

3Dグラフィックスシステムの開発環境

Development Environment of 3D Graphics Systems

● 田上祐也 ● 渡辺 真 ● 山口裕子

あらまし

近年、カーナビゲーションシステムやデジタルメータなど車載機器で3Dグラフィックスを用いるものが増えている。これらの機器は、よりグラフィカルなユーザインタフェースを搭載することによって、今まで表現できなかった視覚的な効果を実現できるようになった。富士通セミコンダクターでは、カメラから得られる実際の映像を3Dグラフィックスと組み合わせることで現実の世界と融合させ、ユーザの視覚支援を行う拡張現実注目し、グラフィックスシステムの開発を推進している。このような3Dグラフィックスを用いたシステムの構築やアプリケーションを開発するためには、多くの専門知識が必要になる。富士通セミコンダクターでは、3Dグラフィックスを用いたアプリケーション開発者の負担を軽減するために開発環境と最適な機能、性能を発揮できる製品を実現するためのソフトウェアを開発、提供している。

本稿では、3Dグラフィックスを用いたシステムを実現するためのソフトウェア構成やアーキテクチャ、およびその応用事例について説明する。

Abstract

Recently, more and more in-vehicle devices have been using 3D graphics in applications such as car navigation systems and digital meters. These devices adopt user interfaces that are more graphical, and they have come to offer visual effects that were previously impossible to produce. Fujitsu Semiconductor is focusing on augmented reality, in which actual images taken by cameras are combined with 3D graphics for integration with the real world to enhance the vision of users, and is moving ahead with the development of graphics systems. The building and development of these systems and applications that use 3D graphics require considerable expertise. Fujitsu Semiconductor is developing and offering a development environment that can reduce the burden on developers of applications using 3D graphics and software so as to realize products capable of delivering optimum functionality and performance. This paper describes the software configuration and architecture needed for a system that uses 3D graphics and gives examples of their application.

まえがき

近年、多くの分野で3D（3-Dimensional）グラフィックスというキーワードが注目されている。タブレットPCなどで使われているタッチパネルでは、液晶画面上で綺麗なGUI（Graphical User Interface）を滑らかに動作させる表現が可能になっている。視覚的な効果とオブジェクトを指で直感的に操作できるアプリケーションは、ユーザーにとって非常に使いやすく、これらを利用した様々なアプリケーションが広まってきた。このような表現を可能としているのが3Dグラフィックス技術である。

更に、3Dグラフィックスによる表現を用いてユーザーの視覚支援を行うシステムの研究や普及が進んでいる。例えば、医療用画像では、MRIやCT、超音波装置などの映像を3Dグラフィックスで表示し、自由な視点から見られるようになっている。

組み込みシステムでは、3Dグラフィックスを描画するための標準API（Application Programming Interface）として、OpenGL ESがKhronosグループによって策定されている。Khronosグループは、

このほかにも様々な業界標準APIを策定している⁽¹⁾。富士通セミコンダクターでは、OpenGL ES2.0に準拠したグラフィックスライブラリを提供している。

本稿では、まず3Dグラフィックス表示を行うためのシステム構成について説明する。次に、ソフトウェアアーキテクチャや実装技術の基本原則、応用事例および課題について説明する。最後に今後の展望について述べる。

グラフィックスシステム

3Dグラフィックスを用いたアプリケーションやシステムの開発には、リアルタイムシステム、画像処理など幾つかの専門知識が必要になる。例えば、3Dグラフィックスで拡張現実（AR：Augmented Reality）を表現する場合、複数のカメラ映像入力により取得したリアルタイムの画像と3Dモデルを合成する。このとき、入力画像全体のバランスを見て、光源によって視認性が低下した領域を補正したり、注目すべき箇所を強調したりするような画像補正処理を行う。その後、画像処理を行った映像を3Dモデルと組み合わせ、見せたい方向からの視点に設定を行い描画する。最後に完成した映像を画面上に表示することになる {図-1 (a)}。こ

