

[招待講演] 車載メディア処理と LSI アーキテクチャ —課題と将来動向—

黒田 一朗

NEC エレクトロニクス 〒211-8668 神奈川県川崎市中原区下沼部 1753

E-mail: ichiro.kuroda@necel.com

あらまし 車載応用のメディア処理として、安全運転支援や高度なユーザーインターフェースのための画像認識や 3D グラフィックス技術について、今後期待されるアプリケーションと、それらを実現するシステム LSI のアーキテクチャ、実現コストについて紹介する。さらにこれらの技術を統合した次の世代の車載メディア処理 LSI のターゲットと課題について検討を行う。

キーワード カーエレクトロニクス、画像認識、3D グラフィックス、並列処理プロセッサ、GPU

Media Processing LSI Architectures for Automotives — Challenges and Future Trends —

Ichiro Kuroda

NEC Electronics Corporation 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa, 211-8668 Japan

E-mail: ichiro.kuroda@necel.com

Abstract This report describes future media processing applications and system LSI architectures with their costs for automotives, especially for image recognition and 3D graphics for assistance of safety driving as well as advanced graphical user interface. After that, next generation media processing LSIs for automotives which will integrate image recognition functions and 3D graphics functions are discussed with their target performance and challenges.

Keyword Car electronics, image recognition, 3D graphics, parallel processor, GPU

1. はじめに

カメラ付携帯電話やデジタルテレビ、デジタルカメラなどの普及により、デジタル画像の入力デバイスとしてのカメラ、及び出力デバイスとしてのディスプレイの量産、低価格化が進み、デジタル画像の活用があらゆるところで進みつつある。特に自動車分野においては、安全運転支援を主な目的として車外、車内に複数のカメラが搭載されるようになり、またドライバーへの各種情報提供がカーナビなどによる画像インターフェースにより実現されるようになってきている。

これまで車載用途にデジタル画像の活用を可能にする技術として、膨大なデータ量の動画の蓄積、伝送（通信）を容易にする画像圧縮技術や、地図をはじめとする大量の知識情報をわかりやすく、あるいはリアリティを付加した形で表現する画像生成（グラフィックス）技術などが利用されてきている。これに加えて、カメラで撮影した画像から空間情報を取得する画像認識技術の安全運転支援への活用が始まっている。

今後、より安全でより効率的なドライビングを実現するために車の周囲の環境を把握し、これをドライバーに伝え、あるいはドライバーの見落としをカバーするための手段として、これら画像を中心としたメディア処理技術の活用が期待されている。

一方において、画像の圧縮、生成、認識などのメディア処理技術はムーアの法則に支えられた LSI の性能向上と共に発達し、性能を向上させてきた。より効率的（画質を保ちながら高圧縮率）な画像圧縮、よりリアルかつ高速な画像生成、より頑健かつ精度の高い画像認識には、より多くの演算量を必要とし、より高速大規模な LSI を必要とする。これら画像メディア処理の膨大な演算性能要求に応えるために、画像処理特有のデータ並列性を活用した様々な LSI アーキテクチャが検討されてきている。

本稿では車載用途に向けた画像メディア処理について、車載特有の制約条件、要求条件のもとでのメディア処理と、LSI アーキテクチャ実現における課題、

将来動向について報告する。特に安全運転支援のための画像認識及び、画像認識と共に車とドライバーとのインターフェースとして性能向上が期待される画像生成(グラフィックス)、さらにその2つを融合したナビアプリケーションも含め、今後期待される技術とLSI実現におけるチャレンジに焦点をあてた議論を行う。

2. 車載メディア処理の機能とアーキテクチャ

車載メディア処理は従来のカーナビによる地図情報、位置情報に加えて、車内外に設置されたカメラにより収集された画像情報から有用な交通関連情報等を収集してドライバーをアシストするサービスを可能にすることを期待されている。

このようなドライバーの安全運転を支援するサービスとしては、

- (1) ナビゲーションなどの情報サービスやドライバーの視覚機能の支援、拡張
 - (2) 衝突やレーン逸脱、あるいはドライバーの居眠りなどの危険を警告、監視をするサービス
 - (3) 衝突回避ブレーキなどのドライバーの運転機能の一部を代行/支援するサービス
- などが考えられている。

