601222 <http://doi.acm.org/10.1145/2601097.2601222>
- Bittner, J., Hapala, M., and F., H. 2013. Fast insertion-based optimization of bounding volume hierarchies. Computer Graphics Forum 32, 1, 85–100.
- Bradski, G., 2000. The OpenCV Library, Dr. Dobb's Journal of Software Tools.
- Chaos Group, 2006. V-Ray – RayTracer. Disponível em: <http://www.chaosgroup.com/en/2/vray.html> [Acessado 10/05/2012].
- Christensen, P.H., Laur, D.M., Fong, J., Wooten, W.L., and Dana Batali, D. 2003. Ray differentials and multiresolution geometry caching for distribution ray tracing in complex scenes. Computer Graphics Forum (Proc. of Eurographics 2003), 22(3), pp. 543–552.

- Christensen P. H., Fong J., Laur D. M., Batali D. 2006. Ray tracing for the movie 'Cars'. In Proc. IEEE Symposium on Interactive Ray Tracing (2006), pp. 1–6.
- Christensen, P. H. 2008. Point-Based Approximate Color Bleeding. Pixar Technical Memo 08 - 01 (Disponível em: <http://graphics.pixar.com/library/PointBasedColorBleeding/paper.pdf>). [Acessando em 13/02/2012]
- Cohen, M. F. & Greenberg, D. P. 1985. The hemi-cube: a radiosity solution for complex environments. In Proceedings of the 12th annual conference on Computer graphics and interactive techniques (SIGGRAPH '85). ACM, New York, NY, USA, 31-40. DOI=10.1145/325334.325171 <http://doi.acm.org/10.1145/325334.325171>
- Cook, R. L., Carpenter, L., And Catmull, E. 1987. The Reyes image rendering architecture. In SIGGRAPH'87, 95–102.
- Cook, R. L., Shade Trees, Computer Graphics (SIGGRAPH '84 Proceedings), pp. 223-231, July 1984.
- Crane, K., 2006. Importance Sampling for Monte Carlo Ray Tracing. Disponível em: <http://www.cs.caltech.edu/~keenan/mc419.pdf> [Acessado em 18/06/2010].
- Dammertz, H., Keller, A., And Lensch, H. P. A. 2010. Progressive point-light-based global illumination. Computer Graphics Forum 29, 8.
- Dean, J. & Ghemawat, S. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107-113. DOI=10.1145/1327452.1327492 <http://doi.acm.org/10.1145/1327452.1327492>
- Foley, Tim and Hanrahan, Pat. 2011. Spark: modular, composable shaders for graphics hardware. ACM Trans. Graph. 30, 4, Article 107 (July 2011), 12 pages. DOI=10.1145/2010324.1965002 <http://doi.acm.org/10.1145/2010324.1965002>

- Foley, T.; Sugerman, J. 2005. Kd-tree acceleration structures for a GPU Raytracer. In: HWWS '05: Proceedings of The ACM Siggraph/Eurographics Conference on Graphics Hardware, p. 15-22, New York, NY, USA, 2005. ACM.
- Fernando, R. 2004. GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics. Pearson Higher Education.
- Gaggioni, H., Dhanendra, Yamashita, J., Kawada, N., Endo, K. Clark, C., 2012. S-Log: A new LUT for digital production mastering and interchange applications. Sony SLog Manual. Disponível em: https://pro.sony.com/bbsccms/assets/files/mkt/cinema/solutions/slog_manual.pdf Acessado em: 29/03/2012.
- Garanzha, Kirill, Pantaleoni, Jacopo, McAllister, David K.: Simpler and Faster HLBVH with Work Queues. High Performance Graphics 2011: 59-64
- Günther, J.; Popov, S.; Seidel, H.-P. ; Slusallek, P. Realtime ray tracing on GPU with BVH-based packet traversal. In: Proceedings of The IEEE/Eurographics Symposium on Interactive Ray Tracing 2007, p. 113-118, Sept. 2007.
- Guntury, S. and Narayanan , P. J.. 2010. Ray tracing dynamic scenes with shadows on GPU. In Proceedings of the 10th Eurographics conference on Parallel Graphics and Visualization (EG PGV'10), James Ahrens, Kurt Debattista, and Renato Pajarola (Eds.). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 27-34. DOI=10.2312/EGPGV/EGPGV10/027-034 <http://dx.doi.org/10.2312/EGPGV/EGPGV10/027-034>
- Havran, V. Heuristic Ray Shooting Algorithms. 2000. MSc thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague.
- Havran, V. 2007. About the relation between spatial subdivision and object hierarchies used in ray tracing. In Proceedings of the Spring Conference on Computer Graphics (SCCG).