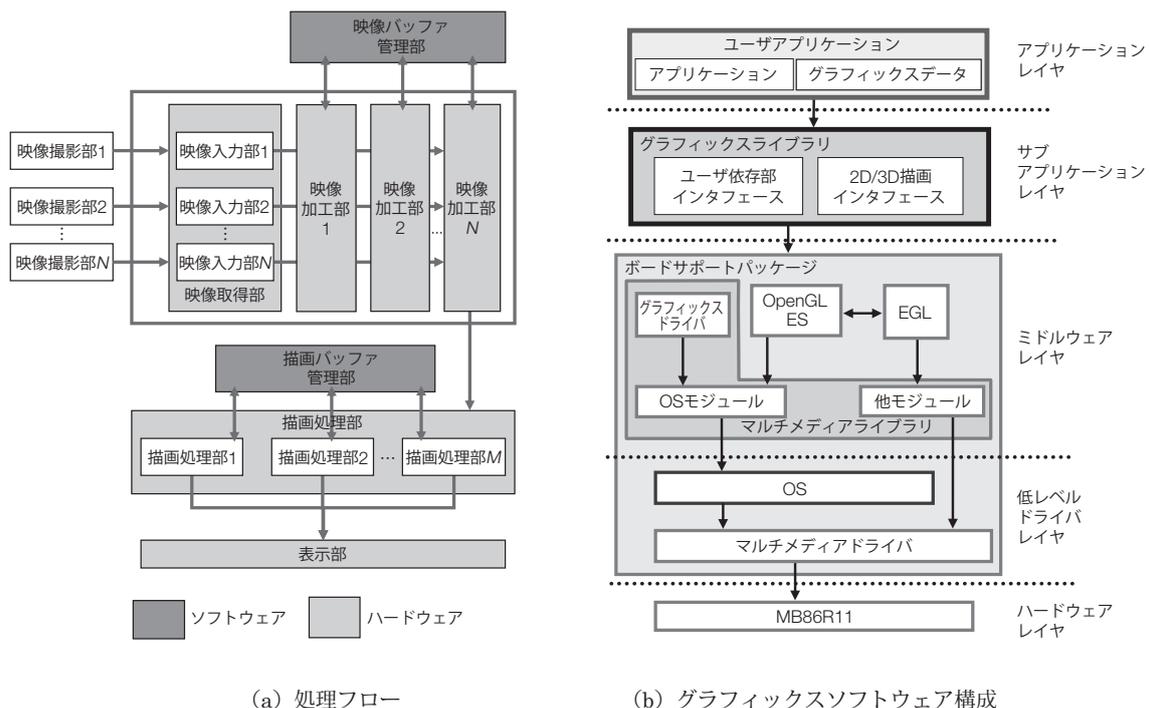


図-1 グラフィックスシステム

れらの一連の処理は、カメラから映像を取り込む時間以内で処理を完了し、遅延を最小限に抑えることで、リアルタイムシステムとして構築しなければならない。

これら全てのことをアプリケーションから制御、構築するには多大な労力が必要となるため、困難である。そのため、ユーザがこれらの知識を持たなくとも容易にアプリケーションを開発することが可能なソフトウェアが期待されている。また、3Dグラフィックスを用いたアプリケーションは高度な作込みが必要である。そのため、組込みシステムと同等の動作をPC上で確認し、事前開発を可能にするシミュレータが期待されている。これらの状況から、富士通セミコンダクターでは、アプリケーション開発におけるユーザの負担を軽減し、容易に3Dグラフィックスを用いたアプリケーションを開発できる開発環境を提供している(図-1(b))。

ユーザアプリケーションでグラフィックスシステムを構築するには多種多様な技術の連携が必要となる。車載向けグラフィックスSoCであるMB86R11⁽²⁾は、様々なグラフィックスアプリケーションを実現するため、3Dグラフィックスエンジン、2Dグラフィックスエンジン、画像処理エンジンといった複数のグラフィックスエンジンを搭載している。

