

## 交通シミュレータのリアルタイム性能に関する評価と考察

植田 健夫<sup>1)</sup> 中村 俊一郎<sup>1)</sup> 宮西 洋太郎<sup>2)</sup>日本工業大学<sup>1)</sup> 宮城大学<sup>2)</sup>

Evaluation and consideration concerning real-time performance of traffic simulator

Takeo Ueta<sup>1)</sup>, Shunichiro Nakamura<sup>1)</sup>, Yohtaro Miyanishi<sup>2)</sup>

1), Nippon Institute of Technology, 2), Miyagi University,

## 1. はじめに

我々は、車 1 台 1 台の動きを重視したマイクロモデル交通シミュレータ NITTS の開発を進めている。NITTS の特徴は、リアルタイムシミュレーションを行えることである。しかし、マイクロモデルだけでなくマクロモデルほどでないが、大きな規模のシミュレーションを行うことが必要となってきた。しかし、シミュレーションの規模が大きくなり時間や表示台数が増加するとリアルタイムで行うことが困難になってくる。そこで、表示台数が多くなる試験でもリアルタイムで行うことができるようにするために、ハードウェア面とソフトウェア面での改良によって性能が向上するかどうか考察した。ハードウェアではビデオカードを変更することにより処理の多くの時間がかかる描画処理速度の向上につながるかを評価した。ソフトウェアではグラフィックハンドルをメモリに格納することによってのフォームへの描画処理の向上につながるかを評価した。

## 2. 試験モデル

評価に使用した試験モデルは NITTS の大規模モデルの横 10000m×縦 7000m で車輦表示台数は約 30000 台の試験である。NITTS での最大車輦発生台数は約 32000 台なので、この試験モデルをリアルタイムで行うことができればすべての試験モデルをリアルタイムで行うことができるので試験モデル 1 を評価用の試験モデルとする。

Fig. 1 の [a] の描画範囲は処理時間の評価として用いたもので、最大まで拡大し描画台数を 0 台にすることによって描画処理に関係しない計算処理の時間と考えることができる。

Fig. 2 の [b] の描画範囲は NITTS の標準的な試験モデルで使われている描画範囲で大規模シミュレーションモデルは主にこの描画範囲まで拡大して評価を行っている。

Fig. 3 の [c] の描画範囲は全描画状態である。

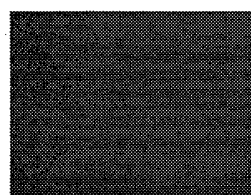
このように描画範囲を Fig. 1, Fig. 2 及び Fig. 3 のように変更し、リアルタイム性能の評価を行う。

Evaluation and consideration concerning real-time performance of traffic simulator

1) Takeo Ueta, Shunichiro Nakamura · Nippon Institute of Technology

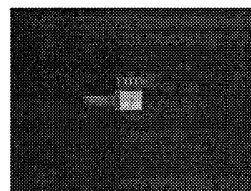
2) Yohtaro Miyanishi · Miyagi University

また、画面すべてに車輦が描画されている状態での 1 分間のシミュレーションにかかった処理時間を測定した。



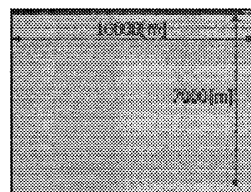
描画範囲  
縦 0 (m)  
横 0 (m)  
表示台数  
0 (台)

Fig. 1 [a] の描画範囲



描画範囲  
縦 700 (m)  
横 1000 (m)  
表示台数  
292 (台)

Fig. 2 [b] の描画範囲



描画範囲  
縦 7000 (m)  
横 10000 (m)  
表示台数  
29241 (台)

Fig. 3 [c] の描画範囲

## 3. ハードウェアでの性能向上

Table. 1 PC. 1 の環境

CPU	Intel Celeron M 1.73GHz
MEM	512MB
VGA	Intel 945GM
OS	Windows XP Professional SP2
CP	Microsoft Visual Basic 6.0

Table. 1 に示すような汎用的なパソコンの環境で行った場合の結果は Table. 2 のようになった。尚、プログラムはコンパイルされたものを使用

Table. 2 PC. 1 環境での 1 分間の処理時間

PC. 1	描画範囲 (m <sup>2</sup> )	[a]	[b]	[c]
	描画台数 (台)		0	292
処理時間 (sec)		330	335	696

そこで、PC.1 より高い描画性能を実現できるパソコンでなら描画処理にかかる時間を短縮できると考えられる。以下の Table.3 に示す PC.2 の環境で高速に処理できるかを評価した。

