

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 6

1994

УДК 681.3.06

И. В. Белого, Ю. Ю. Некрасов, А. В. Романовский, Ю. В. Тарасов
(Новосибирск)

ТИПЫ ДАННЫХ И ОПЕРАЦИИ
ДЛЯ ОПИСАНИЯ ВИЗУАЛЬНЫХ МОДЕЛЕЙ ТРЕХМЕРНЫХ СЦЕН
И ИХ ДИНАМИКИ РЕАЛЬНОГО ВРЕМЕНИ

Обсуждаются типы данных и операции для описания достаточно широкого класса визуальных моделей трехмерных сцен и их динамики реального времени. Набор типов и операций отображает специфику описываемых визуальных моделей, которые используются в авиационных и космических тренажерах. Вместе с тем эти средства могут успешно применяться для получения визуальных моделей в таких областях, как производство компьютерной видеорекламы, машиностроительное проектирование и 3-мерное моделирование и т. д. Изложение ведется в контексте специализированного языка программирования L-, однако предлагаемые типы и операции могут быть легко запрограммированы на любом языке общего назначения, например, в рамках какой-либо системы интерактивного визуального моделирования.

Введение. Целью разработки предлагаемого набора типов данных и операций было обобщение и развитие идей в [1, 2] и объединение в пределах одноязыкового подхода средств для описания геометрии 3-мерных объектов, их визуальных свойств, свойств среды моделирования, регулярной динамики сцены и статической структуры 3-мерной сцены, оптимизируемой для ее отображения в реальном времени. Типы и операции реализованы в рамках языка смешанных вычислений L- [3], который позволяет программировать модели 3-мерных сцен и их динамики для последующей визуализации в реальном времени. Несмотря на специфику языка, который является частью комплекса трехмерного визуального моделирования SoftLab Images (версия 2.3) [4], обсуждаемые далее средства могут быть легко запрограммированы на любом современном языке общего назначения, как, например, показано в [5].

Набор типов данных включает:

- скалярные типы Integer (целый), Real (вещественный);
- типы-массивы Vector (вектор в 3-мерном пространстве), VectArr (массив векторов), Vertices (массив вершин граней);
- типы-записи Color (цвет), DotVis (визуальные характеристики набора точек), FaceVis (визуальные характеристики граней), Feature (особенность местности), Fog (туман), Lamp (источник освещения), Map (текстурная карта), Plane (плоскость в 3-мерном пространстве), Reref (система координат), Sphere (сфера в 3-мерном пространстве), Terrain (местность), Texture (текстурные характеристики), Vertex (вершина грани), Viewer (наблюдатель);
- типы со скрытой структурой объектов Dots (набор точек), Faces (набор граней), Matrix (матрица преобразования координат в 3-мерном пространстве).

Под операциями в данной статье понимаются как непосредственные операции (например, арифметические), так и процедуры и функции.

Скалярные типы данных. К объектам скалярных типов применимы традиционные арифметические операции, операции сравнения, тригонометрические и другие стандартные математические функции. Если операнды двух-

местной операции имеют целый и вещественный типы, то целый операнд автоматически преобразуется в вещественный тип.

Типы-массивы. Объекты данных типа `Vector` представляют собой массивы из трех элементов типа `Real` и используются для задания векторов (точек) в 3-мерном пространстве. К объектам данных этого типа применимы традиционные операции векторной арифметики: "+" (сложение векторов), "-" (вычитание векторов), "*" (скалярное умножение), "**" (векторное умножение), а также векторно-скалярные операции "*" (умножение вектора на скаляр), "/" (деление вектора на скаляр).

Объекты данных типа `VectArr` представляют собой массивы переменной длины элементов типа `Vector` и используются для конструирования наборов точек и массивов вершин (см. далее тип `Vertices`). К объектам данных типа `VectArr` применима операция "&" (конкатенация массивов), одноместная операция `inv` (обращение массива), функции `bezier` и `qSpline` (построение аппроксимирующей ломаной по методу Безье). К объектам данных типа `VectArr` и `Vector` применима операция "+" (добавление вектора в конец массива).

Объекты данных типа `Vertices` представляют собой массивы переменной длины элементов типа `Vertex` (см. типы-записи) и используются для задания геометрических и визуальных характеристик граней. К объектам данных типа `Vertices` применимы операции, аналогичные операциям для `VectArr`.

