

УДК 681.3.513

Е. А. Никулин

ПРОЦЕДУРНЫЙ ШУМ И МУЛЬТИВАРИАНТНЫЙ ТАЙЛИНГ

Нижегородский государственный технический университет им. Р. Е. Алексеева

Тема работы: Создание процедурных бесшовных текстур с неповторяющимися плитками.**Цель работы:** Развитие методов процедурного бесшовного текстурирования.**Метод решения:** Особый выбор узловых элементов текстурной решетки.**Оригинальность:** Обеспечен неперриодичный тайлинг больших мультитекстур.**Выводы:** Метод предназначен для бесшовного покрытия больших площадей с малыми затратами ресурсов.*Ключевые слова:* текстура, клеточный шум, тайлинг.

Статья посвящена совершенствованию *тайлинга* – создания процедурных текстур-тайлов (от *tile* – плитка), способных бесшовно соединяться друг с другом в желаемых направлениях. После краткого изложения и иллюстрации алгоритма синтеза так называемого *клеточного шума* (ограничимся его 2D-вариантом) будет предложен способ вариации его исходных данных, обеспечивающий гладкость сшивки текстур по границам и неповторяемость их внутреннего рисунка.

Работа генератора клеточного шума происходит на решетке, разбивающей текстурную карту с габаритами $\{N, M\}$ на клетки, уложенные в n рядов и m колонок длиной и высотой

$$d=M/m, \quad h=N/n \quad (1)$$

текселов (рис. 1, а). В узлах решетки задаются / вычисляются узловые элементы $e_{00} \div e_{nm}$ – числа или векторы, используемые для расчета значений шумовой функции во всех текселах. В простейшем *численном шуме* они генерируются как случайные числа $e_{ij} = rnd(1) \in [0, 1]$. В самом распространенном виде клеточного шума – *градиентном шуме Перлина* [1] – в узлах решетки генерируются нормированные векторы градиентов $\mathbf{G}_{ij} = [\sin(\varphi_{ij}) \cos(\varphi_{ij})]$ со случайными углами $\varphi_{ij} = rnd(2\pi) \in [0, 2\pi]$, после чего вычисление значения шумовой функции в каждом (v, u) -м текселе карты производится в следующем порядке:

- определяются целочисленные координаты клетки, содержащей тексел:

$$i = \lfloor v/h \rfloor - \{v=N\}, \quad j = \lfloor u/d \rfloor - \{u=M\}; \quad (2)$$

- находятся нормализованные клеточные параметры тексела (рис. 1, б):

$$t = v/h - i \in [0, 1], \quad \tau = u/d - j \in [0, 1]; \quad (3)$$

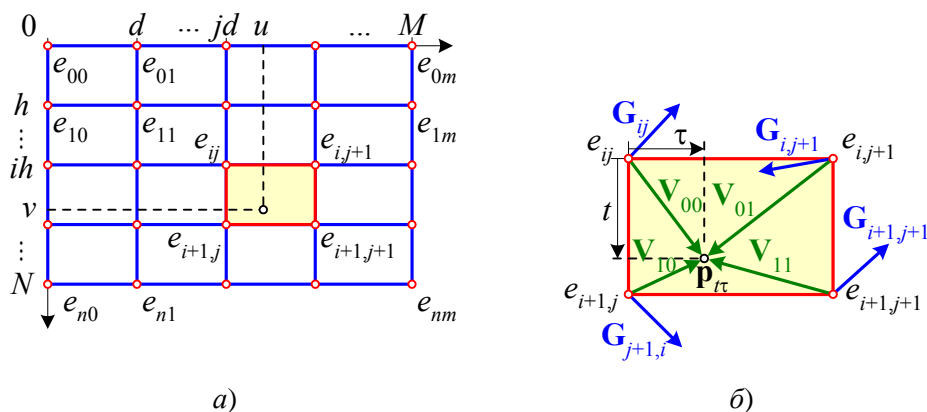


Рис. 1. Решетка и клетка для шума Перлина

- из углов клетки в точку $\mathbf{p}_{t\tau} = [\tau \ t]$ проводятся векторы

$$\mathbf{V}_{00} = [\tau \ t], \quad \mathbf{V}_{01} = [\tau - 1 \ t], \quad \mathbf{V}_{10} = [\tau \ t - 1], \quad \mathbf{V}_{11} = [\tau - 1 \ t - 1]; \quad (4)$$

