64 Камаев А.Н. и др.

# Генерация процедурной модели морского дна для тестирования и отладки систем технического зрения АНПА\*

A.H. Камаев¹, В.А. Сухенко², Д.А. Карманов¹kamaev an@mail.ru|eternallfame@mail.ru|zenbudd@mail.ruХабаровск, Россия, ¹Вычислительный центр ДВО РАН;²Тихоокеанский государственный университет

Разработка систем компьютерного зрения для задач навигации и маневрирования автономных необитаемых подводных аппаратов (АНПА) — перспективное направление развития подводной робототехники. Но разработка таких систем невозможна без тщательного тестирования и отладки. Использование реальных АНПА для таких целей требует больших временных и финансовых затрат. Кроме того, отсутствует возможность оценить точность работы алгоритмов, так как реальные данные о рельефе дна, над которым перемещается АНПА, неизвестны. Использование процедурных моделей дна для тестирования и отладки систем технического зрения АНПА приведёт к повышению скорости и надёжности разрабатываемых систем.

В статье рассматриваются основные этапы построения процедурной модели подводного дна: генерация рельефа на разных уровнях детализации, текстурирование, рендеринг. Рельеф строится на основе низкочастотной карты высот, содержащей такие элементы морского дна, как шельф, материковый склон, ложе океана, каньоны и горные хребты. Далее рельеф дополняется деталями, в том числе утёсами и пещерами на основе предлагаемого в статье «умного шума» с использованием техники марширующих кубов. Во время рендеринга рельеф дополнительно уточняется до пиксельной точности. Для текстурирования применяются процедурные 3D текстуры, имеющие характерный и уникальный рисунок на различных уровнях масштаба.

**Ключевые слова:** процедурный, текстурирование, карта высот, тесселяция, фрактал, шум, рельеф, АНПА

## 1. Введение

Развитие подводной робототехники требует создания надежных систем технического зрения для навигации и маневрирования АНПА вблизи дна, где важна высокая точность. Для тестирования и отладки подобных систем необходимо большое количество данных из акваторий, имеющих различный рельеф и покрытие дна. Также важна возможность контролировать работу тестируемых систем на каждом шаге и знать реальный рельеф акватории для оценки точности результата. Всё эти требования не могут быть удовлетворены в результате реальных экспериментов, поэтому в докладе рассматривается метод для генерации процедурной акватории, которую можно было бы использовать для нужд тестирования и отладки систем технического зрения АНПА.

## 2. Современные методы генерации

Процедурная генерация рельефа — активно развивающаяся область современной компьютерной графики. В настоящее время используются самые различные методы и алгоритмы для решения задачи процедурной генерации. Среди них можно выделить методы вида [3], когда дизайнер с помощью заранее подготовленной кисти рисует карту высот,

Работа выполнена при финансовой поддержке РФФИ грант № 16-31-00187 мол\_а. Работа опубликована при финансовой поддержке РФФИ, грант №16-07-20482

которая затем усложняется с использованием некоторых алгоритмов, метод [1], основанный на генерации карт рек и горных хребтов, генерация на основе фрактального шума, дополняемого моделированием эрозии почвы [2] и многие другие методы [9].

Подводный рельеф значительно отличается от наземного из-за различных физических факторов, влияющих на его формирование. Так для подводного рельефа характерны пещеристые структуры, которые не могут быть построены с использованием большинства современных алгоритмов генерации ландшафтов. Базовые элементы подводного рельефа, такие как шельф, материковый склон, ложе океана также не рассматриваются в современных методах генерации. Использование подводных ландшафтов для отладки систем технического зрения АНПА накладывает дополнительные требования уникальности и высокой детализации каждого элемента ландшафта. В настоящей работе предлагается метод генерации, учитывающий эти требования, а также специфику подводных ландшафтов.

