

VI. 各種支援機能・応用機能

1. 成熟都市シミュレータ

(1) シミュレーションとその階層

例えば「景観シミュレーション」という概念にも、いくつかの異なるレベルのシミュレーションが含まれており、それらが重層的に機能することにより実用的で有用なシステムが実現されます。

① 製図・絵描きのシミュレーション（レンダリング）

最も基礎には、与えられた2次元ないし3次元の実体（空間における線・面・立体）の形状をデータとして表現し（ファイルの形式、メモリ上の構造体等）、これを適当な視点から眺めた時のパースを画面に表示するシミュレーションです。線画として表示するためには、幾何学の座標計算だけで十分ですが、立体的に見せるためには、光源の方位・色彩と、それぞれの物体表面の色彩・反射率・模様（テクスチャ）の情報を用いた膨大な計算を行う必要があります。近年、コンピュータの処理速度の向上、及び定石的な処理をパッケージしたライブラリの充実により開発が楽になり、様々なアプリケーションに組み込まれて急速に実用化・普及しつつあります。

このシミュレーションの中で行われる計算は、図学・光学・幾何学等に基づくものであり、その手順は地域を問わずかなり客観的に規定することができます。

建設省版・景観シミュレータにおけるLSS-Gファイルは、状況に無関係に、物体固有の性質として記述できる情報（モデル）を記述したものです。これに対して、LSS-Sファイルは、このシミュレーションの中で、状況に応じてユーザーが変更することのできる光源、視点・注視点、時刻などの情報を定義することができます。景観シミュレータの中での機能としては、メイン画面における接近・後退、回転、シフトなどの視点移動、編集のメニュー中の視点設定・移動経路設定・可視範囲解析、シャッター機能、光源設定機能、表示の下の諸メニューなどがこのレベルのシミュレーションに対応するものです。

応用として、立体視（左右の目を視点位置として二つのパースを作成し、それぞれを左右の眼に別々に供給して立体的に見せる）のための装置（液晶シャッターメガネ等）を用いた技術、視点位置を少しずつ変化させながら連続的に表示し、動画として見せる技術、マルチスクリーンを用いて臨場感を高める技術などが普及し始めています。

② 設計・工事のシミュレーション（モデリング）

一定の固定された一つの仮想世界の中を歩き回るだけではなく、実際の土木建築施設の建設や除却に伴う景観への影響を検討するためには、風景を構成する諸要素を追加・削除・変形・加工するような機能を実現する必要があります。

実際の景観検討項目に対応した、図形の編集機能の性能の充実により、操作性が決定されます。

建設省版・景観シミュレータにおいては、メイン画面の選択機能、編集の配置機能、移動・回転・スケール機能、削除機能、マテリアル・テクスチャ編集機能、形状生成の諸機能が、これに対応しています。

この機能は、検討しようとする対象物の形状・仕上げ等が、単独または少数の設計・計画主体により予め決定されるようなプロセスに用いることができます。公共土木建築施設の設計、市街地再開発のように、予め形状を決定してから、実際の工事に入るようなプロセスに適しています。

このシミュレーションが根拠とするロジックは、単なる自然法則としての幾何学のみならず、土木

建築技術（施工技術）の体系、土木施設に係る構造令や建築基準法・都市計画法等が関係していますが、その内容はある程度客観的に把握し、アルゴリズムとしてプログラムに取り込むことが可能です。

③都市計画・社会現象のシミュレーション

都市は、都市を構成する個々の建築物（それぞれのオーナーによって計画される）によって構成されています。その増改築・建替等は、一般にはそれぞれのオーナーの自由意志により個々の時点で行われ、設計・計画も個別の条件により、別々に決定されます。

このような状況に対して、都市計画では、建蔽率・容積率、斜線制限等のゆるやかな条件により、個々の建築活動に対する制約条件を設けることにより、全体としての町並の形成に影響を与えています（必ずしもその目的とする所は、景観の形成ではない）。更に、近年では、地域の実情に合わせて、地域固有の将来像を設定し、これに基づいて個々の建築物の形状を規制することのできる地区計画や建築協定という制度が普及し、その中で良好な景観形成も有力な目的の一つとなってきました。

さて、このような都市計画条件の検討段階において、どのような条件を設定すれば、どのような市街地景観が予測されるか、という事項は重要な点です。（しばしば、都市計画が「個人の権利や自由の制約」として理解された場合に、将来像が無視された形で議論される事柄ですが）。

このことは、一つの条件設定を仮定した場合に、それに合致するような建築形状を一つ一つマニュアルで作成し、そのようにして作成したデータを膨大に配置することで市街地景観を検討する、という手順では膨大なコストが必要となります。

そこで、景観シミュレータでは、敷地割のデータを予め用意し、その上に都市計画条件を設定した上で、建築物の概形を自動的に生成する、市街地生成機能を用意しました(1997)。ここでの生成は1回限りであり、生成結果は、LSS-Gファイルとして生成し、景観シミュレータでこれを読み込んで、中を歩き回り眺めるような操作方法でした。

今般、建築研究所の特別研究「成熟社会に向けての都市解析データの整備と都市開発プログラム」において、都市の社会経済的条件も含めた「成熟都市シミュレータの開発を行いました。このシステムにおいては、都市のインフラ、敷地割とその上に建つ初期の建築物、都市人口分布を初期条件として、統計から得られた建築物の平均耐用年数等に基づく確率的な更新を、年次をステップとして行います（その地域全体の平均的な更新過程を、個々の敷地上の更新に適用する）。その際に、都市計画条件に対して限度一杯に建築が行われるのではなく、社会経済条件によっては、それよりもはるかに低い建物しか建たないような状況も反映させます。更に、シミュレーションの結果については、ネットワーク越しに連携した景観シミュレータに、年次をステップとしてオンラインでデータを送出・更新し、シミュレータの側では次第に未来に向かって進んで行く市街地景観の中を歩き回ることができるようになっています。

課題としては、社会現象のアルゴリズム、ロジックをプログラムに落とし込む際の客観性の問題があり、このことは究極的には社会現象そのものに、予測不可能性が常に内在するという問題に帰着します。但しこれは、人工生命等、そのロジックよりも寧ろ、シミュレーション結果の面白さ、尤もらしさで有効性を確認しようとする、近年のシミュレーション研究のあり方と合致した方向ではありません。

（2）共時態と通時態

設計計画において、例えば再開発や集合住宅の建替等での景観シミュレータの利用は、竣工時点における3次元的な空間構成（共時態）に対する評価に他なりません。これは設計者あるいはその集団による計画内容を表現することとなります。

一方、成熟都市シミュレータにおいては、「竣工時点」という概念は存在しません。検討開始時点における初期条件と、生成条件（都市計画条件等）から、毎年を時間単位として変化のシミュレーシ

ョンを行い、逐次景観シミュレータに差分情報を送出して表示を行います。オペレータが停止をかけるまで、原理的には無限の未来までシミュレーションが続きます（通時態）。

区画整理のように、設計・計画段階において、竣工時点における「完成像」を描ききることができず、個々の敷地における各施主による設計・建設に委ねられるようなタイプの事業の場合、事業計画段階では、建蔽率容積率をはじめとする都市計画的条件、あるいは更に詳細な地区計画・景観条例などによる制約条件下での、ランダム性をはらんだ生成プロセスをシミュレートする必要があります。一般地域における都市計画の検討も、この条件に近いでしょう。

更に広く見ると、日本あるいはアジアの都市においては、比較的寿命の長い耐火造の建物であっても、「半永久」と言えるような耐用年数を有するには至らず、物的あるいは社会的な耐用年数に至ると、更新を繰り返しています。従って、上に「共時態」として例示したような場合であっても、より長期的に見ると、通時態としての側面を有しています。言いかえると、再開発が完成・竣工しても、それが最終的な姿なのではなく、その建物が耐用年数に達し、再再開発される状況は内包されているのであり、理想的には、そのような状況が織り込まれたような形で当面の事業計画も考察されるのが真実味があります。このような意味で、都市計画という営為の本質は、日本あるいはアジアの諸都市においては、その完成像を絵（「共時態」）として提示することではなく、予想することに意味のある範囲での有限の未来という視界の中で、当面想定される問題を解決・回避するための当座の方向を見出すこと、さらにそのような営為を、継続的に行う（通時態）こと、という性格を帯びます。これは、視界範囲に感知される情報に基づいてコースを選択する「運転」ないし「操縦」という行為に近いものです。

シミュレーションの技術は、このような営為を支援するものとして期待されるでしょう。

（3）シミュレーションのアルゴリズム

成熟都市シミュレータのロジックの本質的な部分は、以下の機能である。その他に、データ編集・ファイルロード・セーブ、各種条件設定等の編集機能、景観シミュレータとの通信機能、歴史的累積情報のデータベース登録、及びこれに基づく各種評価ロジックなどがシステムには組込まれています。

① 年次ステップによる滅失・生成の処理

既存の建物を全てスキャンし、(2)のアルゴリズムに従って滅失建物をピックアップします。滅失が生じた場合大都市モデルにおいては、直ちに(同年次内に)建替に移行します(空地・駐車場以外のいずれかの建築類型に移行します)。地方都市モデルにおいては、除却が行われた場合、直ちに建替えとするのではなく、ひとまず空地または駐車場とします。一般地域の場合には、滅失のあった土地について、従前からの空地・駐車場と同じ条件で、新規生成の評価を行います。

新規生成に関する評価関数が真値(類型番号)を返した場合、敷地条件及び都市計画条件に従って、新築建物の形状を計算します。除却・滅失にかかる差分データについては、年次別にとりまとめ、表示装置である景観シミュレータに、ネットワーク経由で送じます。同時に、評価に必要な累積データとして追加記憶します。

② 滅失の確率計算

それぞれの建物については、構造及び建設年次のデータが付されているので、現在年次と建設年次の差から、建物の年齢を求め、別途用意した構造別・年齢別滅失率/年に比例した確率で真値を返す、乱数を用いた評価関数を適用して、除却の有無を判定します。除却の確率については、現在までの所、地方都市が対象となっていることから、別プロジェクトで固定資産台帳の調査により建物更新の状況を分析した二本松のデータを用いています(文献2)。

設定された災害条件により、災害が発生すると、その年次に限り、通常の除却処理とは別に、再度災害による滅失の確率計算を行います。

③ 新規生成の確率計算

建築物は、類型+パラメータにより記述します。形状生成は、パラメータにより形態を生成する、建築類型関数によって行います。

既存の空地・駐車場、及び大都市モデルの場合、当該年次に滅失のあった敷地について、新規生成の確率計算を行います。

新規生成の確率、類型選択、及び類型内のパラメータ設定を評価関数が決定します。形状生成においては、敷地条件（形状及び接道条件）、及び都市計画条件により基本的な形状を決定します。

経済状態（生成確率・類型選択）、地盤条件（特殊基礎が必要な場合、高い建物が成立しにくい）、表土条件を加味しています（Ver.1.0ではまだ実装していません）。

これらの条件の内、シミュレーションを行う対象地区全域に係るものについては、デフォルト値を与えるようになっていきます。また、市街化区域・用途地域・防火地域のように、区域を限定して適用されるもの（対象地区が複数の地区区分にまたがる場合）については、個々の敷地に関して、どのアイテムにおけるどの区分に属しているかについてを予め与えておき、シミュレーションの段階で、各地区毎にどのような条件を与えるかを指定できるようになっています（まだ実施事例はない）。

④ 生成シミュレーションの結果に基づく形態生成

生成評価関数が生成することを要求したそれぞれの建築類型について、敷地条件及び都市計画条件との照合を行い、規制内容を満たす所まで、建ぺい率、ヴォリューム、高さ、及び建物壁面線をトーンダウンします。

成熟都市シミュレータVer.1.0においては、メニュー上は、都市計画条件として、以下の項目を用意しています。

- a. 建ぺい率：敷地面積に対する建坪の割合の上限値
- b. 容積率：敷地面積に対する延べ床面積の上限値
- c. 道路斜線：道路境界における上限高さ、そこから敷地内部に向って上る坂の勾配として表現しています
- d. 壁面後退：道路境界からのセットバックとして表現しています。

これらについては、（あ）まず対象地域全体についてのデフォルト値として定めます。他に指定がない場合には、この条件が適用されます。（い）次に、市街化区域・調整区域、都市計画区域外について定めます。（う）更に、用途地域別に指定がある場合には、これを適用します。（え）最後に、同じ用途地域内に、上記の条件の指定の異なるサブ区域がある場合には、これを設定します。優先順位は、

下位 （あ）<（い）<（う）<（え） 上位

であり、上位の定めなき時は、より下位（即ち最も上位の）ものが適用されます。

⑤ 都市施設・周辺景観の処理

成熟都市シミュレータVer.1.0においては、都市施設及び周辺地形・背景は、固定（時間に依存しない）ものとして扱っています。これらは、景観シミュレータのLSS-G形式のデータとして記述されています。これらが指定されている場合、リセット・コマンドを実行した際に、それぞれを景観シミュレータに、そのままの形で送出して表示させた上で、ファイル保存命令を発行し、部品として利用

できるディレクトリにファイル保存します。

道路に関しては、交差点をまず生成し、その間を断面と中心線軌跡で結ぶ方法でパラメトリックに生成します。

シミュレーションのセッションにおいては、これらをファイルとして読み出し、表示するコマンドを発行します。

⑥ 景観シミュレータとのデータ交換

評価セッション実行時には、年次をステップとして、変更部分のみの差分ファイルを、バッファリングによりバースト・モードで転送することにより処理効率を高めています。

成熟都市シミュレータは、3次元表示機能を有する景観シミュレータとは、別のプロセスで動作します。ネットワーク上で結合された二つのコンピュータで分業することを想定して開発されています。

最も単純には、まず成熟都市シミュレータが、[1]ある年次の市街地の3次元的構成要素の全てを発生させ、[2]これを景観シミュレータに転送し、[3]景観シミュレータがそれを読み込み、[4]表示する。という流れが考えられます。しかしながら、この場合、二つのコンピュータ（ないしプロセス）は、互いに相手が処理を終了する間、待っていることとなります。また、大量の形状データを伝送するためには、時間を必要とします。

そこで、成熟都市シミュレータと景観シミュレータの間のデータ交換に関しては、以下のような工夫が加えてあります。

a. 市街地の変容に際して、差分に係るデータのみを転送する。

シミュレーション開始時点で、「リセット」の処理を行うことにより、まず成熟都市シミュレータは、都市の初期状態を記述する形状データを全て、景観シミュレータの側に送付します。

次に、年次毎のステップ処理においては、削除される要素について、DELETE(グループ名称); コマンドを送ることにより、まず削除を行い、次に新たに付加える要素の形状を記述するデータを送出します。

成熟都市シミュレータが、内部で保持する市街地構成要素のデータベースと、景観シミュレータの側で蓄積されている景観構成要素のデータの同一性を保つように処理することにより実現しています。

万一、シミュレータ側のユーザーが、手作業などにより、ある建物データを削除した場合は、一貫性が失われ、転送された削除コマンドがエラーを起し、メッセージが発生すると、景観シミュレータ側の受付処理が止ってしまいます。このような事態を避けるために、成熟都市シミュレータの側から、景観シミュレータのエラー報告モード（エラーが生じても、景観シミュレータのユーザー側にはメッセージを出さない）をコントロールするようにしています。

b. バッファリングにより成熟都市シミュレータによる市街地生成（次の年次）と、景観シミュレータによる読み込み・表示処理を同時並行的に行うことにより、双方の待ち時間を減らす。

成熟都市シミュレータの側（送り側）では、ある年次のデータの生成・シミュレータへの送付を完了すると、相手側の処理完了を待たずに、次の年次のデータの作成に入ります。景観シミュレータの側では、忙しい処理中であっても、タイマー割込みにより定時的に送られてきたデータを監視し、送られてきたら時間のかかる内部処理を棚上げとして、まずバッファに取り込み、受信完了の返事を出してから、処理を続行します。このような協調により、意味のない待ち時間を無くし、マルチCPUまたはネットワーク分業によるパフォーマンスの最適化を図っています。

バッファリングの節目となるのは、描画コマンド CAT();です。通信回線の効率を上げるために、成熟都市シミュレータの側では、このコマンドが到来するまで、送信を保留してメモリ上に削除・形状生成のコマンドを蓄積し、上記コマンドが現れた段階で、まとめてバースト・モードでデータを転送します(パケットよりも小さなデータ単位を送出すると、効率が下がる)。受け側では、描画コマンドを受けるまでは、構文解析などを行っているのみで、時間のかかるレンダリングには入らないので、受信バッファの次の処理工程としては、比較的迅速に処理を完了することが可能です。

- c. 定型化していて繰返し使用可能な部品やパラメトリックな部品は、受け側の景観シミュレータの側で保有しておき(ない場合には、セッション開始に先立って転送しておく)、通信内容は、部品0名称の指定、パラメータのみとする。

これは、形状記述ファイル LSS-G を小さくするのと同じ技術です。

⑦ シミュレーションの結果として得られた都市(将来)像の評価

評価には多数の項目があります。Ver.1.0においては、単体指標を用いて、群の時間的な推移を通時態を辿る統計機能と、空間構成について共時態を評価する図化機能を用意しています。最終的な適否を判定する機能ではなく、ユーザーが評価を行うための参考資料を作成する機能となっています。

シミュレーションの実行時(ステップ実行、及び連続実行)においては、ヒープ領域に確保した配列に、評価に必要な集計値を、累積的に蓄積して行きます。評価機能が要求された場合には、これらのデータを取出し(単体指標による集計、及び空間構成)、ユーザーが要求したフォーマット(図・ファイル・3次元形状修飾)によって出力します。

(4) データ形式と作成方法

成熟都市シミュレータのデータは、基本的には、初期条件設定に係る条件(インフラ、敷地割、建築物の初期状態(類型及び築後年数)、人口配置)など、及び、形成条件に係る条件です。

① 領域条件

a. 地割初期状態

これは、シミュレーションの中で更新後の建築物を自動的に発生させるために重要な条件です。専用住宅、店舗併用住宅などの各類型は、敷地との関係、及び絶対方位に基づいて、それぞれの類型毎の法則性に従って形状を生成します。

地割の初期状態は、地割定義ファイル(*.ldt)によって記述します。これは、景観シミュレータ2.03付属の都市開発シミュレータの形式に準拠していますが、さらに拡充して、Ver.2として、細かな接道条件などを記述できるようにしてあります。また、Ver.2においては、敷地形状を記述するポリゴンの頂点数を無制限にしてあります。

【Ver.2 のフォーマット】

第1行目に

VERSION2

と記述します。これでVer.2を識別します。

‘#’で始まる行は、コメント行です。

Filcon1.exe を用いて、複数の地割データを合成して一つの地割定義ファイルを作成した場合、元の

ファイル名称を示すコメント行が自動的に挿入されます。

一つの敷地の形状が1行で記述されます。このフォーマットは、

“名称” 標高 頂点数 頂点座標

です。頂点座標は、X値 Y値の組が、頂点数だけ1行の中で繰返されます。最後の座標値は、開始点に戻る必要はありません。但し、反時計回りに記述されるものとし、最初の点と次の点を結ぶ線分が主な前面道路に面する辺を表すという約束です。

```
VERSION2
#地区コード:<d:¥@keikan¥入力¥現況¥0000>
"d4001x" 69.1 12 56209.273 194964.669 56212.600 194911.871 56215.726 194892.121 56222.653 194869.316
56292.750 194866.896 56292.801 194927.522 56331.771 194929.617 56326.442 194973.091 56292.724 194970.077
56292.260 194991.703 56279.256 194991.844 56279.233 194970.736
#地区コード:<d:¥@keikan¥入力¥現況¥0001>
"f1301" 65.7 4 56767.176 194939.042 56772.457 194950.445 56760.878 194951.169 56759.746 194939.573
"f1302t" 65.4 4 56761.391 194926.686 56767.235 194938.981 56746.758 194940.483 56746.155 194928.116
"f1303t" 65.8 5 56756.792 194917.250 56761.496 194926.905 56746.156 194928.156 56745.677 194919.067
56753.970 194918.300
"f1304t" 65.7 5 56751.649 194905.744 56756.729 194917.111 56753.869 194918.342 56745.655 194919.007
56745.214 194908.757
"f1305c" 65.7 4 56747.792 194897.155 56751.590 194905.825 56745.113 194908.720 56744.716 194898.668
"f1306" 65.7 4 56760.796 194951.131 56747.287 194952.150 56746.701 194940.683 56759.663 194939.514
#地区コード:<d:¥@keikan¥入力¥現況¥0002>
"f1401" 65.8 4 56739.195 194952.413 56719.547 194953.453 56719.050 194943.854 56738.537 194942.757
"f1402" 65.9 5 56719.605 194953.312 56706.888 194954.019 56706.441
```

実際の敷地条件の入力は、手間のかかる作業になります。そこで、本システムを活用しようとする現場に対しては、デジタイザを用いることを薦めています。これにより、長時間のVDT作業なしに、図面の形で用意された現況地割や換地計画図などから入力作業を行うことが可能となります。スキャナーで下図を読み込み、画面で作業する、という方法も不可能ではないが、現在までのマウスなどのデバイスは、正確な位置情報を入力する手段としては、細部においてやや不安定です。

デジタイザは、現場により機種が異なり、入力データのFORMATは、機種により様々です(例えば、ボタンが複数ある場合、押されたボタンがどれか、に対応する情報が付け加わる)。このために、contour.bas というbasic インタープリタを用いた入力ソフトを用意しています。現場で機種の仕様にあわせて、入力部分を改変できる小さなソフトです。

Fullset でインストールした場合、contour.bas は、

¥keikan¥maju¥contour

のディレクトリにあります。

入力ミスに対する修正機能が貧弱なので、小さな区画に分けてデータを入力することを薦めています。座標系に関しては、1:2500 基本図などに区域割の線を入れたものを用意し、これから分割された各区域(入力の単位)の地図の四隅の座標値を計測しておきます。Contour.bas で一つのファイルを作成する時に、まず四隅の座標値を聞いてくるので、そこで入力を行います。入力された各敷地のデータは、上記の座標系に基づく座標値に変換されて出力されます。

デジタイザ各機種への対応は、

A. 初期化部分

行番号150~210

(デジタイザのモード設定等)ここに掲載した例は、なつかしいNEC社のPC9801シリーズに接続した場合の設定方法です。

B. 1クリック毎のデータ取得部分

行番号250~360

(各機種データ・フォーマットに対応)

を修正することにより行います。Bでは、デジタイザから送り返された情報の内、座標値に関するものを取出して、X、Yの値として返すほか、曲線状の境界などにおいて、ドラッグが行われた時の、待ち時間及びサンプリング間隔(解像度)などを指定しています。

Conteur.bas

```

1 '93/09/2322:13:44
10 DEF FNT(X)=VAL(RIGHT$(TIME$,2))
20 DEF FNX(X)=INT(X/6!)
30 DEF FNY(Y)=400-INT(Y/6!)
35 DEF FND(X1,Y1,X2,Y2) = SQR((X1-X2)*(X1-X2) +
(Y1-Y2)*(Y1-Y2))
36 DEF FNF(X,Y)=PARAMS.A*X+PARAMS.B*Y+PARAMS.C
37 DEF FNFY(X,Y)=PARAMS.D*X+PARAMS.E*Y+PARAMS.F
40 DIM PX(1000),PY(1000)
41 DIM MAT(3,3)
45 XMAX = 3800 : YMAX = 2600 '寒研仕様
50 CONSOLE 0,25,0,1
60 SCREEN 3,0,0,1
70 CLS 3
80 GOSUB*INITIALIZE
90 GOSUB*FILER
100 IF NEW.FLAG THEN GOSUB*GET.THE.FRAME ELSE
GOSUB*RECOVERER
110 GOSUB*MAIN
120 GOSUB*FINE
130 END
140 '
150 *INITIALIZE
160 DELTA1=10 : DELTA2=20 : OUTFILES="D:POLYGON.DAT"
170 OPEN "COM:N72" AS #1
175 PRINT #1."P:"
180 PRINT #1,CHR$(AH1B)+ "F"; "ASCII CODE MODE
COMMAND TO DIG."
190 PRINT #1,CHR$(AH1B)+ "I"; "ONE SHOT STREAM MODE
COMMAND TO DIG."
200 PRINT #1,CHR$(AH1B)+ "U"; "SAMPLING RATE 200
POINTS/SEC COMMAND TO DIG."
210 RETURN
220 '
230 *GET.A.NEW.POINT
240 XSAVE=-100:YSAVE=-100
250 *GET.A.POINT
260 PRINT #1."PZ:" 'START COMMAND TO DIG.
270 INPUT #1,X,Y,BUTTON.FLAG
280 IF XSAVE=X AND YSAVE=Y THEN TSAVE=FNT(0):GOTO 280
290 TIME=FNT(0):DELT=TIME-TSAVE:IF DELT<0 THEN
DELT=-DELT+60
300 IF DELT > 1 THEN 340
310 R=(XSAVE-X)*(XSAVE-X)+(YSAVE-Y)*(YSAVE-Y)
311 R=FND(XSAVE,YSAVE,X,Y)
320 IF R<100 THEN TSAVE=TIME:GOTO 260
321 IF R<DELTA1 THEN TSAVE=TIME:GOTO 260
330 BEEP 1:BEEP 0:PRINT "[X:";X;":Y:";Y;":I:";GOTO 350
340 BEEP 1:PRINT "[X:";X;":Y:";Y;":B BEEP 0
350 XSAVE=X : YSAVE=Y : TSAVE=TIME
360 RETURN
370 '
380 *GET.THE.FRAME:PRINT "図面枠の指定 (右上->左上->左
下->右下)"
390 PRINT " 4点をデジタイズ":
GOSUB*SILENT.DELAY2:GOSUB*BEEP.DELAY1
400 PRINT"右上":GOSUB *GET.A.POINT:UR.X=X:UR.Y=Y
405 INPUT "座標値 X,Y : ",FUR.X,FUR.Y
410 PRINT"左上":GOSUB *GET.A.POINT:UL.X=X:UL.Y=Y
415 INPUT "座標値 X,Y : ",FUL.X,FUL.Y
420 PRINT"右下":GOSUB *GET.A.POINT:LR.X=X:LR.Y=Y
425 INPUT "座標値 X,Y : ",FLL.X,FLL.Y
430 PRINT"右下":GOSUB *GET.A.POINT:LR.X=X:LR.Y=Y

