

УДК 004.021

В. Л. Колесников, доктор технических наук, профессор (БГТУ);
А. И. Бракович, кандидат технических наук, старший преподаватель (БГТУ)

3D-МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ ЗАГРЯЗНЕНИЯ ВЫДЕЛЕННЫХ ЗОН ТЕРРИТОРИИ

В статье приведено описание программного средства для 3D-моделирования и визуализации загрязнения выделенных зон территории, математической основой которого является стандартная методика ОНД-86 и элементы теории построения информационных сетей. Программный продукт позволяет проводить моделирование экологической обстановки выделенной территории, создавать информационную сеть для разработки системы локального мониторинга и проводить анализ полученных данных.

The article describes the software tools for 3D-modeling and visualization of pollution identified areas of the territory. The mathematical basis of this software is the standard method OND-86 and the elements of the theory of constructing information networks. The software product allows the simulation of environmental conditions on the selected areas, the creation of an information network for the development of local monitoring and make the analysis of the received data.

Введение. Локальный мониторинг выбросов загрязняющих веществ в атмосферный воздух стационарными источниками в Республике Беларусь осуществляется экологическими службами предприятий. Контрольные проверки мониторинговых данных проводятся аналитическими лабораториями территориальных органов Министерства природы.

Перечень контролируемых веществ, нормативы допустимых выбросов (ДВ) и периодичность наблюдений, а также место отбора проб определяются территориальными органами Министерства природы для каждого конкретного источника на предприятии с учетом специфики производства и предполагаемого уровня вредного воздействия на атмосферный воздух.

В городе Минске локальный мониторинг проводится на 14 предприятиях, на долю которых приходится около 70% суммарного выброса загрязняющих веществ от стационарных источников города. В течение года выполняется около 5200 определений контролируемых веществ на 157 источниках.

Не подвергая ревизии важность получаемой таким образом информации, нельзя не отметить ее фрагментарность, разбросанность по времени и пространству. Собранные табличные данные не позволяют корректно их обработать с получением математического описания и визуализации загрязнения территории в виде 3D-проекций поверхностей, которые наглядно и объективно давали бы возможность оценить экологическую ситуацию в городе для принятия оперативных управленческих решений.

Для этого нужно знать, какое необходимо и достаточное количество информации нужно иметь, сколько датчиков и как рационально их разместить по подведомственной территории, чтобы отслеживание загрязнения с заданной точностью и в объемном виде осуществлялось

сразу по всей выделенной зоне через определенные промежутки времени [1, 2].

С указанной целью выполнена публикуемая работа.

Основная часть. Процесс формирования изображения должен учитывать две главные сущности. Это визуализируемый объект сцены и наблюдатель. Объект существует в пространстве независимо от кого-либо. Наблюдатель же представляет собой средство формирования изображения наблюдаемых объектов. Именно наблюдатель формирует изображение. Наблюдатель и наблюдаемый объект существуют в одном и том же трехмерном мире, а создаваемое при этом изображение получается двухмерным. Суть процесса формирования изображения состоит в том, чтобы, зная положение наблюдателя и положение объекта, описать (синтезировать) получаемое при этом двухмерное изображение (проекцию).

В трехмерной графике пространственные объекты, заданные в непрерывном виде, как правило, аппроксимируют (приближают) множеством треугольников. Именно треугольник представляет собой элементарный примитив, с помощью которого описываются все элементы сцены, в том числе и те, которые имеют гладкую форму (сфера, цилиндр, параметрические поверхности и др.).

При этом моделируемая трехмерная сцена может описываться тысячами, а иногда миллионами вершин. Для каждой вершины (точки) примитива должны выполняться вычисления по одним и тем же формулам. Для осуществления преобразований с вершинами используют матричный подход. Таким образом, преобразование точки в пространстве сводится к умножению вектор-строки размерности на матрицу преобразования.

Проект создавался на базе MFC (Microsoft Foundation Classes). Такой выбор обоснован многими причинами. За основу архитектуры приложения был взят однодокументный графический

интерфейс Single document interface (SDI). В проекте применялись возможности Direct3D.