- Heckenberg, D., Saam, J., Doncaster, C, Cooper, C. 2011, Deep Compositing, Animal Logic White Paper.
- Herrera, A. 2014. NVidia Grid: Graphics Accelerated VDI with the Visual Performance of a Workstation. NVidia White Paper. Disponível em: http://www.nvidia.com/content/grid/resources/White_paper_graphics_accelerated_VDI_v1.pdf [Acessado 20/05/2013].
- Horn, B.K.P. & Schunck, B.G. 1980. Determining optical flow. *Artificial Intelligence*, vol 17, pp 185–203, 1980.
- Horn, D. R.; Sugerman, J.; Houston, M. ; Hanrahan, P. 2007. Interactive k-d tree gpu raytracing. In: *I3D '07: Proceedings of The 2007 Symposium on Interactive 3d Graphics And Games*, p. 167-174, New York, NY, USA, 2007. ACM.
- Hou,Q., Qin,H., Li,W., Guo,B. and Zhou,K. 2010. Micropolygon ray tracing with defocus and motion Blur. *ACM Transaction on Graphics*, SIGGRAPH 2010. [Disponível em: <http://www.kunzhou.net/2010/mptracing.pdf>].
- Intel, 2012. Embree : Ray Tracer Engine. Disponível em: <https://embree.github.io> [Acessado 10/05/2012].
- Intel, 2015. Intel® Parallel Studio XE 2015. Disponível em: <https://software.intel.com/en-us/intel-parallel-studio-xe/details> [Acessado 10/05/2015].
- Ivson, P. N. S., 2009. Ray Tracing Dynamic Scenes on the GPU. MSc Thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Ivson, P. N. S., Duarte, L. and Celes, W., 2009. GPU-Accelerated Uniform Grid Construction for Ray Tracing Dynamic Scenes. *Monografias em Ciência da Computação*, no. 14/09, ISSN 0103-9741. Pontifícia Universidade Católica do Rio de Janeiro.
- J., H. W. et al, 2003. Monte Carlo Ray Tracing, course notes, Siggraph 2003 Course 44. Disponível em: <http://www.cs.rutgers.edu/decarlo/readings/mcrt-sg03c.pdf> [Acessado em 18/06/2012].

- Jevans, D. Wyvill, B., 1989. Adaptive voxel subdivision for ray tracing. In Proc. of Graphics Interface '89. p. 164-72.
- Kensler, A. 2008. Tree rotations for improving bounding volume hierarchies. In Proc. IEEE Symposium on Interactive RayTracing, 73–76
- Klimaszewski, K. S., 1995. Faster ray tracing using adaptive grids and area sampling. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-14329., Brigham Young University.
- Klimaszewski, K. S.; Sederberg, T., 1997. W. Faster ray tracing using adaptive grids. IEEE Computer Graphics and Applications, 17(1):42-51.
- Kondermann, D., Abraham, S., Brostow, G., Förstner, W., Gehrig, S., Imiya, A., Jähne, B., Klose, F., Magnor, M., Mayer, H., Mester, R., Pajdla, T., Reulke, R., and Zimmer, H.. 2011. On performance analysis of optical flow algorithms. In *Proceedings of the 15th international conference on Theoretical Foundations of Computer Vision: outdoor and large-scale real-world scene analysis*, Frank Dellaert, Jan-Michael Frahm, Marc Pollefeys, Laura Leal-Taixé, and Bodo Rosenhahn (Eds.). Springer-Verlag, Berlin, Heidelberg, 329-355. DOI=10.1007/978-3-642-34091-8_15 http://dx.doi.org/10.1007/978-3-642-34091-8_15
- Kolb, C., Mitchell, D. & Hanrahan, P., 1995. A Realistic Camera Model for Computer Graphics. SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. Pages 317 – 324. ISBN:0-89791-701-4 doi>10.1145/218380.218463
- Kollig, T & Keller, A., 2004. Illumination in the Presence of Weak Singularities. Monte Carlo and Quasi-Monte Carlo Methods 2004, Springer-Verlag, Berlin.
- Krivanek, J., Gautron, P. 2009. Practical Global Illumination With Irradiance Caching. 1st Ed, Morgan & Claypool Publishers, Synthesis Lectures on Computer Graphics And Animation (Barsky, B. A. Editor). ISBN 9781598296440. DOI: 10.2200/S00180ED1V01Y200903CGR010