```

```

435 INPUT "座標値 X,Y : ",FLR.X,FLR.Y
440 GOSUB*BEEP.DELAY3
450 *SHOW.THE.FRAME
460 LINE (FNX(UR.X),FNY(UR.Y))-(FNX(UL.X),
FNY(UL.Y)),5
470 LINE (FNX(UL.X),
FNY(UL.Y))-(FNX(LL.X),FNY(LL.Y)),5
480 LINE (FNX(LL.X),
FNY(LL.Y))-(FNX(LR.X),FNY(LR.Y)),5
490 LINE (FNX(LR.X),
FNY(LR.Y))-(FNX(UR.X),FNY(UR.Y)),5
500 PRINT "枠表示完了"
510 DX=UL.X-LL.X:DY=UL.Y-LL.Y:R=SQR(DX*DX+DY*DY)
520 A1=DY/R:B1=-DX/R:C1=-A1*UL.X-B1*UL.Y
530 DX=UR.X-LR.X:DY=UR.Y-LR.Y:R=SQR(DX*DX+DY*DY)
540 A2=-DY/R:B2=DX/R:C2=-A2*UR.X-B2*UR.Y
550 DX=UR.X-UL.X:DY=UR.Y-UL.Y:R=SQR(DX*DX+DY*DY)
560 A3=DY/R:B3=-DX/R:C3=-A3*UR.X-B3*UR.Y
570 DX=LR.X-LL.X:DY=LR.Y-LL.Y:R=SQR(DX*DX+DY*DY)
580 A4=-DY/R:B4=DX/R:C4=-A4*LL.X-B4*LL.Y
590 RETURN
591 '
599 *EDGE.CHECK : EDGE=0
600 IF A1*X+B1*Y+C1<0 THEN EDGE=EDGE+1
601 IF A2*X+B2*Y+C2<0 THEN EDGE=EDGE+2
602 IF A3*X+B3*Y+C3<0 THEN EDGE=EDGE+4
603 IF A4*X+B4*Y+C4<0 THEN EDGE=EDGE+8
604 RETURN
605 '
610 *GET.A.POLYGON:PRINT"ひとつの敷地を入力します。";
611 PRINT " 道路から見た正面左隅から反時計回りに進んで
下さい。"
612 PRINT " (終了する場合は、デジタイザの右辺付近を指
定)"
620 XSAVE=-1:YSAVE=-1:N=1
630 GOSUB*SILENT.DELAY2:GOSUB*BEEP.DELAY1
640 PRINT"最初の点を入力待ちです。"
650 GOSUB *GET.A.POINT
660 IF X>3750 THEN END FLAG=1:RETURN 'セッション終了
670 START.X=X:START.Y=Y:START.FLAG=1
680 PX(0)=X:PY(0)=Y
690 EDGE.FLAG=0 : EDGE2.FLAG=0
700 GOSUB*EDGE.CHECK:EDGE.FLAG=EDGE '最初の点の範囲
内外判定
710 IF EDGE.FLAG=0 THEN 750 '範囲内である場合
711 GOSUB*BEEP.DELAY1 '範囲外である場合
712 X.SAVE=X:Y.SAVE=Y
713 GOSUB*GET.A.POINT '範囲内に到達するのを待つ
714 GOSUB*EDGE.CHECK
715 IF EDGE THEN 712
716 EDGE=EDGE.FLAG
718 GOSUB *GET.MIDDLEPOINT '壁を越えた地点を計算する
719 X.START=X.MIDDLE.Y.START=Y.MIDDLE
720 PX(0)=X.START:PY(0)=Y.START
721 START.FLAG=0
722 GOTO 870
729 '
730 *GET.MIDDLEPOINT
732 IF EDGE=1 THEN
K1=A1*X.SAVE+B1*Y.SAVE+C1:K2=A1*X+B1*Y+C1:GOTO 745
733 IF EDGE=2 THEN
K1=A2*X.SAVE+B2*Y.SAVE+C2:K2=A2*X+B2*Y+C2:GOTO 745
734 IF EDGE=4 THEN
K1=A3*X.SAVE+B3*Y.SAVE+C3:K2=A3*X+B3*Y+C3:GOTO 745
735 IF EDGE=8 THEN
K1=A4*X.SAVE+B4*Y.SAVE+C4:K2=A4*X+B4*Y+C4:GOTO 745

```



```

K1=A3*X.SAVE+B3*Y.SAVE+C3:K2=A3*X+B3*Y+C3:GOTO 745
735 IF EDGE=8 THEN
K1=A4*X.SAVE+B4*Y.SAVE+C4:K2=A4*X+B4*Y+C4:GOTO 745
737 ERROR 255
745 X.MIDDLE=(K2*X.SAVE-K1*X)/(K2-K1)
746 Y.MIDDLE=(K2*Y.SAVE-K1*Y)/(K2-K1)
747 PRINT
X.SAVE,Y.SAVE,X,Y,"X.MIDDLE=";X.MIDDLE,"Y.MIDDLE=";Y.
MIDDLE
748 RETURN
749
750 GOSUB *GET.A.POINT'枠内の形状記憶開始
760 IF X>3750 THEN EDGE.FLAG=1:GOTO 900 '強制終了の
場合
770 IF START.FLAG=0 THEN 800
780 IF ABS(START.X-X)+ABS(START.Y-Y)>DELTA2 THEN
START.FLAG=0:GOTO 870
781 IF START.FLAG=1 THEN PRINT "START.FLAG=1
782 IF FND(START.X,START.Y,X,Y)>DELTA2 THEN
START.FLAG=0
783 GOTO 870
790 PRINT "EDGE.FLAG=",EDGE.FLAG;
800 IF EDGE.FLAG THEN 810 ELSE 850
810 GOSUB*EDGE.CHECK
820 IF EDGE=0 THEN 850
830 EDGE2.FLAG=EDGE
835 X.SAVE=PX(N-1):Y.SAVE=PY(N-1)
840 GOSUB*GET.MIDDLEPOINT
845 X=X.MIDDLE:Y=Y.MIDDLE:GOTO 900
850 PRINT "FROM START=";ABS(START.X-X)+ABS(START.Y-Y)
860 IF ABS(START.X-X)+ABS(START.Y-Y)<DELTA1 THEN 900
861 IF FND(START.X,START.Y,X,Y)<DELTA1 THEN 900 '出
発点近傍に至る
870 PX(N)=X:PY(N)=Y
875
LINE(FNX(PX(N-1)),FNY(PY(N-1)))-(FNX(PX(N)),FNY(PY(N)
)),7
880 N=N+1
890 GOTO 750
900
PX(N)=X:PY(N)=Y:N=N+1:PX(N)=EDGE.FLAG:PY(N)=EDGE2.FLA
G
910 GOSUB*BEEP.DELAY3
920 RETURN
930
940 *SAVE.THE.POLYGON
950 HN=1:FOR I=0 TO NCONTEUR
951 IF INT(CONTEUR)=INT(H(I)) THEN HN=HN+1
952 NEXT
960 NCONTEUR=NCONTEUR+1
970 H(NCONTEUR)=CONTEUR
975 N$(NCONTEUR)=NOMOR$
980 HN(NCONTEUR)=HN
990 CONJ(NCONTEUR)=EDGE.FLAG OR EDGE2.FLAG
1010 GOSUB*MAKE.CONTNAME
1020 OPEN CONTNAME$ AS #2
1030 IF EOF(2) THEN 1040
1035 PRINT "そのデータは既にあるが、それを消すか(Y-N) ?
";
1036 GOSUB*BACK.WIND
1037 C$=INPUT$(1):PRINT:IF C$<>"Y" THEN 1039
1038 PRINT #2,"###appended###":GOTO 1040
1039 CLOSE #2:KILL CONTNAME$:OPEN CONTNAME$ AS #2
1040 WRITE #2,N,HN
1050 FOR I=0 TO N:WRITE #2,PX(I),PY(I):NEXT
1060 CLOSE #2
1070 RETURN
1080
1090 *MAKE.CONTNAME
1100
CONTNAME$=FILENAME$+RIGHT$("0000"+MID$(STR$(INT(CONTE
UR)),2),4)
1110
CONTNAME$=CONTNAME$+"."+RIGHT$("0000"+MID$(STR$(HN),2)
,3)
1120 RETURN
1130
1140 *CHECK.THE.POLYGON
1150 C=5:GOSUB*DRAW.THE.POLYGON
1200
1210 GOSUB*SILENT.DELAY2:GOSUB*BEEP.DELAY1
1220 PRINT "OK:左半分 NG:右半分 をクリック";:GOSUB
*GET.A.POINT
1230 IF X>XMAX/2 THEN C$="N" ELSE C$="Y":PRINT C$
1240 IF C$="N" THEN

```

```

C=0:GOSUB*BEEP.DELAY2:GOTO*DRAW.THE.POLYGON
1250 GOSUB*BEEP.DELAY3
1255 GOSUB*SILENT.DELAY2:GOSUB*EDGE.BEEP
1260 INPUT "敷地の標高は (m) : ",CONTEUR
1265 INPUT "敷地の地番は : ",NOMOR$
1270 C=3 'C=1+((CONTEUR/2)MOD 5)
1290 GOSUB*DRAW.THE.POLYGON
1320 GOSUB*SILENT.DELAY2:GOSUB*LONG.PROMPT
1321 PRINT "OK:左半分 NG:右半分 をクリック";:GOSUB
*GET.A.POINT
1322 IF X>XMAX/2 THEN C$="N" ELSE C$="Y":PRINT $
1323 IF C$="N" THEN C=0:GOSUB*BEEP.DELAY2:GOTO 1255
1327 GOSUB*SUCCESS
1330 RETURN
1339
1340 *DRAW.THE.POLYGON
1341 FOR I=0 TO N-2
1342 LINE
(FNX(PX(I)),FNY(PY(I)))-(FNX(PX(I+1)),FNY(PY(I+1))),C
1343 NEXT I
1344 IF PX(N) THEN RETURN 'from edge to edg; not closed
open loop
1345 LINE
(FNX(PX(I)),FNY(PY(I)))-(FNX(PX(0)),FNY(PY(0))),C
1346 RETURN
1350 *MAIN
1360 GOSUB *GET.A.POLYGON
1370 IF END.FLAG THEN CLS 3:RETURN
1380 GOSUB *CHECK.THE.POLYGON : IF C$<>"Y" THEN 1360
1390 GOSUB *SAVE.THE.POLYGON
1400 GOTO *MAIN
1410 *WIDOWS
1420 *LONG.PROMPT
1430 *EDGE.BEEP: FOR TIME=1 TO 20:BEEP 1:BEEP 0:BEEP
1:BEEP 0:NEXT:RETURN
1440 *PROMPT1
1450 *BEEP.DELAY1: FOR TIME=1 TO 50:BEEP 1:BEEP
0:NEXT:RETURN 'PROMPT
1460 *SUCCESS
1470 *BEEP.DELAY3
'SUCCESS
1480 FOR TIME=1 TO 5:BEEP 1:FOR TIME=1 TO 100:NEXT
1490 BEEP 0:FOR TIME=1 TO 100:NEXT
1500 NEXT:RETURN
1510 *ALARM
1520 *BEEP.DELAY4:FOR TIME2=1 TO
3:GOSUB*BEEP.DELAY3:GOSUB*SILENT.DELAY
1530 NEXT:RETURN
1540 *SILENT.DELAY2:FOR TIME=1 TO
3:GOSUB*SILENT.DELAY:NEXT:RETURN
1550 *SILENT.DELAY:FOR TIME=1 TO 1000:NEXT:RETURN
1560 *FAILURE
1570 *BEEP.DELAY2:FOR TIME=1 TO 30:BEEP 1:BEEP 0:FOR
TIME=1 TO ABS(15-TIME)
1580 NEXT:NEXT:RETURN
1590 *BACK.WIND
1600 FOR TIME=1 TO 25:BEEP 1:BEEP 0:FOR TIME=TIME TO
40:NEXT:NEXT:RETURN
1610 *FRONT.WIND
1620 FOR TIME=1 TO 35:BEEP 1:BEEP 0:FOR TIME=1 TO
TIME:NEXT:NEXT:RETURN
1630 *FILER
1640 DIM H(1000),HN(1000),CONJ(1000),N$(1000)
1650 PRINT "ドライブは(A-F) : ";
1660 C$=INKEY$ : IF C$=" " THEN 1660
1670 IF C$=" " THEN 1660 ELSE C=ASC(C$)
1680 IF 97<=C AND C<=102 THEN 1710
1690 IF 65<=C AND C<=70 THEN 1720
1700 GOTO 1660
1710 C=C-32
1720 DRIVE$=CHR$(C)+" : ";
1730 PRINT "[";DRIVE$;"]"
1740
1750 PRINT "地図の番号 : ";
1760 INPUT BLOCK.NUMBER
1770 IF BLOCK.NUMBER<0 THEN 1760
1780 IF BLOCK.NUMBER>9999 THEN 1760
1790
FILENAME$=DRIVE$+RIGHT$("0000"+MID$(STR$(BLOCK.NUMBER
),2),4)
1800
1810 OPEN FILENAME$ AS #2
1820 IF EOF(2) THEN NEW.FLAG=1 ELSE NEW.FLAG=0
1830 IF NEW.FLAG THEN CLOSE #2:KILL FILENAME$:RETURN

```

```

1840 INPUT #2,UR.X,UR.Y
1850 INPUT #2,UL.X,UL.Y
1860 INPUT #2,LL.X,LL.Y
1870 INPUT #2,LR.X,LR.Y
1880 INPUT #2,NCONTEUR
1890 FOR I=1 TO NCONTEUR
1900 INPUT #2,H(I),HN(I),CONJ(I),N$(I)
1910 NEXT I
1911 INPUT #2,TEXT$
1912 IF TEXT$<>"COORDS" THEN 1920
1913 INPUT #2,FUR.X,FUR.Y
1914 INPUT #2,FUL.X,FUL.Y
1915 INPUT #2,FLL.X,FLL.Y
1916 INPUT #2,FLR.X,FLR.Y
1920 CLOSE #2
1930 RETURN
1940 '
1950 *RECOVERER
1960 GOSUB*SHOW.THE.FRAME
1970 FOR I=1 TO NCONTEUR
1980 CONTEUR=H(I):HN=HN(I):GOSUB*MAKE.CONTNAME
1990 OPEN CONTNAME$ FOR INPUT AS #2
2000 INPUT #2,N,HN
2010 INPUT #2,XSAVE,YSAVE
2020 FOR I1=1 TO N-1
2030 INPUT #2,X,Y
2040 C=2 'C=1+(CONTEUR/2 MOD 5)
2050 LINE(FNX(XSAVE),FNY(YSAVE))-(FNX(X),FNY(Y)),C
2060 XSAVE=X:YSAVE=Y
2070 NEXT I1
2080 CLOSE #2
2090 NEXT I
2091 ' INPUT #2,TEXT$
2092 ' IF TEXT$<>"COORDS" THEN 2100
2093 ' INPUT #2,FUR.X,FUR.Y
2094 ' INPUT #2,FUL.X,FUL.Y
2095 ' INPUT #2,FLL.X,FLL.Y
2096 ' INPUT #2,FLR.X,FLR.Y
2097 ' CLOSE #2
2100 RETURN
2120 OPEN FILENAME$+"BAK" AS #2 : CLOSE #2
2130 OPEN FILENAME$ AS #2 : CLOSE #2
2140 KILL FILENAME$+"BAK
2150 NAME FILENAME$ AS FILENAME$+"BAK2160 OPEN
FILENAME$ FOR OUTPUT AS #2
2170 WRITE #2,UR.X,UR.Y 2180 WRITE #2,UL.X,UL.Y
2190 WRITE #2,LL.X,LL.Y
2200 WRITE #2,LR.X,LR.Y
2210 PRINT #2,NCONTEUR

```

```

2110 *FINE
2220 FOR I=1 TO NCONTEUR
2230 WRITE #2,H(I),HN(I),CONJ(I),N$(I)
2240 PRINT N, H(I),HN(I),CONJ(I),N$(I)
2250 NEXT I
2251 PRINT #2,"COORDS"
2252 WRITE #2,FUR.X,FUR.Y
2253 WRITE #2,FUL.X,FUL.Y
2254 WRITE #2,FLL.X,FLL.Y
2255 WRITE #2,FLR.X,FLR.Y
2260 CLOSE #2
2270 RETURN
10000 'データの形式について
10001 '(1)インデックスファイル (ファイル名:"NNNN." 但し NNNN は地図番号 4 桁)
10002 ' UR.X,UR.Y
10003 ' UL.X,UL.Y
10004 ' LL.X,LL.Y
10005 ' LR.X,LR.Y
10006 ' NCONTEUR(コンタ数)
10007 ' H(1),HN(1),CONJ(1)
10008 ' . . .
10009 ' H(N),HN(N),CONJ(N)
10010 '
10011 '
10012 '(2)個別コンタラインファイル (ファイル名前:"NNNNKKKK.JJJ"
10013 ' 但し、KKKK は標高 4 桁、JJJ は同標高枝番号)
10014 ' N(点数),HN(同標高枝番号)10015 ' PX1,PY1
10016 ' PX2,PY2
10017 ' . . .
10018 ' PXN,PYN これは、終端条件を示すフラグとなっている。
10019 ' PXN: 0 の場合: 閉じている。
10020 ' PXN,PYN: 1 左辺 2 右辺 3 上辺 4 下辺と交差
10021 '

```

Conteur.bas は、一回のセッション（起動から終了まで）の間に、一つのブロック（複数の敷地から成る）を処理します。起動直後にブロック番号を尋ねてくるので入力します。既存のファイルがあれば、それを読み込み、続けて新たな敷地の入力を求めてきます。指定されたブロック番号に対応するファイルが存在しなければ、新たなブロックのデータを作成します。

一つのブロックに関して、敷地の数だけのファイルと、そのブロックの敷地全体の属性及び敷地リストを含むマスターファイルが作成されます。これは、以下の形式を有しています。

```

ブロックマスターファイル 「0001」
2388.5,2057.5 (四隅のドット座標値)
1562,2044
1594,375.667
2420.67,391.667
6 敷地総数
65.68,1,0,"f1301" 標高,通番,種別,名称
65.44,2,0,"f1302t"
65.76,3,0,"f1303t"
65.74,4,0,"f1304t"
65.65,5,0,"f1305c"
65.71,6,0,"f1306"
COORDS

```

```
56800,195000
56700,195000
56700,194800
56800,194800
```

ここに列挙された敷地数だけ、敷地形状を記述したファイルが作成されます。

個々の敷地形状を記述するファイル

例：「00010065.001」

```
5,1          (頂点数、開始条件)
2127,1545.17 (各頂点のX, Y座標)
2168.83,1641
2073,1645.33
2065.5,1548.5
2127.33,1546
0,0          (EOD、終了条件)
```

個々の敷地形状ファイルの名称は、ブロック番号（4桁）＋標高（4桁）．敷地通し番号（拡張子3桁）です。開始条件、終了条件は、敷地境界線が、その図幅からはみ出て、隣の地図に繋がる場合などに対応するものです。

これを用いて、ブロック単位のデータを作成したあと、filcon1.exe という小さなプログラムにより、地割記述ファイル(*.LDT)を合成します。このためには、メモ帳等のエディタを用いて、o.c という名前のテキストファイルをコントロール・ファイルとして作成し、filcon1.exe と同じディレクトリに格納した上で、dosコマンドとして filcon1.exe を実行します。filcon1.exeは、以前のバージョンで、市街地生成に用いるための地割入力・編集に用いていた filcon.exe を改良したものです。Fullsetでインストールした場合、

¥keikan¥maju¥filcon1

のディレクトリにあります。追加機能等は、以下の通りです。地区全体が複数のブロックから構成されている場合であっても、地区全体の敷地割の状態を示す.ldt ファイルを生成することができます。個々のブロックの入力錯誤訂正や、現況を計画案に差替える場合などは、対応するデータ及び o.c を書換えて、filcon.exe を再実行します。

Filcon.exe の使用方法の拡張について(970625 福島市役所対応)

contour.basによりデジタル入力したファイルを filcon.exe により、地割ファイルに変換する場合、

●従前(ver.1.0)

o.c に、地区の名称を指定する

例：c:¥jiwari¥0001

●変更後(ver.2.0)

[追加した機能]

- ・従来の機能はそのまま維持する
- ・複数のファイルを処理し、一つの地割ファイルに集約する
- ・不整形敷地・隅切のある敷地等のデータを正しく処理する。
- ・敷地条件のデータを、地番情報の中に埋め込む
- ・敷地データを、地番情報に基づいて管理する

[修正ヶ所]

- ・デジタル入力ファイル名を指定する o.c ファイルの仕様を追加する

[使用上での変更点]

o.c ファイルの1行目に、version2 の1行を追加する。

o.c ファイルの2行目に、出力ファイル名を記述する。

3行目以降に、デジタル入力ファイル名を、1行ずつ記述する。

例：

version2
c:\ykeikan\都市開発\仲間町.ldt
c:\ykeikan\データ\0001
c:\ykeikan\データ\0002
c:\ykeikan\データ\0003
c:\ykeikan\データ\0004
c:\ykeikan\データ\0005
c:\ykeikan\データ\0006
c:\ykeikan\データ\0007

970626 区画整理事務所にて追記

- ・本日までの開発内容の使用上の補足説明
- ー市街地生成を、MAJU.EXEにて実施する。
- ー景観シミュレータの側では、5月20日以降のバージョンを使用する。
SIM.EXEとMAJU.EXEが協調して動作するための条件：
それぞれの実行形式(.EXE)が格納されているディレクトリに、
TIMERPATH.TXT
というデータファイルがあり、その最初の2行に、
MAJU.FLG
MAJU.DAT
の情報交換用のファイルの所在が記述され、それが一致していること。
なお、二つの.EXEが同一ディレクトリに存在する場合には、共通1つで良い。
- ーMAJU.EXEの使用方法概略
領域条件設定：敷割条件の入力で、.LDTファイルを指定する。
領域条件設定：建物初期条件で、.ADAファイルを指定する。
敷地を確認するだけの場合、建物初期条件は不要。
- ーユーティリティで、敷地の表示を行うことができる。
- ー建物初期条件を設定する、.ADAファイルの形式概要：
地番 建築年次 床面積
建築タイプ SPD.ADA

補足情報

- 以上の3行単位の繰り返しにより、地番で指定した場所の建物現況を指定する。
建築年次は、西暦年で示す。
床面積は平米数
建築タイプは、目下以下の通り
- 1 なし
 - 0 建築をファイルで指定する(CADで入力した場合等)
その場合、ファイル名称(**.GEO)を続けて記述する。 1 木造平入平屋
 - 2 木造平入2階建
 - 3 店舗併用住宅
 - 4 容積率目一杯ビル
 - 5 寄せ棟平屋(函館)
 - 6 寄せ棟2階家(同上)
- なお、建築タイプは、福島市の実状に合わせて増補する予定。
角地については、タイプ指定は同じとし、敷地の特性を見て区別して形状生成する。

b. 建築物初期条件

建築物の初期条件は、建築物定義ファイル(*.ada)によって記述されます。

【建築物定義ファイルのフォーマット】

2行が個々の建築物を示す1レコードになっています。1レコードは次の構成になっています。

“敷地名称” 建築年次 面積 (改行)

類型コード ファイル名 (改行)

敷地名称は、建物と敷地を関連づけるために用いられます。定義されていない敷地名称が指示される

と、エラーを生じます。

建築年次は、西暦年号（4桁）で示します。

面積は実数値で示します。

類型コードは、建築物の類型を示すものです。概ね、固定資産登録上の類型に従っています。以下のコードを定義しています。

- 1 : 空地
- 0 : 特殊形状の建築物をファイル指定
- 1 : 駐車場
- 2 : 専用住宅平屋
- 3 : 専用住宅二階建
- 4 : 専用住宅寄棟平屋
- 5 : 専用住宅寄棟二階建
- 6 : 店舗
- 7 : ビル

