【土木計画学研究・講演集 No.11 1988年11月】

コンピュータ・グラフィックスを用いた景観シミュレーション・システム（ＬＡＮＳＩＳ）の開発

Development of 'LANSIS', the Landscape Simulation System Using Three Dimensional Computer Graphics

榊原和彦*

By Kazuhiko SAKAKIBARA

This paper outlines LANSIS (the LANdscape SImulation System) developed by the author and others for the use of planners and designers to represent designed landscapes. LANSIS is a total system composed of four subsystems as follows: (1) 3-D rendering system, (2) database management system, (3) modeling system, (4) 2-D rendering system. The 3-D rendering system, adopting the LUMINOUS system that is a 3-D computer graphics software on the market, has been improved so that it can give reaistic representation by means of texture mapping, bump mapping, overlaying actual images onto a computer generated one, generating reflected images, and so on. Other systems were developed to enhance availability of LANSIS that should be fit for various use.

1．はじめに

構造物の建設，自然の改変，都市再開発などに際して，景観アセスメントを実施するのが一般的になってきている．また，より積極的に，新しい景観の構築を目指した「景観設計」を行うことも多い．いずれの場合にも，なんらかの形で，出来上がるであろう景観の「予測」を行うことが必要である．そのためには，端的に，出来上がるであろう景観そのものをなんらかの媒体を介してさし示す，すなわち，「景観シミュレーション」を行うのがよく，表現手段としてコンピュータ・グラフィックスを用いるのが妥当・有効であることも明かである．

こういった背景のもとに，筆者は，文部省の私学研究設備助成金によって導入した，エンジニアリングワークステションのユーステーション／Ｅ２０，フルカラー・グラフィック装置のグラフィカ／Ｍ１０４８，および，スキャンライン・アルゴリズム三次元ＣＧソフトウェア・パッケージ（LUMINOUS）を主要なハード，ソフトウェア資源とする総合的な景観シミュレーション・システムであるＬＡＮＳＩＳ（Landscape Simulation System）の構築を目指している．本論はその内容を紹介するものである．

＊ 正会員 工博 大阪産業大学教授 工学部土木工学科
（〒574 大東市中垣内３－１－１）

2．システムの基本的考え方

（1） ＬＡＮＳＩＳの目的と課題

ＬＡＮＳＩＳは，構造物景観，都市景観，地域・自然景観などの景観シミュレーションに用い，その結果を①計画（設計）者の計画（設計）プロセスの中における判断・評価，②意志決定者・関係者・市民への計画（設計）結果の提示・伝達，③評価実験や意志集約，などに使用するものである．それに耐え得るシステムとするためには，以下のような課題を解決する必要がある．

a) 写実的な表現　景観シミュレーションのアウトプットに必要な表現性の水準は，それを眺め，鑑賞し，利用する人々の要求によって異なるものではあるが，現実的な状況を想起するに十分なだけの写実性を有することが必要である．しかしながら，LUMINOUSは，ＣＧ一般が不得意とする，人，樹木，地形等の自然物の表現の問題を除いても，写実的表現という点で，(1)質感表現の手段を持たない，(2)鏡面反射光を処理できない，(3)（半）透明物体の表現が不可能，(4)平行光線（太陽光），点光源の処理しかできず，微妙な陰・影の表現ができない，など不満足なところが多い．そこで，自然物表現の問題も含めて，種々の手段を講じ，解決をはかる．また，これに伴って必要なLUMINOUSの再構成を行う．

b) 画面合成　写実的な表現の問題とも関わるが，景観シミュレーションのためには，三次元（３Ｄ）グラフィックスだけが行えれば事足りるわけではない．背景のモンタージュや二次元（２Ｄ）画面でのコラージュなど，３Ｄ，２Ｄの両方で画像合成が可能とならねばならない．