システム上でこれらのグラフィックスエンジンはCPUと独立して制御されているため、並列動作が可能である。各グラフィックスエンジンは専用のドライバによって状態管理されており、複数タスクから使用される場合でも、ユーザはほかのタスクを意識することなく動作させることができる。

2D/3Dグラフィックスの連携

本章では、富士通セミコンダクターが実装しているグラフィックス処理技術について紹介する。

アプリケーションの複雑化、高解像度化にに応じて十分なシステム性能を発揮するには、全てのグラフィックスエンジンが効率良く動作できるよう制御しなければならない。そのため、グラフィックスSoCとともに提供されるグラフィックスライブラリ、グラフィックスドライバなどのソフトウェアでは様々な工夫を行っている。

MB86R11用のグラフィックスライブラリで

は、グラフィックスインタフェースの一つとしてOpenGL ES 2.0を採用している。富士通セミコンダクターのグラフィックスSoCでは、シェーディングアルゴリズムの進化に追従するために固定パイプラインを廃したOpenGL ES 2.0に対応し、3DグラフィックスエンジンにProgrammable Shaderを搭載している。Programmable Shaderはピクセル単位での処理が可能となっているため、特別なグラフィックスエンジンがなくても、自由な表現を実現できる。

Programmable Shaderは、GPGPU (General-Purpose computing on Graphics Processing Units)として画像解析や数値演算などにも利用することができ、アプリケーション性能を向上させる技術の一つとして期待されている。

3Dグラフィックスエンジンでは、高度な表現はもちろん、2Dグラフィックスと同等の表現も実現できる。しかし、あらかじめPCで作成した2Dデータの方が高解像度、かつ高速に処理できることもある。そのため、2Dグラフィックスエンジンを別に搭載することによって2Dグラフィックスを高速描画し、システム性能の向上を図っている。

通常2D/3Dグラフィックスエンジンは、単独の描画処理命令を受け付け、実行することしかできない。そのため、複数のグラフィックスエンジンを制御するシステム構成で、CPUがグラフィックスエンジンを並列処理させるための制御が複雑になっていた。これは、複数のアプリケーションが動作する場合、処理速度が遅くなるという制約になる。

これに対し、MB86R11では、グラフィックスエンジンを制御する負荷を軽減させるために、描画処理命令のキューイングを行うコマンドシーケンサを搭載している。コマンドシーケンサは、グラフィックスエンジンを制御する一連のコマンドを複数受け付けることができるため、グラフィックスエンジンを間断なく動作させることができる。

この仕組みによって、これまでCPUで行われていた割込み処理の一部をコマンドシーケンサに任せられるため、CPUの負荷を軽減できる。ソフトウェアでは、このコマンドシーケンサを効率良く動作させるためにグラフィックスエンジンの描画完了を検知しながら動作できるコマンドを組み立

てて制御している。

MB86R11では、グラフィックスエンジンと同じ数のコマンドシーケンサを搭載している。ソフトウェアでは、それぞれのコマンドシーケンサに対して、グラフィックスエンジン同士の並列動作を妨げないように処理命令を構築している。また、一つの描画バッファに2D/3Dグラフィックスが混在するような場合には、並列動作させずにコマンドシーケンサ間で互いの描画完了を待ち合わせるようにコマンドで制御できる（図-2）。

このように2D/3Dグラフィックスエンジンで描画した結果は、表示コントローラを使用して最終的にディスプレイに出力される。一般に、複数の画像を重ね合わせて表現したい場合は、グラフィックスエンジンを使用してメモリ上で重ね合わせる方法と、表示コントローラ上で重ね合わせる方法がある。

富士通セミコンダクターのグラフィックスSoCは、従来から複数の画像を複数のレイヤに分けて描画し、表示コントローラによってレイヤを重ねさせる機能を搭載している。表示コントローラを使った重畳は、特に更新頻度の異なる画像を重ね合わせる場合に有効である（図-3）。