2.1. 安全運転支援機能と画像認識

安全運転支援を実現する画像認識としては、路上の他の車や障害物や歩行者の認識、信号、標識、その他道路標示(横断歩道などの路面マーク、工事中などの看板)の認識など、車外の環境を認識する処理がある。また車内では、ドライバーの居眠り(まぶたの状況)、顔向き/視線、姿勢の検知、認識などがある。これらのアプリケーション実現に必要な計算量と予想実現時期を図1に示す。

特に精度が要求される安全目的の画像認識処理は人の生命に関わるものであるから、非常に高い信頼性、認識精度が特に要求され、中には数百 GOPS 以上の演算性能が必要になるものもある。

このように高い認識精度を実現しながら、膨大なデータを限られた時間内でリアルタイム処理するには、高い演算性能が必要になるが、同時に車に搭載するためには、これを低コストで実現する必要がある。さらに、熱や振動という悪条件のもとでの耐久性が求められ、低電力かつコンパクトに実現する必要がある。

一方、車載の画像認識では、多様な外部条件や、環境の変化への対応が必要になる。具体的には、道路のコンディションや混み具合、様々な照明条件(昼、夜、トンネル)や天候条件(晴れ、曇り、雨)、及びそれらの急変に対応できる必要がある。

様々な認識対象や環境条件に対応するためには様々な認識アルゴリズムを利用する必要がある。例え

ば車の認識では、上記の照明条件や天候条件に加え、認識する車との相対位置(前方、後方、側方)によって、それぞれに適した認識アルゴリズムを採用する必要がある。さらに同じ認識対象(例えば前方車の認識)に対しても認識精度を高めるために、様々な画像処理(前処理)、画像認識手法が開発されている。

図2に示す路上の車に対する一般的な認識手法では、画面内の認識候補の探索/検出(仮説生成)及び(車であるかどうかの)確認(仮説検証)の2つのステップから構成されている。仮説生成のステップに対しては、対称性、色、影、角、縦/横エッジ、テクスチャ、ライトなどの車の特徴を示す知識を利用して車を検出する方法や、ステレオ視による視差マップや逆投影マップを利用する方法、あるいはオプティカルフローなど計算による動きを利用する方法などが提案されている。

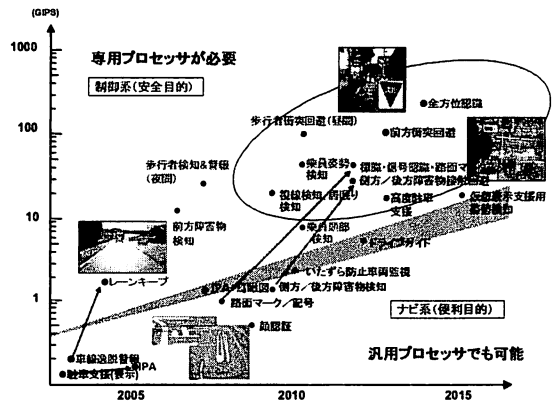


図1 車載画像認識のニーズと計算量予測

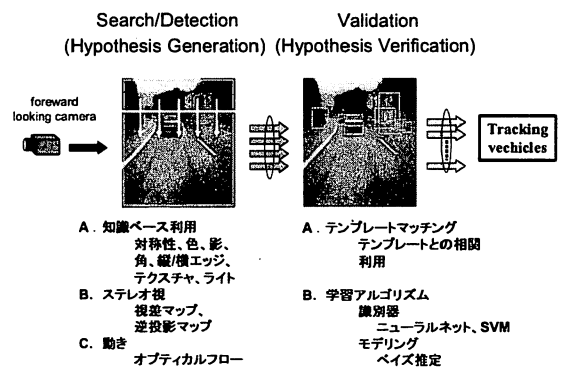


図2 車載画像認識アルゴリズムの多様性

一方、仮説検証ステップに対しては、テンプレートマッチングを利用する方法や、パターン識別問題として学習アルゴリズムを利用する方法、例えばPCAなど

を利用して特徴抽出を行い、ニューラルネットやSVMなどを利用して識別を行う方法などが提案されている。さらにこのような認識、検出処理とトラッキング（追跡）処理を組み合わせる場合が多い[1][2][3]。