Типы-записи. Для краткости и наглядности изложения структура типа-записи (набор полей) далее изображается в следующем формате:

```
(структура типа-записи) : :=  
"TYPE" (имя типа-записи) ":" "RECORD"  
"{" (имя типа поля) [":" ] (имя поля) {" (имя поля) " ;" "}"
```

В этом формате в кавычках изображаются терминальные символы, фигурные скобки означают повторение их содержимого 0 или более раз, а квадратные скобки — возможное отсутствие их содержимого. Наличие ":" после имени типа поля означает, что все последующие поля являются указателями на объекты данного типа.

```
TYPE Color : RECORD { Real red, green, blue; }
```

Объекты данных типа `Color` используются для задания цвета в терминах трехкомпонентной (RGB) модели цвета. Значение поля `red` задает уровень красной составляющей цвета, значение поля `green` — зеленой, значение поля `blue` — синей. Значения полей должны быть в интервале 0—1.

```
TYPE DotVis : RECORD {  
  Color : color;  
  Real radius, intensity, visibilityRange;  
  Vector direction;  
  Real apertureAngle, verticalApertureAngle;  
  Vector zenith;  
  Integer : switch;  
}
```

Объекты данных типа `DotVis` используются для задания визуальных характеристик точек. Поле `color` задает цвет точек, `radius` — радиус точек, `intensity` — интенсивность цвета точек, `visibilityRange` — дальность видимости точек. Поле `direction` вместе с координатами точек задает полупространства видимости точек, поле `apertureAngle` — угол конуса видимости точек в радианах, поле `verticalApertureAngle` — угол в вертикальной плоскости симметрии пирамиды видимости точек в радианах, поле `zenith` — вертикальную ось пирамиды видимости точек, поле `switch` — указатель на «выключатель» точек. Если в поле хранится `NIL` или значение указываемого `Integer`-объекта не равно 0, то точки, к которым относятся визуальные характеристики, отображаются в окне обзора наблюдателя при условии их видимости. Если значение указы-

ваемого Integer-объекта равно 0, то соответствующие точки не отображаются в окне обзора наблюдателя.

```
TYPE FaceVis : RECORD {  
  Color : directLightDiffusionColor, ambientLightDiffusionColor;  
  Real intensity;  
  Color : reflectionColor;  
  Real highLightSize, frontalOpacity, obliqueOpacity;  
  Integer : switch;  
}
```

Объекты данных типа FaceVis используются для задания визуальных характеристик граней. Поле directLightDiffusionColor задает цвет диффузно отражаемого гранью света от освещающих ее источников, Color : reflectionColor — цвет отражения в перпендикулярном плоскости грани и параллельном ей. Поле switch задает указатель на «выключатель» граней.

```
TYPE Feature : RECORD {  
  Reper : reper;  
  Sphere sphere;  
  Faces : implementation;  
  Feature : next;  
}
```

Объекты данных типа Feature используются для задания особенностей (набора объектов) местности, находящейся в моделируемой сцене. Объекты типа Feature связываются в списки. Поле reper задает систему координат, в которой находится данная особенность (набор объектов) местности, поле sphere — сферу, которая полностью или почти полностью содержит набор объектов местности, implementation — набор граней, представляющих данную особенность, next — NIL или указатель на следующий в списке объект типа Feature.

```
TYPE Fog : RECORD { Color : color; Real visibilityRange; }
```

Объекты данных типа Fog используются для задания визуальных характеристик среды, в которой находится моделируемая сцена. Поле color задает цвет среды, в которой располагается моделируемая сцена, visibilityRange — дальность видимости в среде моделирования.

```
TYPE Lamp : RECORD {  
  Vector direction;  
  Reper : reper;  
  Vector : position;  
  Color : color;  
  Real illuminationRange, apertureAngle;  
  Integer : switch;  
}
```

Объекты данных типа Lamp используются для задания невидимых источников освещения моделируемой сцены. Поле direction задает ориентацию источника освещения в его системе координат, reper — систему координат источника освещения, position — положение источника освещения в его системе координат, color — цвет источника освещения, illuminationRange — дальность свечения источника освещения в метрах, apertureAngle — угол конуса

свечения источника освещения в радианах. Поле `switch` задает указатель на «выключатель» источника освещения. Источник освещения включен и учитывается при расчете освещенности соответствующих граней, если в поле хранится `NIL` или значение указываемого объекта данных не равно 0.

```
TYPE Map : RECORD { String fileName; }
```