例えば、アプリケーションのある時点では、更新が必要なレイヤと更新が不要なレイヤが混在し

ている場合がある。このとき、不要な更新を行わないようにソフトウェアで制御することができる。

一方、表示コントローラに負荷がかかりすぎる場合には、メモリ上で重ね合わせる方法が有効である。どのグラフィックスモジュールに負荷がかかるかはアプリケーションによって異なる。そのため、特定のグラフィックスモジュールに負荷が偏らないように、システムの負荷に応じてアプリケーションを設計できるソフトウェアとなっている。

以上のように、富士通セミコンダクターでは、限られたリソースの中で、各種デバイスとの連携や描画、表示の更新が最適なバランスになるように考慮したグラフィックスソフトウェアを提供している。

カメラ映像とグラフィックスの連携

カメラ映像入力を含んだシステムの場合、内部で行われる画像処理の負荷に関わらず、定められたフレームレートで遅延なく処理結果を出力する必要がある。

グラフィックス描画処理は毎フレームの負荷が変動するため、システム動作を単純にグラフィックス描画処理のタイミングに合わせてしまうと、カメラ入力映像のサンプリング間隔にゆらぎが生

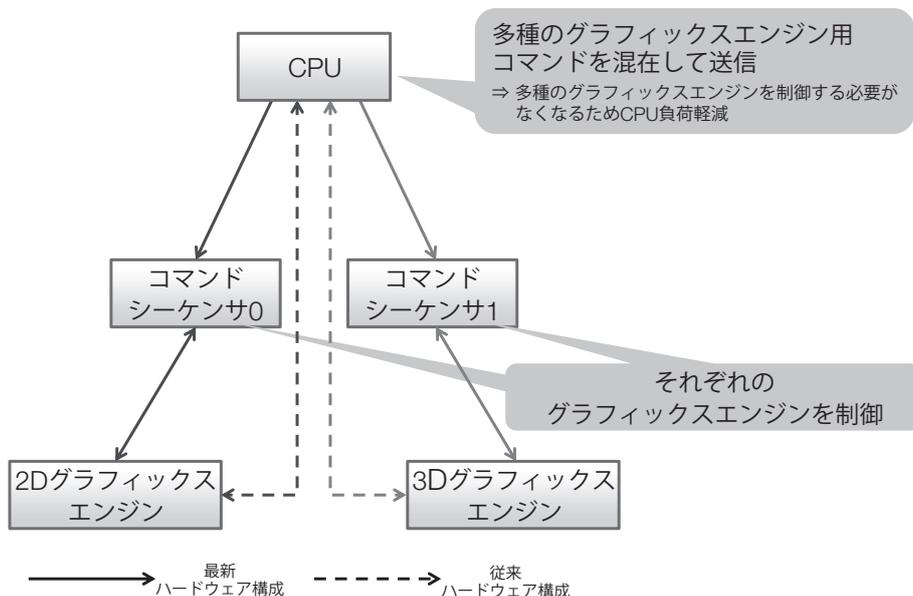


図-2 コマンドシーケンサの動作

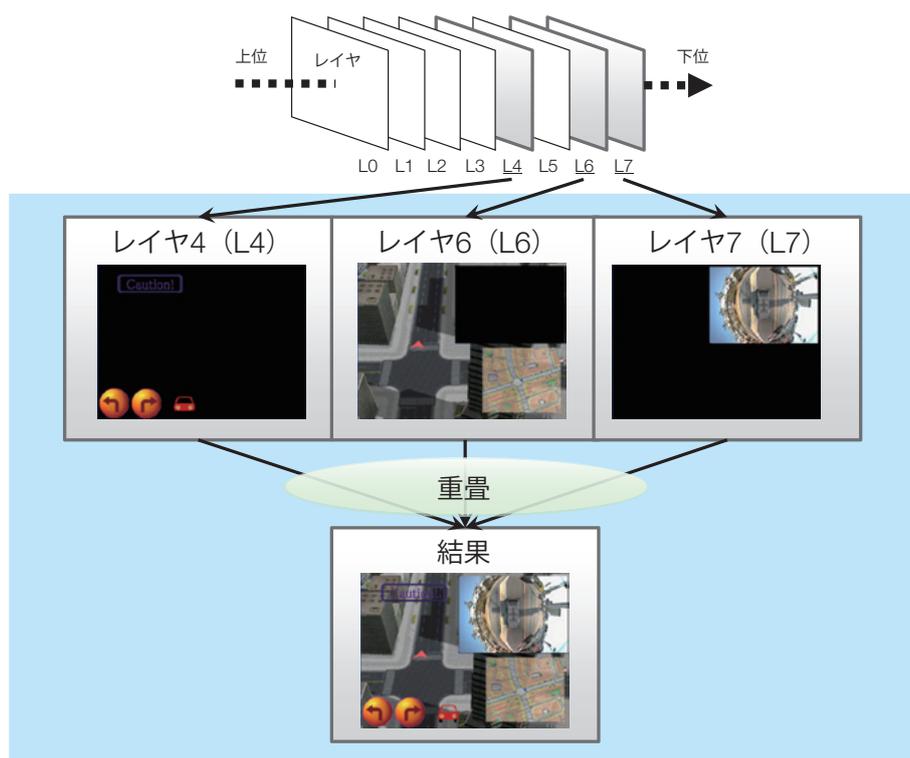


図-3 表示コントローラによるレイヤ重畳

じる。これを回避し、一定の間隔でサンプリングされたカメラ映像を入力画像として使用するために、キャプチャコントローラに参照用バッファ+ダブルバッファのトリプルバッファ構造を持たせている。

