

マルチプロセッサ型交通シミュレータの開発*

4 Z B - 0 2 五十嵐 智也† 中村 俊一郎‡ 宮西 洋太郎‡ 斎藤 成一¶
 日本工業大学† 公立はこだて未来大学‡ 三菱電機(株)¶

1. はじめに

都市及びその近郊における交通渋滞の解消を目的に、交通渋滞を解析するための交通シミュレータを作成している。2000 年度に NITTS を開発した。これは 1000 m x 700m の領域をアニメーション表示するマクロシミュレーションである。2001 年度はより広域なシミュレートを行うための手段としてマルチプロセッサによるマルチ画面表示のアニメーションの手法を開発中である。パソコンを複数用い、各パソコンに 1000m x 700m の範囲のシミュレーションを分担させ、それらのディスプレイを連結して配置することにより、より広域な範囲を同時に画面表示させようとするものである。図 1 はこの手法の概要を示す図である。

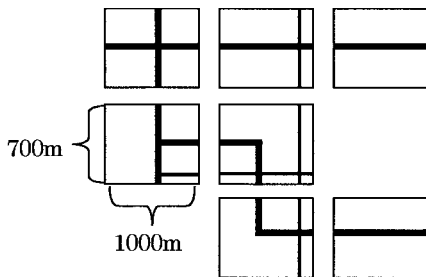


図 1 パソコン配置例

この方法の場合、パソコン一台の表示範囲を拡大したシミュレーションに比べ、車一台一台の動き、そしてシミュレーションマップ全体の動きを拡大、縮小操作なしに、容易に把握することができ、パソコン一台にかかる負荷も非常に軽くてすむ。又この方法はディスプレイの配置を自由に変えられるため、シミュレーションを行いたい場所の形に合わせてディスプレイを配置でき、必要なパソコンの台数を最小限に押さえる

*Development of traffic simulator which uses multiprocessing
 †Tomoya Igarashi, Shunichiro Nakamura
 ‡Yohtro Miyanishi
 ¶Seiich Saito
 † Nippon Institute of Technology
 ‡ Future University Hakodate
 ¶ Mitsubishi Electric Corp

ことが出来るという利点もある。

以下にこのシミュレータの概要を述べる。

2. システム構成

図 2 にハードウェア構成を示す。図示のようにパソコン群は、相互の通信手段として、100Mbps イーサネットに接続される。通信プロトコルは安定性より速度を重視した UDP/IP とし、通信ソフトウェアは Visual Basic の Winsock コントロールを使用した。

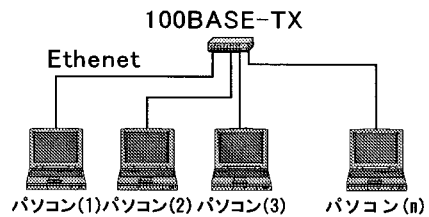


図 2 ハードウェア構成

3. シミュレータ概要

3.1 プログラム構造

データ入力プログラムはスタンドアロンで動作するが、その他はメインルーチンに統括されシミュレーション実行時に動作する。図 3 にそのフローを示す。

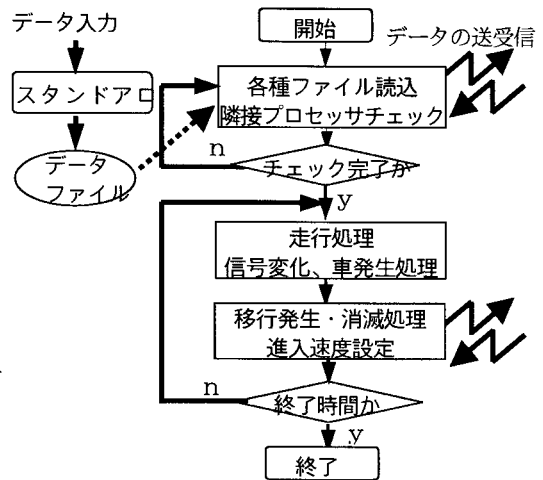


図 3 プログラムフロー

シミュレーション本体は単体でのシミュレーションに必要な各プログラム、複数のパソコンと通信を行いつつ各制御を行うプログラムに2分される。以下マルチプロセッサに対応させるに当たり追加したプログラムについて述べる。