## 3. Формирование рельефа

Разработанный в результате данного исследования алгоритм предполагает четыре этапа формирования модели: генерация низкочастотной карты (см. раздел 3.1), наложение фрактального шума для уточнения этой карты (см. раздел 3.2), построение сеточной модели (см. раздел 3.3), уточнение модели в процессе рендеринга (см. раздел 3.4).

## 3.1 Формирование карты рельефа

Карта рельефа  $\mathbf{H}=H(i,j),\ i=1,2,\ldots,h,\ j=1,2,\ldots,w,$  где w и h — размеры карты, определяет высоты рельефа  $H_{i,j}$  в точке с координатами  $(j\cdot res,i\cdot res),$  где — разрешение карты, заданное в метрах на точку (м/т). В настоящей работе использовались карты с разрешением res=10 м/т и линейными размерами  $w,h\leqslant 1000$  точек. Такие карты могут описать акватории размером до 10 км, что более чем достаточно для отладки алгоритмов технического зрения АНПА.

Карта рельефа содержит следующие базовые элементы, определяемые небольшим набором параметров: берег, шельф, материковый склон, каньоны, ложе океана и горные хребты (рис. 1а). Контуры этих элементов определяются с помощью фрактальных линий, а высоты в точках получаются в результате вычисления различных функций, зависящих от удалённости точек от контурных линий (рис 1б). Пример низкочастотной составляющей модели дна, построенной на основе карты рельефа (рис 1б), представлен на рис. 1в.

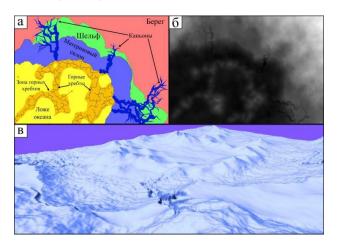


Рис. 1: Карта рельефа: а – компоненты карты, б – графическое отображение высот, в – низкочастотная модель, построенная на основе карты рельефа

## 3.2 Фрактальный шум

Для современного ПК в памяти мы можем разместить треугольники, покрывающие акваторию с точностью до 1–2 метров. При заданном разрешении res карты  $\mathbf{H}$  частоты рельефа от  $1/10000~m^{-1}$  до  $1/20~m^{-1}$  представлены самой картой рельефа. Такой частоты недостаточно для реалистичного представления ландшафта. Генерация оставшихся частот от  $1/20~m^{-1}$  осуществляется при помощи шумовой функции.

Определим два набора частот (октав) шума:

$$\mathbf{f}_l = (f_{li}) = \frac{2^{i-1}}{res}$$

$$\mathbf{f}_h = (f_{hj}) = \frac{2^{j+n_l}}{res}$$

где  $i=0,1,\dots,n_l,\ j=0,1,\dots,n_h,\ n_l=\lfloor\log_2(2\cdot res)\rfloor,\ n_h=\lfloor\log_2(p_w^{-1}2^{-n_l}res)\rfloor,\ p_w$  – размер выводимого фрагмента (пикселя экрана) в метрах.

До того, как мы определим вид шумовой функции, введем следующие обозначения:  $\mathbf{S}=(S_{ij})$  – карта уклона (рис. 2a) и  $\mathbf{R}=(R_{ij})$  – карта неровностей (рис. 2б), где  $i=1,2,\ldots,h,\ j=1,2,\ldots,w,\ R_{ij}$  – среднеквадратическое отклонение значений высокочастотной составляющей высот в окрестности точки  $H_{ij}$  размером  $9\times 9$  точек,

$$S_{ij} = \frac{2\arccos(n_{ij,z})}{\pi},$$

где  $n_{ij}=(n_{ij,x},n_{ij,y},n_{ij,z})$  — нормаль к низкочастотной составляющей  ${\bf H}$  в точке (i,j). Пусть  ${\bf P}({\bf x})$ , где  ${\bf x}=(x_1,x_2,x_3)$  — трёхмерный шум Перлина [8],  ${\bf I}_S(f,x_1,x_2)$  и  ${\bf I}_R(f,x_1,x_2)$  — функции влияния уклона и неровностей:

$$I_S(f, x_1, x_2) = \left(1 - \sqrt{\frac{\min(f, f_{\text{max}}) - f_{l0}}{f_{\text{max}} - f_{l0}}}\right) \left(1 - \cos(\pi S'_{x_1 x_2})\right)^2,$$

$${\rm I}_{R}(f,x_{1},x_{2}) = \sqrt{\frac{\min(f,f_{\rm max}) - f_{l0}}{f_{\rm max} - f_{l0}}} \left(0.9 R_{x_{1}x_{2}}^{'} + 0.1\right),$$

где  $S_{x_1x_2}'$  и  $R_{x_1x_2}'$  – значения, получаемые из  ${\bf S}$  и  ${\bf R}$  посредством билинейной интерполяции,  $f_{\rm max}=20~{\it M}^{-1}$ , т.к влияние описанных выше карт на частоты, большие, чем  $f_{\rm max}$ , можно считать незначительным.

Используя введенные обозначения, определим шумовую функцию:

$$N(\mathbf{x},t) = \sum_{i=0}^{n_t} \frac{2}{f_{ti}} P(f_{ti}\mathbf{x}) \left( I_S(f, x_1, x_2) + I_R(f, x_1, x_2) \right).$$

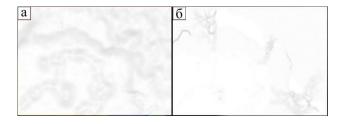


Рис. 2: а – карта уклона, б – карта неровностей

#### 3.3 Построение сеточной модели

Имея карту высот  $\mathbf{H}$ , шумовую функцию  $\mathbf{N}(\mathbf{x},t)$  и набор низких частот  $\mathbf{f}_l$ , определим функцию плотности поверхности  $\mathbf{v}(\mathbf{x}) = x_3 - H_{x_1x_2}^{'} + \mathbf{N}(\mathbf{x},l)$ , где значение  $H_{x_1x_2}^{'}$  получено из  $\mathbf{H}$  посредством билинейной интерполяции. Будем считать  $\mathbf{v}(\mathbf{x}) = 0$  поверхностью генерируемого рельефа. Если  $\mathbf{v}(\mathbf{x}) > 0$ ,

66 Камаев А.Н. и др.

то точка  $\mathbf{x}$  находится над поверхностью ландшафта, в противном случае — внутри, или на поверхности.

Аппроксимируем поверхность  $v(\mathbf{x}) = 0$  треугольными гранями. В качестве алгоритма триангуляции выберем алгоритм марширующих кубов [6]. Результатом работы этого алгоритма является набор вершин, образующих треугольные грани, аппроксимирующие искомую поверхность. Пример такой аппроксимации можно видеть на рис. 3.

В результате работы алгоритма могут образоваться несколько изолированных поверхностей. Все поверхности, кроме той, которая содержит наибольшее число треугольников, удаляются. Также для каждой вершины вычисляется нормаль к поверхности.

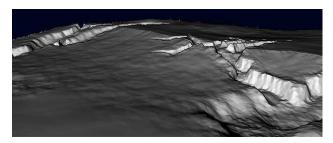


Рис. 3: Сеточная модель

# 3.4 Рендеринг геометрии

В качестве метода достижения пиксельной точности предлагается тесселяция видимых треугольников такого порядка, чтобы на каждый пиксель визуализируемого изображения приходился один треугольник. Аппаратная тесселяция при использовании OpenGL позволяет разбивать каждую сторону треугольника всего на  $2^6$  частей, что недостаточно для некоторых расстояний от камеры до треугольника при длине стороны треугольника в один метр. Предлагается разбивать такие треугольники по заранее подготовленному массиву разбиения.