Table.3 PC.2 の環境

CPU	Intel Xeon 5110 1.60GHz
MEM	512MB
VGA	Radeon HD 3870
OS	Windows XP Professional SP2
CP	Microsoft Visual Basic 6.0

CPU は Xeon のデュアルコアになっているが、NITTS のプログラムはマルチスレッドに対応していないので、その他のパソコンより評価用としては適していたのでこの環境になった。

Table.4 PC.2 環境での 1 分間の処理時間

PC.2	描画範囲 (m <sup>2</sup> )	[a]	[b]	[c]
	描画台数 (台)	0	292	29241
	処理時間 (sec)	263	269	865

Table.4 の結果ハードウェアの変更ではあまりいい効果が得られなかった。しかし、描画範囲を狭くすると処理時間が PC.1 時より向上している。これは、デュアルコア CPU によって Windows による処理部分が高速化したためと考察できる。

#### 4. ソフトウェアでの性能向上

NITTS のプログラムでの問題点は実際に描画するところは Windows に処理を任せている部分が上げられる。Windows に描画処理を任せるのではなく NITTS のプログラム上からウィンドウを指定しメモリに格納することでメモリアクセスに変更でき処理速度の向上につながると言える。

そこで、まずシミュレーション画面部分のウィンドウフォームのウィンドウハンドルを取得しメモリに格納してみた。

Table.5 PC.2 環境でメモリ格納した処理時間

PC.2 Memory	描画範囲 (m <sup>2</sup> )	[a]	[b]	[c]
	描画台数 (台)	0	292	29241
	処理時間 (sec)	484	492	864

ウィンドウハンドルをメモリに格納するだけでは性能向上には至らなかった。さらにデバイスコンテキストを使ってグラフィック描画命令を行えばグラフィック処理の高速化につながるはずである。しかし、NITTS のグラフィック処理は様々な拡張がなされており、プログラムの大部分に変更を施さねばならないものであることが分った。よって、今回は変更を見送ることとする。

大規模シミュレーションは交通流を評価するものなのできめ細かくシミュレーションを行う必要は無く、描画範囲も 100 倍の大きさとなるので 0.1 秒間隔で処理しても変化が少なく実用的ではない。さらに演算も多くなり CPU の負担も増すことになる。よって、通常のシミュレーションの様に 0.1 秒間隔で評価を行うのは困難である。なので、大規模シミュレーションを評価する時に限り 1.0 秒間隔で処理を行えばリアルタイムで処理を行うことができると言える。

Table.6 PC.2 環境で処理間隔変更した処理時間

PC.2 clock	描画範囲 (m <sup>2</sup> )	[a]	[b]	[c]
	描画台数 (台)	0	292	29241
	処理時間 (sec)	60	60	87

完全なリアルタイムにはならなかったが、大規模シミュレーションの全画面でも処理を向上させるには、演算間隔を変更するのが小幅で簡単に変更できるので仕様に組み込むこととする。また、もう少し高性能な CPU を使用すれば大規模シミュレーションでもリアルタイムで行えると思う。

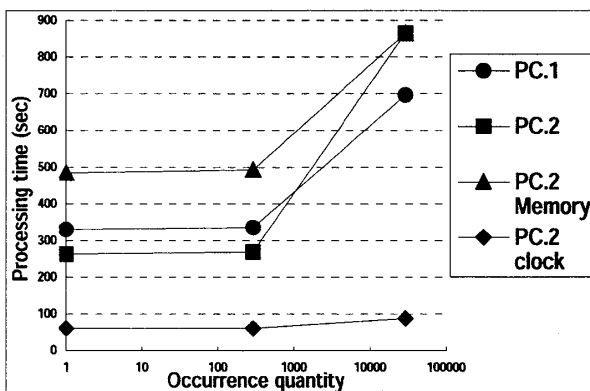


Fig.4 各 PC 環境の処理時間

#### 6. まとめ

ウィンドウのデバイスコンテキストを取得し API 関数を使用し描画を命令すれば処理時間の短縮につながることはなるはずである。しかし、このプログラムを完成させるにはプログラム上の大部分に変更を適用する必要がある。

大部分に変更を適用するならば VB6.0 環境でなく VB.NET の環境に変更してしまえば近年のマルチコア化によるソフトウェア環境にも対応し、その他機能の実装も可能となるので今後の開発が必要となると考える。

#### 参考文献

- 1) 神保尚義他：「交通シミュレータの限定描画による処理の高速化の研究」，情報処理学会第 68 回全国大会 (2006)