ファイル名称は、特殊な建物等で、具体的な形状を記述するファイル(1ss-g形式)が別途用意されている場合に用います。それ以外の場合は、類型に応じて形状が自動的に生成されるため、このファイル名称は無視されますので、適当な文字列を入力します。

例：建築物定義ファイル

```
"d4001x" 1932 30.000000
-1 spd.ada
no info
"f1301" 1906 30.000000
3 spd.ada
no info
"f1302t" 1936 30.000000
0 spd.ada
no info
"f1303t" 1908 30.000000
```

```
0 spd.ada c
no info
"f1304t" 1976 30.000000
0 spd.ada
no info
"f1305c" 1957 30.000000
0 spd.ada
no info
"f1306" 1927 30.000000
3 spd.ada
no info
```

c. 人口初期状態

人口の初期状態は、住民定義ファイル(*.man)により記述します。

【住民定義ファイルのフォーマット】

“敷地名称” 性別(M/F) 年齢

d. 都市施設配置

都市施設としては、当面、交差点リストデータ(*.crs)を使用しています。これは、[ユーティリティ][交差点の編集]により自動的に生成・編集できます。

【交差点リストデータのフォーマット】

ncross=交差点の数

交差点名称

セグメント数

セグメント・データ（3点の座標値と6のパラメータから成る）を、セグメント数だけ繰り返す。

（次の）交差点名称

（以下交差点の数だけくり返し）

例：交差点リスト

```
ncross=5;
交差点 1
4
  0.000    0.000    0.002
  0.000   20.000    0.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000   20.000    0.000
 20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000  -20.000    0.000
 20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000  -20.000    0.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
十字路（平面交差）
4
  0.000    0.000    0.002
  0.000   20.000    0.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000   20.000    0.000
 20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000  -20.000    0.000
 20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000  -20.000    0.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
```

```
三叉路
3
  50.000    50.000    0.002
  50.000    80.000    0.000
  30.000    20.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  50.000    50.000    0.002
  50.000    80.000    0.000
  70.000    20.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  50.000    50.000    0.002
  30.000    20.000    0.000
  70.000    20.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000   30.000    2.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
  0.000    0.000    0.002
  0.000  -30.000   -2.000
 -20.000    0.000    0.002
2.000 10.000  9.000  1.000  0.900  0.150
一一町・一一通り交差点
3
  0.000    0.000    0.002
 -10.000    0.000    0.000
  0.000   10.000    0.002
2.000 10.000  6.000  1.000  0.500  0.150
  0.000    0.010    0.002
 -10.000    0.000    0.000
 10.000    0.000    0.002
2.000 10.000 10.000  1.000  1.000  0.150
  0.000    0.000    0.002
 10.000    0.000    0.000
  0.000   10.000    0.002
2.000 10.000  6.000  1.000  0.500  0.150
```

e. 商業業務分布

敷地単位で定義します。現在の所、まだ実装していません。

f. 周辺地形・景観

これは、LSS-Gデータとして提供されます。

g. 表土条件（未実装）

植栽の成長等に影響する表土を、計画的に移動した土地区画整理の例があります。

h. 地盤条件（未実装）

地盤条件により、高層建物が建ちにくい、という影響がある場合があることが知られています。

② 形成条件

a. 建ぺい率

敷地面積に対する、建坪面積の上限値を制限するものです。

b. 容積率

敷地面積に対する建物の延べ床面積の割合の上限値を定めます。ビルなどの場合には、階高を4mと近似し、ピロティや吹き抜けはないものとして、容積率を計算しています。

c. 高さ制限

ビルの最大高さを規定しています。

d. 道路斜線制限

当面、敷地前面における立ち上り高さ、勾配で定義するようになっています。

e. 壁面後退

敷地前面からの後退距離をm単位で指定します。

（5）景観シミュレータへのデータ出力

景観シミュレータと連携して、ステップ（1年単位）毎に、その時点での市街地の状況を3次元的に表示するために、景観シミュレータの側に、ネットワーク経由で外部からコマンドやデータを受けて、それに対応した動作をするようなインターフェースを増設しました。これを前提として、成熟都市シミュレータを開発し、年次毎に景観シミュレータにシミュレーションの結果のうち、市街地の3次元形状に関する部分を、データとして送出し、表示を行うようにしました。更に、データの授受の効率を高めるために、一定の意味あるデータのまとまりを、ファイルとして授受できるようなバッファリングの機能を作成してあります。



【図VI-1-1】現況の表示

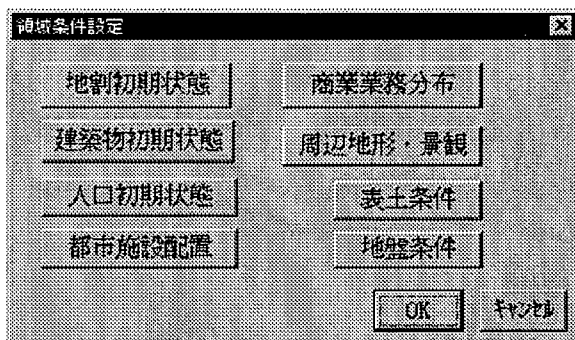
(6) シミュレーションの条件設定

シミュレーションの過程では、まず初期条件としての市街地の現況（多くの場合は、現時点）を設定します。次にシミュレーションの条件（様々の都市計画条件など）を設定し、シミュレーションを行います。シミュレーションを途中で中断し、形成条件を変更して、更にシミュレーションを続行するような操作も可能としています。

【操作方法】

① 初期条件の入力

初期条件は、メイン画面の[領域条件]ボタンによりダイアログを起動します。



【図VI-1-2】初期条件—領域条件の設定

[地割初期条件]、[建物初期条件]等のボタンにより、ファイル選択メニューを起動し、前述のように用意してある定義ファイルを指定します。

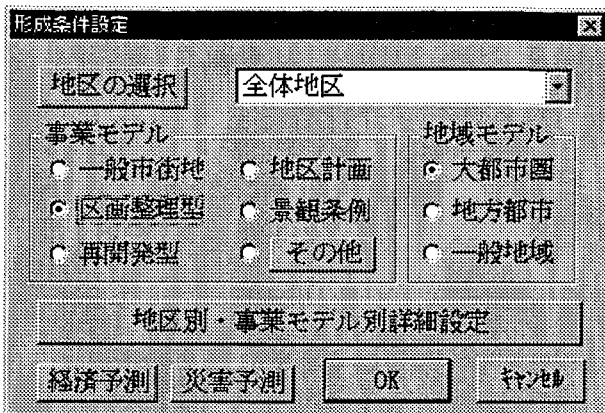
② 形成条件の設定

形成条件は、メイン画面の[形成条件]ボタンを押すことにより、ダイアログを起動します。

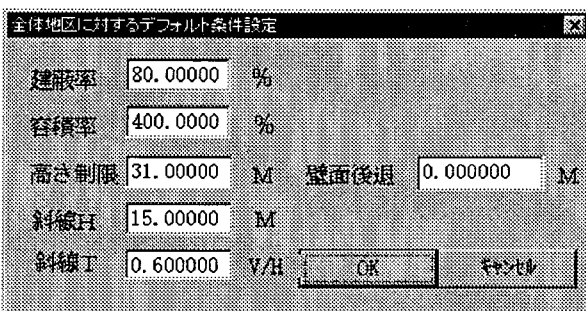
地区の選択は、領域条件の設定の中で地域地区、事業対象区域などが定義されている場合に、ここで設定しようとする地区を選択し、その上で、細かな条件を設定します。

具体的な条件の設定ダイアログは、[地区別・事業モデル別詳細設定]のボタンにより起動します。

下の例では、事業モデルを区画整理型とし、設定対象を全体地区として、詳細条件を設定します。



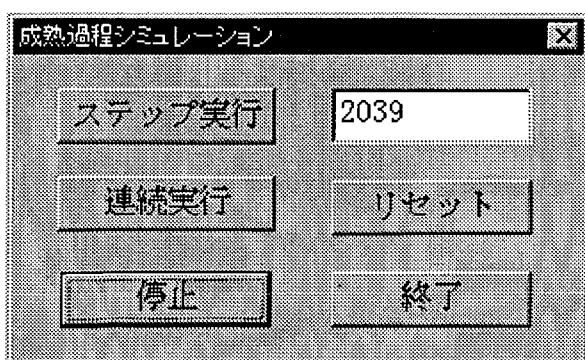
[図VI-1-3] 初期条件—形成条件の設定



[図VI-1-4] 都市計画上のパラメータの設定

③ シミュレーションの実行

シミュレーションの実行ダイアログは、メインの[シミュレーション]ボタンにより起動します。



[図VI-1-5] シミュレーション実行の制御

まず、[リセット]ボタンにより、現況の地割と建物を表示します。

ステップ実行により、1年間だけのシミュレーションを行います。前述のように確率計算により、古い建物を取除くと共に、空地に新しい建物を発生します。その際に形成条件によって指定された都市計画条件等に対応します。

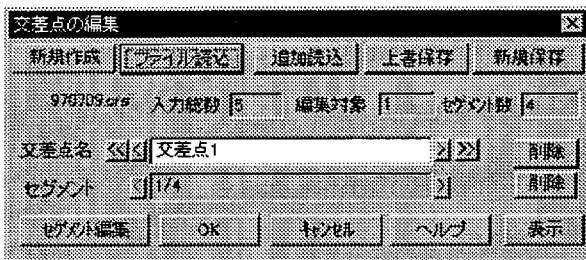
連続実行により、ユーザーが[停止]をかけるまで継続的にシミュレーションを行います。



【図VI-1-6】シミュレーションの結果出力例

停止をかけた状態で、[終了]により一度メイン画面に抜け、[形成条件設定]により都市計画条件等を変更した上で、再びシミュレーションを行うと、停止した時点から先のシミュレーションを行うことができます。これにより、当面の都市計画条件と、将来の都市計画条件が変化する場合予測などを行うことができます。

(7) 応用機能：道路構造令に基づく交差点形状の生成



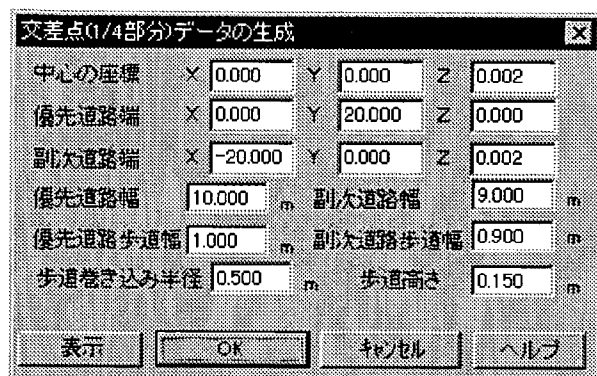
【図VI-1-7】道路構造令による交差点形状生成

前述の、景観シミュレータをネットワーク越しに外部から操作するインターフェースは、成熟都市シミュレータ以外に様々な応用システム作成の途を開くものであり、様々な工夫が可能となります。その一例として、成熟都市シミュレータを操作する上でも関係の深い、交差点の3次元形状を作成するツールを、成熟都市シミュレータの附属機能として作成してみました。これは、検討対象地区全体の交差点のリストを、座標値と様々なパラメータ（歩道幅・高さ、道路の勾配、歩道巻き込み半径など）のセットとして表現し、これに基づき交差点の3次元形状を生成し、景観シミュレータに送出して表示する機能です。これにより、従来手間のかかった交差点の形状生成が簡単にできるようになります。

一つの交差点は、そこに接続する道路の数と同数のセグメントにより記述します。一つのセグメントは、隣接して接続する道路の半分を合せた形状で、建物からなる街区に沿って折れ曲る歩道を含んだ部分です。このように定義することにより、道の両側で車道や歩道の幅が異なる交差点も表現が可能

となります。編集[表示]ボタンにより景観シミュレータに3次元形状を出力して確認することができます。

作成したデータは、ファイル保存し、都市施設配置により読込んで用います。



【図VI-1-8】交差点セグメントのパラメータ指定

(8) 応用機能：マルチ・スクリーンの評価セッションの実行

景観シミュレータの外部インターフェースを応用したもう一つの例として、評価セッション用に、予め用意してあるシーンファイルを用いて、所定の時刻を予めプログラムしておき、これに従ってシーンを自動的に切り替える機能(シーケンサ機能)です。この場合、コントロールを受ける景観シミュレータ(プロセス)を最大5までセットすることができます。即ち、液晶プロジェクタなどの投影画面中にマルチ・スクリーンで、複数の景観シミュレータを開いて並べ、相互に関係のある表示内容を協調的に切り替えるような使用方法を可能としています。

【操作方法】

①マルチスクリーンの使用時における環境設定

マルチスクリーンを使用する際には、maju.exeと同じディレクトリに置かれる環境設定ファイル

Timerpath.txt

の中に、複数のデータ・ファイル及びフラッグ・ファイルの名称を並べます。

例：Timerpath.txt の内容

```
C:¥maju1.flg
C:¥maju1.dat
C:¥maju2.flg
C:¥maju2.dat
C:¥maju3.flg
C:¥maju3.dat
C:¥maju4.flg
C:¥maju4.dat
C:¥maju5.flg
C:¥maju5.dat
```

一方、景観シミュレータの側では、例えば、ksim¥binの下に、必要なスクリーンの数だけ、サブ

ディレクトリを作成します。

例：

ksim¥bin¥session1

ksim¥bin¥session2

ksim¥bin¥session3

ksim¥bin¥session4

ksim¥bin¥session5

そして、それぞれの中に sim.exe を一つずつコピーすると共に、それぞれの sim.exe のための環境設定ファイル

timerpath.txt

を作成、それぞれの中で、ユニークなコントロール・ファイル名称を指定します。

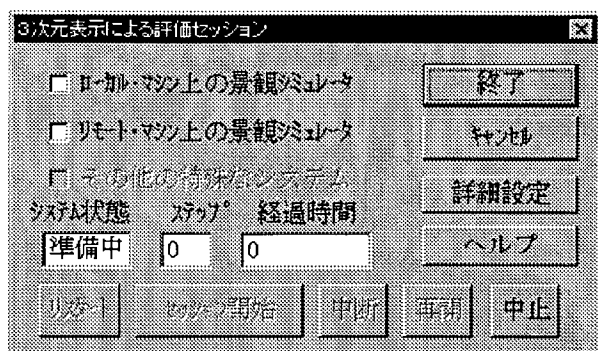
例：session3 の timerpath.txt

C:¥maju3.flg

C:¥maju3.dat

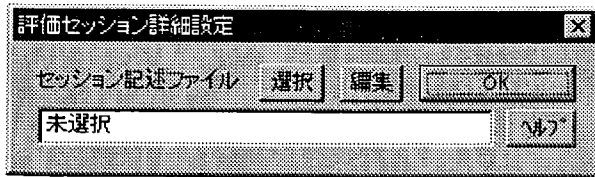
② プログラム方法

メイン画面の[評価セッション]ボタンを押します。すると、3次元表示による評価セッションのダイアログが表示されます。



【図VI-1-9】評価セッションの実行

そこで、[詳細設定]のボタンを押すと、評価セッション詳細設定のダイアログが表示されます。ここで、既に作成してあるシーケンス記述ファイル(*.sdc)を再編集する場合には、選択ボタンを押して、そのファイルを選択します。新たなファイルを作成する場合には、そのファイル名称を直接入力します。



【図VI-1-10】評価セッションの条件編集

次に、[編集]ボタンを押すと、メモ帳が開いて、そのシーケンス記述ファイルを編集する状態になります。新たなファイル名称を指定した場合には、「ファイルXXXX.sdcが見つかりません。新しく作成しますか」と聞いてくるので、[はい]と答えます。

シーケンス記述ファイルは、以下の文法に従って記述します。

- ・ 1行の構成：

開始時刻 コード 対象番号 コマンド

開始時刻は、「時：分‘秒」の形式で記述します。セッション開始からの経過時間で表現します。

コードは、sとbがあります。

sは、景観シミュレータの制御に関する記述であることを示します。

bは、メモ帳でテキスト・ファイルを開くことを示します。

hは、セッションの終了を示します。

対象番号は、マルチ・スクリーンの場合のどのスクリーンを切替えるかを指定するID番号です（1～5）。但し、bの場合には、メモ帳による表示の中止時点を表現します。

コマンドは、sの場合には、景観シミュレータのシーンの切替えに関するコマンドを記述します。bの場合には、表示するテキストのファイル名称を記述します。

景観シミュレータのシーン切替えに関しては、以下のコマンドがあります。

SCENE(N); N番目のシーンを表示する。

SCENE_KEMUDIAN(); 次のシーンを表示。

SCENE_YANGTADI(); 前のシーンを表示。

例：評価セッションのプログラム

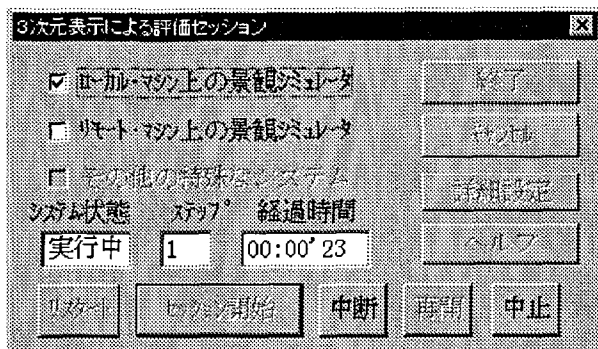
00:00' 03 s	1	SCENE(0);
00:00' 05 s	2	SCENE(0);
00:00' 06 s	3	SCENE(0);
00:00' 07 s	4	SCENE(0);
00:00' 09 s	5	SCENE(0);
00:00' 11 s	4	SCENE(6);
00:00' 13 s	1	SCENE(1);
00:00' 15 s	1	SCENE(0);
00:00' 17 s	3	SCENE(0);
00:00' 19 s	3	SCENE(2);
00:00' 21 s	3	SCENE(4);
00:00' 23 s	3	SCENE(6);
00:00' 32 s	4	SCENE(9);
00:00' 35 s	4	SCENE(11);
00:00' 43 s	4	SCENE(13);
00:00' 51 s	2	SCENE(0);
00:00' 53 s	2	SCENE_KEMUDIAN();
00:00' 55 s	2	SCENE_KEMUDIAN();
00:00' 57 s	2	SCENE_KEMUDIAN();
00:01' 03 s	2	SCENE_KEMUDIAN();
00:01' 05 s	2	SCENE_KEMUDIAN();
00:01' 12 s	5	SCENE_KEMUDIAN();
00:01' 14 s	5	SCENE_KEMUDIAN();
00:01' 16 s	5	SCENE_KEMUDIAN();
00:01' 18 s	1	SCENE(0);
00:01' 19 s	2	SCENE(0);
00:01' 21 s	4	SCENE(0);
00:01' 25 s	1	SCENE_KEMUDIAN();
00:01' 30 s	1	SCENE_KEMUDIAN();
00:01' 40 s	1	SCENE_KEMUDIAN();
00:01' 50 s	2	SCENE(0);
00:01' 55 s	2	SCENE_KEMUDIAN();
00:02' 00 s	2	SCENE_KEMUDIAN();
00:02' 05 s	2	SCENE_KEMUDIAN();
00:02' 10 s	2	SCENE_KEMUDIAN();
00:02' 15 s	2	SCENE_KEMUDIAN();
00:02' 20 s	2	SCENE_KEMUDIAN();
00:02' 25 s	2	SCENE_KEMUDIAN();
00:02' 35 s	2	SCENE_KEMUDIAN();
00:02' 40 s	2	SCENE_KEMUDIAN();
00:02' 50 s	2	SCENE_KEMUDIAN();
00:02' 55 s	2	SCENE_KEMUDIAN();
00:03' 00 s	2	SCENE_KEMUDIAN();
00:03' 05 s	2	SCENE_KEMUDIAN();

00:03' 10 s	2	SCENE_KEMUDIAN();
00:03' 15 s	2	SCENE_KEMUDIAN();
00:03' 20 s	3	SCENE(0);
00:03' 25 s	2	SCENE(9);
00:03' 30 s	3	SCENE_KEMUDIAN();
00:03' 40 s	3	SCENE_KEMUDIAN();
00:03' 45 s	3	SCENE_KEMUDIAN();
00:03' 50 s	3	SCENE_KEMUDIAN();
00:03' 55 s	3	SCENE_KEMUDIAN();
00:04' 00 s	3	SCENE_KEMUDIAN();
00:04' 10 s	3	SCENE(5);
00:04' 20 s	4	SCENE(0);
00:04' 33 s	4	SCENE_KEMUDIAN();
00:04' 42 s	4	SCENE_KEMUDIAN();
00:04' 51 s	4	SCENE_KEMUDIAN();
00:05' 00 s	4	SCENE_KEMUDIAN();
00:05' 10 s	4	SCENE_KEMUDIAN();
00:05' 20 s	4	SCENE_KEMUDIAN();
00:05' 30 s	4	SCENE_KEMUDIAN();
00:05' 40 s	4	SCENE_KEMUDIAN();
00:05' 45 s	4	SCENE_KEMUDIAN();
00:05' 50 s	4	SCENE_KEMUDIAN();
00:05' 55 s	4	SCENE_KEMUDIAN();
00:06' 00 s	4	SCENE_KEMUDIAN();
00:06' 05 s	4	SCENE_KEMUDIAN();
00:06' 10 s	4	SCENE_KEMUDIAN();
00:06' 15 s	4	SCENE(14);
00:06' 28 s	5	SCENE(0);
00:06' 35 s	5	SCENE_KEMUDIAN();
00:06' 40 s	5	SCENE_KEMUDIAN();
00:06' 45 s	5	SCENE_KEMUDIAN();
00:06' 50 s	5	SCENE_KEMUDIAN();
00:07' 00 s	5	SCENE_KEMUDIAN();
00:07' 10 s	5	SCENE_KEMUDIAN();
00:07' 20 s	5	SCENE_KEMUDIAN();
00:07' 30 s	5	SCENE_KEMUDIAN();
00:00' 01 b	00:00' 30	@restop.txt
00:00' 31 b	00:01' 00	@resrch1.txt
00:01' 01 b	00:01' 30	@resrch2.txt
00:01' 31 b	00:04' 30	@resrch3.txt
00:04' 31 b	00:06' 30	@resrch4.txt
00:06' 31 b	00:09' 00	@resrch5.txt
00:08' 00 b	00:08' 01	@warn1.txt
00:09' 00 h	0	-

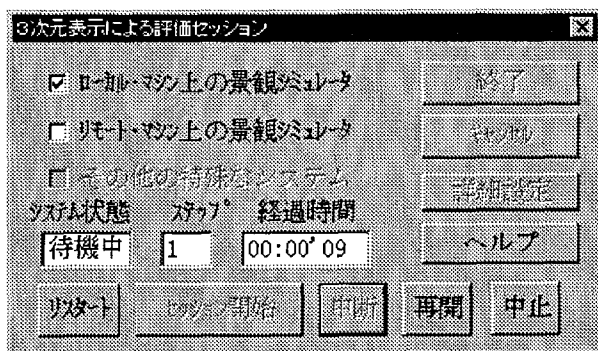
編集が終わったら、[OK]ボタンにより、[3次元表示による評価セッション]のメニューに戻ります。

③ 評価セッションの実行

[3次元表示による評価セッション]のメニューで、[セッション開始]ボタンを押すことにより開始します。[中断]は、内部のクロックを停止させます。[再開]で再開します。[リスタート]で、内部のクロックをゼロに戻します。



【図VI-1-1 1】実行中の表示



【図VI-1-1 2】中断中の表示

2. リアル・モデラー

(1) 概要

平成9～10年度に、建築研究所では、官民共同研究「地域の3次元将来像の予測システムの開発」を一般公募しました。その目的は、完成しつつあった景観シミュレータをベースとして、実際の事業に応用する上での実用性を高めるために、各種の民間技術・製品等とのインターフェースを図り、連携して使用できるようにすることでした。

公募の結果、下記の2社が応募・採択され、1～2年間にわたりファイル・コンバータの作成および連携機能の強化・改善を行うと共に、実際の再開発や区画整理に実験的に応用することを通じて、データ作成期間の短縮や、シミュレーションの質の向上を達成することができました。

富士通では、1枚の写真のデータから、テキスト付きの3次元モデルを作成する機能を持ったリアル・モデラー(RealModeler)を開発し、平成9年には既に商品として発売していました。

アジア航測では、ステレオ空中写真を自動解析し、地形と市街地建築物の3次元データを取得する技術に基づき、3次元データベースのサービス(作成済み分のライセンス販売、及び新規作成受注)を開始していました。

官民共同研究においては、これらの解析結果を、景観シミュレータの中で配置することのできる部品の形に変換する、ファイル・コンバータを作成しました。このコンバータについては、無償で提供しています。また、景観シミュレーションに用いるのに便利のように、建物の座標軸の設定やスケールの指定を行うことができるように、システム自体も改善されました。

官民共同研究として連携機能を開発することの効用として、景観シミュレータのユーザーの側では、この商品ソフトやサービスと連携して用いることにより、市街地現況の3次元データなどを作成する手間とコストを大幅に軽減することが期待できます。一方、RealModelerあるいは3次元データベースの側では、従来想定されていた応用範囲・ユーザー層を拡大することが期待できます。