Для начала работы с библиотекой Direct3D подключаем заголовочные файлы и статистические библиотеки.

Далее объявляем в классе указатели на главный интерфейс IDirect3D9, интерфейс устройства и метод (InitD3D), в котором их проинициализируем:

```
class C3DSys
{protected:
LPDIRECT3D9 g_pD3D; // интерфейс
LPDIRECT3DDEVICE9 g_pd3dDevice; // устройство public:
HRESULT InitD3D( HWND hWnd );};

Инициализация будет выглядеть следующим образом:

HRESULT C3DSys::InitD3D( HWND hWnd )
{// основной интерфейс
if( NULL == ( g_pD3D = Direct3DCreate9
( D3D_SDK_VERSION ) ) )
return E_FAIL; }
// описание параметров устройства
D3DPRESENT_PARAMETERS d3dpp;
// заполняет указанную в качестве первого параметра структуру нулями
ZeroMemory( &d3dpp, sizeof(d3dpp) );
// вывод будет производиться в окно
d3dpp.Windowed = TRUE;
// задает режим работы механизма двойной буферизации
d3dpp.SwapEffect =
D3DSWAPEFFECT_DISCARD;
// указывает, какой формат буфера будет использоваться
d3dpp.BackBufferFormat = D3DFMT_UNKNOWN;
d3dpp.EnableAutoDepthStencil = true;
d3dpp.AutoDepthStencilFormat = D3DFMT_D16;
if( FAILED( g_pD3D->CreateDevice(
D3DADAPTER_DEFAULT, D3DDEVTYPE_HAL, hWnd,
D3DCREATE_SOFTWARE_VERTEXPROCESSING,
&d3dpp, &g_pd3dDevice ) ) )
{return E_FAIL;}
```

После инициализации и настройки устройства добавляем метод построения и отображения сцены на экране:

```
VOID C3DSys::Render() // метод, отвечающий за визуализацию
{// функция очистки экрана
g_pd3dDevice->Clear(0,NULL,D3DCLEAR_TARGET|D3DCLEAR_Z
BUFFER, D3DCOLOR_XRGB(128,128,128), 1.0f, 0 );
// начало формирования сцены
if( SUCCEEDED( g_pd3dDevice->BeginScene() ) )
```

```
{// здесь строим графику
g_pd3dDevice->EndScene();// завершение }
// показываем сформированное изображение
g_pd3dDevice->Present( NULL, NULL, NULL, NULL ); }
```

В Direct3D есть такое понятие, как «сцена» (scene). В сцену входят со своими свойствами все 3D-объекты, которые отображаются в окне приложения. Перед началом сцены вызываем:

```
g_pd3dDevice->BeginScene();
g_pd3dDevice->EndScene();
```

Другими словами, построение всех объектов происходит между этими двумя методами.

Поскольку все изображение формируется в невидимом буфере (backbuffer), необходимо после формирования изображения показать этот буфер. Для этого нужно вызвать метод:

```
HRESULT Present
{// координаты прямоугольника
CONST RECT* pSourceRect, // координаты прямоугольника клиентского окна
CONST RECT * pDestRect, HWND
hDestWindowOverride, // handle клиентского окна, которое предназначено для визуализации сцены
CONST RGNDATA* pDirtyRegion // не используется и должен быть установлен в NULL);
```

В терминах систем координат процесс получения проекции может быть описан следующей упрощенной блок-схемой:

- 1) локальная система координат (система координат объекта);
- 2) мировая система координат (глобальная система координат);
- 3) система координат камеры (система координат обзора);
- 4) система координат экрана (дисплей).

Переход от одной системы координат к другой эквивалентен смене базиса. Когда нужно преобразовать объект из одного базиса в другой, следует умножить все вершины объекта на соответствующую матрицу приведения базиса.