- Lauterbach, C., Garland, M., Sengupta, S., Luebke, D. & Manocha, D. 2009. Fast BVH Construction on GPUs, In: Proceedings of Eurographics '09.
- Lucas Digital Ltd, 2006. Technical Introduction to OpenEXR. Available from: <http://www.openexr.com/TechnicalIntroduction.pdf> [Accessed 20/05/2013].
- Lucas, B. D. & Kanade, T. 1981, An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pages 121-130.
- Lokovic, T., And Veach, E. 2000. Deep shadow maps. SIGGRAPH Computer Graphics, 385–392.
- Luebke, D., Reddy, M., Cohen, J. D., Varshney, A., Watson, B., Huebner R., 2002. Level of Detail for 3D Graphics. 1st Ed, The Morgan Kaufmann Series in Computer Graphics.
- Machado, L.E., 2010. Parallel game engine techniques for multiprocessors. Ph.D. Thesis. Pontifícia Universidade Católica do Rio de Janeiro.
- Madeira, D. , Montenegro, A., Clua, E., Lewiner, T., 2009. GPU Octrees and Optimized Search. In: Proc. VIII Brazilian Symposium on Games and Digital Entertainment.
- Marques, Rodrigo, Feijo, Bruno, Breitman, Karin, Gomes, Thieberson, Ferracioli, Laercio, Lopes, Hélio, A cloud computing based framework for general 2D and 3D cellular automata simulation, Advances in Engineering Software, Volume 65, November 2013, Pages 78-89, ISSN 0965-9978, <http://dx.doi.org/10.1016/j.advengsoft.2013.05.014>.
- Mcguire, M. et al. Abstract Shade Trees, ACM Symposium on Interactive 3D Graphics and Games (I3D 2006), pp. 79-86, Março 2006.
- Molnar, Steven, Cox, Michael, Ellsworth, David, and Fuchs, Henry. 1994. A Sorting Classification of Parallel Rendering. *IEEE Comput. Graph. Appl.* 14, 4 (July 1994), 23-32. DOI=10.1109/38.291528 <http://dx.doi.org/10.1109/38.291528>

- Mono, 2001. Mono Project History. Disponível em: <http://www.mono-project.com/docs/about-mono/history/> [Acessado 10/05/2012].
- Nam, B. and Sussman, A. 2004. A Comparative Study of Spatial Indexing Techniques for Multidimensional Scientific Datasets. In Proc. 16th Int. Conf. on Scientific and Statistical Database Management (SSDBM 2004), pp. 171-180.
- NVidia, 2011. Optix : Ray Tracer Engine. Disponível em: <http://www.nvidia.com/object/optix.html> [Acessado 10/05/2012].
- Novák, J., Nowrouzezahrai, D., Dachsbacher, C. & Jarosz, W. 2012. Virtual ray lights for rendering scenes with participating media. In ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012). v. 31, pp. 60-71.
- Ogre, 1999. Brief history of OGRE . Disponível em: <http://www.ogre3d.org/tikiwiki/tiki-index.php?page=Brief+history+of+OGRE> [Acessado 10/05/2012].
- Pantaleoni, J. and Luebke, D.. 2010. HLBVH: hierarchical LBVH construction for real-time ray tracing of dynamic geometry. In *Proceedings of the Conference on High Performance Graphics* (HPG '10). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 87-95.
- Parker, Steven G., Bigler, James, Dietrich, Andreas, Friedrich, Heiko, Hoberock, Jared, Luebke, David, McAllister, David, McGuire, Morgan, Morley, Keith, Robison, Austin and Stich, Martin. 2010. OptiX: A General Purpose Ray Tracing Engine.. ACM Transactions on Graphics.
- Parker, Steven G., Friedrich, Heiko, Luebke, David P., Morley, R. Keith, Bigler, James, Hoberock, Jared, McAllister, David K., Robison, Austin, Dietrich, Andreas, Humphreys, Greg, McGuire, Morgan, Stich, Martin: GPU ray tracing. Commun. ACM 56(5): 93-101 (2013)
- Pharr, M. & Fernando, R. 2005. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems). Addison-Wesley Professional.