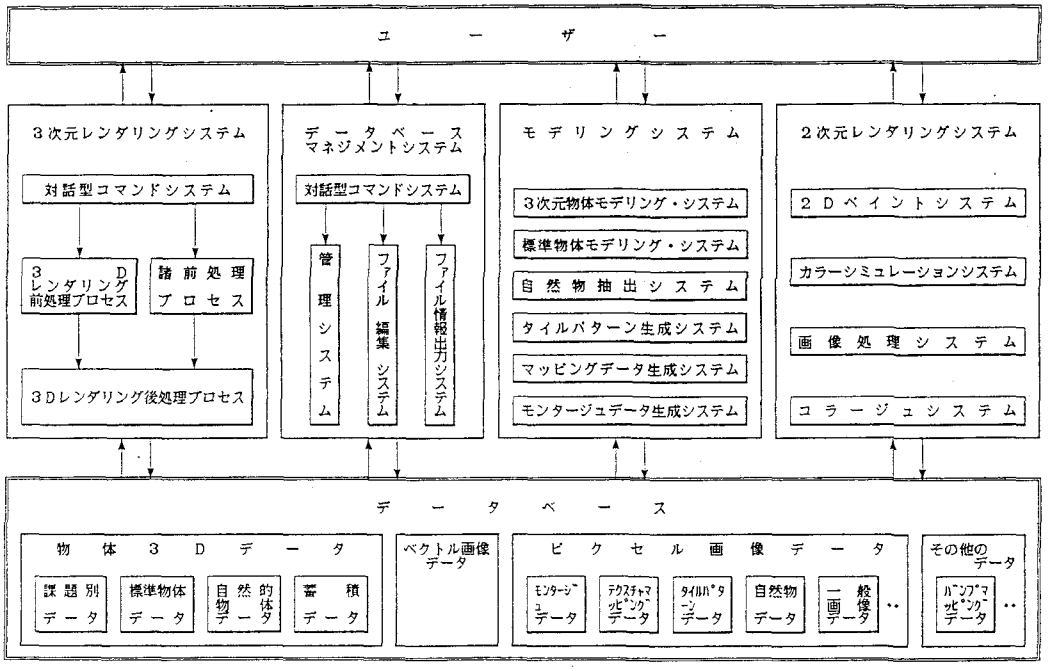
c) データベース・システム　システムを利用する過程で，数多くの，多種多様なデータを利用する．それらのデータには，シミュレーションの課題毎につくるものもあれば，既存のものを利用することもある．あるいは，将来の利用を想定して予め用意しておくべきものもある．このためにデータベースを構築し，その管理・運用のためにデータベース・マネジメント・システムをつくることは不可欠である．

d) モデリング（物体・画像データ生成）・システム　上記c)で述べた多くのデータは，いずれにせよ何らかの手段で作成，生成，抽出しなければならない．これをモデリングと総称するが，モデリングを効率的に行うことがシステムのavailabilityを高める．したがって，種々のデータの各々についてモデリング・システムを考える必要がある．

e) 二次元レンダリング　二次元での画像表現および画像操作は，上記b)で述べた場合の外，３Ｄ画像の操作，構成要素のカラー・シミュレーション，データ画像のプロセシング等々，多くの場合に必要となる．それらの各々に対応できるようにしておかなければならない．