共同研究期間中に、これを実際の現場に導入し、現況市街地の建物の3次元データ作成、事業後の市街地将来像を構築するための、テキスト付き現況地形+市街地、住宅や店舗、ビルなどの部品の作成、及び、標準的なテキスト・データの作成などに活用し、成果を上げました。

特に近年、デジタル・カメラが普及してきたため、ノート・パソコンと組み合わせることにより、現場で直ちに現況建物の3次元データを起こすことも可能となってきました。

開発の成果として、ファイル・コンバータであるSKV2LSS.exe(リアル・モデラー)及び3ddb.exe(3次元データベース)が、フリーウェアとして無償で公開されることとなりました。但し、ソースコードは現在のところ非公開です。

改良されたRealModeler自体は、引続き、同社の製品として販売されています。同製品に関する情報は、同社サイト

<http://www.fsel.co.jp/skv>

から公開されています。リアルモデラー自体は、商品ソフトであり、官民共同研究の成果ではないため、操作例について開発元から提供されたマニュアルを(3)として採録し解説することとし、まずファイル・コンバータの操作方法を解説します。

① インストール

本資料付属のCD-ROMにsetupが収録されているので、これにより、画面の指示に従って、skv2lss.exe 適当な場所にインストールして下さい。

② ファイル変換の実際

RealModeler を操作すると、*.skv という拡張子を有するRealModeler独自の形式の3次元形状を記述したファイルが作られます。これはテキスト形式なので、メモ帳等で開けば、どのような形式か、すぐに分ります。

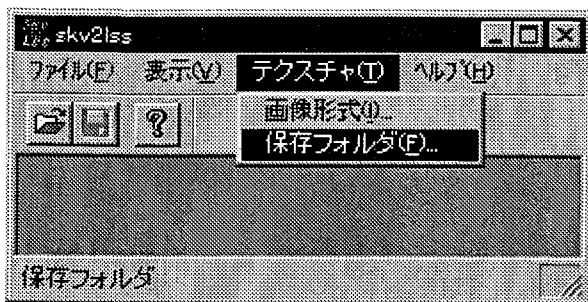
このファイルが記述するモデルは、元の写真から切出した、イメージ・データとリンクする情報を持っています。

RealModelerの一つの特徴として、元のイメージ・データを、モデルを構成する面毎に、小さな長

方形領域に切出してあります。これらのファイルは、.bmp形式で保存されています。

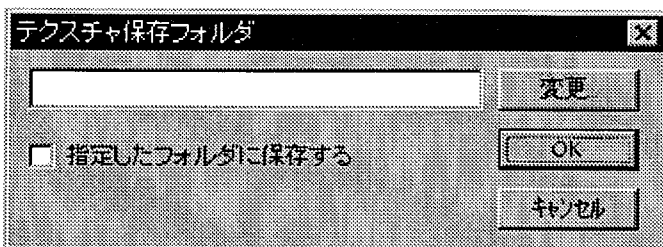
従って、ファイル・コンバータの仕事としては、skv形式のファイルが有する3次元形状（テクスチャ座標を含む）に関する情報を、景観シミュレータが用いるLSS-G形式に変換すると共に、テクスチャ・データを、bmp形式からsgi形式に変換しています。

作業効率を良くするために、変換されたsgi形式を格納するディレクトリ（通常は、景観シミュレータのテクスチャデータを格納するkdb\texture\sgi）を指定します。一度指定すると、再度立上げた時にもこれを記憶しています。従って、skv2lssを最初に起動した時に、これを設定しておくのが便利でしょう。



[図VI-2-1]テクスチャ保存フォルダをまず指定する

メニューの[テクスチャ][保存フォルダ]を選択し、ディレクトリを指定するダイアログを開きます。



[図VI-2-2]テクスチャ保存フォルダの指定

ここで、[指定したフォルダに保存する]をチェックし、変更ボタンを押すと、フォルダ指定ダイアログが開くので、テクスチャのフォルダ（通常kdb/texture/sgi）を指定します。

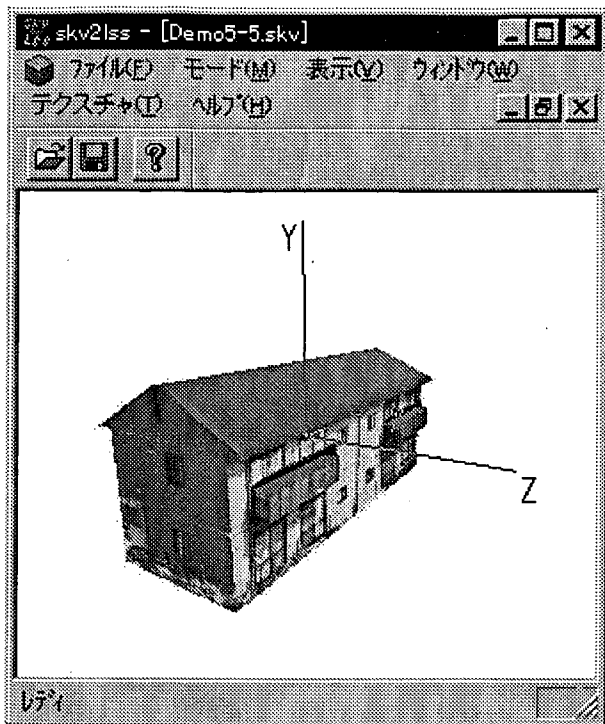


[図VI-2-3]テクスチャ保存フォルダの参照・選択

なお、このダイアログは、Windows95以降、あるいはNT4.0以降の部品を使っているもので、それ以前のバージョンでは使用できません。

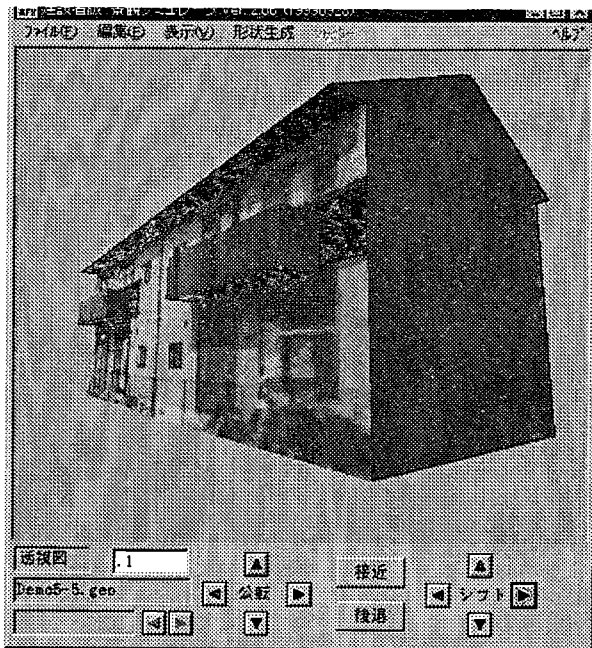
次に、[ファイル][開く]でskv形式のファイルを開くと、画面に3次元表示が現れ、マウสดラッグで回転させて確認することができます。

[ファイル][名前を付けて保存]で、LSS-Gファイルとして保存することができます。



[図VI-2-4]コンバータ skv2lss 中でのモデルの表示・確認

なお、実務マニュアルの出版に当り、富士通社よりリアルモデラー3.0体験版の提供がありましたので、CD-ROMに掲載しています。操作方法は、基本的に以下(3)に掲載するマニュアルと同様です。但し、体験版では、読込んで解析することのできる画像ファイルを制限しています。画像の形式として、.bmp形式で用意されているものが、同RealModelerの練習用として元々用意されていたものです。また、.jpg形式で用意されているもの(ImageSample01-08)は、景観シミュレーションで実際に用いたことのある、市街地の建物等です。



[図VI-2-5] 景観シミュレータでモデルを見る

この他、[モード]メニューで、ワイヤー表示/テクスチャ表示を切換えたり、座標軸の表示のON/OFFを切換えることができます。

保存したファイルを、景観シミュレータで開いて編集したり、配置コマンドで市街地の中に配置したりできます。

(2) 作業手順

① 写真をスキャナーから読み込む

a. 写真について

写真は、焦点が中心にあるものを利用します。写真を撮影する場合は、焦点を中心にするように気をつけてください。(通常、わざと焦点をずらさない限りは中心になります。)

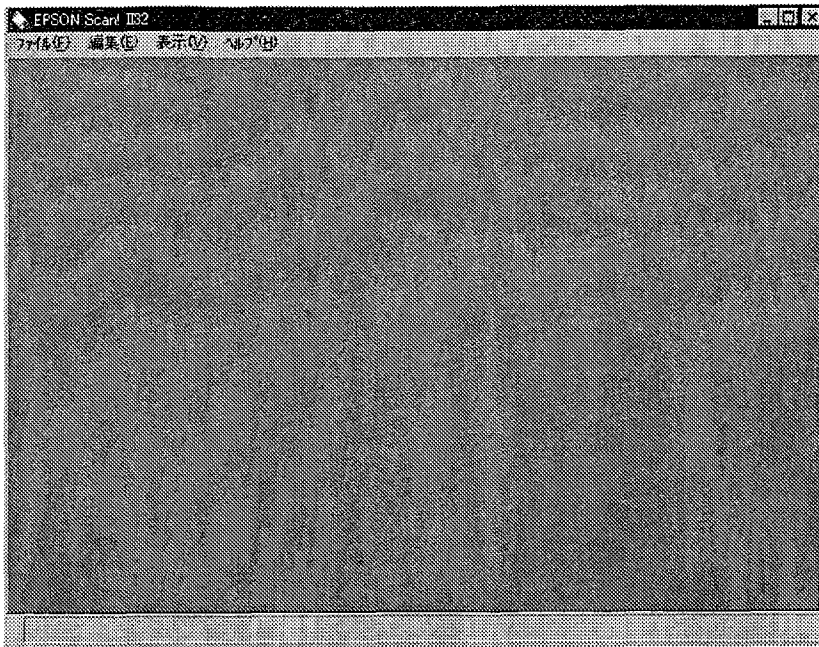
パンフレット、カタログ、雑誌などに掲載されている写真は、焦点が中心にない場合が多く、モデルの作成時に多少ずれる可能性があります。なるべく、自分で撮影した写真をご利用頂くことをお勧めします。

b. プレビュー

背景にしたい写真から、スキャナーを使ってパソコンに画像ファイルとして取り込みます。

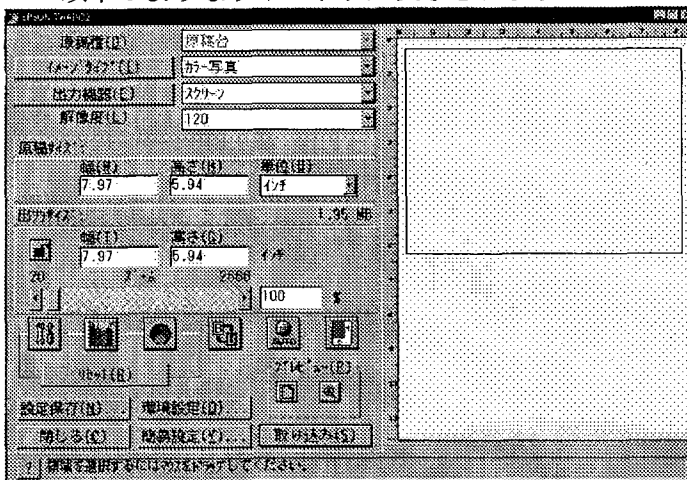
<1> スキャナーに写真をセットし、EPSON Scan! IIプログラムを起動します。

以下のようなウィンドウが現れます。



【図VI-2-6】

- <2> メニューバーの「ファイル」から「イメージ取得」を選択します。
 以下のようなウィンドウが表示されます



【図VI-2-7】

- <3> 「プレビュー」ボタン（右側）をクリックすることにより、スキャナーにセットされた写真が読み込まれ、ウィンドウ右側にプレビュー表示されます。

もし、斜めになっているなどしたら、写真をセットし直して再度行ってください。

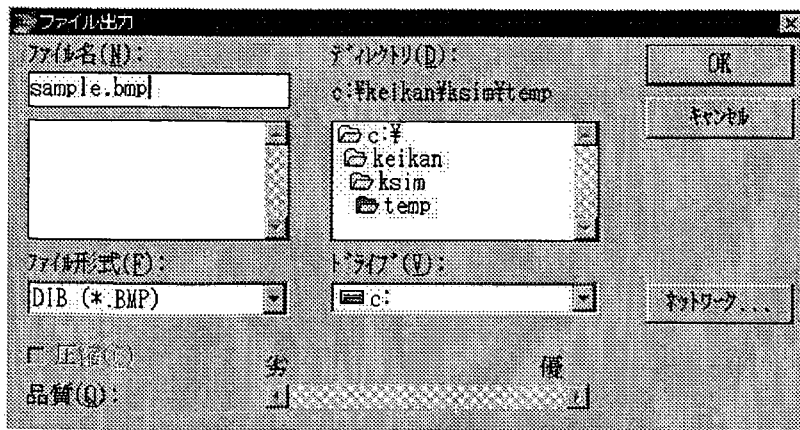
- <4> プレビュー画面上で必要な範囲を矩形で指定します。左上から右下へドラッグして指定します。よければ、「取り込み」ボタンを押してください。取り込みが終わったら、「閉じる」ボタンを押して下さい。

注) 解像度については、通常「72」となっています。ここの値を大きくすると、出力サイズもそれに比例して大きくなるのであまり大きな値を設定しないように注意してください。

c. 画像ファイルの保存

- <1> 取り込んだ画像をファイルに保存します。

メニューバーの「ファイル」の「ファイル出力」を選ぶと、ファイル選択ウィンドウが現れます。



【図VI-2-8】

<2> 任意のフォルダに移動し、ファイル名を付けて保存して下さい。

また、その時のファイル形式は、「DIB(*.BMP)」を選択して下さい。

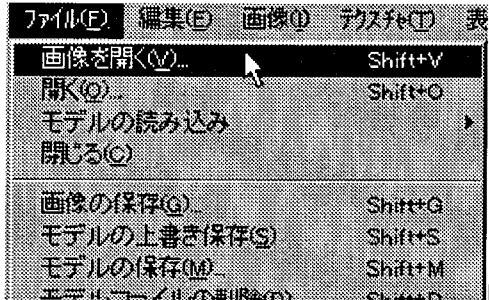
保存するディレクトリはいつも同じところにした方が良いでしょう。

例えば、景観シミュレータがインストールされているディレクトリの下に「keikan\ksim\temp」というディレクトリがあるのでそこを使用するようにします。

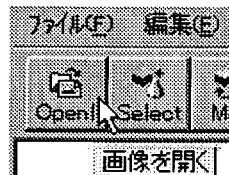
②モデル作成

a. 画像データの選択

<1> 「ファイル」メニューから「画像を開く」コマンドを選択します。



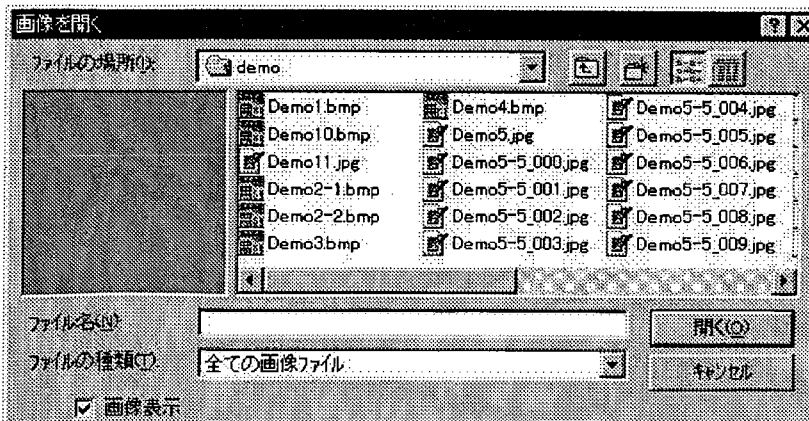
【図VI-2-9】



または、ツールバーのボタンをクリックします。

【図VI-2-10】

<2> 「画像を開く」ダイアログボックスが表示されます。

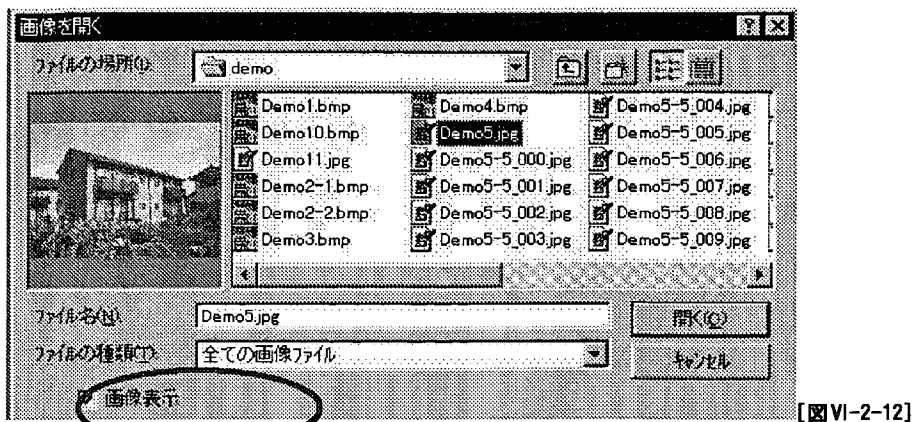


【図VI-2-11】

<3> 今回はDemo5.jpgの画像データを使用し、モデリングします。

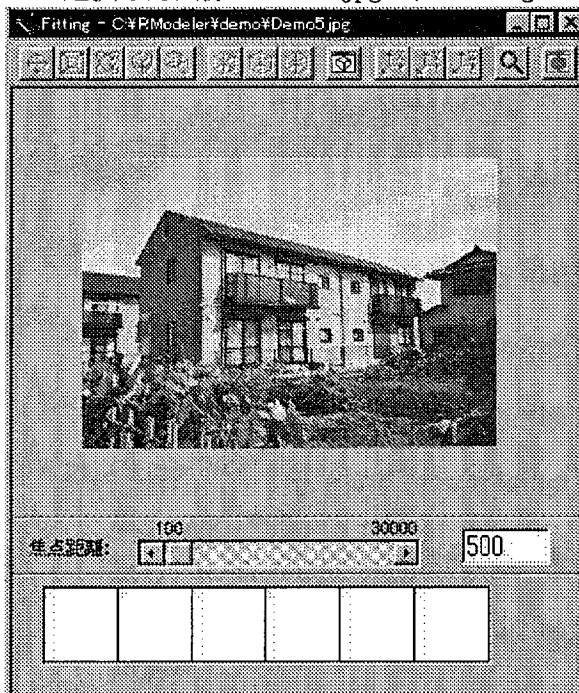
Demo5.jpgを選択します。

画像表示の隣にあるチェックボックスをチェックすると、選択した画像を見ることが出来ます。



<4> Demo5.jpgを選択したら、「開く」ボタンを押します。

<5> 選択した画像「Demo5.jpg」が、Fittingウィンドウに表示されます。



b. 本体の作成

<1> 「画像」メニューから「向き合わせ」を選択します。画像上に3本の線が表示されます。



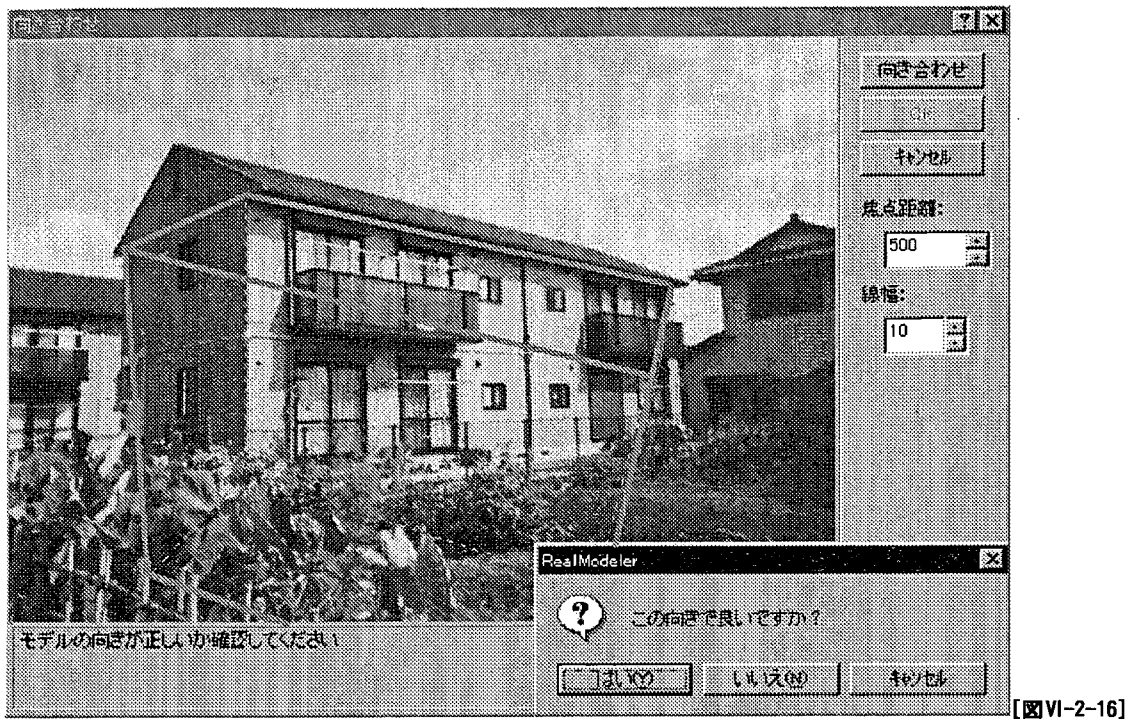
【図VI-2-14】

<2> 軸の先端および中心をマウスで選択してモデルの手前の角に合わせます。合わせる角は、直方体の角でなくてはなりません。軸の中で黄色で表現されている軸は上下方向の軸とします。



【図VI-2-15】

<1> 指定した3軸の向きを選択します。（考えられる2パターンの向きから選択します。下図の場合違うので「いいえ」を選択します。

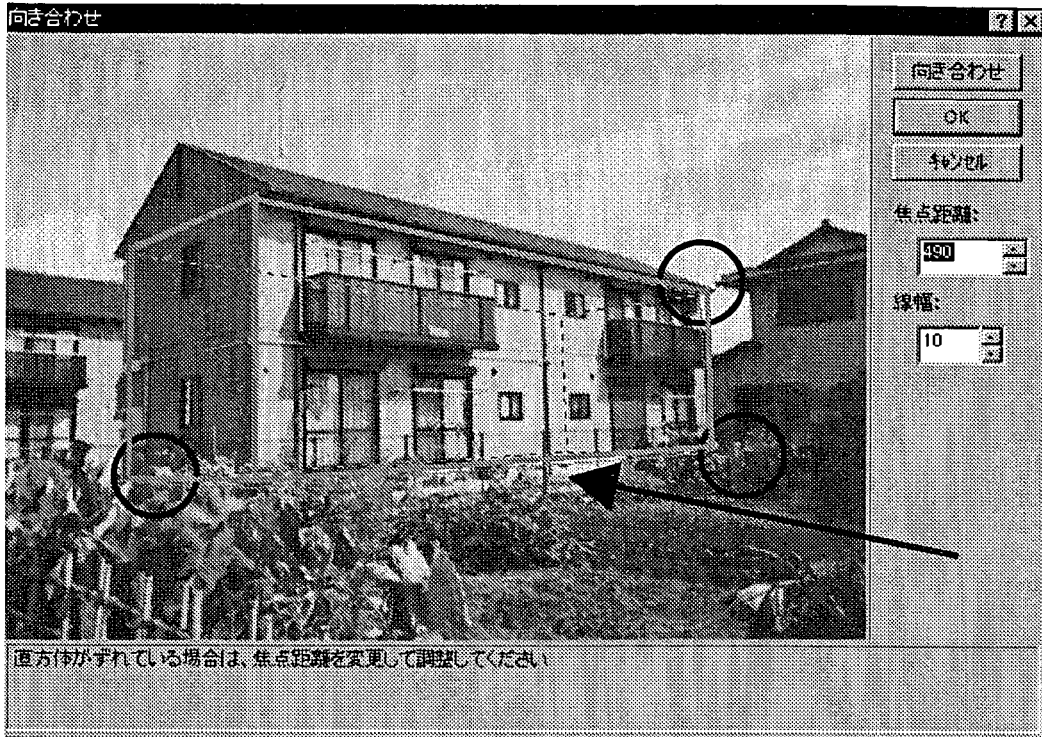


【図VI-2-16】



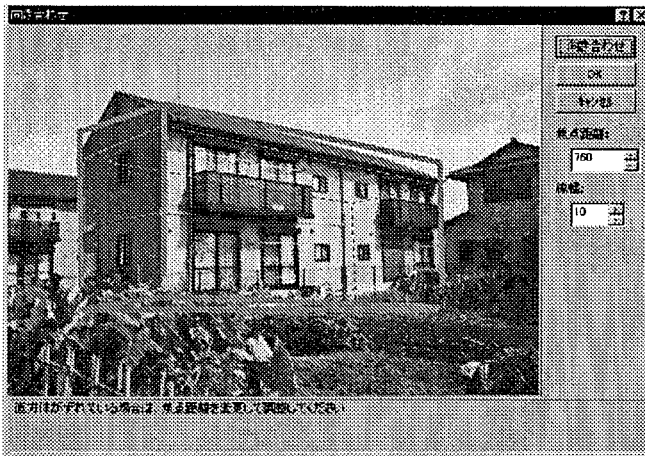
【図VI-2-17】

- <4> 指定した3軸が正しい向きになりました。「はい」を選択して次に進みます。
- <5> 焦点距離を操作して指定した3軸からなる直方体が正しい形になるようところまで変更します。
(焦点距離の値を上下させます。)すべての角がピッタリあってきたところが焦点距離が正しい値と判断します。