- Pharr, M., Humphreys, G. 2010. Physically Based Rendering, From Theory to Implementation. 2nd Ed, Morgan Kaufmann. ISBN: 9780123750792
- PipelineFX, 2012. Qube!: Render Farm Manager. Disponível em: <https://pipelinefx.com> [Acessado 10/05/2012].
- Pixar, 1970. A Brief Introduction To RenderMan - Pixar. Disponível em: <http://renderman.pixar.com/view/brief-introduction-to-renderman> [Acessado 10/05/2012].
- Pixar, 2006. Cars. Disponível em: <http://www.pixar.com/featurefilms/cars/> [Acessado 18/06/2012].
- Pixar, 2010. Tractor: Renderman Job distribution. Disponível em: <http://renderman.pixar.com/view/pixars-tractor> [Acessado 10/05/2012].
- Pixar, 2011. Oscar - Scientific & Engineering Award. Available from: <http://renderman.pixar.com/view/renderman-awards> [Accessed 15/08/14].
- Popov, S.; Günther, J.; Seidel, H.-P. ; Slusallek, P. Stackless kd-tree traversal for high performance gpu ray tracing. Computer Graphics Forum, 26(3):415-424, September 2007.
- Purcell, T. J.; Buck, I.; Mark, W. R. ; Hanrahan, P. Ray tracing on programmable graphics hardware. ACM Transactions on Graphics, 21(3):703-712, July 2002. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).
- Reinhard, E., Smits, B. and Hansen, C., 2000. Dynamic Acceleration Structures for Interactive Ray Tracing. In Proceedings of the Eurographics Workshop on Rendering. Disponível em: <http://www.cs.utah.edu/~reinhard/papers/egwr2k.pdf> [Acessado 18/06/2012].
- Ridolfi, L. 2012. Construção de Espaços de Cor Euclidianos e Perceptualmente Uniformes com base na Fórmula CIEDE2000. Tese de doutorado do Programa de Pós-Graduação da Pontifícia Universidade Católica do Rio de Janeiro.

- Roccia, J., Paulin, M. and Coustet, C., 2012. Hybrid CPU/GPU KD-Tree Construction for Versatile Ray Tracing, Proceedings of Eurographics Symposium on Rendering.
- Sabino, T. L. R. ; Andrade, P. F. ; Clua, E. W. G. ; Montenegro, A. A. ; Pagliosa, P. . A Hybrid GPU Rasterized and Ray Traced Rendering Pipeline for Real Time Rendering of Per Pixel Effects. Lecture Notes in Computer Science, v. 7522, p. 292-305, 2012.
- Samet, H. 1984. The Quadtree and Related Hierarchical Data Structures. *ACM Comput. Surv.* 16, 2 (June 1984), 187-260. DOI=10.1145/356924.356930 <http://doi.acm.org/10.1145/356924.356930>
- Schlick, C. 1994. An Inexpensive BRDF Model for Physically-based Rendering. *Computer Graphics Forum* 13 (3): 233. doi:10.1111/1467-8659.1330233
- Shevtsov, M., Soupikov, A., and Kapustin, A. 2007. Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. *Comput. Graph. Forum* 26, 3, 395 404.
- Sintorn, E., Kämpe, V., Olsson, O. & Assarsson, U.. 2014. Compact precomputed voxelized shadows. *ACM Trans. Graph.* 33, 4, Article 150 (July 2014), 8 pages. DOI=10.1145/2601097.2601221 <http://doi.acm.org/10.1145/2601097.2601221>
- Squyres, J. 2008. Open MPI: 10¹⁵ Flops Can't Be Wrong. In *Supercomputing 2008 Poster Proceedings*. Disponível em: <http://www.open-mpi.org/papers/sc-2008/jsquyres-cisco-booth-talk-1up.pdf> Acessado em: 12/04/2014.
- Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; Wiegand, T. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology (IEEE)* 22 (12). [Acessado 22/03/2013].
- Stich M., Friedrich, H., & Dietrich, A.. 2009. Spatial splits in bounding volume hierarchies. In *Proceedings of the Conference on High*