（2） システムの基本的構成

上記(1)の課題を考慮して構成したソフトウェア・システムの基本的構成を図－１に示す．システム



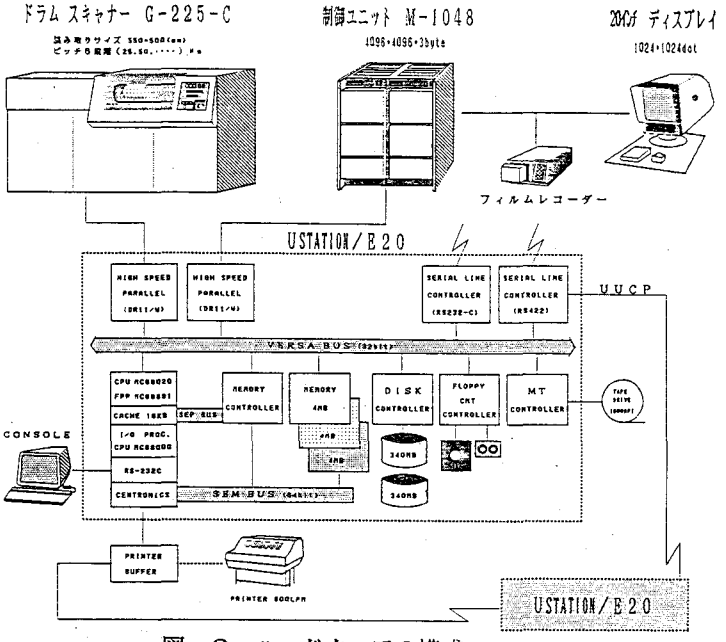
図－１　ＬＡＮＳＩＳの基本的構成

は，４つの主要部分からなり，各々は独立しているが，データベースを共有し，それぞれに操作する．ハードウェア・システムの構成は図－２に示す．

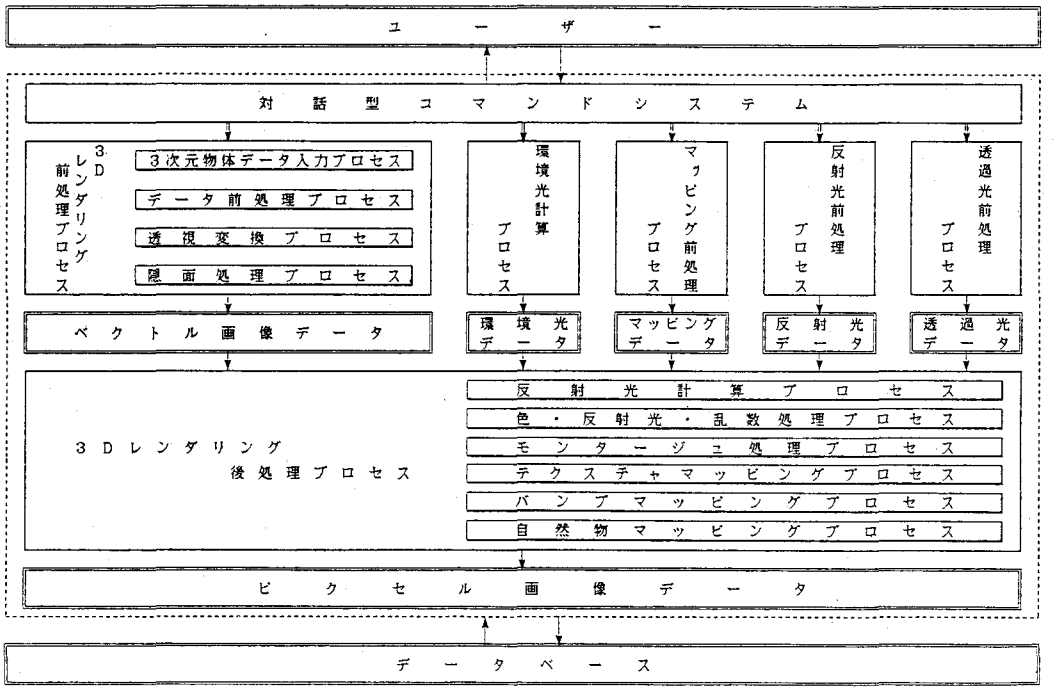
３．ＬＡＮＳＩＳの３Ｄレンダリング・システム

（１） システムの概要

システムの概要を図－３に示す．大別して，(1)３Ｄレンダリング前処理プロセス，(2)３Ｄレンダリング後処理プロセス，(3)その他の処理プロセス，の３つの部分から成る．前２者がシステムの基幹的部分で，(1)では，３次元物体データを入力し，透視変換，隠面処理を行って，画像データをつくる．このデータは，図－４に示すようにスキャンライン毎の物体の位置データなどから成る．以後ベクトル画像情報と呼ぶことにする．元のluminousでは，(1)，(2)のプロセスの分離が判然としておらず，画像情報の出力も行っていなかったのをこのように改良したのである．３Ｄ画像は全てこのデータを基にしてつくられるので，こうすることにより，写実的表現のための処理など，種々の画像加工・処理が可能とな



図－２ ハードウェアの構成



図－３ 三次元レンダリング・システムの構成

った．

⑵は，ベクトル画像データおよび⑶でつくられる種々のデータに基づいて，ピクセル（画素）毎の輝度すなわちＲＧＢの値を算出し，出力する（このようなデータをピクセル画像データと呼ぶ）．