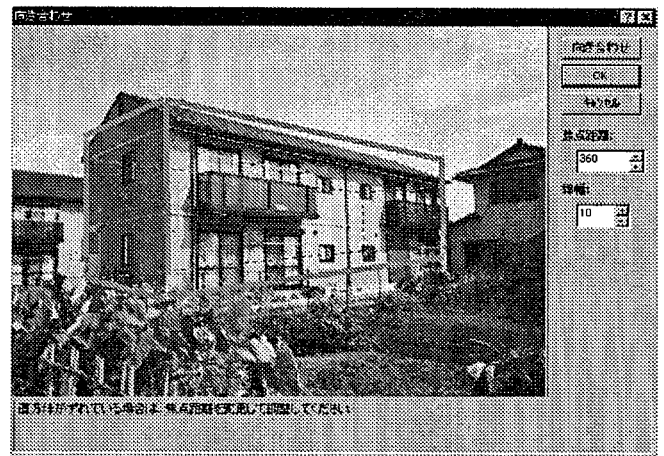


【図VI-2-18】

悪い例

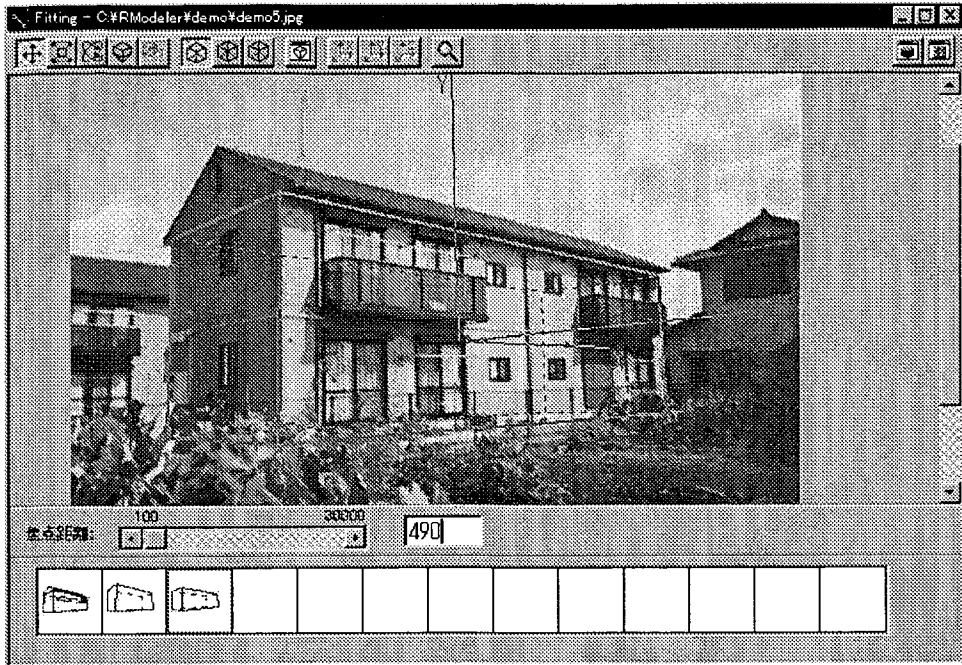


【図VI-2-19】



【図VI-2-20】

<6> 最後に「OK」を選択すると図のように直方体モデルの合わせ込みが完了します。これで、建物の本体部のモデリングができました。

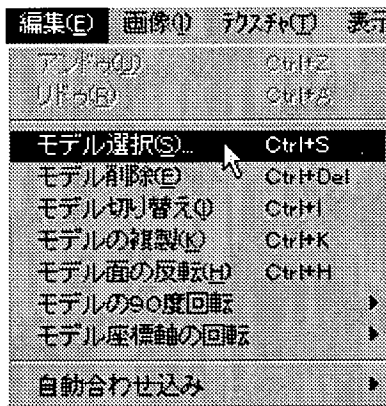


【図VI-2-21】

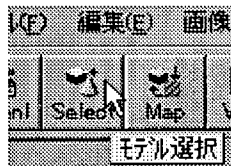
C. ベランダの作成

建物の本体部を作成後、次にベランダを作成します。

<1> 「編集」メニューから「モデル選択」コマンドを選択します。

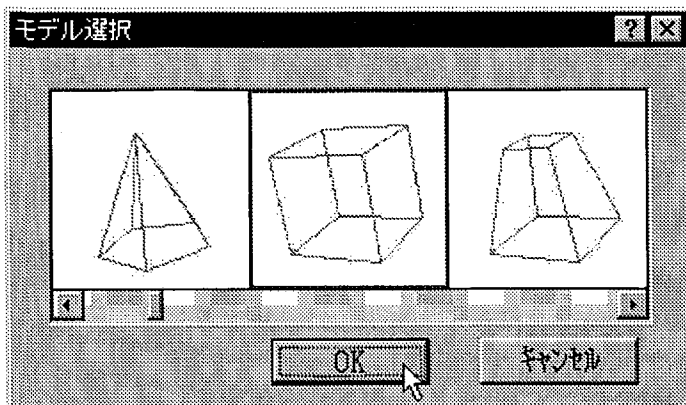


【図VI-2-22】



【図VI-2-23】 または、ツールバーのボタンをクリックします。

<2> 「モデル選択」ダイアログボックスが表示されます。

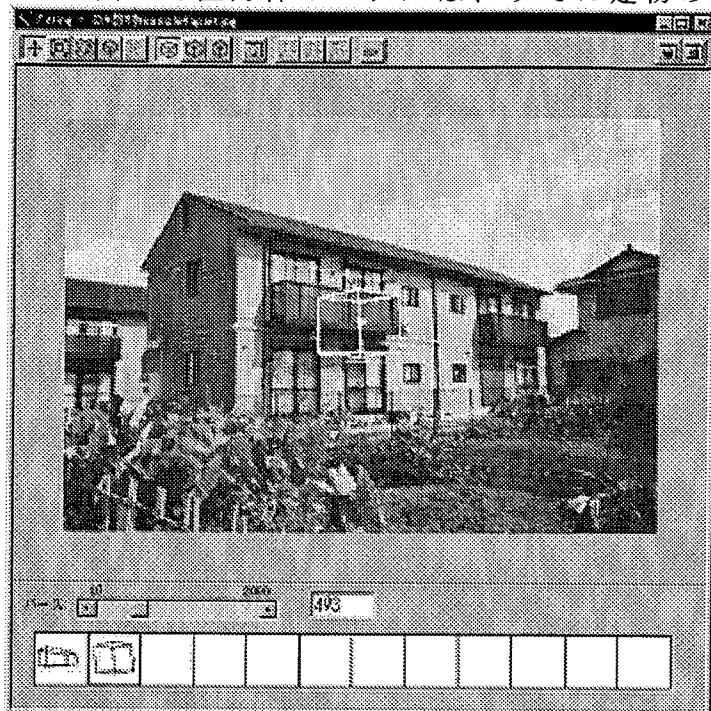


【図VI-2-24】


ベランダの形状に近い形のモデルを選択します。直方体のモデルを選択して(赤い枠で囲まれた状態)、「OK」ボタンをクリックします。

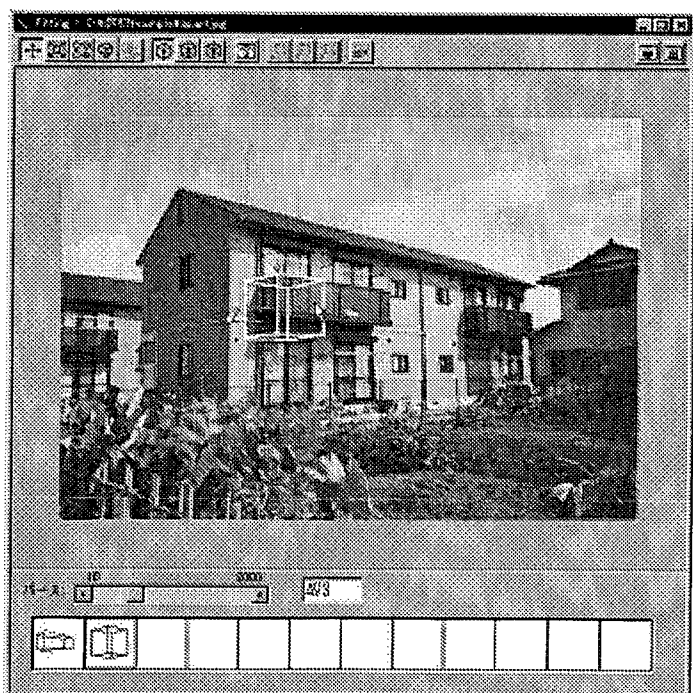
<3> Fittingウィンドウに選択したモデルが表示されます。

呼び出した直方体のモデルは、すでに建物の向きに合った状態になっています。




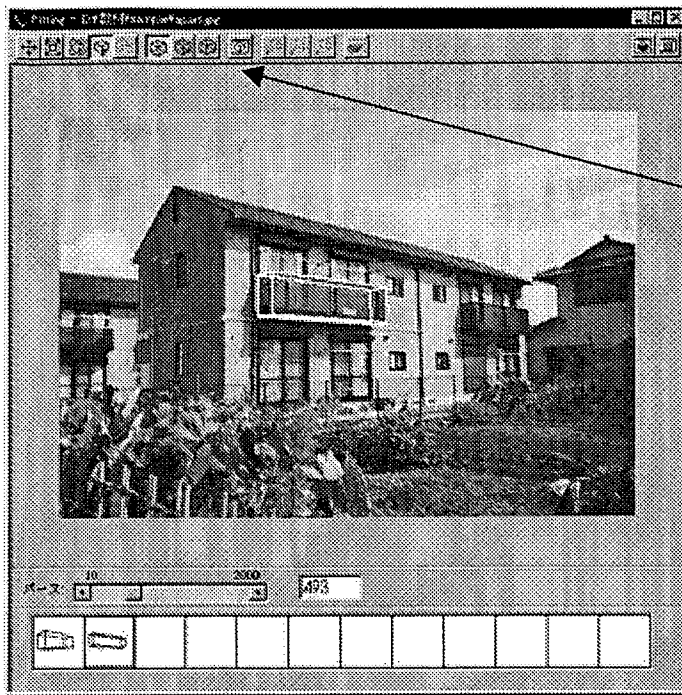
【図VI-2-25】

<4> 「移動」ボタン  を押して移動モードにしてからマウスを左ボタンを押しながらドラッグし、ベランダを移動します。ベランダの角に合わせてモデルを移動します。



【図VI-2-26】

<5> 「変形」ボタンを押して変形モードにして面を変形させ、ベランダの形に合わせます。

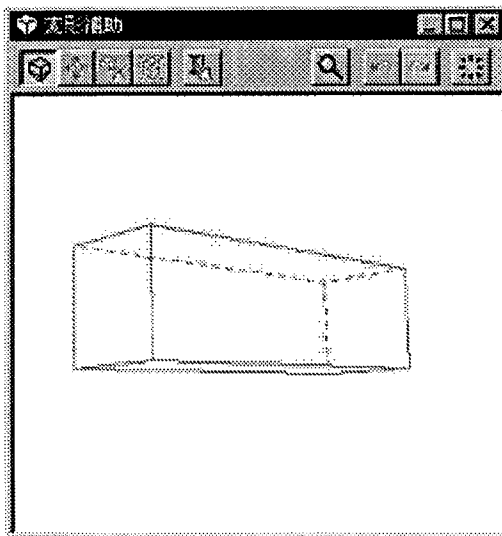


変形補助ウィンドウ
表示ボタン

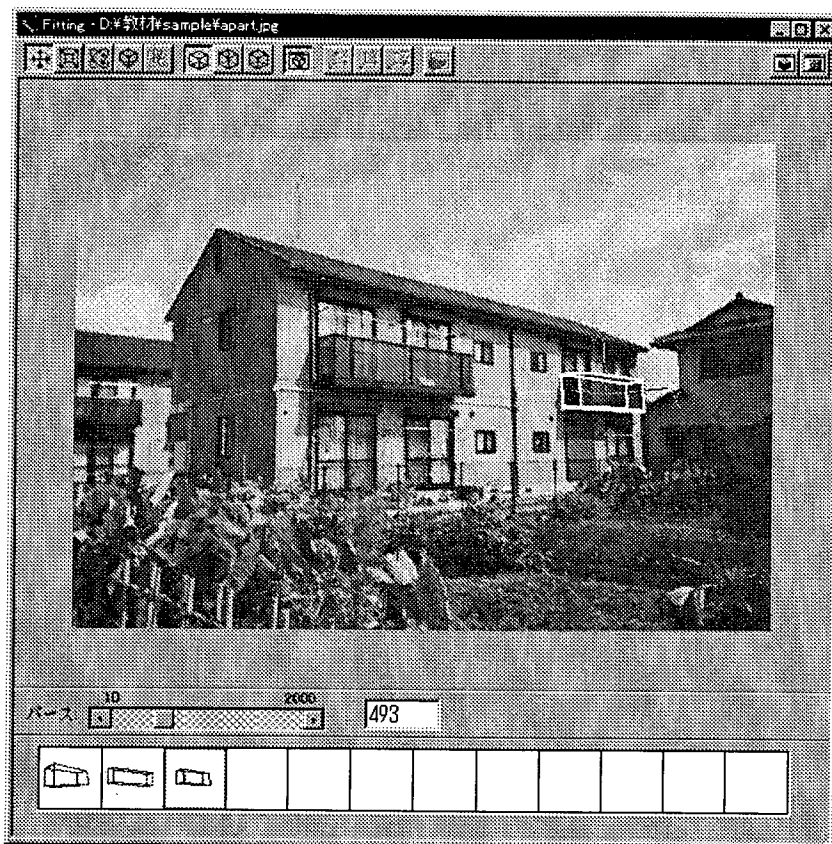
[図VI-2-27]

変形操作は、変形補助ウィンドウを利用すると便利です。変形補助ウィンドウは、変形モード時に「変形補助ウィンドウ」ボタンを押すと表示されます。

<6> <1>～<5>までの過程と同様にもう一つのベランダを作成します。



[図VI-2-28]



【図VI-2-29】

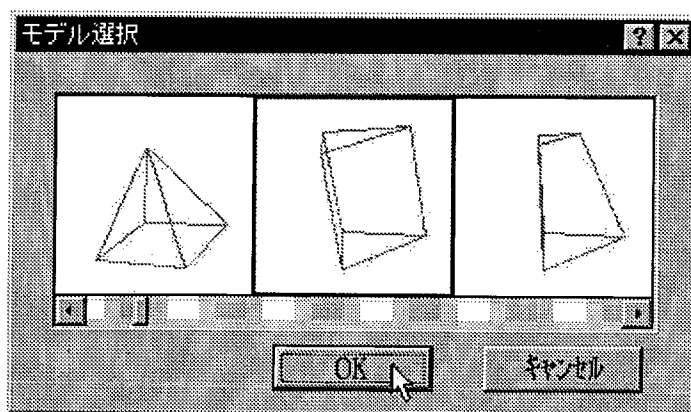
d. 屋根の作成

建物の本体部、ベランダを作成後、最後に屋根を作成します。

屋根のモデル作成には、三角柱の基本モデルを使用します。

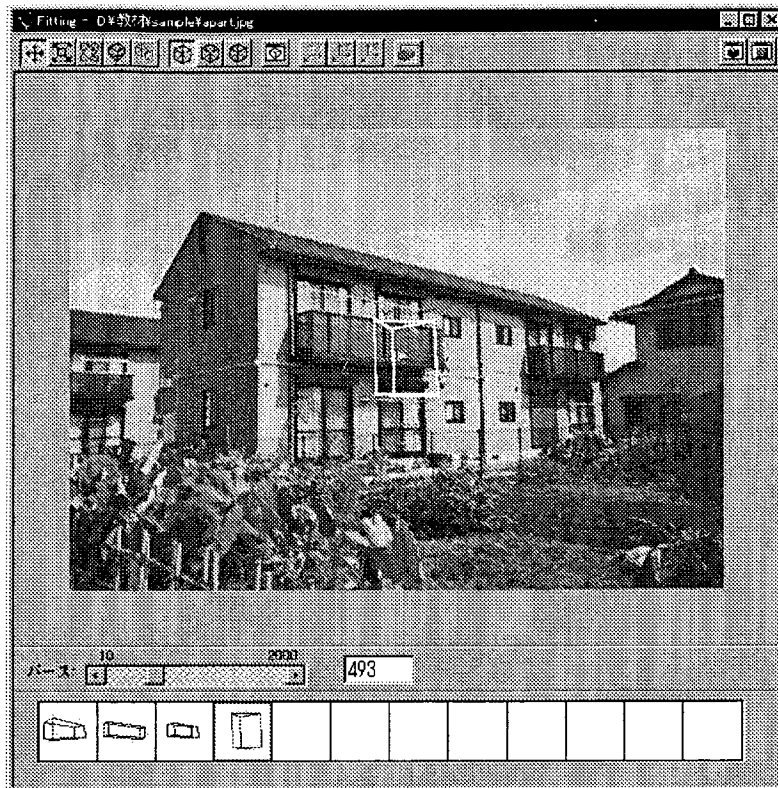
<1> 「編集」メニューから「モデル選択」コマンドを選択します。

<2> 三角柱のモデルを選択します。



【図VI-2-30】

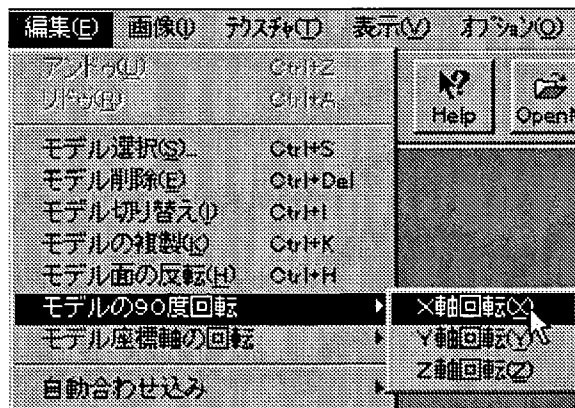
<3> Fittingウィンドウに選択したモデルが表示されます。



【図VI-2-31】

<4> 三角柱のモデルの向きが違うので、回転します。

「編集」メニューから「モデルの90度回転」コマンドを選択し、サブメニューから「X軸回転」コマンドを選択します。



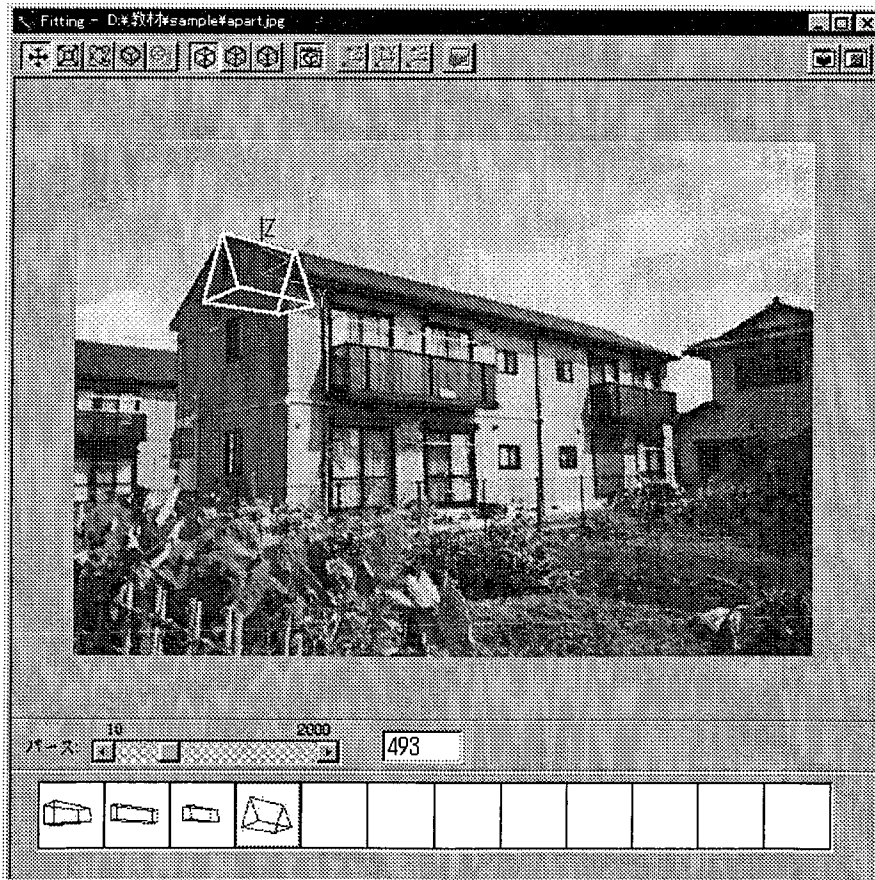
【図VI-2-32】

もう一度、今度は「Z軸回転」を行います。これで、三角柱の向きが屋根の向きと同じになります。あとは、屋根の形に合わせて大きさを合わせていきます。



[図VI-2-33]



<5> 三角柱のモデルを屋根の角に合わせます。



[図VI-2-34]

<6> 屋根の角度にモデルを合わせます。

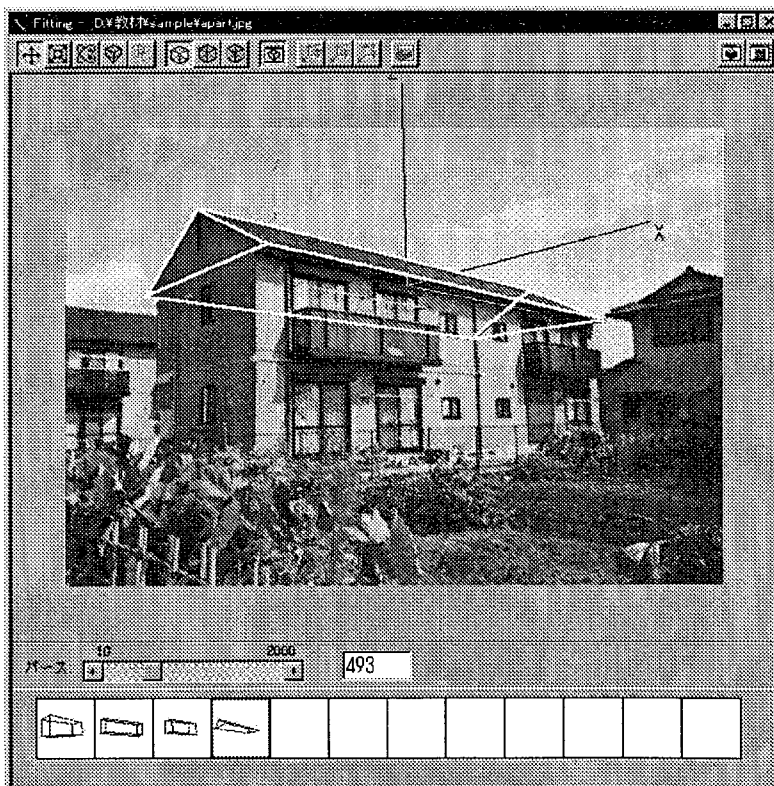
三角柱を左右に伸ばします（X軸方向のみ拡大）。

「拡大」ボタンを押し、X軸を固定します。マウスで拡大し、屋根のすそに合わせます。

<7> 屋根の形に合うように変形させます。



【図VI-2-35】



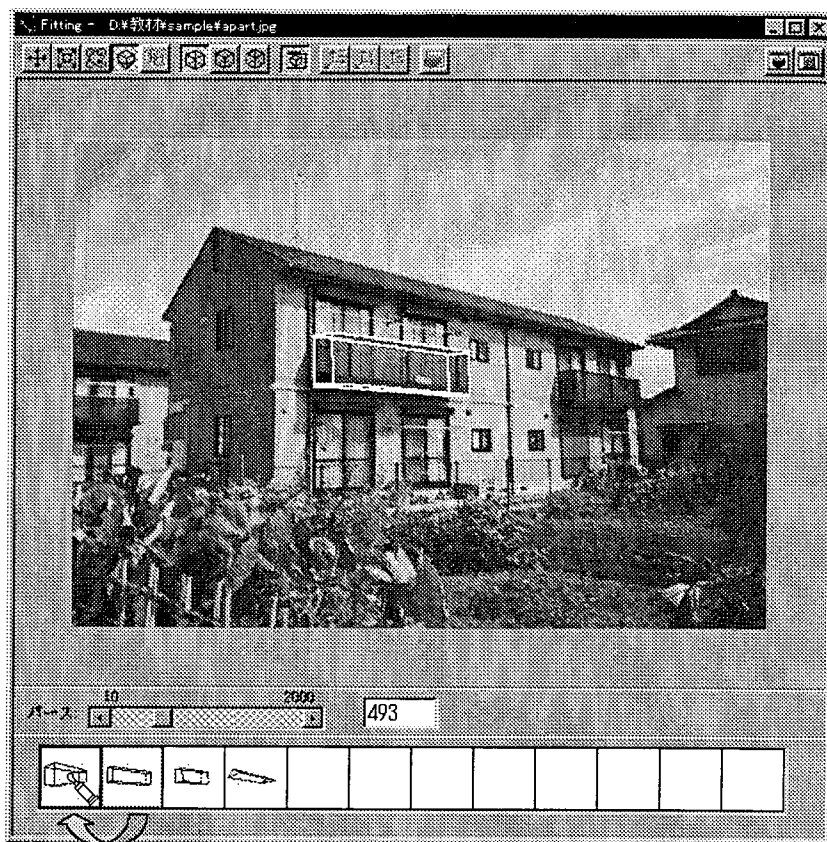
【図VI-2-36】

e. モデル結合

作成した建物の本体部、ベランダ、屋根は、それぞれ個別に作成しているため、最後に結合し建物を完成させます。

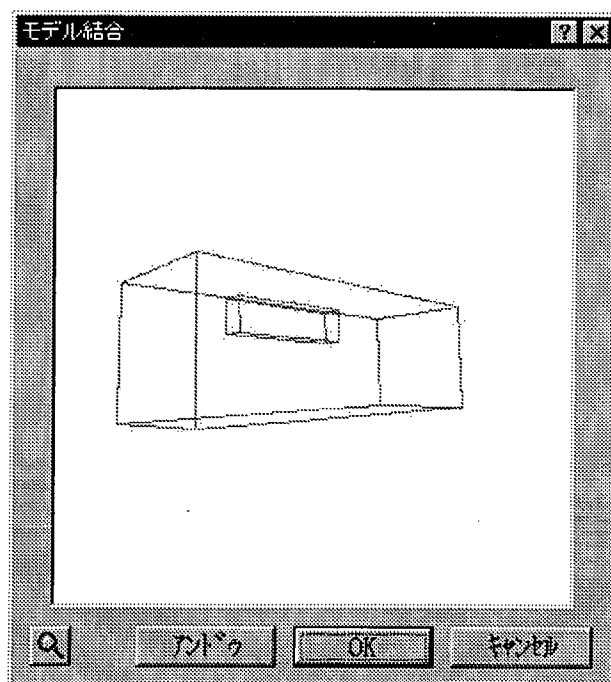
<1> Fittingウィンドウの下部にあるモデル一覧内で、一つ目のベランダのモデルをドラッグし