Объекты данных типа `Map` используются для задания текстурных карт граней. Поле `fileName` задает имя файла, в котором находятся данные, соответствующие одной или нескольким текстурным картам.

```
TYPE Plane : RECORD { Vector normal, point }
```

Объекты данных типа `Plane` используются для задания плоскостей в 3-мерном пространстве. Поле `normal` задает нормаль к плоскости — вектор длиной, равной 1, поле `point` — координаты одной из точек плоскости.

```
TYPE Reper : RECORD { Matrix locationInBasis; Reper : basis }
```

Объекты данных типа `Reper` используются для задания древовидных иерархий правосторонних ортонормированных систем (пространств) координат моделируемой сцены. Поле `locationInBasis` задает положение данной системы координат в базисной системе координат, поле `basis` — базисную систему координат для данной системы координат.

К объектам данных типа `Reper` и `Matrix` применимы процедуры `locateInBasis` (задание положения системы координат в базисной системе), `relocateInBasis` (задание перемещения системы координат относительно базисной системы), `relocateInItself` (задание перемещения системы координат относительно самой себя).

```
TYPE Sphere : RECORD { Vector center; Real radius }
```

Объекты данных типа `Sphere` используются для задания сфер в 3-мерном пространстве. Поле `center` задает координаты центра сферы, поле `radius` — радиус сферы.

```
TYPE Tile : RECORD {  
  Integer diagonal;  
  Color : color1, color2;  
  Feature : featureList;  
};
```

```
TYPE Heights : ARRAY [25] {Real};  
TYPE Terrain : RECORD {  
  Integer xTilesTotal, yTilesTotal, tilesToSee;  
  Matrix locationInWorld;  
  Real tileSideSize;  
  ARRAY [xTilesTotal * yTilesTotal] {Tile} tiles;  
  ARRAY [xTilesTotal * yTilesTotal] {Heights} heights;  
}
```

Объекты данных типа `Terrain` используются для задания поверхностей, с помощью которых в моделируемой сцене может представляться местность или что-то другое. Поверхность `Terrain` состоит из квадратных в проекции на плоскость `OXY` участков, каждый из которых задается квадратной регулярной решеткой и состоит из 16 ячеек. Узлы решетки, расстояния между которыми по осям `OX` и `OY` равны, соответствуют точкам на местности. Для каждой точки задается высота над плоскостью `OXY` в собственной системе координат. Ячейка участка представлена двумя треугольниками, вершинами которых являют-

ся точки в узлах квадратной регулярной решетки. Каждой ячейке поверхности могут соответствовать особенности местности — объекты данных типа Feature.

Поле `xTilesTotal` задает количество участков поверхности вдоль оси *OX*, поле `yTilesTotal` — вдоль оси *OY*, поле `tilesToSee` — дальность видимости в участках для наблюдателя, поле `locationInWorld` — матрицу преобразования собственной системы координат Terrain в мировую систему координат, поле `tileSideSize` — длину стороны квадратного участка местности, поле `tiles` — массив, элементами которого являются объекты данных типа Tile, соответствующие квадратным участкам поверхности Terrain. Поле `heights` представляет массив, элементами которого являются массивы высот для ячеек каждого участка поверхности Terrain.

К объекту данных типа Terrain применимы функции `bothVisible` (определение взаимной видимости двух точек над поверхностью Terrain), `height` (определение высоты от точки до поверхности Terrain), `intersection` (определение точки пересечения луча с поверхностью Terrain), процедуры `insertFeature` (связывание особенности поверхности Feature с поверхностью Terrain так, что при отображении поверхности особенность Feature отображается в окне обзора текущего наблюдателя в правильном приоритетном порядке относительно поверхности), `showTerrain` (отображение граней поверхности Terrain и особенности на ней в окне обзора текущего наблюдателя при условии их видимости).

```
TYPE Texture : RECORD {  
  Map : colorMap, opacityMap, bumpMap, intensityMap, mixMap;  
  Color : mixedColor;  
  Reper : mapReper; }
```