- скалярные произведения градиентных и угловых векторов

$$e_{ij} = \mathbf{G}_{ij} \cdot \mathbf{V}_{00}, \quad e_{i,j+1} = \mathbf{G}_{i,j+1} \cdot \mathbf{V}_{01}, \quad e_{i+1,j} = \mathbf{G}_{i+1,j} \cdot \mathbf{V}_{10}, \quad e_{i+1,j+1} = \mathbf{G}_{i+1,j+1} \cdot \mathbf{V}_{11}; \quad (5)$$

- эти числа используются для вычисления значения шума методом бипараметрической интерполяции:

$$F_{vu} = \text{lin}(\text{lin}(e_{ij}, e_{i+1,j}, L(t)), \text{lin}(e_{i,j+1}, e_{i+1,j+1}, L(t)), L(\tau)) \quad \forall v = \overline{0, N}, \quad u = \overline{0, M}. \quad (6)$$

В (6) использована функция $\text{lin}(A, B, t) = A \cdot (1 - t) + B \cdot t$ линейной интерполяции от объекта A до объекта B по нормализованному параметру $t \in [0, 1]$. Таким параметром могут быть значения функции $L(t) \in [0, 1]$, которая должна обладать обязательными свойствами $L(0) = 0$ и $L(1) = 1$, обеспечивающими непрерывность соединения смежных клеток шумовой функции, и желательными свойствами $L'(1) = L'(0) = 0$ и $L''(1) = -L''(0)$ первого и второго порядков гладкости шивки. На рис. 2 построены графики и приведены примеры цветовой интерполяции для трех полиномиальных функций интерполяции:

- простейшей линейной функции $L_1(t) = t$, при которой и при $L_1(\tau) = \tau$ (6) превращается в билинейную интерполяцию;
- нелинейных функций $L_3(t) = 3t^2 - 2t^3$ и $L_5(t) = 10t^3 - 15t^4 + 6t^5$, создающих двойную гладкость стыковки и большой контраст изображения;
- возможно использование косинусной (не полиномиальной) функции интерполяции $L_{co}(t) = 0.5(1 - \cos(\pi t))$, отличающейся от $L_3(t)$ максимум на 0.01, но вычислительно более трудоемкой. Поэтому эту функцию следует признать *неэффективной*.

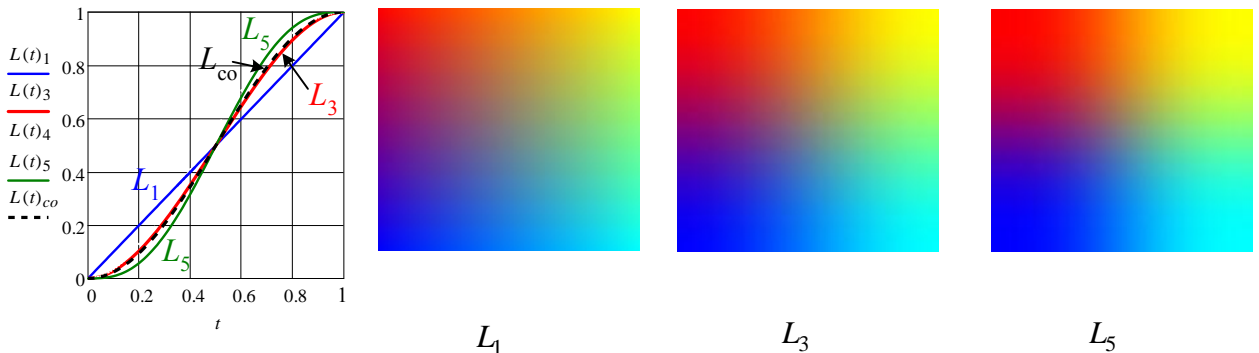


Рис. 2. Функции интерполяции

Благодаря интерполяции создается коррелированность клеточного шума, приводящая к образованию на текстуре причудливых фигур. Важно, что благодаря перебору в (6) текселов всей карты, а не в отдельных клетках по номерам (2), текстура будет закрашена без наложений и пропусков, а длина и высота клетки в (1) не обязаны быть степенями двойки и даже целочисленными! Это особенно ценно при выборе размеров решетки, которые могут быть произвольными целыми числами $n \leq N$ и $m \leq M$.