アパートの部屋部分のモデルにドロップします。



【図VI-2-37】

<2> 「モデル結合」ダイアログが表示されます。

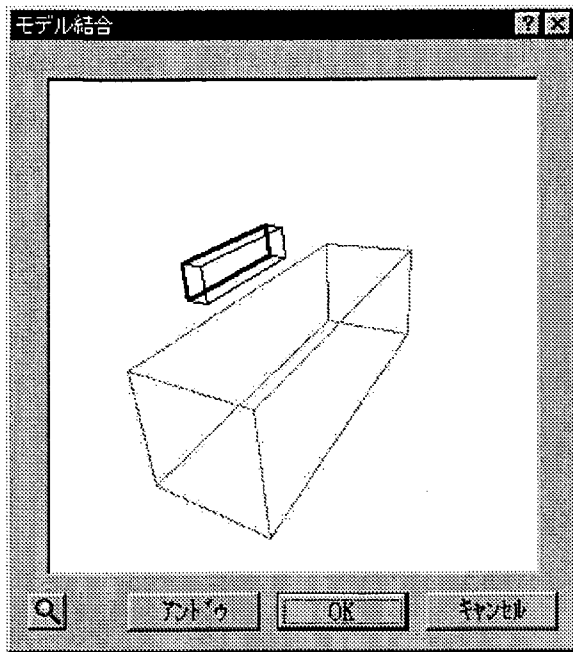


【図VI-2-38】

<3> 結合させたい面を指定します。

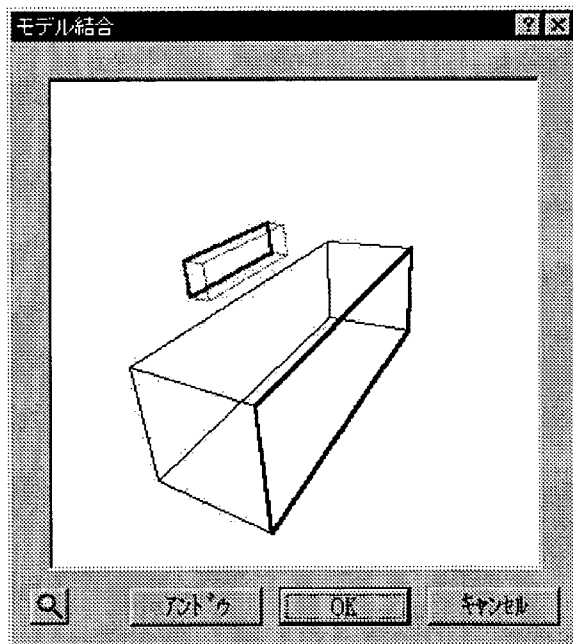
最初にベランダの奥側の面を選択します。選みたい面上にマウスを移動し、マウスの右ボタンで面を

切替えます。



【図VI-2-39】

<4> 次にアパート部屋部分の手前の面を選択します。

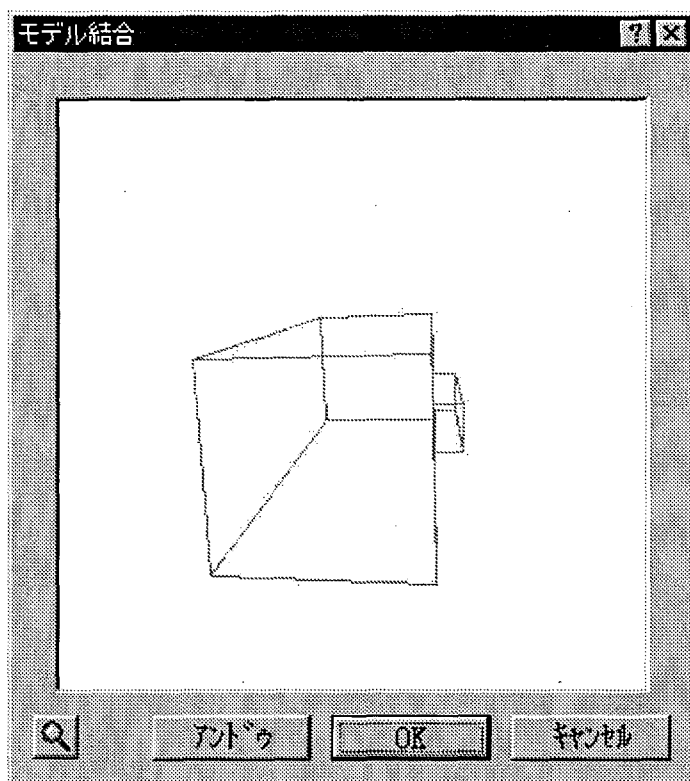


【図VI-2-40】

この時点で、指定した面同士が結合されます。

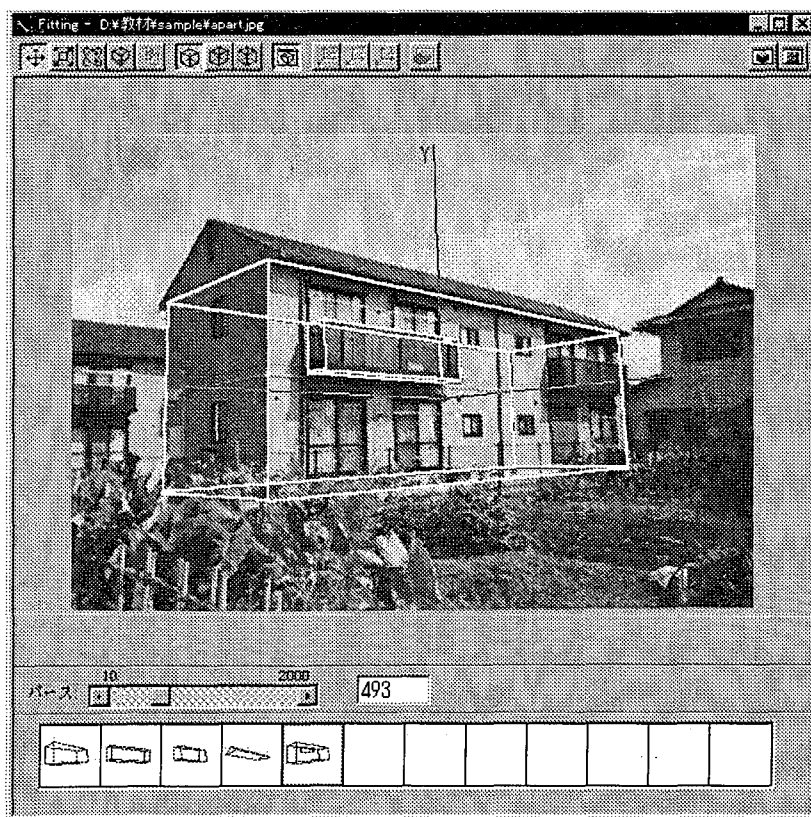
※ 面を選択する時は、マウスの左ボタンをクリックします。
またマウスの右ボタンをクリックすることにより、カーソル位置にある面が順番に切り替わります。

<5> 回転させて確認します。



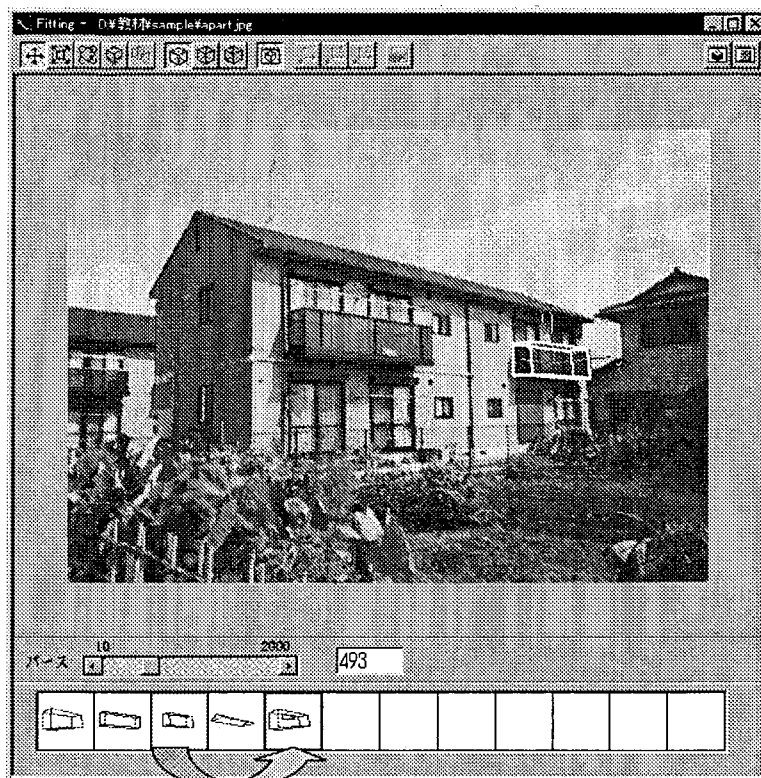
【図VI-2-41】

OKボタンをクリックすると「モデル結合」ダイアログが閉じられ、Fittingウィンドウに結合されたモデルが表示されます。



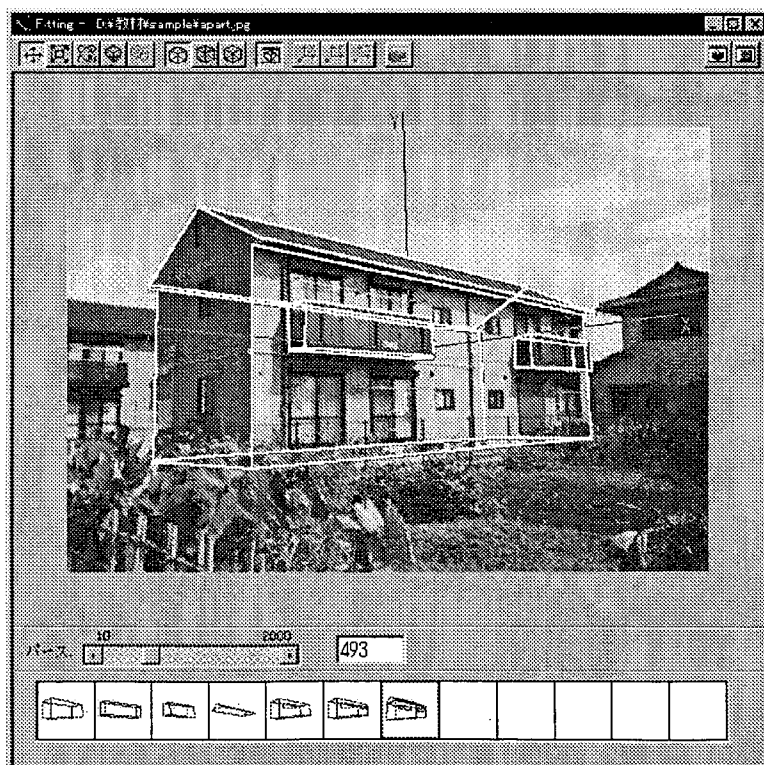
【図VI-2-42】

<6> もう片方のベランダも同じように結合します。





【図VI-2-43】

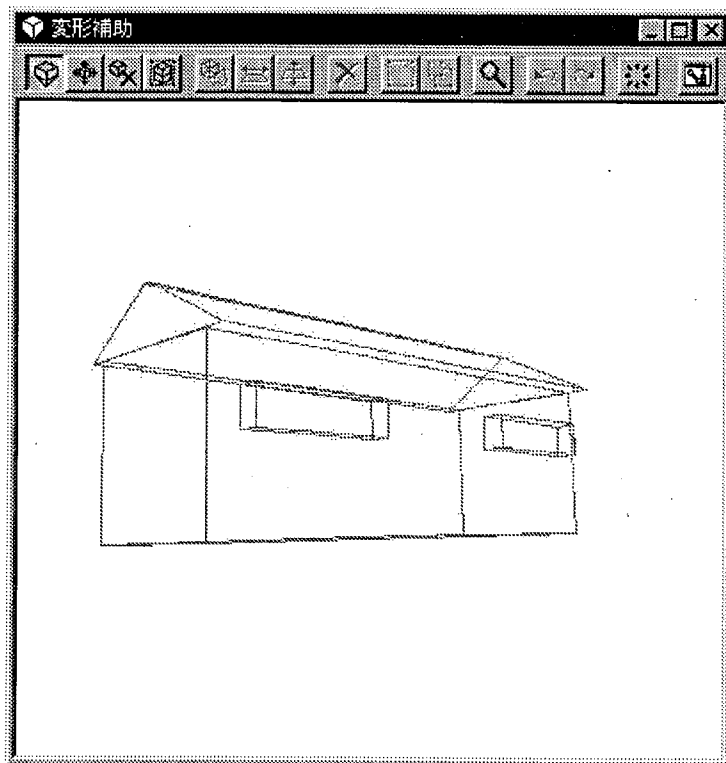
<7> 屋根も同様に結合します。



【図VI-2-44】

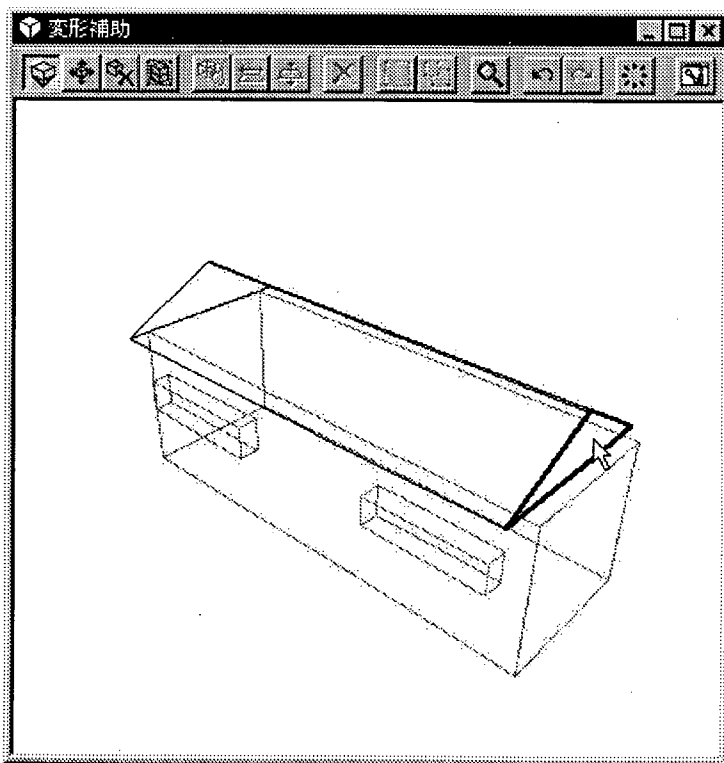
<8> 結合したモデルをチェックします。変形補助ウィンドウのボタンが押されている状態で、変形ボタンをクリックします。

<9> 変形補助ウィンドウが開かれます。



【図VI-2-45】

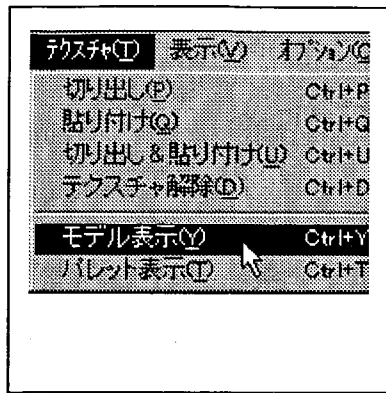
<10> 回転させてモデルをチェックし、修正します。



【図VI-2-46】

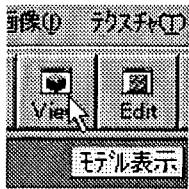
回転させてみると屋根の長さが合っていないのが分かります。

<11> 「モデル表示」ウィンドウで作成したモデルを確認します。
「テキスト」メニューから「モデル表示」コマンドを選択します。



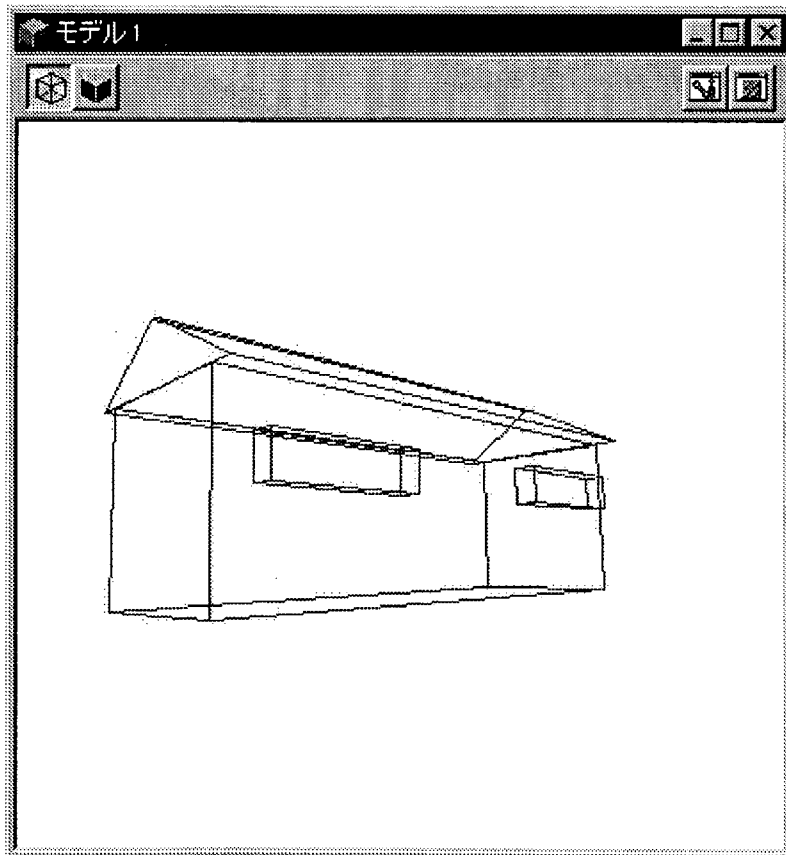
[図VI-2-47]

またはツールバーのボタンをクリックします。



[図VI-2-48]

<12> 作成したモデルが「モデル表示」ウィンドウに表示されます。



[図VI-2-49]

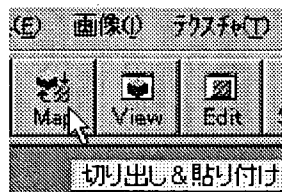
f. テクスチャ切り出し、貼り付け

合わせ込みが完了したモデルに元画像からテクスチャを切り出し、貼り付けます。

<1> 「テクスチャ」メニューから「切り出し&貼り付け」コマンドを選択します。



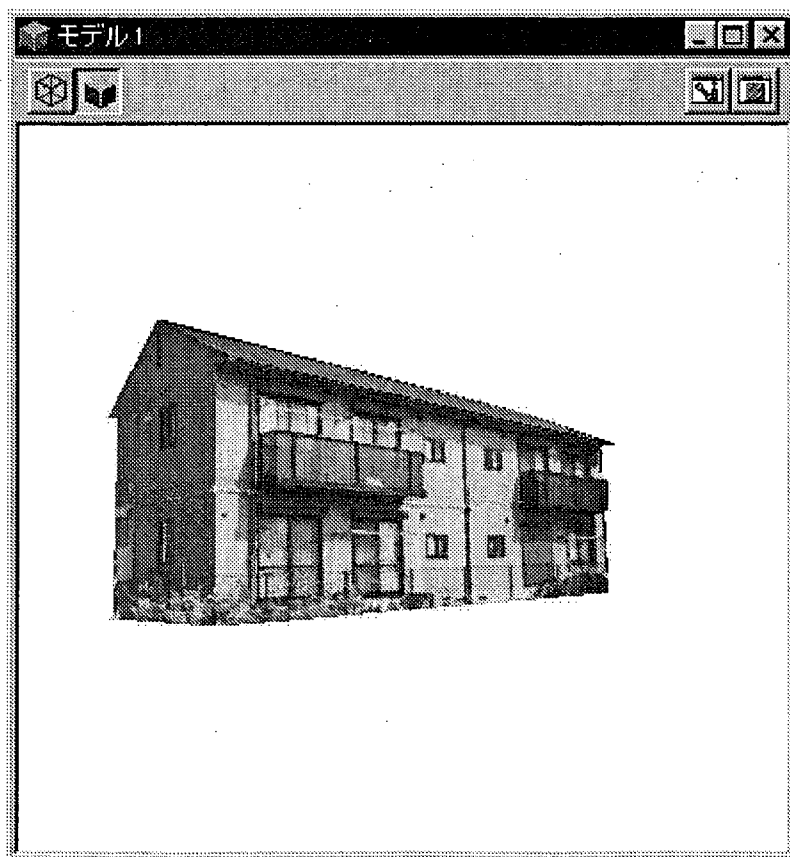
[図VI-2-50]



[図VI-2-51]

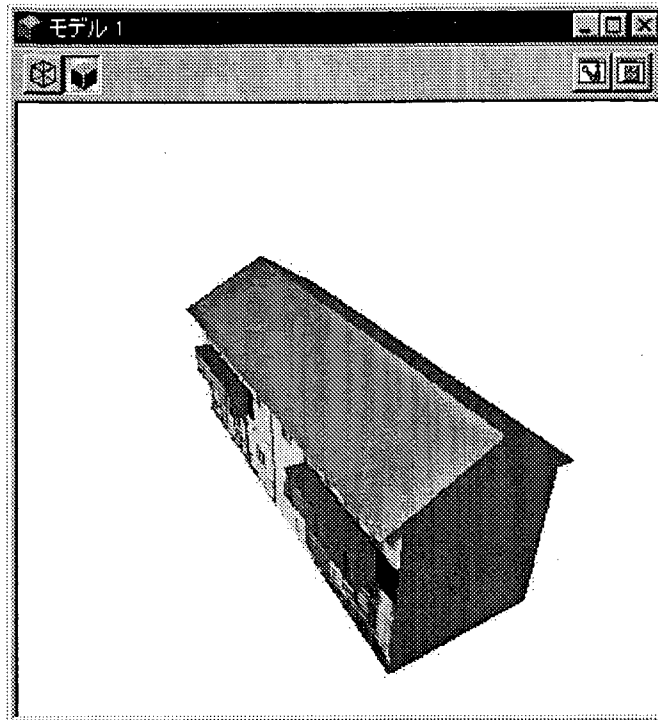
または、ツールバーのボタンをクリックします。

<2> 「モデル表示」ウィンドウに表示されているモデルにテクスチャが貼り付けられます。



[図VI-2-52]