⑶は，後述するように，写実的表現のためのデータ処理・加工をおこなう前処理プロセスである．

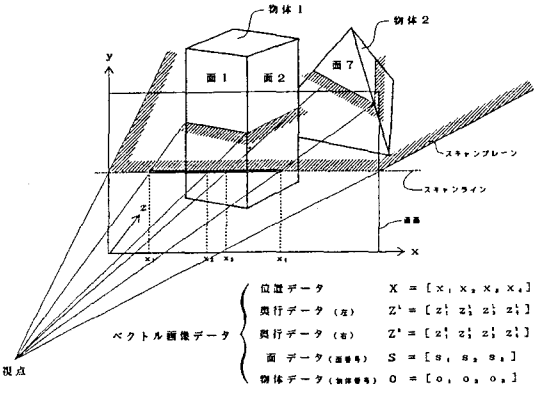
（2） 後処理プロセスのアルゴリズム

ピクセル画像データを求める際，アンチ・エリアシングを行う必要がある．そのために，ピクセル中の各図形の輝度の，面積を重みとする加重平均をピクセル輝度とする方法がある．このシステムでは，サブスキャンライン[1]（図－５参照）によって図形の面積を求め，輝度計算する．すなわち，ピクセルｉのサブスキャンラインｌ（ｌ=0,1,2,...,n）にかかわる輝度を$c_{i,l}$，サブスキャンラインと面ｆ（ｆ=1,2,...,m）の交差部分の長さを$s_{f,l}$，面ｆの輝度をI_fとすると，輝度c_iは次式になる．

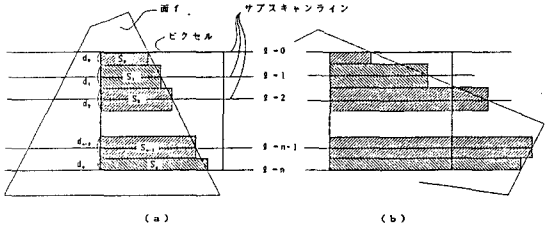
$$c_i=\sum_{l=0}^{n} c_{i,l}=\sum_{l=0}^{n}\sum_{f=1}^{m} d_l \cdot s_{f,l} \cdot I_f \quad (1)$$

ここで，d_l= 1/(2n)　（l=0,n のとき）

1/n　　（0<l<n のとき）



図－４　スキャンラインとベクトル画像データ



図－５　サブスキャンラインによる面積計算

この方法では，図－５,(b)のような場合には，正確な面積を計算出来ないが，計算時間も速く，ｎ＝３もとれば実用的には十分な精度をもつ．

面の輝度I_fは次式のように定める．

$$I_f=\left(I_c+\sum_{p=1}^{Q} I_{d,p}\cos\alpha_p\right) R_f \quad (2)$$

ここに，I_cは環境光の色（光の強さ，反射光と考えてもよい，一般には，物体の種類や場所に関わらず一定で，かつ，同じ値のＲ，Ｇ，Ｂ成分で与える），$I_{d,p}$は，光源ｐ（p=1,2,...,Q，点光源と平行光源に限る）の面ｆにおける色（光の強さ，反射光，Ｒ，Ｇ，Ｂ成分で与える），α_pは光源ｐの方向ベクトルと面ｆの法線ベクトルとのなす角である．また，R_fは面ｆの色（あるいは反射率，Ｒ，Ｇ，Ｂ成分で与える）である．通常は，太陽光（平行光源）の色$I_{d,1}$を，$I_c+I_{d,1}=1$となるように与えるので，R_fは，太陽光が面に直角にあたった時の面の輝度となり，光のコントラストは，I_cと$I_{d,1}$の比で定まる．

モンタージュ処理とは，予め画面にピクセル画像データが入っているとき，ベクトル画像データを用いて，ピクセル輝度を再計算するような処理である．したがって，既にあるデータをc_i^0とすると，$c_{i,1}$は，次式のように計算する．

$$c_{i,1}=c_i^0 \cdot d_1\left(1-\sum_{f=1}^{m} s_{f,1}\right)+\sum_{f=1}^{m} d_l \cdot s_{f,1} \cdot I_f \quad (3)$$

（3） 質感表現の方法

３Ｄレンダリングによって描かれる物体の質感を表現するために，まず，テクスチャ・マッピングを可能にした．これには，いくつかの考え方，計算方法があるが，本システムでは，次のようにした．

①テクスチャ・データは，通常使われているような離散的にサンプリングされた配列データではなく，ピクセル画像データの形のものとした．データの収集・生成の便（現実にあるテクスチャは写真やスライドを介してドラムスキャナで読み取るが，ピクセル画像データの形でデータが得られる）や計算時の精度などを考慮したためである．