2.2. 安全運転支援のためのアーキテクチャ

車載の画像認識を実現する LSI アーキテクチャは、多様なアルゴリズムを実現できるプログラマビリティを確保しながら、低電力で高い性能を実現する必要がある。その為には、いかに画像処理アルゴリズム特有のデータ並列性を利用するかが重要になる。

画像処理あるいは画像認識（コンピュータビジョン）のアルゴリズムは一般に、低レベル処理、中間レベル処理、高レベル処理に分類される[4]。このうち、処理量が多いのは主に低レベル処理や中間レベル処理の一部である。

低レベル処理としては例えば画素レベルでの雑音除去やひずみ補正などの前処理、エッジや色、動きなどの検出などの局所特徴抽出を行う処理があり、これらは画素レベルでのデータ並列性が高く、各画素に対して同じ処理を行い、局所情報しか用いないため、高並列アーキテクチャによる実現が可能である。

また、中間レベルの画像処理においても、ニューラルネットワークなどの局所特徴抽出の結果に対する判別処理やヒストグラムなどの統計量計算、ステレオ計算、テンプレートマッチング計算などがあり、これも並列処理アーキテクチャによる高速実現が可能である。

一方、中間レベルの処理で得られた判別情報を用いる高レベルの処理は、低・中間レベルの処理と比較すると並列性、演算量とも少なく CPU で実現されることが多い。

2.3. ナビゲーション機能と3D 処理

ナビゲーション機能の強化としては、グラフィックス技術を用いてドライバーに交通情報を提供するユーザーインターフェースの高度化がある。従来の地図を利用したカーナビでは、2次元の地図表示と実際の車外の状況との関係付けが容易でない場合もあり、よりわかりやすい表現方法として、鳥瞰図（2.5次元）や、ドライバーの視点から見た3D（都市など）の景観表示が利用されるようになってきている。特に撮影された画像を用いたリアル（実写）3D景観表示では、実際に車外に見える景観との対応付けが容易になるため、ドライバーがこれから見えてくる道路景観上でのルート指示を容易に理解できるようになる。

例えば、衛星画像と標高情報から3D地形図を再構成する方法(Terrain Rendering)があり、テクスチャマッピングを利用した3Dグラフィックス技術が利用されている[5]。

また、特に都市、市街地のリアル3D景観表示は、

事前に撮影された画像（ビデオ）をそのまま再生して利用することもできるが、撮影した視点が限られてしまうため、様々な運転状況（道路上での位置）に対応できない。そこで(都市)景観の3Dモデルと、撮影画像から得られるテクスチャ画像を利用して、3D景観を再構成する方法が考えられている[6][7]。これにより、より少ない情報量（3Dモデル）で様々な状況（ドライバー視点）に対応して撮影画像から得られたテクスチャ画像をマッピングして3D景観を合成することによりリアルな表示が可能になる。

この際、3Dモデルやテクスチャは撮影された（動）画像を利用して抽出することができる。この抽出処理（動画から3D再構成）には高度な画像認識処理が必要であり多大な演算量が必要であるが、ナビゲーション用には、このような抽出処理をすべての車に搭載する必要はない（データ収集車を走らせてオフラインで処理してもよい）。但し、このような処理が車に搭載され、実時間処理が可能になると、走行中に車載カメラ画像から周囲の（未知の）環境の3Dモデルの再構成が可能になり、自律走行が可能なロボットカーに一步近づくことができる。

2.4. GPUによる3Dモデル生成、認識処理の実現

リアルな3D表示処理や3Dモデルを扱う高度な画像認識処理は近年高性能GPU(グラフィックプロセッサ)を利用してリアルタイム実現されるようになってきている。

90年代以降グラフィックスハードウェア技術はグラフィックスAPIと共に進化してきた(図3)が、2000年前後からそれまでの固定機能ハードウェアベースのグラフィックAPI(OpenGL1.xなど)からプログラマブルGPUベースのAPI(OpenGL2.x、DirectX8以降)へと大きく変化し、これにより多様でリアルな画像の生成が可能になった。