Локальная система координат (локальное пространство) определяет исходные координаты объекта, т. е. в тех, в которых он задан. Моделирование объекта в локальной (собственной) системе координат удобнее, чем напрямую в мировой системе координат. Локальная система позволяет описывать объект, не обращая внимания на положение, размер, ориентацию других объектов в мировой системе координат.

После того как заданы все объекты в своих собственных (локальных) системах координат, необходимо привязать их к общей мировой системе координат. Процесс трансформации координат объектов, заданных в локальных системах в мировую (общую), называют мировым

преобразованием (world transform). Для применения мирового преобразования метод SetTransform вызывается с параметром D3DTS_WORLD.

Для установки матрицы вида (преобразование в пространство камеры) используется метод SetTransform с первым параметром D3DTS_VIEW.

Последним этапом преобразования при получении 2D-изображения является операция проекции. Для работы с перспективной проекцией предназначена функция:

```
D3DXMATRIX* WINAPI D3DXMatrixPerspective FovLH (
D3DXMATRIX *pOut, // результат
FLOAT fovy, // вертикальный угол обзора
FLOAT Aspect, // отношение ширины окна к высоте
FLOAT zn, // расстояние до передней отсекающей
плоскости
FLOAT zf // расстояние до задней отсекающей
плоскости
);
D3DXMatrixPerspectiveFovLH ( &matProj,
D3DX_PI/4, 1.33f, 1.0f, 10000.0f);
g_pd3dDevice->SetTransform (
D3DTS_PROJECTION, &matProj);
```

Все это собрано в методе SetupMatrix нашего класса:

```
VOID C3DSys::SetupMatrix()
{
D3DXMatrixIdentity( &matWorld );
g_pd3dDevice->SetTransform( D3DTS_WORLD,
&matWorld );
matView = matVo*matVa*matVz*matVd;
g_pd3dDevice->SetTransform( D3DTS_VIEW,
&matView );
D3DVIEWPORT9 pViewport;
g_pd3dDevice->GetViewport(&pViewport);
float Aspect = 1.0f*pViewport.Width
/pViewport.Height;
D3DXMatrixPerspectiveFovLH ( &matProj,
D3DX_PI/4, Aspect, 1.0f, 10000.0f );
g_pd3dDevice->SetTransform ( D3DTS_PROJECTION,
&matProj );
}
```

Рис. 1 представляет собой диаграмму классов, предназначенных для сбора, хранения и переработки основной информации программного продукта.