②これを，通常行われているようなピクセル毎の変換ではなく，図－６,(a)のように，マッピングする面に対応するピクセルの全てに関して一括変換して

色合成し，記憶しておく（ここまでがマッピング前処理プロセス）．これは，画面上の面の輝度を予めピクセル毎に求めておくことに等しく，レンダリング後処理プロセスにおいてマッピングするには，(2)式のRf をピクセルｉ毎に与えるようにすればよい．③変換は，図－６,(b)のように，行毎に２段階で行う．すなわち，まず，列方向の色合成を，行方向のピクセルの全てに渡って行い（図－６,(b)の(3)のピクセルｊの色は，(2)のj_1，j_2 の部分の色の，各々の面積を重みとする加重平均である），かつ，変換前後のピクセルの四隅の位置を把握しておく．④次に，行方向の色合成により，ピクセルの色を求める（図－６,(b)において，(1)のピクセルｉの色は，(3)のi_1，i_2 の部分の色の各々の面積を重みとする加重平均である）．

以上の方法により，精密に，しかも，比較的短時間にテクスチャ・データ変換（前処理）を行うことが可能となり，かつ，後処理プロセスの最小限の変更で，簡単にマッピング処理を行えるようになった．

なお，バンプ・マッピングも上記と同じ考え方で，出来る限り共通のルーチンを使って行うようにした．

面の色R_f や光の色（強さ）Ｉを画素毎に乱数によって変化させるのも質感を表現する上で効果が高い．今，面の色R_f のＲ，Ｇ，Ｂの各々に同じ値ｅを加えてやると，ＨＬＳカラーモデル[2]を用いた計算によれば，色は次のように変わる．

①色相は変わらない．
②ｅ＜0 のとき，明度は下がり，彩度は上がる．
③ｅ＞0 のとき，明度は上がり，彩度は下がる．

したがって，面の色の明度，彩度が何らかの理由で微妙に変化している時には，ピクセル毎に正規乱数などによりｅを求めて，R_f に加えてやればよい．

また，Ｒ，Ｇ，Ｂに正の値ｇを乗じてやると，色は次のようになる．

①色相は変わらない．
②ｇ＜1 のとき，明度は下がり，彩度は，明度が0.5以下のとき変わらず，0.5以上のとき下がる．
②ｇ＞1 のとき，明度は上がり，彩度は，明度が0.5以下のとき変わらず，0.5以上のとき上がる．

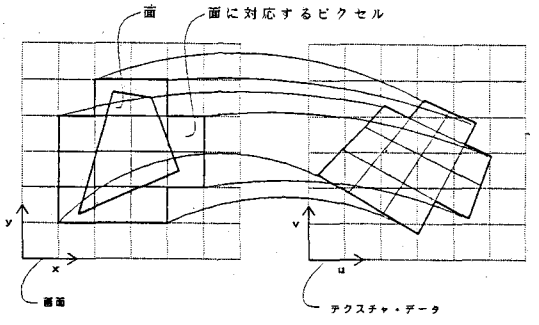
そこで，面が微細に汚れているようなことを想定する場合は，0＜ｇ＜1 になる乱数ををRf に乗じてやるとよい．面に微細な凹凸があるような場合には，面と光のなす角度が一様でなくなり，反射光の強さが変わる．そこで，(2) 式の$\sum_{D=1}^{Q} I_{d,D}\infty\alpha$。に，正の乱数を乗じてやればよい．この他に，３原色の内の特定のものだけを変えても面白い効果が生まれる．

ところで，より遠くに面があれば，テクスチャ密度は小さくなり，色の変化の巾（分散）は，視点からの距離が遠くなる程小さくなる．そこで，本システムでは，視点から距離Ｈ（基準距離と呼ぶ）のところにある面の色や光が，平均 0，標準偏差σの正規分布にしたがって増減するとき，Ｌの距離にある面の標準偏差は，$\sigma_L = \sigma\exp(1-L/H)$ となるものとした．

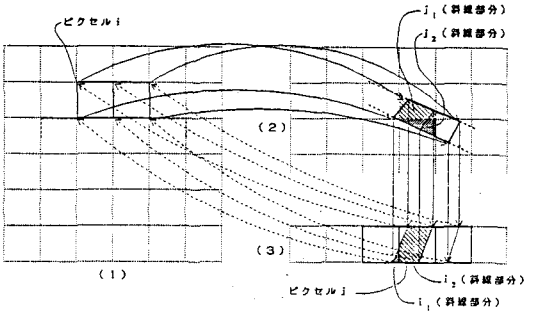
この方法は，マッピングに比べて計算時間もはるかに速く，データを持つ必要もない割には，効果的である．