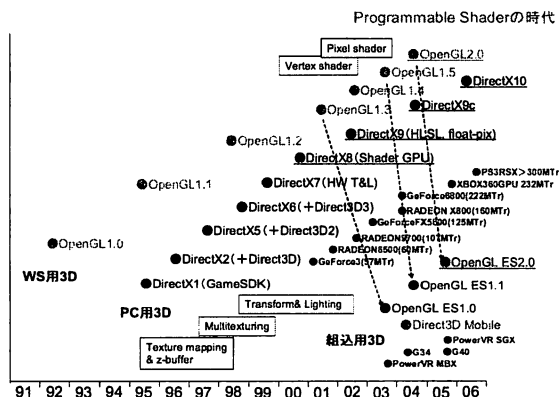


図3 グラフィックスAPIとハードウェアの進展

GPUによるグラフィックス処理は三角形(あるいはポリゴン)パッチにテクスチャを貼り付けて3Dグラフィックス画像を実現するものであるが、処理としては(1)Vertex処理(各頂点の位置、色、方向ベクトルの計算)、(2)ラスタライズ(各三角形パッチから各画素の生成)、(3)Fragment処理(各画素(Fragment)のテクスチャ、色、奥行き、ブレンディング処理の計算)、等を順に実行する。プログラマブルGPUではこれらの各段階の処理において、頂点レベルの並列性や、画素レベルの並列性を利用して、Vertex処理とFragment処理を並列プロセッサで実現している。

プログラマブルGPUの例として、NVIDIAのGeForce6800は図4に示す。MIMD並列の6個のVertexプロセッサとSIMD並列の16個のFragmentプロセッサを内蔵し、チップ規模は222MTr(130nm)で、400MHz動作で600MTriangle/sec、6.4GPixel/secの描画性能を実現している[8]。プロセッサ部の演算性能は120GFLOPSと推定される。他のGPUの規模と性能例を表1に示す。

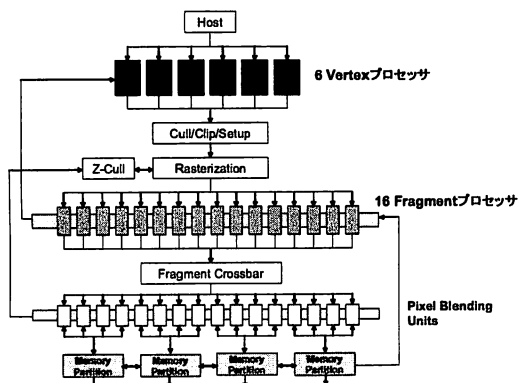


図4 GPU (NVIDIA GeForce6800)

	nm	MTr	MHz	V	F	Tri/s	Pix/s
RADEON X1900XT	90	384	624	8	48	1.25G	10G
GeForce7800	110	302	430	8	24	860M	10.3G
RADEON X800	110	160	400	6	16	600M	4.8G
GeForceGo6800	130	140	300	5	12	375M	3.6G

表1 GPUの規模と性能

(V: vertexプロセッサの数、F: fragmentプロセッサの数)

プログラマブルGPUを使った都市景観や地形(Terrain)の3Dモデルのレンダリング例として(1)Pentium4-3GHzとX800により1280x1024の都市景観の画像を毎秒15~20枚描画した例[9]や、(2)GeForce6800により1024x768の地形画像を毎秒30枚描画した例[5]などが報告されている。

一方、プログラマブルGPUはGPGPU(General-Purpose GPU)と呼ばれるGPUによる汎用数値計算技術の研究が進んでおり、特にGPUのデータ並列演算向きアーキテクチャが画像処理、コンピュータビジョン処理に向いているため、この分野への適用が盛んに研究されている[10]。ビジョン処理の実現例としては、Plane sweep法をベースにしたDepth map(奥行き情報)計算(726x520, ~30fps)をNVIDIA GeForce6800GTで実現した報告[11]や、KLT(Kanade-Lucas-Tomasi)特徴追跡処理(1024x768, 30fps、特徴点数1000)をATI X1900XTで実現した報告があり、後者のケースではCPU(詳細不明)の20倍高速としている[12]。この特徴追跡処理を利用して、市街地を走行して撮影したビデオから市街地の3Dモデルを再構成する研究も進められている(本処理は全体の一部であり、全体をリアルタイムで実現するためにはさらに数倍高い性能が必要である)[7]。

但し、表1で示したようなGPUの消費電力は20~100Wのレンジにあり、現時点では車載用途向きではない。一方、近年車載用LSIに搭載されるような組込み用グラフィックスコアも同じくプログラマブルGPUに移行しつつあるが[13]、性能、規模とも1桁から2桁小規模なものに留まっている。