また、各グラフィックスモジュール間のデータの受渡しはダブルバッファまたはトリプルバッファの構成となっている。あるグラフィックスモジュールの出力結果をほかのグラフィックスモジュールが参照している間も並列に描画や書込みができる(図-4)。

この仕組みにより、アプリケーションから任意のタイミングでカメラ入力画像を取得可能となるため、例えばアプリケーションの動作を特定のクロックに合わせて動作させれば、定期的なカメラ入力映像を使用した出力画像を生成することができる。

このようなカメラ入力画像をアプリケーションで使用するシステムの場合、キャプチャ部からグラフィックス描画、表示モジュールに至るまで、複数のソフトウェアで連携することになる。ソフトウェアでは、ハードウェアの仕様や特徴を考慮

し、アプリケーションレイヤからドライバレイヤまで一貫した設計を行うことで、性能低下の原因となるバッファコピーを発生させないようにしている。そして、常に各グラフィックスモジュールが並列に動作するよう最適な構造となっている。

応用事例

富士通セミコンダクターでは、富士通研究所で研究発表された全周囲立体モニタシステム⁽³⁾、⁽⁴⁾をMB86R11上で製品化した(図-5)。この技術は先に記述した画像処理技術を駆使し、統合的なシステムとして構築したものである⁽⁵⁾、⁽⁶⁾。

全周囲立体モニタとは、車両周囲の映像を立体的に合成し、自由な視点から見た結果をディスプレイに表示する車載向け視界補助映像処理技術である。更に、表示映像を切り換える際には滑らかに視点を移動することもできる(図-5)。

全周囲立体モニタでは、大量のポリゴンをリアルタイムで描画、表示することが求められるため、効率良く処理できるようにVBO(Vertex Buffer Object)を用いてメモリの転送量を減らした。一方、毎フレーム変更が必要なカメラ映像の処理は