<3>モデルを回転させてみます。写真上で見えていない部分にはテクスチャが貼られず赤色または灰色で表示されます。

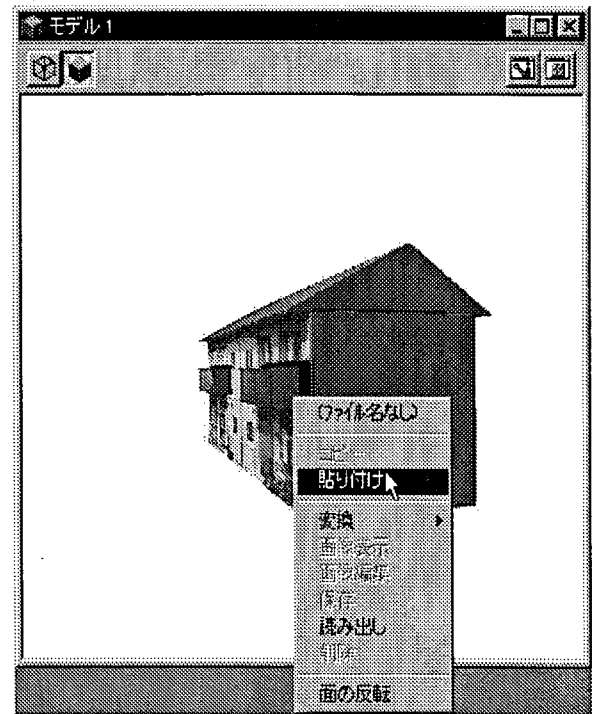


【図VI-2-52】

<4>見えない部分のテクスチャは、見えている部分のテクスチャをコピー&貼り付けます。



【図VI-2-54】



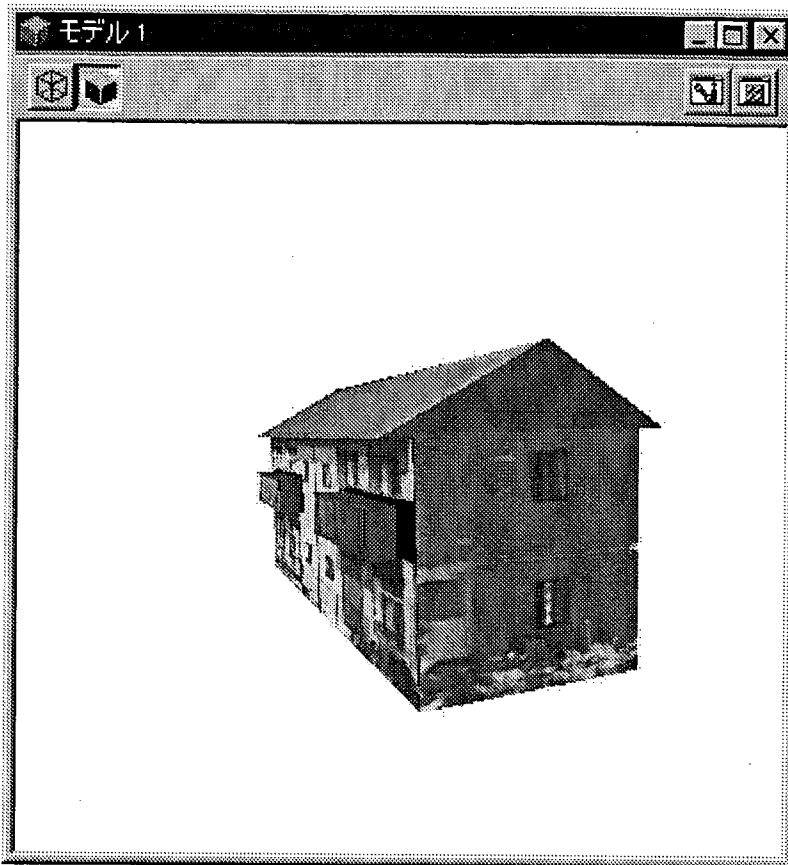
【図VI-2-55】

テクスチャをコピーするには、「モデル表示」ウィンドウ内でコピーしたいテクスチャの貼り付けられている面でマウスの右ボタンをクリックし、ポップアップメニューから「コピー」コマンドを選択します。

コピーしたテクスチャを他の面に貼り付けるには、貼り付ける面でマウスの右ボタンをクリックし、ポップアップメニューから「貼り付け」コマンドを選択します。

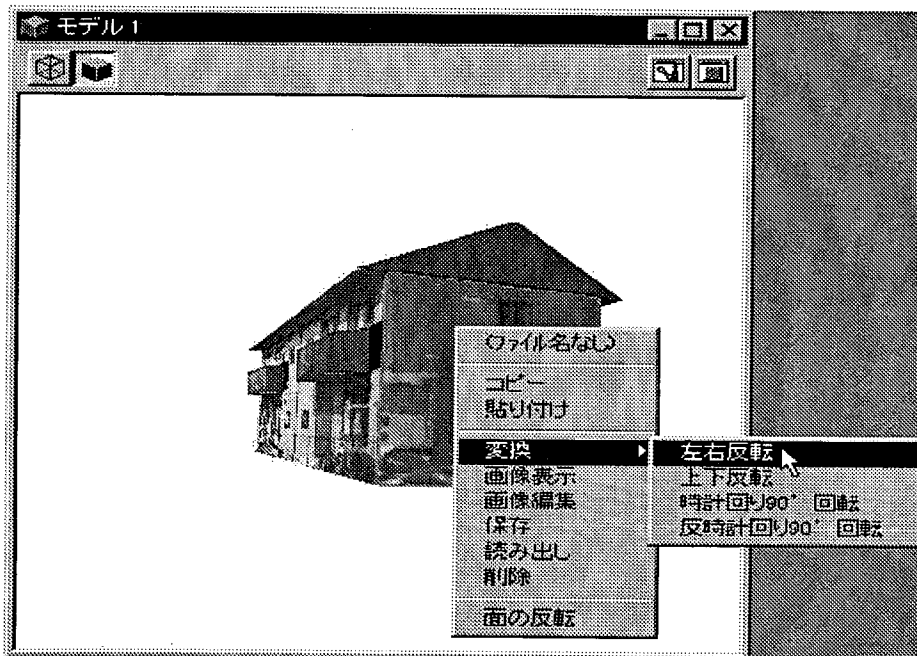
<5> テクスチャが貼り付けられました。

この場合テクスチャが左右反対になってしまったので、向きを変えます。



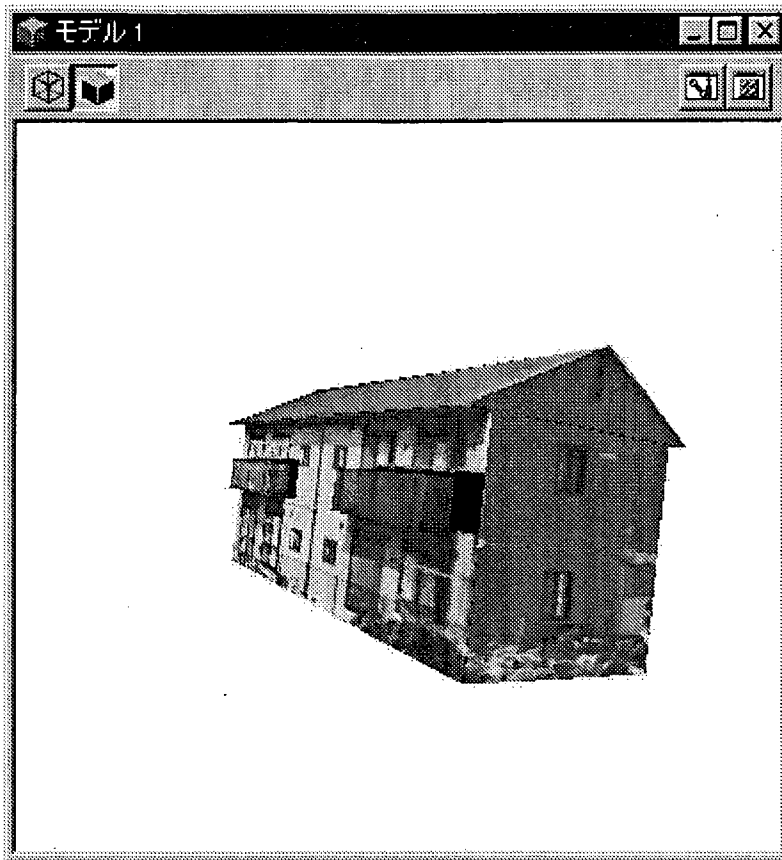
【図VI-2-56】

<6> テクスチャの向きを変えるには、マウスの右ボタンをクリックし、ポップアップメニューから「変換」コマンドを選択します。その「変換」メニューの中から「左右反転」を選択します。



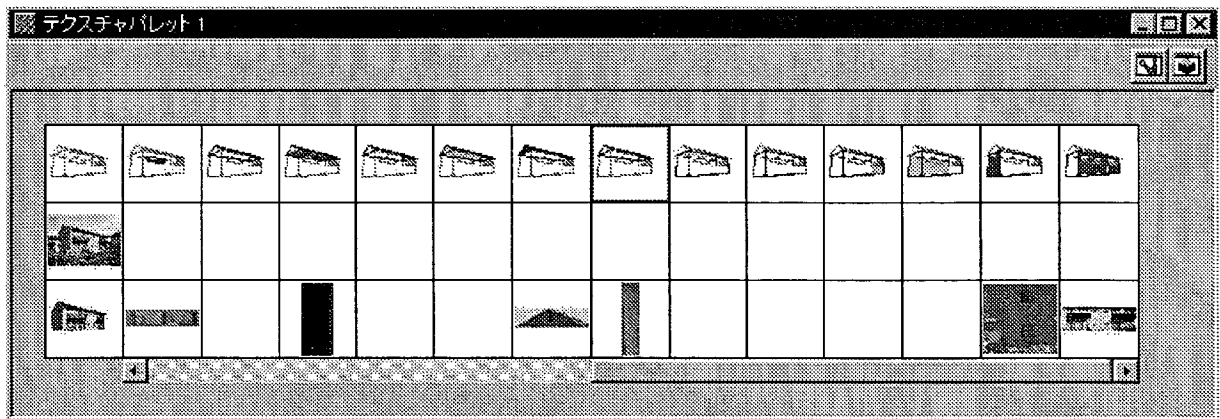
【図VI-2-57】

<7> 屋根やベランダ部分も同様にテクスチャを貼っていきます。



【図VI-2-58】

<4>～<7>の操作は、テクスチャパレットウィンドウを利用しても行うことができます。



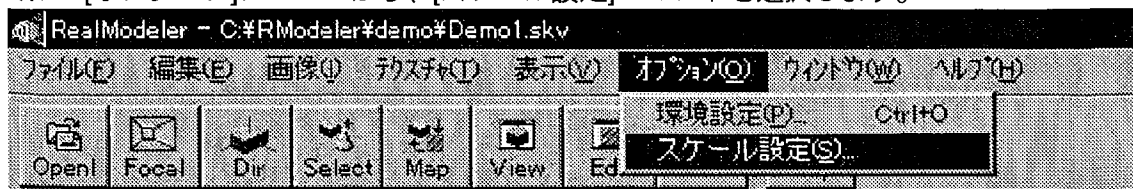
③モデルの保存

【図VI-2-59】

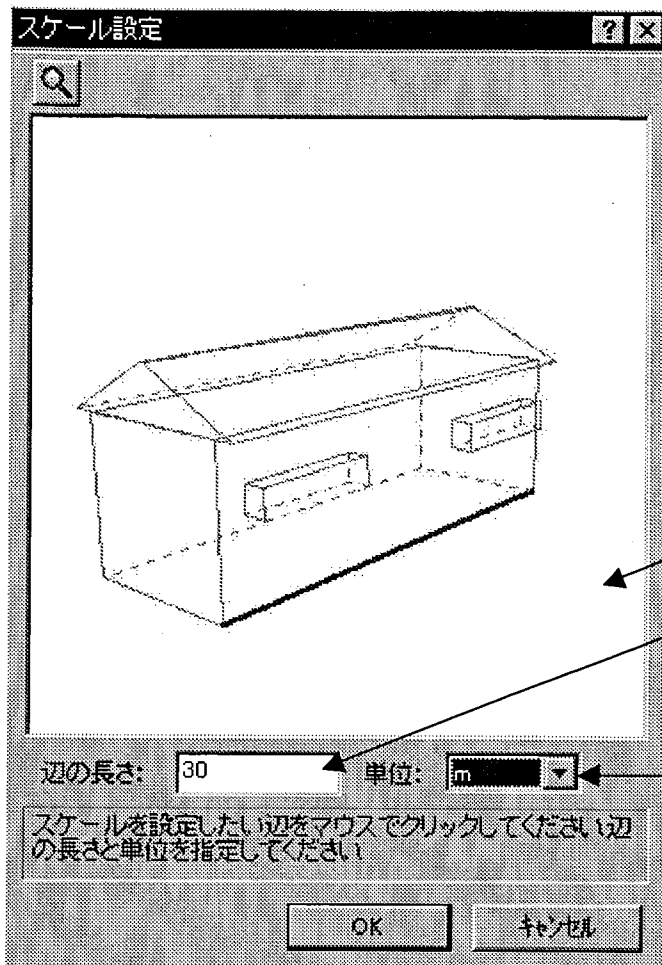
作成したモデルにスケールを設定してから保存します。

a. モデルのスケール設定

<1> [オプション]メニューから、[スケール設定]コマンドを選択します。



【図VI-2-60】



スケールを設定する辺をマウス (左ボタン) で選択します。選択された辺は赤色で表示されます。
 選択した辺 (赤色表示) のスケールを指定します。選択されていない辺は、選択された辺のスケールに比例した値を自動的に計算し、設定されます。

景観シミュレータのスケールの単位に合わせます。

【図VI-2-61】

<2> [スケール設定]ダイアログで、作成したモデルのスケール値を入力し[OK]ボタンを押します。「スケール設定」ダイアログでスケールを設定しない場合、スケールは1未満の値になります。

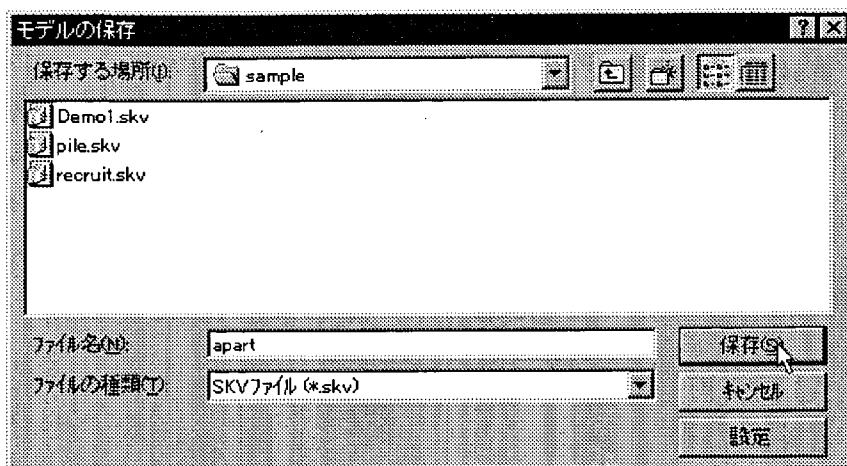
b. モデルの保存

<1> [ファイル]メニューから、[モデルの保存]コマンドを選択します。

ファイル(F)	編集(E)	画像(I)	視点(V)	表示(S)
画像を開く(O)				Shift+V
開く(O)				Shift+O
モデルの読み込み				
閉じる(C)				
画像の保存(S)				Shift+G
モデルの上書き保存(S)				Shift+S
モデルの保存(M)				Shift+M
モデルファイルの削除(D)				Shift+D

<2> [モデルの保存]ダイアログボックスで、ファイル名を入力し[保存]ボタンを押します。

【図VI-2-62】



【図VI-2-63】

Apartというファイル名で保存しました。

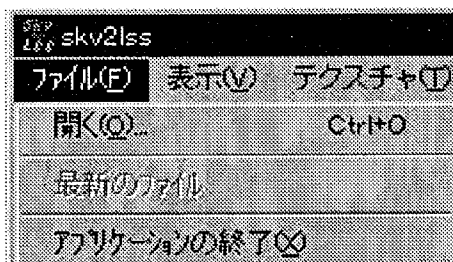
④ 連携ツール

a. 連携ツール (Skv2lss) でデータを変換

Skv2lssでskv形式をLSS形式に変換する手順を説明します。

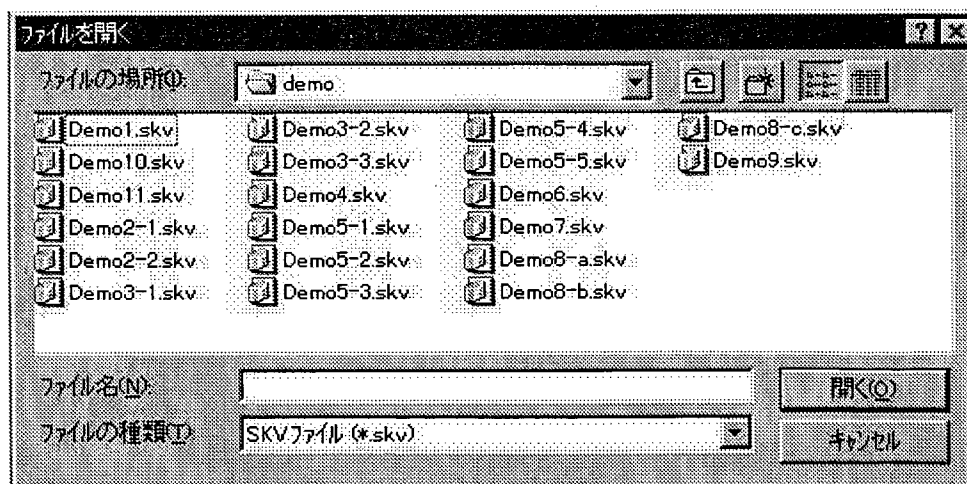
<1> Skv2lssを起動します。

<2> 「ファイル」メニューから「開く」コマンドを選択します。



【図VI-2-64】

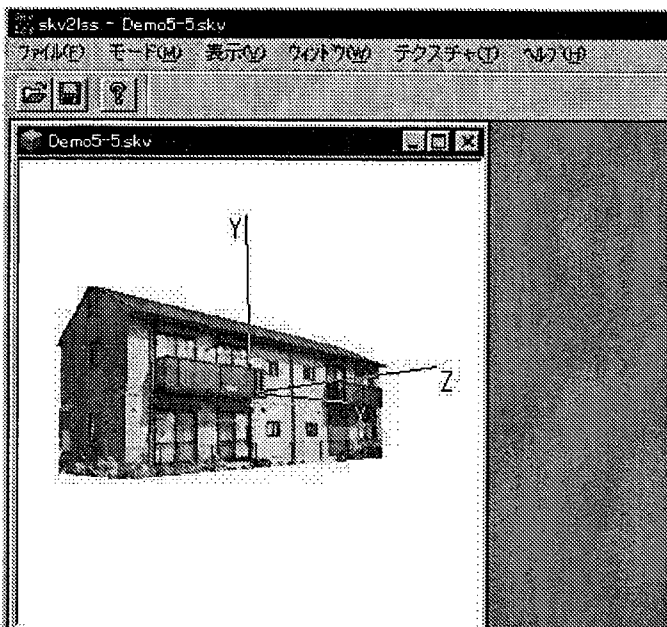
<3> 「画像を開く」ダイアログボックスが表示されます。



【図VI-2-65】

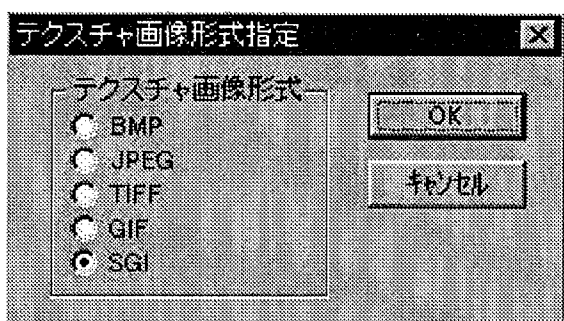
<4> RealModelerで作成、保存したskvファイルを選択します。(例ではDemo5-5.skvを選択)

<5> 画像の保存形式を指定します。(何度も設定する必要はありません。)



【図VI-2-66】

「画像」メニューから「画像形式」を選択します。



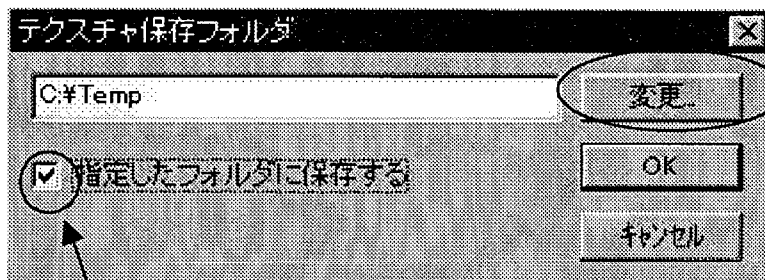
【図VI-2-67】

景観シミュレータで使用する画像形式「SGI形式」を選択します。

<6> 画像（モデルのテクスチャデータ）の保存先ディレクトリを指定します。（何度も指定する必要はありません。）

「画像」メニューから「保存フォルダ」を選択します。

「テクスチャ保存フォルダ」指定ダイアログが表示されます。

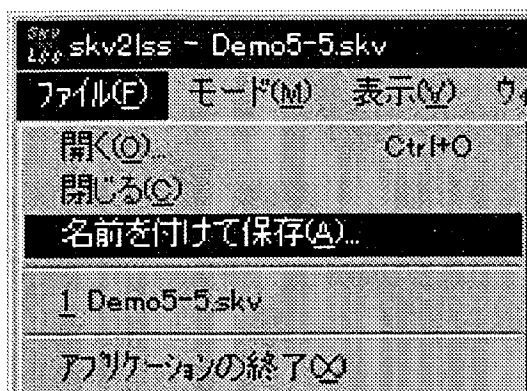


「変更」ボタンを選択し、フォルダを指定します。指定するフォルダは、景観シミュレータで画像（SGI形式）を保存する固定のディレクトリです。

【図VI-2-68】

フォルダを指定したらチェックします。
「OK」ボタンを選択してダイアログを終了します。

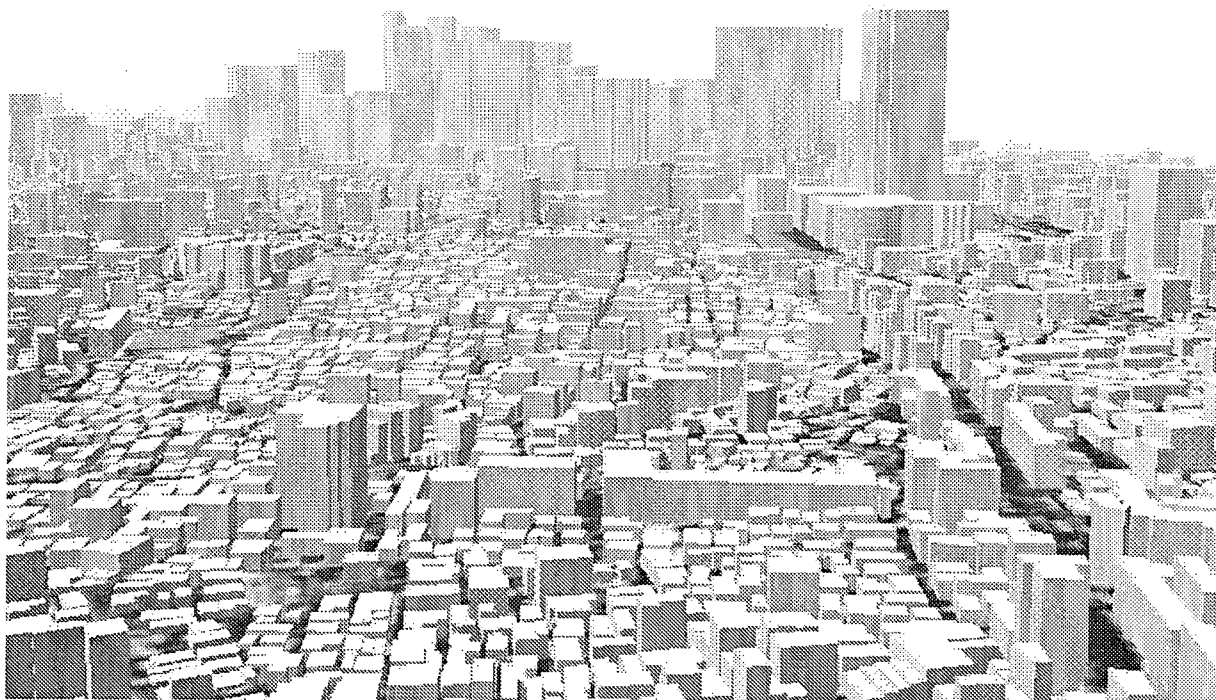
＜7＞「ファイル」メニューから「名前を付けて保存」を選択し、LSS形式で保存します。