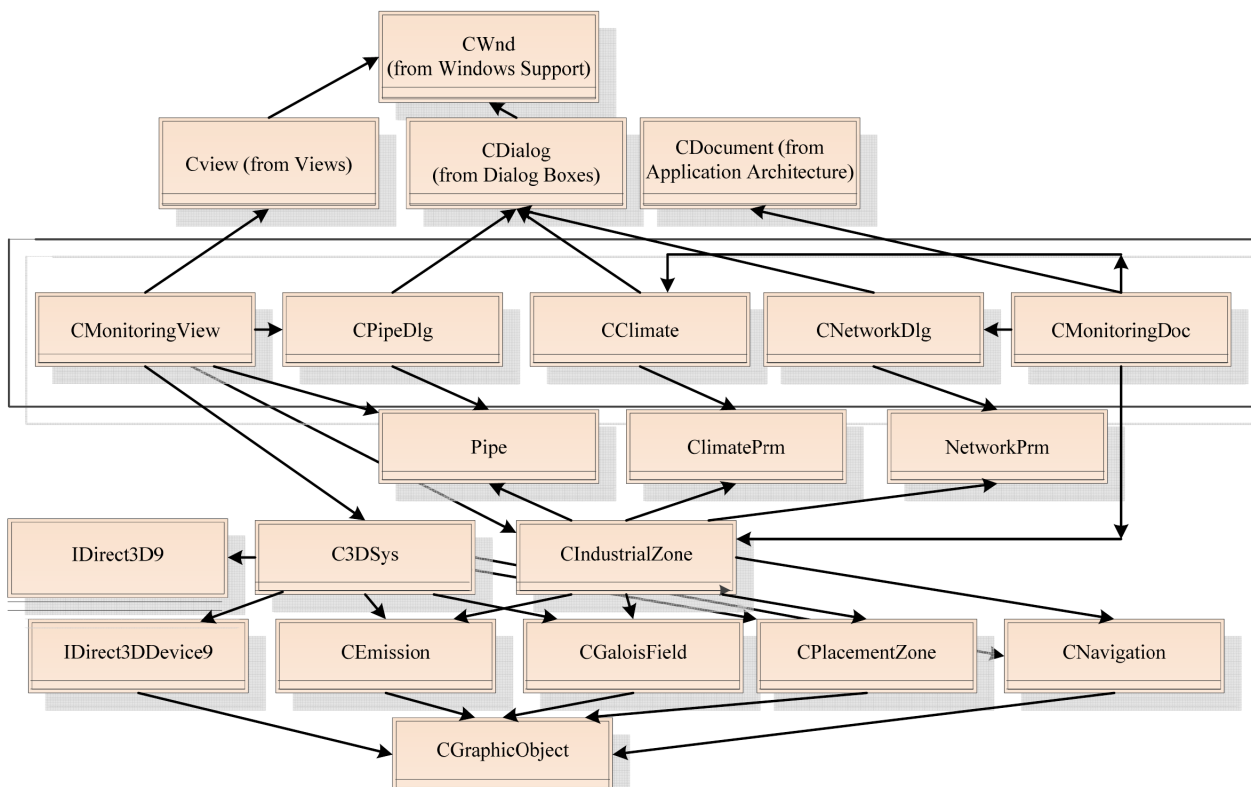


Рис. 1. Диаграмма классов:

- Pipe – характеристики предприятий как источников выбросов;
- ClimatePrm – ветровой и климатический режим; NetworkPrm – параметры информационной сети;
- CIndustrialZone – содержит Pipe, ClimatePrm, NetworkPrm и функции «добавить трубу», «удалить трубу», «проверить коллизию», «установить фокус на трубу»;
- CMonitoringDoc – специальный класс для хранения данных;
- CMonitoringView – предназначен для рисования данных, которые находятся в CMonitoringDoc

Сам по себе этот класс может рисовать только примитивную графику. Для того чтобы рисовать 3D, нужно инициализировать DirectX. Его инициализирует класс C3DSys, его метод Init().

После инициализации создается IDirect3DDevice9, который отвечает за рисование (программное), CEmission, CGaloisField, CPlacementZone, CNavigation предназначены для рисования «график выброса», «поле Гауза», «трубы на плоскости», «компас с флюгером».

Устройство для рисования IDirect3DDevice9 обеспечивает всякие шрифты, палитры и прочее, все рисуется в функции Render() класс C3DSys по средствам IDirect3DDevice9, только «график выброса» рисует CEmission (метод Draw), «поле Гауза» – CGaloisField.

Render() вызывается классом CMonitoringView каждый раз, когда происходит событие (нажата/отпущена клавиша мыши, перемещен курсор и др.).

На рис. 2 и 3 представлены два варианта выделенной зоны территории с различным расположением на ней промышленных предприятий. Каждое предприятие обладает индивидуальными свойствами. Программа моделирует шлейфы загрязнения в соответствии с лицензионной методикой рассеивания вредных веществ в атмосфере. Пользователю доступна возможность обзора моделируемой территории, он может легко приближать и менять угол обзора. Каждое из предприятий можно быстро передвигать по территории, и, если в этом есть необходимость, добавлять или удалять любое из предприятий. Под каждым из предприятий выводятся цифровые значения координат.

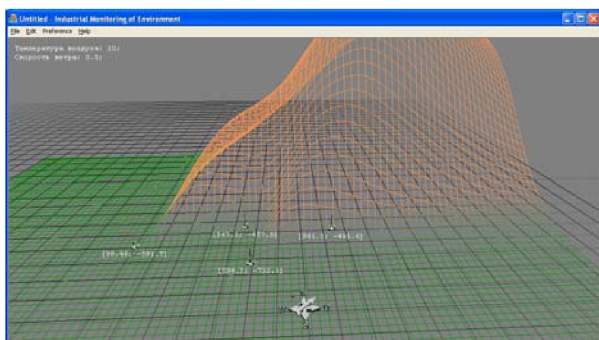


Рис. 2. Первый вариант расположения предприятий на выделенной территории

Возможно, например, воссоздать конфигурацию промышленной площадки Минского тракторного завода или расположения всех ТЭЦ в городе.