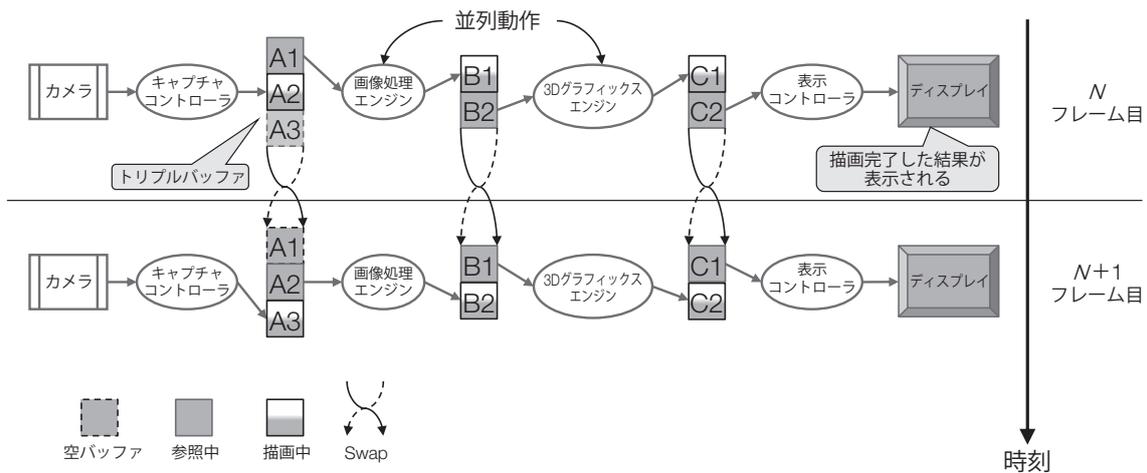


図-4 グラフィックスモジュール並列動作のイメージ

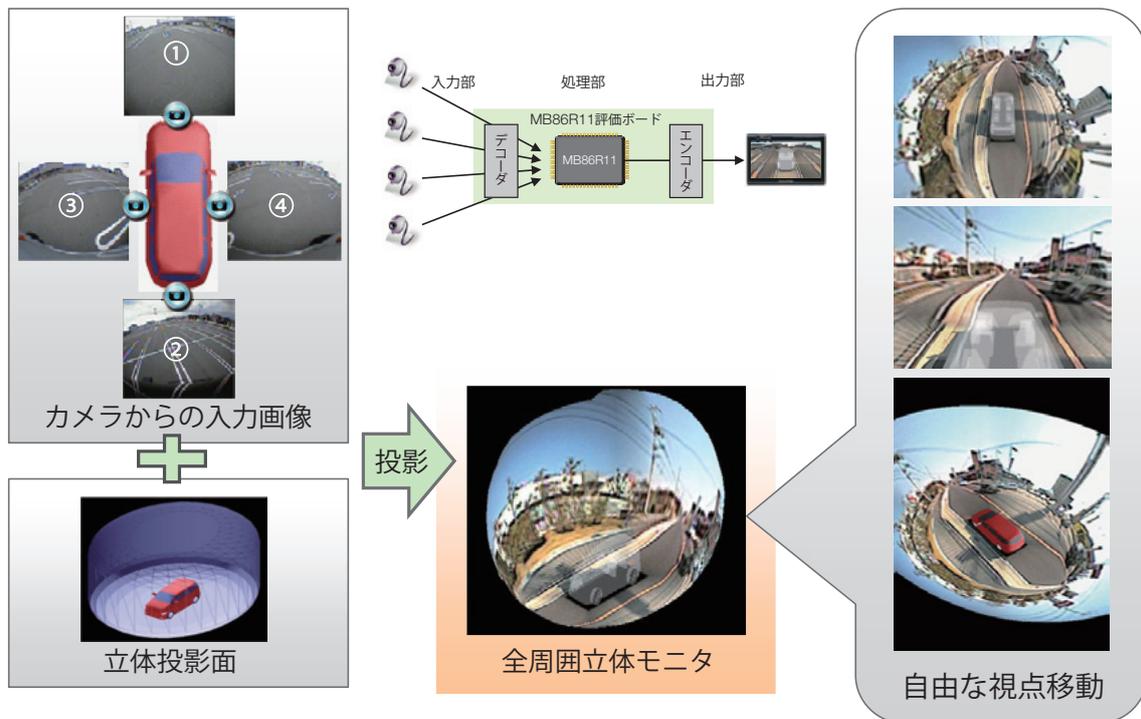


図-5 全周囲立体モニタシステム

Programmable Shaderで行う。これらにより、速度と描画品質を確保している。

また、全周囲立体モニタでは複数のカメラ映像をつなぎ合わせて画像を構成しているが、日射方向によりカメラ間の輝度に大きな差が生まれ、不自然な画像となる場合がある。そこで、各カメラ間の輝度を平均化し、つなぎ目をProgrammable Shaderで滑らかに見せることにより、より視認性の良い映像を作り出している。