【図VI-2-69】

保存と同時に、画像はSGI形式で指定した画像の保存フォルダへ保存します。
以上の変換操作により、景観シミュレータで利用することができる形式で登録されます。

3. 三次元データベース



アジア航測では、従来、地形データ (d t m) 作成などに利用されてきたステレオ空中写真自動解析技術を進めて、市街地を構成する建物の3次元データを作成するシステムを開発し、これを用いて、東京都区部などにおいては既にデータを作成しています。この技術を適用することにより、建設現場周辺の地形+市街地の3次元データを簡便に製作することが可能です。

但し、同社ではシステムそのものをユーザーに供給することはせず、あくまでも「空中写真からデータを作成する」というサービス、あるいは「作成されたデータ」を商品として扱っています。従って、ユーザーとしては、作成された地形+市街地のデータを、目的に応じた範囲・形態で取出し、様々な応用する、という使い方になります。

官民共同研究では、解析されたデータ (CD-ROM) から、必要な範囲を切り出して、景観シミュレータでテクスチャ付き地形+建物のデータとして利用できるファイル形式で出力する、検索+ファイル変換のツールを開発しました。このツールと、サンプルデータは、同社のホームページから自由に公開されています。ダウンロードして、試すことができます

<http://www.ajiko.co.jp/infocom/3ddb/index.html>

実務上は、既にデータが作成されている区域については、同社からデータを購入し、また、それ以外の区域については、既存または新たに撮影した空中写真を解析するサービスを利用することができます。

景観シミュレータの側でも、このシステムで作成した地面の扱い等に工夫を加えました。例えば、

大きなデータの市街地を編集する作業の能率を向上するために、全体をワイヤーフレーム表示にしながら、地面だけをテクスチャ付きで表示する機能などです。

本書のⅠ．に、幕張付近のデータを用いた操作例を掲載しておきました。また、3次元データベース及び幕張のサンプルデータを、付録のCD-ROMに掲載しておりますので、色々と試して下さい。

建物のデータは、壁と屋根（多角形の水平面）でできており、それぞれ別のグループとなっています。従って、屋根だけを削除することができます。

時として、建物の屋根面のある頂点が、微細なループ（病的な形）になっている場合があります。このような場合には、表示段階で、景観シミュレータからエラー表示が出ます。このような場合、CONCAVE属性を外す、として当面对処するのが良いでしょう。また、その屋根面を選択しておいて、平面生成を行うと、この平面を座標値を見ながら、再編集して修正することができます（Ⅳ5(12)参照）。

また、このデータを出発点として、屋根面を切妻・寄棟などの形状に置換えたり（形状生成の機能を使用→Ⅳ5(13)）、削除した上で前述のリアルモデラーを用いて、地上写真から作成したデータに置換えることができます。原理を理解したら、配置機能（Ⅳ5(8)）の中でパラメトリックな屋根を作成するのが、最も能率的と思います。

幕張の再開発の現場では、計画案ができるまでの間、このデータの建物に、建物の種別毎に異なる色を付けて（Ⅳ5(3)参照）テーマ別の現況分析に用いたり、また将来の高層住宅のバルコニー等の位置に視点を設定して、将来の眺望を検討する、といった使い方を工夫しました。

平成13年度には、15のモデル現場の内、13ヶ所について、現況地形+市街地を3次元データベースとして作成しました。

（1）機能

アジア航測株式会社で作成した3次元データベースを建設省版「景観シミュレータ」用LSS-Gデータに変換します。3次元データベースに含まれるデータは以下の通りです。

- ・建物形状（壁面および屋根）
- ・土地形状（メッシュ状標高データ）
- ・土地画像（航空写真データ）

（2）稼働条件

- ・コンピュータ IBM PC/AT互換機, NEC PC-98NX, NEC PC-9821
- ・OS Windows95, Windows98, WindowsNT4.0
- ・メモリ 16MBytes以上。
- ・ディスク 50MBytes以上の空き容量（データを除く）。
- ・ディスプレイ 640×480ドット以上, 65536色以上。
- ・その他 「景観シミュレータ」がインストールされている環境が必要です

（3）インストール

インストーラはありません。「3ddb.exe」ファイルを適当なフォルダにコピーしてお使い下さい。

レジストリ等は書き換えません。

(4) アンインストール

アンインストーラはありません。「3ddb.exe」ファイルを削除して下さい。

(5) 変換操作説明 (ファイル直接指定)

変換する1/2500図葉ファイル名が分かっている場合に以下のように操作します。

① 3DDB の起動

「3ddb.exe」を起動させます。アイコンをダブルクリックする等で起動させて下さい。

② 1/2500 図葉ファイルを開く

「ファイル(F)」メニューから「1/2500 図葉ファイルを開く(O)...」を選んで 1/2500 図葉ファイルを開きます。ファイル拡張子は「.3db」です。

この 1/2500 図葉ファイルには 1/2500 図葉単位の 3次元建物形状と土地形状、土地画像が収められています。

データは真北を上にして東西方向 2000m、南北方向 1500m が範囲です。

③ LSS-G データに変換する区画を選ぶ

1/2500 図葉ファイルを 25 等分した区画から LSS-G データに変換する区画をマウスの左ボタンをクリックして選択します。選択された区画は青色の枠が表示されます。選択した区画をもう一度クリックすると選択が解除されます。

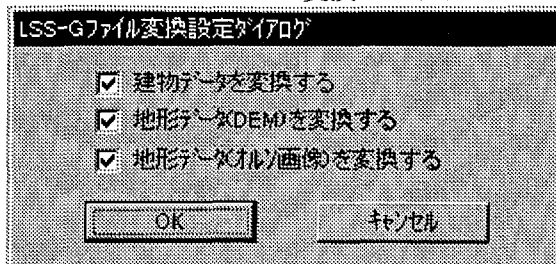
1つの区画は東西方向 400m、南北方向 300m が範囲です。



[図VI-3-1] 選択できる区画は1度に1区画のみです。

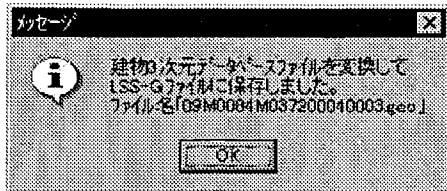
④ LSS-G ファイルに変換する

区画が選択された状態で「ファイル(F)」メニューから「LSS-G ファイルに変換して保存(G)...」を選んで LSS-G ファイルに変換します。



[図VI-3-2] LSS-G ファイル変換設定ダイアログ

この操作後「LSS-G ファイル変換設定ダイアログ」が表示されます。
変換させたい項目にチェックマークを付けて「OK」ボタンをクリックします。
変換が成功すれば以下のようなメッセージダイアログが表示されます。



【図VI-3-3】変換成功の表示

変換された LSS-G ファイル名の規則は次のようになっています。

09M0084M037200040003.geo の場合

- ① ② ③ ④ ⑤

①新国家座標系の番号

01 から 19 まで
09 は第 9 系を表しています。

②変換された範囲の新国家座標系の左下座標値（東西方向）

M はマイナス、P はプラス。
数値は 100m 単位で表記。
M0084 は 8400m を表しています。

③変換された範囲の新国家座標系の左下座標値（南北方向）

M はマイナス、P はプラス。
数値は 100m 単位で表記。
M0372 は 37200m を表しています。

④変換された範囲の長さ（東西方向）

数値は 100m 単位で表記。
0004 は 400m を表しています。

⑤変換された範囲の長さ（南北方向）

数値は 100m 単位で表記。
0003 は 300m を表しています。

「地形データ(判)画像を変換する」にチェックマークを付けて変換させた場合は、土地画像をテクスチャーとした SGI 画像ファイルも生成されます。SGI 画像ファイル名の規則も LSS-G ファイル名の規則に準じています。

例 **09M0084M037200040003.sgi**

LSS-G ファイル (*.geo) は kdbms.set ファイル内の FILE_PATH_GEOMETRY で指定されたフォルダに書き込まれます。

SGI 画像ファイル (*.sgi) は kdbms.set ファイル内の FILE_PATH_TEXTURE で指定されたフォルダの下にある「sgi」フォルダに書き込まれます。

(6) 変換操作説明 (フォルダ指定)

変換する 1/2500 図葉ファイル名が分からない場合に、以下の操作を行います。

① 3DDB の起動

「3ddb.exe」を起動させます。アイコンをダブルクリックする等で起動させて下さい。

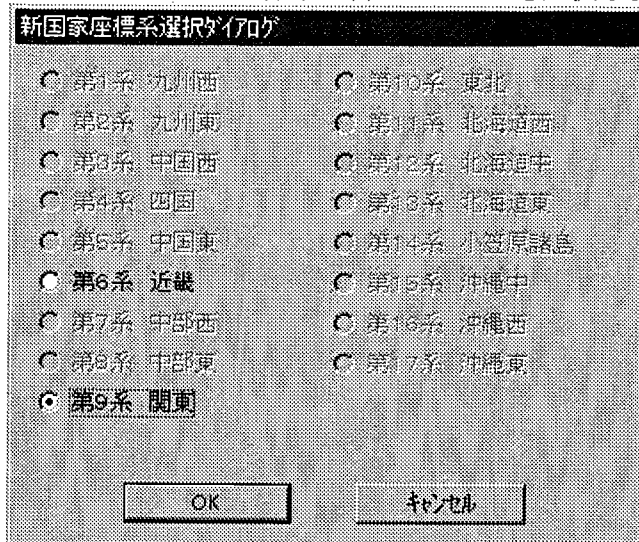
② 1/2500 図葉ファイルがあるフォルダを指定する

「ファイル(F)」メニューから「フォルダ内のファイルデータ位置を表示...」を選んで 1/2500 図葉ファイ

ルがあるフォルダを指定します。フォルダ内にある 1/2500 図葉ファイルのデータを読み込んで表示します。

③ 新国家座標系を選択する

新国家座標系単位でインデックスマップを表示するために、読み込みたい 1/2500 図葉ファイルがどの新国家座標系に属しているかを選択します。

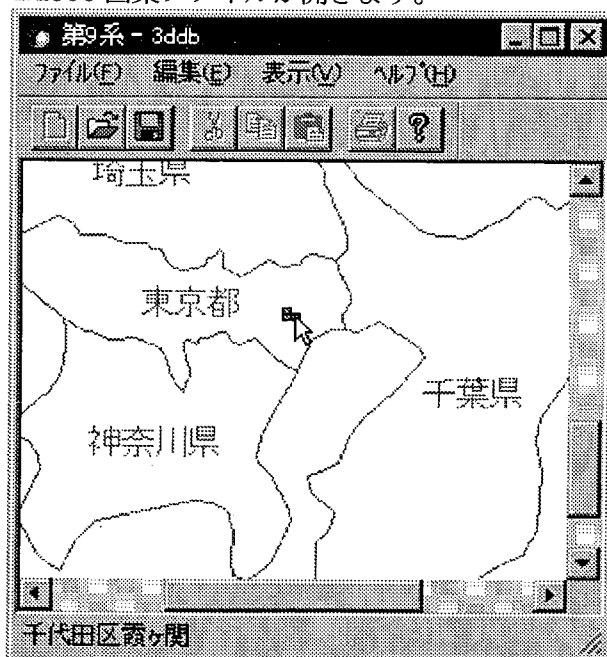


【図VI-3-4】新国家座標系の選択

黒丸のチェックを入れて「OK」ボタンをクリックするとインデックスマップが表示されます。現在用意されているインデックスマップは第6系と第9系です。

④ 1/2500 図葉ファイルを開く

1/2500 図葉ファイルがある領域は緑色で表示されます。1/2500 図葉ファイルに地域名が書き込まれている場合、マウスカーソルをその領域に当てるとステータスバーの部分に地域名が表示されます。その状態のままマウスボタンをダブルクリックするとその領域の 1/2500 図葉ファイルが開きます。



【図VI-3-5】図葉ファイルがある領域の確認と選択

⑤ LSS-G データに変換する区画を選ぶ

ファイル直接指定方法の「3. LSS-G データに変換する区画を選ぶ」をご覧ください。

⑥ LSS-G ファイルに変換する

ファイル直接指定方法の「4. LSS-G ファイルに変換する」をご覧ください。

(7) 終了

「ファイル(F)」メニューから「3DDBの終了(X)」を選んで終了させます。



【図VI-3-6】 終了の方法

4. 景観シミュレータの多国語対応

(1) ハングル版景観シミュレータ

大韓民国農村振興庁韓国農漁村振興公社農漁村研究院では、1996年度より、景観シミュレーションシステムの開発をすべく、日本で事前調査を行い、建設省版の存在を知る所となりました。シミュレーションの対象となる事業・地域は異なるものの、システムの基本的な原理には共通する点が多いため、共同で役割分担を行うことにより、より豊かな成果を共有することができる、との見通しを得ました。そこで、1996年9月10日、11日にソウルで開催された第9回日韓科学技術委員会に提案した所、採択されました。

協定下の課題名を「地域開発における景観シミュレーション技術の応用」（旧コードJ21、新コードE109）とし、内容として

①目的：コンピュータを用いた景観シミュレーション・システムの研究開発を共同で行い、地域開発における、より良い設計、意志決定、市民参加を支援する。

②協力内容：①同じ技術の重複する開発を避け、役割分担により実用的な景観シミュレーション・システムを開発し、より豊かな成果を目指す。②まず二国語に対応することで、国際化に柔軟に対応できる景観シミュレーション・システムを開発する。③評価技術、市民参加に基づく計画・開発手法等、景観シミュレーションシステムを実際の現場に適用するための技術と知識を集約する。

③協力形態：共同研究の実行、研究者の相互訪問、ソフトウェア及びデータなどの研究。成果を含む研究情報・技術情報の交換、共同のセミナー・会議の推進。

④その他：両者は、積極的に成果を活用する努力を行いつつ、商品化を避け、あくまで公共財産としての普及を図る。

という了解の下に、着手する所となりました。なお、日韓科学技術協力協定への採択を承け、1997年6月2日に、日本国建設省建築研究所（島崎勉所長）と、大韓民国農漁村振興公社農漁村研究院（黄奎泰 院長）の間で、具体的な内容に関する Record of Discussions を取り交し、これに基づいて、西暦2000年12月31日を期限として、共同研究を実施することとなりました。

1997年5月に、建築研究所のホームページからVer.2.03を公開したのを期に、同年8月には、ハングル文字への翻訳移植を試み、1週間の予定のところ、ほぼ3日間で完了することができました。

この頃、韓国側で通貨危機・IMF介入という異常事態となり、公務員等の海外渡航禁止がかかる中、上記協定に採択された事が幸いし、当研究に関しては、海外渡航が特別に認められ、日本の現場での使用例等を紹介することができた上、韓国側で研究課題が評価され、上記公社の自主研究から、国費による研究に格上され、重点的な予算配分を得ることができました。これにより、VRML対応や、地形入力などの韓国側独自の開発成果が積上げられつつあります。1999年度には、日本側でのデバッグが進み、安定性・信頼性・操作性を高めたVer.2.05が概成したため、これをベースに8月に再度翻訳・移植を行い、インストーラまで作成することができ、韓国においてもデモから普及段階に到達することができました。

この作業を通じ、言語に依存する部分を、ヘッダーファイル等に集約することにより、翻訳移植を容易化するような整理整頓が進み、後述のインドネシア語への対応も、スムーズに進めることができました。

現在では、利川市内の文化村整備事業や、唐津郡の干拓地における自然生態公園整備事業などに導入しています。韓国側での開発成果を、日本語版に逆翻訳・移植し、成果を共有することが、今後の課題として残されています。

農漁村振興公社及び農漁村研究院に関しては、以下のサイトから情報を得ることができます。但し、ハングル表示を行うためには、WindowsNT4.0の場合には、ハングルのフォント等を追加インストールする必要があります。またWindows95,98等のシステムでは、AsiaSurf等の補助ツールを購入してインストールする必要があります（韓国には、日本のホームページを閲覧したり、自動翻訳するような手段が既に豊富にあります。日本から韓国等のアジア圏のホームページ等にアクセスする環境は、未だ発展途上の観があります）。

<http://www.karico.co.kr/rri>

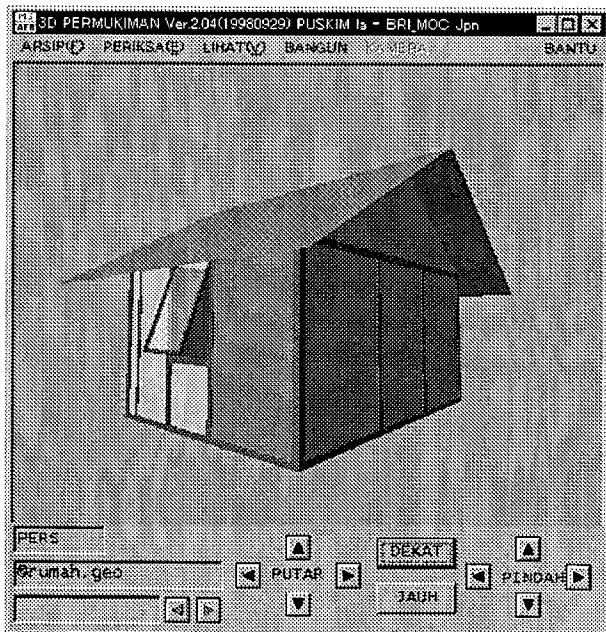
(2) インドネシア語版景観シミュレータ

1993～98年度に、JICAの社会開発事業として、インドネシアにおける集合住宅適正技術開発プロジェクトが行われました。このプロジェクトは、都心に近い公有地などに、不法占拠に由来する低層高密度の住宅地が広範に存在するインドネシアの大都市において、住民に強制退去を強いることなく住環境の改善を進めるために不可欠な、安全・快適かつローコストの中高層集合住宅のプロトタイプを開発することが目的でした。この研究開発は、インドネシア公共事業省・人間居住研究所が中心となっており、建設省建築研究所からも多くのスタッフが専門家として派遣されました。

この中で、計画案の検討及びプレゼンテーションのために試用することを目的として、インドネシア語化の作業が行われました。

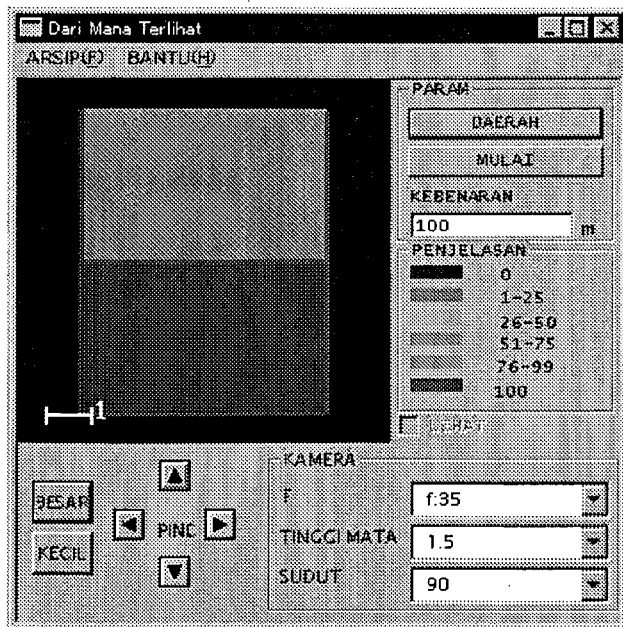
結果的に、インドネシアではまだパソコンが相対的に高価であるのに対し、模型や原寸のモックアップを作成する費用がより低廉であったため、原寸模型による体験入居の方が有効であることが判明しました。

しかしながら、メニュー、ヘルプ、エラーメッセージ等が全てインドネシア語で表示される本格的なソフトウェアとしては、第一号と考えられ、今後の普及が期待されます。また、当地においては、様々な英語版のソフトウェアが、不法コピーにより出回っており、ウィルスによる被害も水や空気のようにあたりまえに存在している中、自由に合法的にコピーできるフリーウェアとしての本システム(インドネシア名：3D Permukiman)は、一つの新しい方向を提示していると考えます。



[図VI-4-1] 実験住宅（メイン画面のインドネシア語表示）

同プロジェクトを行った人間居住研究所等に関しては、以下のサイトから情報を得ることができます。



【図VI-4-2】視点設定画面のインドネシア語表示

<http://www.kimpraswil.go.id>

(3) その他の言語への翻訳・移植の方法

以上の取組みから、国際化対応とは、英語などの国際的に用いられている言語に翻訳することではなく、実際のユーザー（地方の現場担当者を想定）が使えるように、それぞれの地域の言語に容易に翻訳できることである、という考え方に到達します。これらの経験を通じて、言語に依存する部分（メニューや各種コントロールなどのユーザー・インターフェース、ヘルプ・ファイル、エラー・メッセージ等）の整理が進み、新たな言語への翻訳作業がより容易に行えるようになってきました。

Ver.2.05をベースとして翻訳・移植を行う場合、以下のような手順となります。

①リソース・ファイルの翻訳

リソース・ファイルの中には、様々のダイアログのデザインに関する情報が入っています。ボタンの名称などには日本語等のローカル言語が多く含まれるので、これを翻訳しなければなりません。異なる言語に持っていくと、まずここでエラーが発生します。そこで、日本語から出発する場合には、日本語のシステムにおいて、まずこのリソース・ファイルをテキスト・ファイルとして開き、日本語の部分すべてを仮のコードに翻訳します。現在ではインドネシア語版があり、これはアルファベットのみなので、これから出発するのが楽でしょう。

次に各国の開発環境の上で、ボタンを始め各種のコントロールについて、サイズや位置を調整します。これは、ボタン名称など、使用する言語によって実際に表示される文字列の長さが変わるからです。インドネシア語は長いので、大概の言語は、より小さくすることで解決すると思います。注意しなければならないのは、コンボボックスの初期表示メニューのように、リソース・ファイルの中で16進コードで漢字文字列が表現されている部分で、エディターで作業していて見落としやすい部分です。

リソース・ファイルの最初の方にある、言語の指定やフォントの指定の部分については、開発環境によっては、メニューで変更しても反映されない場合がある（開発環境のバグ）ので、テキスト表示したリソース・ファイルで一応確認しておいた方が良いでしょう。

②漢字ヘッダー・ファイルの翻訳

kj_sjis.hに、プログラム中で使用される文字列が #define 文の集合体として集約されています。従って、これを目的言語に翻訳したものを作成し、これをincludeするように変更すれば、基本的にはソースコード中の言語依存部分の翻訳作業は完了します。

③ERR_MSG.txt の翻訳

エラーメッセージは、ソースコード中にはなく、このファイルの中に記述されており、各種プログラムが起動した時点で読み込まれます。そこで、これを翻訳すれば、異なる言語でエラーメッセージを表示することができます。

④ヘルプ・ファイルの翻訳

ヘルプファイルは、全てテキスト形式(*.txt)です。これを翻訳すれば、ヘルプが、目的言語で表示されるようになります。

⑤Autotex.set の翻訳

テクスチャ自動貼付けに用いるこのファイルの中にも日本語があります。sim.exeが起動後に読み込まれるファイルなので翻訳しておきます。

⑥外部関数のダイアログの翻訳

sim.exeからユーザー定義によるパラメトリック部品を作成する際のダイアログ部分にも、日本語が含まれています。ソースコードは別なので、同様に翻訳します。

⑦ライブラリのヘッダーファイルの中にある日本語部分を翻訳します。

dbms.h

⑧景観データベースの翻訳

3種類のデータベースについても、上記と同様にリソース・ファイルを翻訳します。

起動後に読み込まれるものとして、.cls ファイル（プルダウン・メニューを定義）、及びcom.txt ファイルがあります。

⑨サンプル・データの翻訳

LSS・G、LSS・S形式のデータのうち、別のファイルを引用しているもので、引用しているファイル名に日本語が用いられている場合、他の言語のシステム上でうまく開かない場合があります。障害があれば、適宜アルファベットのファイル名称等に置換えます。対応する部分の対応が保たれるように注意して下さい。

⑩その他

ハンゲル化の作業は、開発環境が操作できるプログラマ、日本語版景観シミュレータの操作を一通り理解している者、及び数名の翻訳担当者（電算の知識は必ずしも必要でないが、出現する専門用語は理解できる必要がある）、デバッグ等の経験を豊富にもつベテラン等をうまく組み合わせ、Ver.2.03の時は8名で3日間、Ver.2.05では、4名で3日間程度で翻訳・移植することができました。

しかし、一通りの翻訳・移植作業を行い、ビルドを行ったシステムを起動した時に、正常に動作しない場合、あとどれだけかかるか先が読めなくなります。コンパイラのバージョンの違い（ハンゲルでは英語版がベースだが、日本では日本語化された開発環境が普及している）なども影響しています。現在までに遭遇した、かなり困難な問題は、以下のようなものでした。

- ・座標変換等の実数計算の結果が異なる

→ダミーのステップを加えて回避

- ・GUIで修正を行ったリソースが、ソースをテキストとして打出して見ると、正しく修正されていない (リソース・エディタのバグ)

→テキスト・エディタでソースを直接編集

- ・リンクの設定を同じにしても、リンク・エラーが発生する

→一部のライブラリを、デバッグ・モードのままとすることで漸くリンクが通る

このように、単に翻訳・移植とは言っても、ありえない筈の事まで疑ってかからねば、通らない事もあるのは自然言語の場合と同様です。