3. 車載メディア処理LSIの実現

3.1. 車載画像認識プロセッサ

車載用のLSIは機械的あるいは熱的に厳しい条件下で長期にわたり利用される。このため通常に比べ、動作温度範囲が広く(-40度から85度)、不良率(PPM)も低く抑えられている。同時に既に述べたように厳しいコスト条件、消費電力なども課せられる。

これらの条件を満たしながら高い画像処理性能を実現するために、特に処理量の多い低レベル、中間レベル処理の並列性を利用した専用の画像処理プロセッサが提案されている。

IMAP-CEは図5に示す128個の4-way VLIWエンジンと2kBのローカルメモリからなるプロセッサエレメントを1つのコントロールプロセッサによりSIMD型制御を行うアーキテクチャを採用し、100MHz動作で51.2GOPSの演算性能を実現する[14]。チップ規模は32.7MTrで、消費電力は1.8V動作時で2~4Wである。画像認識性能としては前方車認識を20msで実現する。

Viscontiは5つの8並列(8-bit x 8)SIMD演算器からなる3-way VLIWコプロセッサを搭載したMePモジュール3つから構成され、150MHz動作で18GOPSの演算性能を実現する[15]。チップ規模は21MTrで消費電力は1.5V動作で1Wである。画像認識性能としては、前方車認識を30ms/モジュールで実現している。

以上の画像認識プロセッサは、プロセッサレベルあるいは演算器レベルでの SIMD 並列処理を採用して、低レベル、中間レベルの画像処理の高速化を行い、高性能低電力かつコンパクトな LSI として実現している。

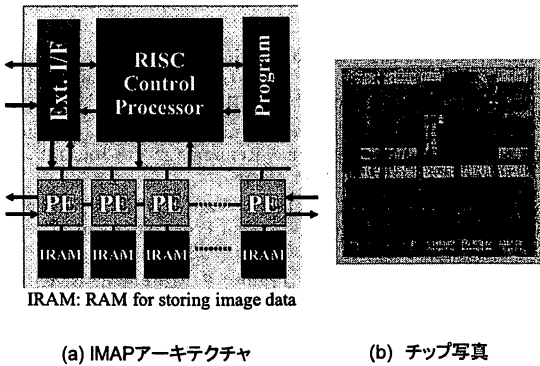


図5 画像認識プロセッサ IMAP-CE

3.2. 車載メディア処理 LSI の課題と将来動向

安全運転支援のための画像認識として、追突防止のための前方車認識、レーン変更時の側方監視、(特に市街地における)自動車周辺の歩行者や自転車の検知など、自動車の前後左右を含めた全方位の監視が望ましいとされている。全方位の監視のために各方向に複数台のカメラを設置したり、双曲線ミラーや魚眼レンズ等を利用した全方位(Omni)カメラの利用などが提案されている[16]。これに対応して、全方位に対する画像認識処理を同時に行う必要がある。例えば、前方、側方、後方の障害物(車、自転車、人)検知、信号・標識検知、さらにドライバーの視線や居眠りの検知まで同時に行うと、これまで主に単一処理向けに利用されてきた画像認識プロセッサの少なくとも 10 倍程度(200~500GOPS)の性能が必要になってくる(図6)。

■ マルチ画像認識

□ 走行時同時認識処理

- 前方、側方/後方障害物(車)検知
 - 歩行者検知
 - 信号、標識認識
 - 視線、居眠り検知
 - レーンキープ
- 単一処理の10倍の性能が必要

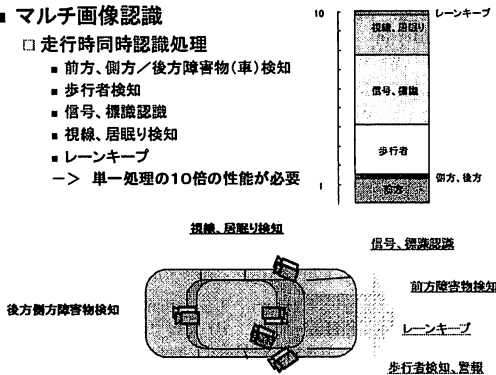


図6 マルチ画像認識と計算量

さらに、画像認識処理のみならず、3Dグラフィックス処理へのニーズが増大することが予想され、例えば、これらの処理の同時実行(ナビ表示をしながら、歩行者接近警告)や、連携した処理(ナビ画像・地図とカメラ画像とのマッチングにより位置修正)が期待される。また、前に示したように3Dモデルを扱う高度な画像認識処理ではGPUが利用されていることから消費電力、コストの問題が解決されれば、今後、画像認識と3Dグラフィックスのそれぞれのプロセッサアーキテクチャの融合、統合が期待される。