Перед выводом на экран карты, заполненной значениями шумовой функции F_{vu} $\forall v = \overline{0, N}, \quad u = \overline{0, M}$, она подвергается преобразованиям, имеющим следующие цели:

- приведение карты к виду, удобному для её дальнейших преобразований;
- преднамеренное искажение карты для достижения желаемых визуальных эффектов;
- конвертация значений карты в используемую цветовую палитру дисплея.

Хорошо, когда экстремальные значения карты равны 0 и 1 (такие *нормализованные* карты будем обозначать как H). Это желание реализуется потексельной операцией *нормали-*

защиты $H = \text{norm}(F)$ произвольной карты F со значениями матричной функции

$$\text{norm}(F)_{vu} = \frac{F_{vu} - \min(F)}{\max(F) - \min(F)} \in [0, 1] \quad \forall v = \overline{0, N}, u = \overline{0, M}. \quad (7)$$

При этом обеспечивается максимальное растяжение значений карты в единичном интервале $[0, 1]$ с $\min(H) = 0$ и $\max(H) = 1$.

Нормализация минимизирует вычислительные затраты, а то и обеспечивает саму возможность выполнения других операций с картами:

- $k_m H$ — масштабирование карты H , позволяющее при $k_m < 1$ уменьшить яркость её изображения, а также преобразовать все её значения в оттенки серого цвета с цветовыми матрицами $R = G = B = 255H$ палитры «True Color». Для получения разноцветной текстуры нужно сгенерировать три карты H_r, H_g, H_b и выводить на экран смесь цветовых матриц $R = 255H_r, G = 255H_g$ и $B = 255H_b$;
- $F + k_c$ — смещение карты F , позволяющее совместно с масштабированием повысить яркость изображения нормализованной карты в k_y раз:

$$\text{bright}(H, k_y)_{vu} = \frac{H_{vu} + k_y - 1}{k_y} \quad \forall v = \overline{0, N}, u = \overline{0, M}; \quad (8)$$

- в различных графических программах имеется великое множество *нелинейных фильтров*, потексельно, аналогично фильтрам (7), (8), обрабатывающих изображения по определенным алгоритмам. Из-за очевидного факта, что «нельзя объять необъятное», ограничимся лишь несколькими фильтрами:

◇ *гамма-коррекция*

$$\text{gamma}(H, p)_{vu} = H_{vu}^p \quad \forall v = \overline{0, N}, u = \overline{0, M}; \quad (9)$$

изменяет в нормализованной карте соотношение между темными и светлыми участками согласно графику степенной функции на интервале $[0, 1]$. При степени $p < 1$ яркость темных участков растет сильнее, чем светлых, а при $p > 1$ яркость светлых участков снижается больше, чем темных;

- ◇ фильтр «Штамп» с уровнем f_1 заменяет в карте F все значения $F_{vu} \geq f_1$ на 1, а все остальные — на 0, делая карту нормализованной:

$$\text{stamp}(F, f_1)_{vu} = \{F_{vu} \geq f_1\} \quad \forall v = \overline{0, N}, u = \overline{0, M}; \quad (10)$$

- ◇ *абсолютный* фильтр заменяет в карте F все имеющиеся отрицательные значения на положительные:

$$\text{abs}(F)_{vu} = |F_{vu}| \quad \forall v = \overline{0, N}, u = \overline{0, M}; \quad (11)$$

- ◇ периодический *модульный* фильтр заменяет все значения карты, лежащие вне диапазона $[0, 1)$, на значения

$$\text{modul}(F)_{vu} = \text{mod}(\text{mod}(F_{vu}, 1) + 1, 1) \in [0, 1] \quad \forall v = \overline{0, N}, u = \overline{0, M}; \quad (12)$$