Для всех возможных вариантов разбиений заранее рассчитываются наборы треугольников. Массив разбиения индексируется переменными  $n_i = \log_2 m_i$ , где,  $i = 0, 1, 2, m_i$  — число разбиений i-й стороны треугольника. Различные комбинации значений  $n_0$ ,  $n_1$ ,  $n_2$  при  $n_{\rm max} = 9$  дают до 1000 возможных наборов. Каждый набор представляет собой массив вершин, составляющих треугольник с заданным разбиением. Пусть  $M_i$  — количество вершин в i-ом наборе, где  $i = 0, 1, \ldots, (n_{\rm max} + 1)^3 - 1$ ,  $\mathbf{v}_{ik} = (v_{ik0}, v_{ik1})$  — вершины i-го набора, где  $k = 0, 1, \ldots, M_i - 1$ .

Вершины  $\mathbf{v}_{ik}$  задаются в системе координат, образованной 0-й и 2-й стороной треугольника. Для преобразования координат вершин из i-го набора в

мировую систему координат необходимо знать следующие данные о треугольнике, который предполагается заменить i-ым набором: координаты 3-х вершин этого треугольника  $\mathbf{p}_0$ ,  $\mathbf{p}_1$ ,  $\mathbf{p}_2$  и усреднённые нормали в вершинах  $\mathbf{\acute{n}}_0$ ,  $\mathbf{\acute{n}}_1$ ,  $\mathbf{\acute{n}}_2$  (для расчёта дополнительных октав шума). Преобразование в мировую систему координат для k-ой вершины i-го набора записывается следующим образом:

$$\mathbf{v}_{ik}' = v_{ik0} \cdot (\mathbf{p}_1 - \mathbf{p}_0) + v_{ik1} \cdot (\mathbf{p}_2 - \mathbf{p}_0) + \mathbf{p}_0$$

Таким образом, вершины и индексы всех наборов могут быть заранее размещены в вершинных и индексных буферах. Тогда отрисовка треугольника будет заключаться в том, чтобы передать в шейдер 6 3-х компонентных uniform векторов  $\mathbf{p}_0$ ,  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ ,  $\mathbf{\acute{n}}_0$ ,  $\mathbf{\acute{n}}_1$ ,  $\mathbf{\acute{n}}_2$  определить индекс набора, рассчитав количество сегментов разбиения  $n_0$ ,  $n_1$ ,  $n_2$  и вызвать команды отрисовки набора с нужным индексом. При этом вершинный шейдер всегда сможет рассчитать реальные координаты точек набора  $\mathbf{v}_{ik}^{'}$  и нормали в них  $\mathbf{n}_{ik}^{''}$  переданные uniform параметры.

Для вывода геометрии её необходимо разбить на блоки, например,  $100 \times 100$  метров. В этом случае блок будет содержать 20—40 тысяч больших треугольников, а всего будет не более 10000 блоков. Предлагается выводить только те блоки, хоть один угол которых ближе к камере, чем 500 метров. Общее количество блоков позволяет перед каждым выводом перебирать все блоки и выводить только нужные. При этом с помощью тесселяции выводятся только те блоки, все треугольники которых могут быть разбиты средствами аппаратной тесселяции. Треугольники ближайших к камере блоков (от 1 до 4) выводятся с помощью заранее рассчитанных наборов. Треугольники вне пирамиды камеры отбрасываются.

После разбиения (аппаратного или ручного) к каждой полученной вершине добавляется шум, описанный в разделе 3.2 (см рис. 4).

$$\mathbf{v}_{ik}^{"} = \mathbf{v}_{ik}^{'} + \mathbf{n}_{ik}^{"} \cdot \mathbf{N}(\mathbf{v}_{ik}^{'}, h)$$

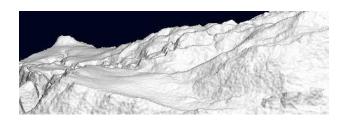


Рис. 4: Сеточная модель с наложенным высокочастотным шумом

## 4. Текстурирование

Текстурирование ландшафтов, разрабатываемых для тестирования систем технического зрения АН-ПА, имеет ряд специфических особенностей:

- 1. Возможно использование лишь процедурных текстур, так как любое применение растровых изображений в большихсценах приводит к возникновению повторяющихся паттернов, которые делают невозможным тестирование алгоритмов, использующих особые точки [5] [7] [10];
- Сложный рельеф делает невозможным корректный расчёт текстурных координат для 2D текстур, поэтому применяются только 3D текстуры;
- 3. Текстуры должны иметь корректное представление на всех уровнях масштаба, на которых проводится тестирование.