Для получения 3D-изображений на рис. 2 и 3 источниками информации служили условия выброса газозадушной смеси из устья дымовых труб. Картины загрязнения на рисунках

получены расчетным путем и только имитируют то, что нам требуется смоделировать. Данные нужно собрать не на трубе, а в приземном слое, там, где живут и дышат люди, расставив на земле необходимое количество датчиков в узлах информационной сети, которую сначала нужно построить.

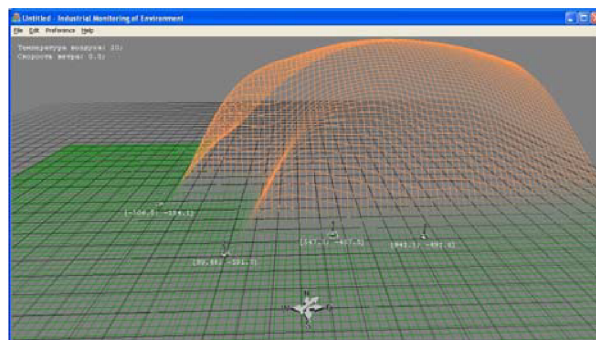


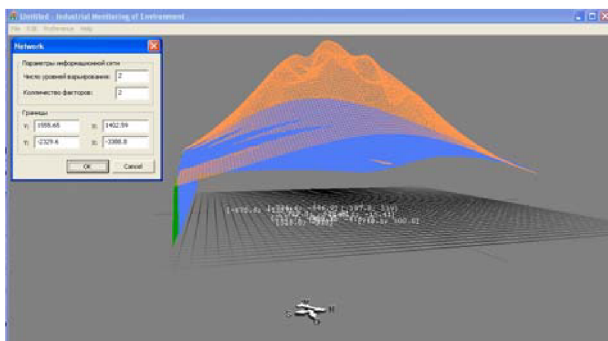
Рис. 3. Второй вариант расположения предприятий на выделенной территории

Интерактивный выбор территории факторного пространства позволяет быстро очертить местность для построения информационной сети. Степень точности получаемой информации о загрязнении окружающей среды можно наглядно сравнивать с имеющимся оригиналом и оценивать степень адекватности математического описания по соответствующим статистическим критериям. Точность описания определяется количеством датчиков и их размещением на территории.

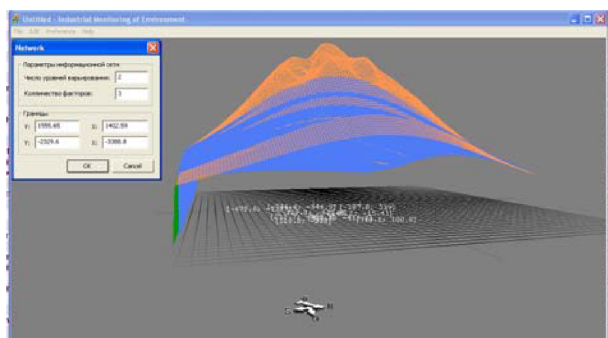
Рациональность размещения датчиков обеспечивается использованием теории конечных полей. Сначала формируется ортогональная таблица путем умножения и сложения элементов поля Гауза в кольце классов вычетов по модулю s в соответствии с координатами групп вершин связок плоскостей на бесконечности. Затем ортогональная таблица трансформируется в информационную сеть, а та, в свою очередь, превращается в таблично заданную функцию, которая с помощью метода наименьших квадратов дает возможность получить различные разновидности полиномиальных моделей в виде отрезка ряда Тейлора.