- Performance Graphics 2009 (HPG '09), Stephen N. Spencer, David McAllister, Matt Pharr, and Ingo Wald (Eds.). ACM, New York, NY, USA, 7-13. DOI=10.1145/1572769.1572771
<http://doi.acm.org/10.1145/1572769.1572771>
- Szirmay-Kalos, L., Szécsi, L. & Sbert, M. 2008. GPU Based Techniques for Global Illumination Effects. 1st Ed, Morgan & Claypool Publishers, Synthesis Lectures on Computer Graphics And Animation (Barsky, B. A. Editor). ISBN 1598295594. DOI: 10.2200/S00107ED1V01Y200801CGR004
- Talbot, J. F., 2005. Importance Resampling for Global Illumination, MSc Thesis, Brigham Young University. Available from: <http://research.justintalbot.org/papers/thesis.pdf> [Accessed 18/Jun/10].
- Tanenbaum, A. S. 2004. Sistemas Operacionais Modernos. São Paulo : Prentice-Hall.
- The Foundry, 2011. Katana. Disponível em: <https://www.thefoundry.co.uk/products/katana/> [Acessado 10/05/2012].
- ThinkBox, 2008. Deadline: Render Farm Management System. Disponível em: <http://www.thinkboxsoftware.com/deadline/> [Acessado 10/05/2012].
- Veach, E., 1997. Robust Monte Carlo Methods for Light Transport Simulation Eric, PhD Dissertation, Stanford University, Available from: http://graphics.stanford.edu/papers/veach_thesis/ [Acessado 18/06/2012].
- Wald, I., Benthin, C. & Slusallek, P. 2003. Interactive Global Illumination in Complex and Highly Occluded Environments. In Proceedings of the 14th EUROGRAPHICS Symposium on Rendering, P.H. Christensen and D. Cohen-Or (2003), 74-81
- Wald, I. 2004. Realtime ray tracing and interactive global illumination. PhD thesis, Saarland University.

- Wald, I., Ize, T., Kensler, A., Knoll, A., and Parker, S. G. 2006. Ray Tracing Animated Scenes using Coherent Grid Traversal. *ACM Transactions on Graphics*, 485-493. *Proceedings of ACM SIGGRAPH 2006*.
- Wald, I., Boulos, S., Shirley, P. 2007. Ray Tracing Deformable Scenes using Dynamic Bounding Volume Hierarchies. *ACM Transactions on Graphics* 26, 1 (2007), 1–18.
- Wald, I., Mark, W.R. Günther, J., Boulos, S., Ize, T., Hunt, W., Parker, S.G. and Shirley, P. 2007. State of the art in ray tracing animated scenes. In D. Schmalstieg and J. Bittner (Eds), *STAR Proceedings of Eurographics 2007*, pp. 89-116.
- Wald, I. 2010. The Utah 3D Animation Repository: Fairy Forest Available from: <http://www.sci.utah.edu/~wald/animrep> [Acessado 18/06/2013].
- Wald, I., Woop, S., Benthin, C., Johnson, G. S., & Ernst, M. 2014. Embree: a kernel framework for efficient CPU ray tracing. *ACM Trans. Graph.* 33, 4, Article 143 (July 2014), 8 pages. DOI=10.1145/2601097.2601199 <http://doi.acm.org/10.1145/2601097.2601199>
- Walter, B., Drettakis, G. & Parker, S.. 1999. Interactive rendering using the render cache. In *Proceedings of the 10th Eurographics conference on Rendering (EGWR'99)*, G. Ward and D. Lischinsky (Eds.). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 19-30. DOI=10.2312/EGWR/EGWR99/019-030 <http://dx.doi.org/10.2312/EGWR/EGWR99/019-030>
- Walter, B., Khungurn, P. & Bala, K., 2012, Bidirectional Lightcuts. *ACM Transactions on Graphics*, Vol. 31, No. 4, Article 59,
- Zhou, K., Hou, Q., Wang, R., and Guo, B. 2008. Real-time kd-tree construction on graphics hardware. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, ACM, New York, NY, USA, 1-11.
- Zhou, K., Hou, Q., Ren, Z., Gong, M., Sun, X. and Guo, B. 2009. RenderAnts: Interactive Reyes Rendering on GPUs, *ACM Transactions on Graphics (SIGGRAPH Asia 2009)*.