（４） 自然物表現の方法

樹木等の自然物は，物体を三次元モデルで表現することがそれほど簡単ではなく，また，リアルなものが出来たとしても，物体数が膨大になる．したが



（ａ） ピクセルとテクスチャ・データの対応



（ｂ） １行分のピクセルデータの計算過程

図－６ テクスチャ・マッピング

って，記憶容量や計算速度を考えれば，３Ｄモデルにもとづいて計算により描画するのは，実用的ではないと考えられる．そこで，自然物の写真からドラムスキャナにより，ピクセル画像データをとり，これを画面にマッピングすることにした．マッピング用データは，予め背景をマスキングし，必要な部分のデータだけを用いる．データ抽出についてはモデリングのところで詳述するが，データのフォーマットは，不要な（マスクされた）部分も含む二次元配列で，ただし無駄の無いように，そのサイズは，マッピング対象物（すなわちデータとして必要な部分）の各次元の最大サイズに等しくしておく（これをマッピング平面と呼ぶ）．マッピングは，次のように行う．

①視点・注視点データ，マッピング平面の位置・大きさのデータから，マッピング平面は画面に対して常に平行であるものと仮定して，それの画面内での行（ｘ）方向，列（ｙ）方向の範囲，奥行きをピクセル単位で求める．

②マッピング平面と画面上の画像とはテクスチャ・マッピングの場合と同じように対応するものとし，やはり同様の手順で，上から順に画面ピクセルの１行毎（ただし，①で求めた範囲内で）に，ピクセル輝度を求める（ピクセルｉの輝度をc_i^Mとする）．

③このとき，画面ピクセルに対応するデータ上のピクセルの全体の中に占めるマスクされた部分の面積の割合q_iを求めておく．

④もとの画面のピクセル輝度をc_i°，とすると，マッピング後の輝度をc_iは，

$$c_i = c_i^\circ \cdot q_i + c_i^M (1 - q_i) \qquad (4)$$

となる．

⑤他の物体に隠される部分は，ピクセル毎に，マッピング平面の奥行きとベクトル画像情報の奥行きを比べて判定する．

⑥影は，マッピング平面を，水平面に対して直角，かつ，マッピング平面の中心部から光源に向かう方向ベクトルの水平面への投影ベクトルに直角になるように回転してから求める．

この方法により，比較的簡単に，リアリティの高い自然物の表現が出来るようになった．ただし，現時点では，水平面に対して直立する物体を，通常の視線（たとえば同一平面上にいる人の視線）から見たときのみ自然に見えるような画像しかつくれず，鳥瞰的な視線などからでは，不自然な形になってしまい，使うことが出来ない．

（５） 鏡面反射表現の方法

鏡面反射を表現するには，先に当該の面（平面である）にあたる反射光を求め，それをその平面にマッピングしてやればよい．そのためには，以下のようにいくつかの方法が考えられる．

① 当該の平面に面対称となる視点・注視点，平面の傾きに応じた視線回転角を求め，それによって透視画像を描く．これを左右逆転したものが全反射光となるので，反射率を乗じて実際の反射光を算出し，平面の元の輝度と併せてマッピングする．

② レイトレーシングにより必要な部分の反射・透過光を同時に求め，マッピングする．

③ 下記(７)の計算結果を利用し，画素の大きさを考慮した，一回の反射のみを追跡するレイトレーシングに等しい計算を行い反射光を求めてマッピングする．

本システムでは，①の方法をとっている．計算時間が速く，既存ルーチンの多くを利用できるからである．なお，マッピングは，下記のように２通りのやり方をとる．

a)全反射光データとしてピクセル画像データを用いる場合　一旦ピクセル画像を描いてしまい，記憶した上でマッピングする．ピクセル毎の輝度がデータであるので後処理プロセスにおける計算が簡単であり，特に，テクスチャ・マッピングしながら鏡面反射光も求めるような場合によい．また，反射光データ自体に自然物やテクスチャのマッピングをしておくことが出来るので便利である．