На рис. 4–7 показаны попарные 3D-изображения моделируемой (светлые линии) и моделированной (темные линии) поверхностей загрязнения территории при различных значениях параметров информационной сети.

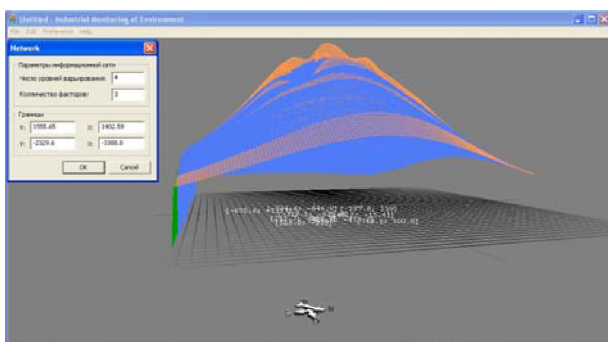
На рис. 4 для двух псевдофакторов и двух уровней варьирования (модуль $s = 2$) аппроксимирующая поверхность представлена одним экстремумом (парабола), тогда как на исходной поверхности можно разглядеть не менее восьми экстремумов.

Рис. 4. Аппроксимация $f = 2; s = 2$

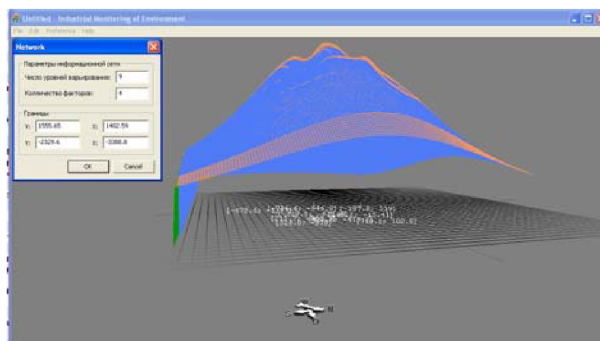
Для трех псевдофакторов и четырех уровней варьирования (рис. 5) потихоньку начинают прорисовываться детали, а трехфакторная девятимодульная сеть практически полностью воспроизводит ту поверхность, которую нужно получить по модели.

Рис. 5. Аппроксимация $f = 3; s = 4$

Естественно, что чем детальнее прорисовка эталона, тем больше требуется информации (количества датчиков) для получения модели.

Рис. 6. Аппроксимация $f = 3; s = 7$

Описание методики и программы рационального размещения датчиков на описываемой территории – материал отдельной статьи.

Рис. 7. Аппроксимация $f = 3; s = 9$

Заключение. В статье приведено описание программного средства для 3D-моделирования и визуализации загрязнения выделенных зон территории. До настоящего момента предпринималось значительное число попыток создания программных средств, предназначенных для мониторинга выбросов загрязняющих веществ в атмосферу от стационарных источников, однако в силу различных причин эти средства не получили должного практического применения и, кроме того, ограничиваются моделированием загрязнения территории в двухмерном пространстве. В области 3D-моделирования выделенных зон территорий весьма актуальной является идея совмещения стандартной методики ОНД-86 [3] и элементов теории построения информационных сетей. Разработанный на этой математической основе программный продукт позволяет проводить 3D-моделирование и визуализацию экологической обстановки выделенной территории, создавать информационную сеть для разработки системы локального мониторинга и проводить анализ полученных данных.

Литература

1. Колесников, В. Л. Компьютерное моделирование и оптимизация химико-технологических систем: учеб. пособие для вузов / В. Л. Колесников, П. П. Урбанович, И. М. Жарский. – Минск: БГТУ, 2004. – 532 с.
2. Колесников, В. Л. Математические основы компьютерного моделирования химико-технологических систем / В. Л. Колесников. – Минск: БГТУ, 2003. – 312 с.
3. ОНД-86 – Методика расчета концентраций в атмосферном воздухе вредных веществ, содержащихся в выбросах предприятий. – Л: Гидрометеиздат, 1987. – 82 с.

Поступила 02.03.2011