Glossário

A _____

AABB (*Axis-aligned bounding box* em inglês, ou Caixa alinhada com os eixos cartesianos) é um volume envolvente de um objeto 3D.

Abstract Factory é um padrão de projeto de software. Este padrão permite a criação de famílias de objetos relacionados ou dependentes, através de uma única interface e sem que a classe concreta seja especificada.

API, de *Application Programming Interface* (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades por programas aplicativos - isto é: programas que não querem se envolver em detalhes da implementação do software, mas apenas usar seus serviços.

B _____

Buffer é uma região de memória usada para armazenar informações.

C _____

CPU (*Central Processing Unit* em inglês, ou Unidade Central de Processamento), é a parte de um computador que interpreta e leva as instruções contidas no software.

CSV (*Comma-separated values* em inglês, ou Valores separados por vírgula) é um formato de arquivo que armazena dados tabelados, cujo grande uso data da época dos *mainframes*. Por serem bastante simples, arquivos .csv são comuns em todas as plataformas de computador.

CUDA é uma API de programação da NVIDIA, que utiliza uma linguagem derivada do ANSI C para suporte ao processamento utilizando os recursos da

GPU.

D

DLL (*Dynamic-link library* em inglês, ou Biblioteca de ligação dinâmica) é a implementação feita pela Microsoft para o conceito de bibliotecas compartilhadas nos sistemas operacionais *Microsoft Windows* e *OS/2*.

F

Far Plane Plano de corte do volume de visão de uma câmera virtual. Consiste na base maior do *Frustum*.

FPS (*Frames per Second* em inglês, ou Quadros por segundo) é a métrica mais usada para teste de desempenho de sistema gráfico. Ela consiste em calcular quantos quadros um sistema/algoritmo é capaz de gerar em 1 segundo.

FFT (*Fast Fourier Transform* em inglês, ou Transformada Rápida de Fourier) é uma ferramenta matemática utilizada para transformar sinais e funções do espaço do tempo para espaço da frequência.

Fragments ou Fragmentos são os possíveis pixels a serem mostrados na tela, sua exibição depende do tratamento feito pelo *Pixel Shader*.

Framework ou arcabouço é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido. Um *framework* pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto de software.

Frustum é o volume de visão de uma câmera virtual. Sua representação pode ser entendida como o tronco de uma pirâmide.

G

Game Engine, ou Motor de Jogos é um programa de computador e/ou conjunto de bibliotecas, para simplificar e abstrair o desenvolvimento de jogos ou outras aplicações com gráficos em tempo real, para videogames e/ou computadores rodando sistemas operacionais

GPGPU (*General-purpose computing on graphics processing units*), ou também referido como GP²U é uma unidade de processamento gráfico que pode executar código específico, assim como uma CPU.

GUI (no Brasil, interface gráfica do usuário; abreviadamente, a sigla GUI, do inglês *Graphical User Interface*) é um mecanismo de interação homem-computador. Com um mouse ou teclado o usuário é capaz de selecionar esses símbolos e manipulá-los de forma a obter algum resultado prático. Esses signos são designados de *widgets* e são agrupados em kits.

GPU (*Graphics Processing Unit* em inglês, ou Unidade de Processamento Gráficos) é o núcleo de processamento das placas gráficas.

H _____

HLSL (*High Level Shader Language*) é uma linguagem utilizada pelo *Microsoft DirectX* para programar *vertex* e *pixel shaders*.

I _____

I/O é uma sigla para Input/Output, em português E/S ou Entrada/Saída. Este termo é utilizado quase que exclusivamente no ramo da computação, indicando entrada de dados por meio de algum código ou programa, para algum outro programa ou hardware, bem como a sua saída ou retorno de dados, como resultado de alguma operação de algum programa.