3.2 全プロセッサ探索プログラム

任意のプロセッサをマスタープロセッサとして、マスタープロセッサが全プロセッサにデータを送る手法を採った。このためにはマスタープロセッサが全てのプロセッサのIPアドレスを知っている必要がある。本プログラムはこれらの全IPアドレスを探索するものである。

シミュレーションを開始する前に、それぞれのプロセッサの上下左右に位置するプロセッサのIPアドレスを指定し、任意の一台(マスタープロセッサ)でシミュレーションのスタートボタンを押す。まずマスタープロセッサは上下左右に位置する各プロセッサ(以下スレーブプロセッサ)に、上下左右にあるパソコンのIPアドレスを要求し、スレーブプロセッサはこれに返答する。そしてマスタープロセッサは返ってきたIPアドレス宛てに、同様にIPアドレスを要求する。以上を繰り返して得られたIPアドレスを隣接プロセッサ情報テーブルに格納し、返ってくるIPアドレスがユニークでなくなるまで繰り返す。

3.3 継続道路番号付与プログラム

隣接するプロセッサをまたがって道路が存在する場合、それぞれのプロセッサはこの道路が、お互いの相手プロセッサまでつながっていることを意識しなければならない。旧NITTSからの変更を最小限にするため道路番号はプロセッサごとに独立しているが、新規に隣接道路テーブルを作成することにした。図4は図1の中央の段を図示したもので継続道路が2本ある。円で囲んである数字が継続道路番号であり、左の道路番号とは無関係に、座標順に隣接道路番号を付与してい

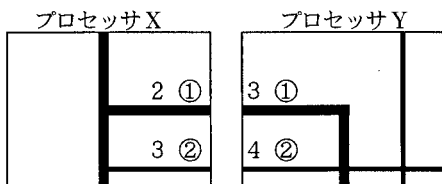


図4 隣接道路

く。以後道路に対しての通信は全てこの隣接道路番号によって行われる。

3.4 継続道路チェックプログラム

継続道路番号を基に道路をそれぞれ見比べ、位置、車線数、道路数が等しいかをチェックする。この時点で上記の情報が一致しない場合エラーとなる。

3.5 進入速度指定プログラム

複数のプロセッサでシミュレーションを行う場合、1つのプロセッサの時とは違い、シミュレーション中のあらゆるデータに自由にアクセスすることは大変である。勿論他プロセッサの情報が必要になる度、それを得るようにすることは可能であるが通信量が膨大になってしまう。この問題を解決するために、我々は他プロセッサへの進入速度を指定する、という近似的手法を考案した。プロセッサは隣接する全道路の全レーンへの進入速度を、状況(信号や継続点付近の車の距離)から設定し相手プロセッサに送信する。現在このデータは1秒おきに送信している。

4. おわりに

NITTSを用い、マイクロシミュレータを基にマクロなシミュレータを可能とするマルチプロセッサ型交通シミュレータの提案を行った。現在このシミュレータは開発の最終段階にあり、プロセッサ間を車がスムーズに移動する様子が確認されている。

参考文献

- [1] 桑原雅夫：広域ネットワーク交通流シミュレーション、自動車技術、Vol.52、No.1、1998
- [2] 猪飼國夫他：ファジィモデルに基づく市街地での渋滞予測用微視的シミュレータ、日本ファジィ学会誌、Vol.11、No.2、1999
- [3] 五十嵐他：道路交通シミュレータ NITTS の開発、日本シミュレーション学会、第20回シミュレーションテクノロジーコンファレンス、10-4、2001
- [4] 中村俊一郎他：交通シミュレータ NITTS とそのマルチプロセッサ化について、情報処理学会、高度交通システム(ITS)研究報告 No.7、2001