Объекты данных типа Texture используются для задания текстуры граней. Поле `colorMap` задает текстурную карту распределения цвета на поверхностях граней, поле `opacityMap` — текстурную карту распределения непрозрачности на поверхности грани, поле `bumpMap` — текстурную карту распределения нормалей на поверхности грани, поле `intensityMap` — текстурную карту распределения интенсивности цвета на поверхности грани, поле `mixMap` — текстурную карту смеси цветов на поверхностях граней, поле `mixedColor` дополняет поле `mixMap` и задает цвет, с которым смешивается цвет всего отражаемого гранью света в пропорциях, определяемых текстурной картой смеси цветов на поверхности грани и текстурными координатами вершин грани. Поле `mapReper` задает систему координат, в которой находится текстурная плоскость грани.

```
TYPE Vertex : RECORD {  
  Vector xyz, normal;  
  Color : color;  
  Real opacity, intensity, tx, ty;  
}
```

Объекты данных типа Vertex используются для задания геометрических и визуальных характеристик вершин граней. Поле `xyz` задает координаты вершины, поле `normal` — нормаль в вершине. Значение нормали используется при интерполяции интенсивности цвета (освещенности) вдоль поверхности грани в зависимости от положения вершины относительно источника освещения. Поле `color` содержит указатель на цвет поверхности грани, поле `opacity` — величину непрозрачности в вершине грани, поле `intensity` — коэффициент для интерполяции интенсивности цвета (освещенности) вдоль поверхности грани независимо от положения грани относительно источника освещения. Пара полей `tx`, `ty` задает текстурные координаты вершины грани на текстурной плоскости.

```
TYPE Viewer : RECORD {  
  Reper : reper;
```

```

Faces backFaces;
String device;
Real distanceToScreen, screenHeight, screenWidth,
    windowX, windowY,
    windowHeight, windowWidth;
}

```

Объекты данных типа *Viewer* используются при построении изображения моделируемой сцены на экране дисплея компьютера. Поле *device* задает систему координат наблюдателя. Считается, что наблюдатель находится в начале своей системы координат, смотрит вдоль направления оси *OY* (ось *OZ* расположена вертикально и направлена вверх, а ось *OX* — горизонтально и направлена вправо относительно наблюдателя). Поле *backFaces* задает набор граней, которые будут служить фоном при построении изображения в окне обзора для данного наблюдателя. Считается, что набор фоновых граней находится в собственной системе координат наблюдателя. Поле *device* задает имя устройства отображения, на экране которого изображается часть моделируемой сцены, видимой наблюдателем, поле *distanceToScreen* — расстояние в метрах от начала системы координат наблюдателя до экрана устройства отображения, поле *screenHeight* — высоту экрана устройства отображения, поле *screenWidth* — ширину экрана устройства отображения в метрах, поле *windowX* — горизонтальную координату верхнего левого угла окна обзора на экране устройства отображения, поле *windowY* — вертикальную координату верхнего левого угла окна обзора на экране устройства отображения, поле *windowHeight* — высоту виртуального окна обзора, через которое наблюдатель смотрит на моделируемую сцену, поле *windowWidth* — ширину виртуального окна обзора.

Отношение текущего наблюдателя к объектам моделируемой сцены устанавливается с помощью встроенных динамических функций *above* (определение положения начала системы координат текущего наблюдателя относительно плоскости), *distanceTo* (определение расстояния от начала системы координат текущего наблюдателя до точки), *inside* (определение положения начала системы координат текущего наблюдателя относительно сферы), *inView* (определение положения сферы относительно пирамиды видимости текущего наблюдателя), *viewerUsedNow* (тест на совпадение с текущим наблюдателем), *radiusLess* (определение отношения между размером проекции радиуса сферы на плоскость экрана и вещественным числом).

Типы со скрытой структурой объектов. Эти типы отличаются от остальных типов отсутствием непосредственных операций конструирования, раскрывающих структуру их объектов. Объекты данных типов со скрытой структурой могут быть получены в результате вызова встроенных функций-конструкторов.

Тип Dots. Объекты данных типа *Dots* задают наборы точек (огней) и могут быть созданы в результате вызова встроенных функций-конструкторов *chain* (создание линейной цепочки точек), *set* (создание неупорядоченного набора точек). К объектам данных этого типа применима операция "&" (объединение наборов), функция *sphereOf* (вычисление охватывающей набор сферы), процедуры *show* (отображение набора точек), *showRanged* (отображение набора точек с приоритетным упорядочиванием относительно граней сцены).