IFFT (*Inverse Fast Fourier Transform* em inglês, ou Transformada Inversa Rápida de Fourier) é uma ferramenta matemática utilizada para transformar sinais e funções do espaço da frequência para espaço do tempo.

K _____

Kernel é um termo com vários significados, contudo, no contexto presente, ele é usado como a definição de uma função central de processamento, ou seja, o núcleo de processamento de dados.

L _____

LoD (*Level of Detail*) são técnicas que envolvem a redução da complexidade de cenas e objetos 3D a fim de reduzir o consumo de processamento.

Logs são registro de atividades gerado por programas de computador.

M _____

Memory leak, ou vazamento de memória, é um fenômeno que ocorre em sistemas computacionais quando uma porção de memória, alocada para uma determinada operação, não é liberada quando não é mais necessária. A ocorrência de vazamentos de memória é quase sempre relacionada a erros de programação e pode levar a falhas no sistema se a memória for completamente consumida.

Microsoft DirectX é uma coleção de APIs que tratam de tarefas relacionadas a programação de jogos para o sistema operacional Microsoft Windows, ou seja, é quem padroniza a comunicação entre software e hardware. O DirectX foi inicialmente distribuído pelos criadores de jogos junto com seus produtos, mas depois foi incluído no Windows.

MipMap são coleções de imagens pré-calculadas de uma imagem maior, que visam acelerar o processo de desenho e reduzir artefatos visuais.

Motor de jogo (Idem Game Engine).

MVC (*Model-view-controller*) é um padrão de arquitetura de software, nele é realizada a separação entre os dados (*Model*) e o layout (*View*). Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, e estes poderão

ser reorganizados sem alterar o *layout*.

N

Near Plane Plano de projeção do volume de visão de uma câmera virtual. Consiste na base menor do *Frustum*.

O

OpenGL (*Open Graphics Library*) é uma especificação definindo uma API multiplataforma e multi-linguagem para a escrita de aplicações capazes de produzir gráficos computacionais 3D (bem como gráficos computacionais 2D).

P

Pixel (*Picture Element*, ou seja, elemento de imagem) é o menor elemento num dispositivo de exibição ao qual é possível atribuir-se uma cor. De uma forma mais simples, um *pixel* é o menor ponto que forma uma imagem digital.

Pixel Shader é um *shader* que manipula pixels por meio de efeitos aplicados a cada um deles na tela.

Plotar é desenhar (uma imagem, especialmente um gráfico) baseando-se em informação fornecida como uma série de coordenadas.

R

Rasterizar é a operação de converter uma imagem vetorial em uma imagem *bitmap*.

Renderização é o processo pelo qual se podem obter imagens digitais. Este processo aplica-se essencialmente em programas de modelagem e animação (3ds Max, Maya etc.), como forma de visualizar a imagem final do projeto bidimensional ou tridimensional.

S

Shader na área da computação gráfica significa um conjunto de instruções de software, que serão utilizadas em uma GPGPU para a produção de efeitos de renderização.

SIMD é um método de operação de computadores com várias unidades operacionais em computação paralela. Neste modo, a mesma instrução é aplicada simultaneamente a diversos dados para produzir mais resultados. O modelo SIMD é adequado para o tratamento, cuja estrutura é muito regular, como as matrizes e vetores.

Stream é um fluxo de dados. Quando um arquivo é carregado para ser editado, esta carga ocorre num fluxo, ou seja, linha a linha até o carregamento total do arquivo, como água a correr num cano ou bytes sendo lidos por um programa.

T _____

Toolkit é um conjunto de *widgets*, elementos básicos de uma GUI. Normalmente são implementados como uma biblioteca de rotinas ou uma plataforma para aplicativos que auxiliam numa tarefa.

U _____

U.D. é um acrônimo de unidades de distância.

V _____

Vertex shader é um *shader* capaz de trabalhar na estrutura de vértices do modelo 3D.

W _____

Widget é um termo sem tradução que designa componentes de interface gráfica com o usuário (GUI). Qualquer item de uma interface gráfica é chamada de *widget*, por exemplo: janelas, botões, menus e itens de menus, ícones, barras de rolagem, etc.

X _____

XML (*eXtensible Markup Language*) é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais. Ela é capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações através da Internet.