Тип покрытия, а соответственно и тип текстуры определяется на основе карт неровностей и карт уклона (см раздел 3.2), при этом используются дополнительные шумовые функции для создания фрактальных границ перехода от одного типа поверхности к другому.

Основные используемые типы поверхности — сыпучие и твёрдые. Создание процедурной текстуры для каждого типа поверхности требует индивидуального подхода. В качестве примера, рассмотрим текстуру сыпучей поверхности — песка. Песок имеет два различных характерных рисунка на двух уровнях масштаба: зерно (рис. 5а) и рябь (рис. 5б), имеющих частоты  $s_g=160$  и  $s_b=2.5$   $m^{-1}$  соответственно.

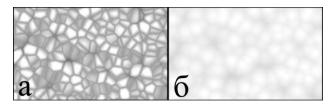


Рис. 5: Основные рисунки песка: а – зерно, б – рябь

Подобные рисунки могут быть получены с использованием функции  $d_k(\mathbf{x})$ , которая возвращает расстояние от точки с координатами  $\mathbf{x}$ до k-й ближайшей точки пространства. При этом всё пространство равномерно заполнено точками со случайными положениями таким образом, что в каждой единице объёма находится в среднем две точки (cellular textures [4]). Для получения рисунка зерна (рис. 5а) использована следующая функция:

$$l_g(\mathbf{x}) = \sum_{i=0}^{2} \sum_{j=0}^{2} a_{3i+j} d_1^j(\mathbf{x}) d_2^j(\mathbf{x}),$$

где коэффициенты  $a_i$ ,  $i=0,2,\ldots,8$  подобраны экспериментально и равны соответственно: 0.6, 0.64,

0.56, -0.45, 0.13, -0.65, -0.75, 0.5, -0.17. Рисунок ряби (рис. 56) получен следующим образом:

$$l_b(\mathbf{x}) = (1 - 0.7d_1(\mathbf{x}))^{0.15}$$

Кроме того, для рисунка зерна определяется цвет каждой частицы. Для этого вводится функция  $n_k(\mathbf{x})$ , которая определяет целое число в интервале [0..255], закреплённое за k-ой ближайшей к  $\mathbf{x}$  точкой пространства. Также заполняется таблица цветов  $\mathbf{C}=(\mathbf{c}_i)$ , где  $i=0,1,\ldots,255$ . Каждый цвет  $\mathbf{c}_i=(c_{ir},c_{ig},c_{ib})$  определяет один из возможных цветов частицы песка. В настоящей работе использовано семь различных цветов, которые многократно повторяются в таблице  $\mathbf{C}$  в зависимости от требуемой частоты встречаемости. Таким образом, цвет зерна в точке  $\mathbf{x}$ 

$$\mathbf{c}_q(\mathbf{x}) = \mathbf{c}_{n_1(\mathbf{x})}$$

Цвет песка  $\mathbf{c}_s$  определяется как среднее значение цветов из таблицы  $\mathbf{C}$ :

$$\mathbf{c}_s = \frac{1}{256} \sum_{i=0}^{255} \mathbf{c}_i$$

Процесс формирования цвета фрагмента песка, имеющего положение  ${\bf x}$  и занимающего площадь  $s \times s$  метров, определяется схемой на рис. 6.