Тип Faces. Объекты данных этого типа задают наборы плоских выпукло-ориентированных граней. Каждая грань имеет не менее трех вершин, которые определяют плоскую выпуклую границу грани. Ориентация граней определяется так, что поверхность грани должна находиться все время слева относительно направления обхода границы грани в порядке перечисления вершин. Наборы граней используются для аппроксимации поверхностей моделируемой 3-мерной сцены.

Объекты данных типа *Faces* могут быть созданы в результате вызова функций-конструкторов *box* (шесть граней, параллельных координатным плоскостям), *circle* (полигональная аппроксимация круга), *convex* (выпуклая оболочка набора вершин), *polygon* (замощение гранями области на плоскости, ограниченной замкнутым контуром), *revolved* (полигональная аппроксима-

ция поверхности вращения), strip (замощение гранями полосы между двумя ломаными), tube (полигональное представление отрезка трубы). К объектам данных этого типа применимы операции "&" (конкатенация наборов граней), "&&" (конкатенация с «заглаживанием» ребер смежных граней при помощи нормалей), операции конструктивной геометрии для твердых тел "+" (объединение), "-" (вычитание), "*" (пересечение), "/" (отсечение), одноместные операции inv («выворачивание наизнанку»), smo («заглаживание» поверхности посредством расстановки нормалей в смежных вершинах граней), функции mapOnCylinder (расчет текстурных координат посредством отображения вершин граней параметра на текстурный цилиндр), mapOnPlane (расчет текстурных координат с отображением вершин граней параметра на текстурную плоскость), mapOnSphere (расчет текстурных координат посредством отображения вершин граней параметра на текстурную сферу), sphereOf (расчет сферы, содержащей грани параметра), процедуры show (отображение грани и точки в окне обзора текущего наблюдателя при условии их видимости), showRanged (приоритетное упорядочивание для произвольного положения наблюдателя и отображение грани и точки в окне обзора текущего наблюдателя при условии их видимости). К объектам данных типа Faces и Vector применима операция "%" (масштабирование граней первого операнда с использованием в качестве масштабных коэффициентов значений координат второго операнда). К объектам данных типа Faces и Plane применима операция mir (построение зеркального отражения граней первого операнда относительно плоскости, заданной вторым операндом).

Tun Matrix. Объекты данных типа Matrix задают преобразования координат в 3-мерном пространстве. Такие объекты могут быть созданы в результате вызова функций-конструкторов around (преобразование поворота вокруг произвольной оси), move (преобразование сдвига), XtoY (преобразование поворота вокруг оси Z в направлении от оси X к оси Y), YtoZ (преобразование поворота вокруг оси X в направлении от оси Y к оси Z), ZtoX (преобразование поворота вокруг оси Y в направлении от оси Z к оси X). К объектам данных типа Matrix применимы операция композиции двух преобразований координат "*", одноместная операция получения обратного значения inv. К объекту данных типа Matrix и объекту любого из типов: Dots, Faces, Plane, Sphere, Vector, VectArr, Vertex, Vertices — применима операция "*" (преобразование координат).

Заключение. Предлагаемый набор типов и операций отображает специфику описываемых визуальных моделей, которые используются в авиационных и космических тренажерах. Однако, как показывает практика, эти средства могут успешно использоваться для получения визуальных моделей в таких областях, как производство компьютерной видеорекламы, машиностроительное проектирование и 3-мерное моделирование и т. д. Представляется интересным реализация рассмотренных типов и операций в рамках системы интерактивного визуального моделирования.

СПИСОК ЛИТЕРАТУРЫ

1. Романовский А. В. VML — язык описания геометрических моделей трехмерных сцен для синтезирующих систем визуализации // Автометрия. — 1993. — № 5.
2. Белаго И. В., Романовский А. В., Тарасов Ю. В. SDL — язык описания визуальных моделей трехмерных сцен // Там же.
3. Романовский А. В. Язык смешанных вычислений для визуального моделирования трехмерных сцен и их динамики реального времени // Автометрия. — 1994. — № 6.
4. Белаго И. В., Некрасов Ю. Ю., Романовский А. В., Тарасов Ю. В. Комплекс трехмерного визуального моделирования SoftLab Images 1.1 // Автометрия. — 1993. — № 5.
5. Романовский А. В. Использование типов данных для визуального моделирования динамики трехмерных сцен // Автометрия. — 1994. — № 2.

Поступила в редакцию 26 мая 1994 г.