b)全反射光データとしてベクトル画像データを用いる場合　直接にベクトル画像データから反射光計算する．計算時間が速くなる．

（６） 透過光表現の方法

透過光は当該物体がないものとして画像を求め，透過率を乗じてマッピングする．ただし，この方法では，屈折が考慮できない．

（７） 環境光による陰影表現の方法

陰影の微妙な変化の表現のためには，天空光を含む環境光による照度の計算を行う必要がある．その手法については，参考文献5)に詳しいのでここでは

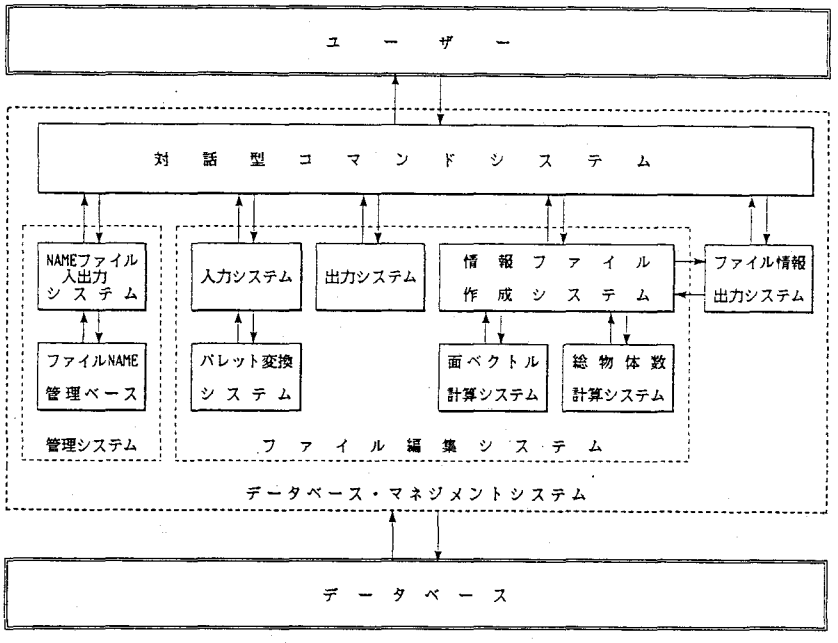
述べないが，それと同様の基本的考え方をとっている．ただし，アルゴリズムは，上記とは全く異なる．すなわち，天空ドームを，底面への投影面積が等しくなるような大きさのセルをもつメッシュに分割し，これを，各セルが１つの画素となっているような平面の画面と見立てて，透視画像を描く．この方法には次のような利点がある．

①計算が簡略．

②参考文献5)と全く同様の計算が可能．

③必要であれば物体別の，影も考慮した輝度，反射率を勘案することが可能．

④　前述の反射光の計算に結果を利用可能．



図－７　データベース・マネジメント・システム

４．ＬＡＮＳＩＳのサブシステム

（１）　データベース・マネジメント・システム

データをその様態によって区分すると，(1)物体３Ｄデータ，(2)ビクセル画像データ，(3)３Ｄレンダリング・システムの計算過程で出来るベクトル画像データ，(4)その他のデータ，になる．現時点でのデータベース・マネジメント・システムは，この内の(1)だけを取り扱うものであるが，このシステムの概要を図－７に示す．

このシステムの主要部分は，ファイル編集システムである．上記(1)のデータには，(1)課題別のデータ，(2)標準的物体のデータ，(3)蓄積された物体のデータ（これまでの課題でつくられたものなど），(4)自然的物体のデータ（地形情報など），があるが，これらは，たとえば建物毎といった小さな単位で１つのファイルになっている．ファイル編集システムは，これらからシミュレーション課題に必要なものを取り出してまとめあげ，くいちがいのある情報を調整し，必要な情報を付加して，一つのファイルとして

表－１　自動生成する標準物体

No.	種別				No.	標準物体
1	ストリートファニチャ				1	ベンチタイプⅠ
					2	ベンチタイプⅡ
					3	灰皿
					4	電話ＢＯＸ
					5	ポスト
					6	ゴミ箱タイプⅠ
					7	ゴミ箱タイプⅡ
					8	道路照明灯
2	道路にペイントされた標識				1	直進指示
					2	右折指示
					3	直進及び右折指示
					4	左折指示
					5	直進及び左折指示
					6	Ｕターン禁止指示
					7	交差点注意指示
					8	２０ｋｍ／ｈ制限指示
					9	３０ｋｍ／ｈ制限指示
					10	４０ｋｍ／ｈ制限指示
					11	５０ｋｍ／ｈ制限指示
					12	横断歩道
3	道路交通標識	看板の種類	1	円形	1	車両進入禁止
					2	車両通行止め
					3	通行止め
					4	駐車禁止
					5	駐停車禁止
					6	２０ｋｍ／ｈ制限
					7	３０ｋｍ／ｈ制限
					8	４０ｋｍ／ｈ制限
					9	５０ｋｍ／ｈ制限
	ボールの種類は以下の３種を選択できる。 1．まっすぐのポール 2．上部が曲がり張り出したポール 3．電柱に付設したポール		2	逆三角形	1	一時停止
			3	四角形警戒標識	1	危険あり
					2	十字形道路交差点あり
					3	ト字形道路交差点あり
					4	Ｔ字形道路交差点あり
			4	四角形規制標識	1	一方通行左
					2	一方通行右
			5	五角形	1	横断歩道