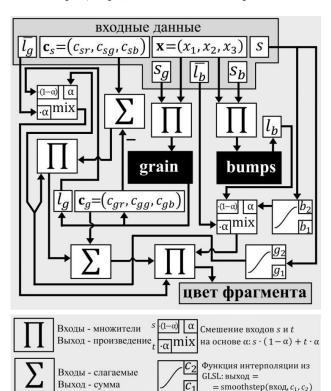


Рис. 6: Схема вычисления цвета фрагмента песка

К входным данным, кроме описанных, добавляются значения средней яркости рисунка зерна  $\overline{l_g}$  и

68 Камаев А.Н. и др.

ряби  $\overline{l_b}$ . Вычисления функций  $\mathbf{c}_g(\mathbf{x}),\ l_g(\mathbf{x})$  и  $l_b(\mathbf{x})$  представлены блоками «grain» и «bumps», принимающими на вход координаты и выдающими значения  $\mathbf{c}_g,\ l_g$  и  $l_b$  соответствующие полученным координатам. Для констант перехода выбраны следующие значения:  $g_1=0.0025$ м,  $g_2=0.01$ м,  $b_1=0.3$ м,  $b_2=0.5$ м.

В результате применения схем, подобных представленной на рис. 6, текстуры получают корректное представление на разных уровнях масштаба и не подвергаются эффекту алиасинга (частота выводимых данных никогда не превышает частоту дискретизации). Пример рассмотренной текстуры песка, а также текстура камня, покрытого растительностью различного цвета, на разных уровнях масштаба представлен на рис. 7.

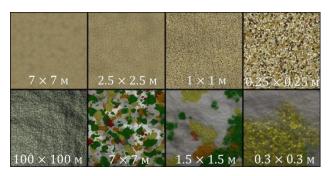


Рис. 7: Примеры текстуры песка (сверху) и камня (снизу) на различных уровнях масштаба

## Заключение

С использованием описанных методов и алгоритмов удалось создать уникальный рельеф сантиметровой точности с неповторяющимся текстурным рисунком. Полученный ландшафт может выводиться в режиме реального времени и быть использован для отладки и тестирования систем технического зрения АНПА. Дальнейшими направлениями развития данной работы станет добавление большего числа процедурно генерируемых текстур, процедурных моделей обитателей дна, размещение моделей различных затонувших объектов, имитация оптических искажений, вносимых линзами камер АНПА и подводной средой, дальнейшее усложнение рельефа путем добавления сети пещер, как базового элемента рельефа.

# Литература

- [1] Farès Belhadj, Pierre Audibert. Modeling Landscapes with Ridges and Rivers: bottom up approach. // GRAPHITE '05. ACM 2005. pp. 447-450.
- [2] Farès Belhadj. Terrain modeling: a constrained fractal model. // AFRIGRAPH '07. ACM 2007. pp. 197-204
- [3] Giliam J.P. de Carpentier, Rafael Bidarra. Interactive GPU-based procedural heightfield brushes. // FDG '09. ACM 2009. pp. 55-62.
- [4] Ebert et al. Texturing and Modeling A Procedural Approach // Morgan Kaufmann, 2003.
- Herbert B. et al. SURF: Speeded Up Robust Features
  // CVIU. 2008. Vol. 110, № 3. P. 346-369.
- [6] William E. Lorensen, Harvey E.Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm // – SIG '87
- Mikolajczyk K., Schmid C. Scale & affine invariant interest point detectors // IJCV. - 2004. - № 60(1).
   - P. 63-86.
- [8] Ken Perlin. Improving noise // ACM Transactions on Graphics (TOG), v.21 n.3, July 2002
- [9] Ruben M Smelik et al. A Survey of Procedural Methods for Terrain Modelling. // 2009., URL: http://graphics.tudelft.nl/ruben/RMSmelik3AMIGAS09.pdf
- [10] A. Verma et al. A New Color SIFT Descriptor and Methods for Image Category Classification // IRAST International Congress CACS 2010. – Singapore, 2010. – P. 819-822.