2つのアーキテクチャの融合、統合を検討するにあたり、GPUおよび画像認識プロセッサの画像処理、画像認識(ビジョン処理)性能の比較を行う。GPUによる画像処理、ビジョン処理ベンチマークとしては、GeForce7800GTXとCPU(Pentium4, 3.2GHz, SSE/HTチューニングなし)との比較や[17]、X800XT(500MHz)とCPU(Pentium4, 3GHz)との比較[18]が報告がされている。一方、動画認識プロセッサ IMAP-CEのCPU(Pentium4, 2.4GHz)との比較による画像処理、ビジョン処理ベンチマークも報告されている[4][14]。主なビジョン処理についてのGPUや動画認識プロセッサとCPU(Pentium4, 3GHz)とのベンチマーク性能比をまとめて示したものを図7に示す。

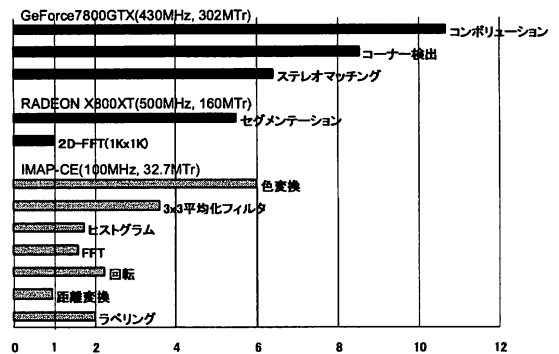


図7 ビジョン処理性能の比較(対P4-3GHz比)

また、より複雑な処理の例として動画認識プロセッサ(IMAP-CE)では、前述した前方車認識(レーン検出も含む)の例ではCPU(Pentium4, 2.8GHz)比で4倍弱の高速化を実現している[14]。

以上よりアルゴリズムにより差はあるが、IMAP-CEがCPU(Pentium4, 3GHz)の約3~4倍、ハイエンドGPUで6~8倍程度の性能と想定される。GPUの性能がIMAP-CEの2~2.5倍とすると、GPUはハードウェア(Tr数x周波数)比、消費電力比でIMAPの40倍程度となり、GPUはハードウェア効率、電力効率で

IMAP-CE と 15~20 倍の開きがある。理由としては、画像認識プロセッサが最小限の演算精度(8~16-bit)で処理を実現しているのに対し、GPU は 32-bit 浮動小数点演算を採用し、高速に動作させ、さらに 3D レンダリング用の専用回路を搭載している点が考えられる。

このように GPGPU は一つの解ではあるが、ハードウェア効率、電力効率が悪いため、車載へのそのままでの適用は困難と思われる。しかしながら、今後、画像認識処理が高度化するにつれ、一部の処理で GPU レベルの演算精度が必要になるとと思われる。そのため、一つの解として 10 倍程度の画像認識プロセッサの性能向上とミドルエンドクラスの(ノート PC 用、例えば GeForceGo6800) Mobile-GPU の組合せが考えられる。車載の目標スペックを 10 ミリ角程度、消費電力 2W として以上のスペックのプロセッサが実現できる時期(世代)をムーアの法則(あるいは ITRS ロードマップ[19])によって予測すると図 8 に示すように 65nm 世代でそれぞれ(画像認識、GPU)別チップとして実現でき、45nm 世代で SOC に統合されると予想される。

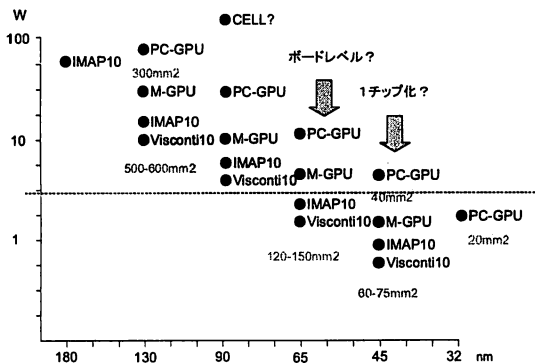


図 8 ムーアの法則による実現時期予測 (M-GPU: Mobile GPU、IMAP10: IMAP-CE x 10)