今後の課題

3Dグラフィックスの分野にはまだまだ多くの可能性がある。例えば、ユーザが見て直感的に分かりやすい表現や、エンタテインメント性のある新しい表現などである。全周囲立体モニタでは、自車に搭載されたカメラからの映像だけでなく、他車や監視カメラの映像を取り込み合成することによって、自車から見えない死角も見ることができ

る技術の研究が進んでいる。また、カメラで撮影された映像は、3次元の現実世界を2次元の平面に投影したものである。この映像から実際の距離感や立体感をリアルタイムに復元することで、より現実に近い立体映像を作り出す技術の研究も進んでいる。

このように、3Dグラフィックスを用いて拡張現実や仮想現実（VR：Virtual Reality）を表現する場合、ユーザが快適と感じられる速度性能や画質が必須条件となる。これを達成するための課題として以下が考えられる。

- ・ネットワークを経由した大量データ処理
- ・画像処理、画像認識技術の連携と高速化
- ・高解像度カメラ、高精細モニタなどによる使用メモリ帯域の改善

これらの課題を解決するには、ハードウェアレベルでの性能向上を待つだけでなく、同時に、ハードウェアやドライバを熟知した上で、より高性能な基本グラフィックスライブラリに拡張していく必要がある。現在、これらをユーザが使用しやすいAPIとして提供できるよう検討している。

む す び

本稿では、MB86R11評価ボード上で動作するグラフィックスソフトウェアのアーキテクチャとアプリケーションの事例について述べた。

今後、グラフィックスSoCは、マルチコアとマルチGPUといった構成が主流となり、アプリケー

ションはより高い並列度で実行されるようになる。これによってアプリケーション間の連携もより活発になり、3Dグラフィックスを含むHMI（Human Machine Interface）の活用が予想される。このようなシステム環境においても、グラフィックスSoCの性能を最大限に引き出すことができるソフトウェアを開発し、アプリケーションの開発を容易にする開発環境を提供していく。

参考文献

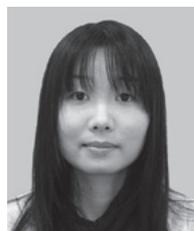
- (1) The Khronos Group.
<http://www.khronos.org/>
- (2) 富士通セミコンダクター：MB86R11.
<http://jp.fujitsu.com/microelectronics/products/assp/gdc/>
- (3) 富士通研究所：全周囲立体モニタシステム。
<http://pr.fujitsu.com/jp/news/2008/11/17.html>
http://jp.fujitsu.com/group/labs/techinfo/techguide/list/car_p07.html
- (4) 清水誠也ほか：全周囲立体モニタシステム。
FUJITSU, Vol.60, No.5, p.496-501 (2009).
- (5) 富士通セミコンダクター：全周囲立体モニタシステム。
<http://jp.fujitsu.com/microelectronics/products/assp/gdc/3dmonitor.html>
- (6) Tech-On! SPECIAL：車載グラフィックス・ソリューション。
<http://special.nikkeibp.co.jp/ts/article/aaa0/112983/>

著者紹介



田上祐也 (たがみ ゆうや)

富士通セミコンダクター（株）
ソフトウェア技術統括部 所属
現在、グラフィックス関連ソフトウェア（OMNIVIEW Library）の開発に従事。



山口裕子 (やまぐち ゆうこ)

富士通セミコンダクター（株）
ソフトウェア技術統括部 所属
現在、グラフィックス関連ソフトウェア（OMNIVIEW Library）の開発に従事。



渡辺 真 (わたなべ まこと)

富士通セミコンダクター（株）
ソフトウェア技術統括部 所属
現在、グラフィックス関連ソフトウェア（OMNIVIEW Library）の開発に従事。