出力する．３Ｄレンダリング・システムは，このデータによって画像を生成する．

このようなシステムにつくることによって，たとえば，一度に入力できる物体個数に制限がある（現時点で，物体数にして 300，面数にして 1,800）３Ｄレンダリング・システムを効率よく使いこなすことが可能になり，あるいは，複数の人間で混乱なく効率的に課題を処理したり出来るようになった．

（2） モデリング・システム

a） 標準的物体モデリング・システム　標準的物体として，表－１にあるものの３Ｄ物体データを，寸法指定だけすれば出力するシステムである．

b） 自然物データ抽出システム　ドラムスキャナで読み取ったピクセル画像データから，必要な部分だけを抽出するシステムである．その方法は，以下の通りである．

①画像中に，ある領域を指定し，さらにその中に小領域を定める．小領域は，データとして必要で残すべき部分か，あるいは反対に，除外（マスク）するべき部分とする．

②小領域中の全ての色の変動範囲を，色相，明度，彩度の別に，ＨＬＳカラーモデルによって計算する．

③小領域の色の範囲に当てはまるピクセルを指定領域中でさがし，残すか，除外する．

④以上を，全てのピクセルについて残すべきか除外すべきか定まるまで繰り返す．そして，除外すべきピクセルにはＲ，Ｇ，Ｂの全てに０の値を入れておく．精密に行うときは，必要な画像の２倍（面積比４倍）の大きさのもので行い，圧縮しておく．

c） タイルパターン生成システム　以下の２種類をつくった．

①通し目地パターンの正方形タイルに，色指定を行ってタイルパターンを発生するもの（システムＡ）．タイル内に幾種類かのテクスチャを発生できる．

②タイル内にテクスチャ設定しないもの（システムＢ）．配置パターンは現状で７種類指定できる．基準色と変化の巾または特定の数種類の色を指定して乱数によりタイルの色を定める．

d） その他のモデリング・システム　課題毎の３Ｄ物体のモデリング・システムは，必要なものであるが，現在のところ開発していない．他の適当なＣＡＤシステムを，データのフォーマット変換を行えるようにして使うことも考えられる．マッピング・データ，モンタージュ・データについては，ドラム・スキャナ・ハンドリングシステムで操作する．

（3） ２Ｄレンダリング・システム

既製の２Ｄペイント・システム（グラフィカ社製）を主要なシステムとしている．コラージュにも主としてこれを用いる．カラー・シミュレーション・システムでは，構成要素のマスキングには自然物データ抽出のルーチンを使い，必要な機能を付加している．画像処理システムは，現状ではドラムスキャナからの読み取りデータの色調整機能などを有する．

５．おわりに

以上，景観シミュレーション・システム，ＬＡＮＳＩＳの概要を述べた．システムは開発途上であり，かなりの程度まで実用に耐え得るものであるとはいえ，未だ十全なものではない．たとえば，写実的表現という点でも，曲面の平滑シェーディング，鏡面反射（ハイライト），霧など天候条件の取り込み，等々付け加えるべき機能は多い．また，テクスチャ・マッピング，自然物マッピングなどでも改良の余地がある．他のサブシステムについてもまだまだなすべきことは多い．これらを今後の課題として，個々のグラフィック技術に拘泥するのではなく，総合的に見て使いやすい，実用的な，優れた景観シミュレーション・システムの構築を目指したい．

＜ 参 考 文 献 ＞

1） 中前・西田：「３次元コンピュータグラフィックス」，昭晃堂，1986.

2） 町田正彦：「コンピュータイメージング」，pp110～122，コロナ社，1984.

3） 榊原和彦：「コンピュータ・グラフィックスによる景観シミュレーションにおける写実的表現に関する研究」，土木学会第42回年次学術講演会講演概要集，1987.

4） 石崎貴夫：「エリアシングの除去を考慮した３次元物体の合成法」，広島大学修士論文，1987.

5） 中前・西田：「天空光を考慮した建造物の表現手法」，日経ＣＧ，No.2，PP.102～112，1986.

6） 山田　学：「景観シミュレーション」，都市計画，No.138，PP.40～45,1985.