4. おわりに

車載応用のメディア処理において、安全運転支援やナビゲーションのための画像認識や 3D グラフィックスについて今後期待されるアプリケーション、これを実現するシステム LSI、実現コストについて紹介した。次の世代のターゲットとして現世代の 10 倍の画像認識性能とノート PC 並みの GPU 性能を車載向けに実現する時期の予測を行い、ボードレベルでは 65nm 世代、1 チップレベルでは 45nm 世代での実現を予想した。

参考文献

[1] M.Bertozzi, A.Broggi, M.Cellario, A.Fascioli, P.Lombardi, M.Porta, Artificial vision in road

vehicles, Proc. IEEE, Vol.90-7, July 2002 pp.1258 - 1271.

[2] V.Kastrinaki, M.Zervakis, K.Kalaitzakis: A survey of video processing techniques for traffic applications. Image Vision Comput. 21(4), 2003, pp.359-381.

[3] Z.Sun, G.Bebis, R.Miller, On-road vehicle detection: a review; IEEE Trans. PAMI, Vol28-5, May 2006, pp. 694 - 711.

[4] S.Kyo, Programmable SIMD Highly Parallel Processor Chips for Real-time Video Recognition Applications, Cool Chips VII, 2004.

[5] J.Schneider R.Westermann, GPU-Friendly High-Quality Terrain Rendering, Journal of WSCG,2006,vol.14,pp.49-56.

[6] N. Cornelis, K. Cornelis, L. Van Gool, Fast Compact City Modeling for Navigation Pre-Visualization CVPR 2006, Vol.2, 2006, pp. 1339 - 1344.

[7] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nistér and M. Pollefeys, Towards Urban 3D Reconstruction From Video, Invited paper, 3DPVT 2006.

[8] J. Montrym, H.Moreton, The GeForce 6800, IEEE Micro, Vol.25-2, 2005, pp.41-51.

[9] M.Kada, D.Klinec, N.Haala., Facade Texturing for rendering 3D city models. In: Proceedings of the ASPRS 2005 Annual Conference, Baltimore, USA. 2005. pp. 78-85.

[10] GPGPU.ORG

[11] N.Cornelis, L.V.Gool, Real-Time Connectivity Constrained Depth Map Computation using Programmable Graphics Hardware, CVPR'05, vol1, 2005, pp.1099-1104.

[12] S.Sinha, J.-M.Frahm, and M. Pollefeys, "GPU-based. Video Feature Tracking and Matching," Tech. Rep. TR06-012, University of North Carolina at Chapel Hill, May 2006.

[13] www.powervr.com.

[14] S.Kyo, T.Koga, S.Okazaki, I.Kuroda, A 51.2-GOPS scalable video recognition processor for intelligent cruise control based on a linear array of 128 four-way VLIW processing elements, IEEE JSSC Vol.38-11, Nov. 2003, pp.1992-2000.

[15] J.Tanabe, Y.Taniguchi, T.Miyamori, Y.Miyamoto, H.Takeda, M.Tarui, H.Nakayama, N.Takeda, K.Maeda, M.Matsui, Visconti: multi-VLIW image recognition processor based on configurable processor CICC 2003, pp. 185 - 188.

[16] T.Gandhi, M.Trivedi, Vehicle Surround Capture: Survey of Techniques and a Novel Omni-Video-Based Approach for Dynamic Panoramic Surround Maps, IEEE Trans. ITS, Vol 7-3, 2006, pp.293-308.

[17] D.Tarditi, S.Puri, J.Oglesby, Accelerator: simplified programming of graphics processing units for general-purpose uses via data parallelism, MSR-TR-2005-184, Microsoft, 2005.

[18] I.Buck, T.Foley, D.Horn, J.Sugerman, K.Fatahalian, M.Houston, P.Hanrahan, Brook for GPUs: Stream Computing on Graphics Hardware, SIGGRAPH 2004

[19] International Technology Roadmap for Semiconductors (ITRS): 2005 Edition.