- *цветовое смешивание* нормализованной карты H , создающее RGB-вектор цвета (v, u) -го тексела как линейную интерполяцию от цвета c_0 до цвета c_1 по нормализованному параметру $H_{vu} \in [0, 1]$:

$$\text{mix}2(c_0, c_1, H)_{vu} = \text{lin}(c_0, c_1, H_{vu}) \quad \forall v = \overline{0, N}, u = \overline{0, M}. \quad (13)$$

Применение в (13) нелинейных функций интерполяции $L_3(H_{vu})$ и $L_5(H_{vu})$ приве-

дет к обострению контраста цветового перехода (см. рис. 2). Более красочные эффекты дает трехцветное смешивание

$$\text{mix}3(c_0, c_1, c_2, H)_{vu} = (1-t)c_0 + 2t(1-t)c_1 + t^2c_2 \quad \forall v=\overline{0, N}, u=\overline{0, M}, t=H_{vu}, \quad (14)$$

устроенное аналогично квадратичной линии Безье. Благодаря свойству любой линии Безье лежать в выпуклой оболочке ее узловых точек, трехцветная смесь всегда будет находиться в цветовом *RGB*-пространстве между точками c_0 , c_1 и c_2 . Отметим, что квадратичная *интерполяция* таким свойством не обладает. На основе полиномиальной модели Безье можно смешивать любое количество цветов;

- *фрактализация* (от *fraction* — дробить) — создание карты $F^{<vn \times \mu m>}$ с теми же размерами $\{N, M\}$, но на более частой решетке с кратным числом рядов vn и/или колонок μm (вот здесь-то и возможны решетки с нецелочисленными длиной и высотой клетки!). Подражая музыкальной октаве, в которой отношение частот верхнего и нижнего звуков равно двум, процедурную карту с частотами $v=\mu=2^k$ также называют *k*-й *октавой*. Ничто не мешает использовать гармоники с разными и не равными степеням 2 частотами v и μ , что позволяет создать анизотропные текстуры с разными свойствами по горизонтали и вертикали;
- *смешивание текстур* с разными частотами позволяет добавить к пространственно протяженным деталям изображения текстуры $H^{<n, m>}$ более мелкие детали гармоник $H^{<vn \times \mu m>}$. В формуле фрактального смешивания

$$C = k_0 H^{<n \times m>} + k_1 H^{<v_1 n \times \mu_1 m>} + k_2 H^{<v_2 n \times \mu_2 m>} + \dots \quad (15)$$

сумма коэффициентов $k_0, k_1, k_2 \dots$ должна быть равна 1 с целью удержания значений карты C в диапазоне $[0, 1]$. Если задавать веса в порядке убывания $k_0 \geq k_1 \geq k_2 \geq \dots$, то низкочастотные гармоники (крупные детали текстуры) будут иметь относительно большую яркость, чем высокочастотные (мелкие детали).

Очень простой способ сделать генерируемую текстуру тайловой — это уравнивать граничные узловые элементы:

- верхнего ряда решетки с нижним для тайлинга плиток по вертикали:

$$e_{n0} = e_{00}, e_{n1} = e_{01}, \dots, e_{nm} = e_{0m}; \quad (16)$$

- левой колонки с правой для бесшовного соединения по горизонтали:

$$e_{0m} = e_{00}, e_{1m} = e_{10}, \dots, e_{nm} = e_{n0}; \quad (17)$$

- для бесшовного размножения во все стороны актуальны оба условия (16), (17).

На рис. 3 показана работа генератора численного шума и результаты операций (7)–(12) и (13)–(15) при исходных данных $N=120, M=150, n=4, m=5$ и размерах клеток $d=h=30$ текселов:

- а) на случайных узловых элементах тайловой решетки

$$E = \begin{bmatrix} 0.737 & 0.797 & 0.148 & 0.318 & 0.01 & 0.737 \\ 0.259 & 0.647 & 0.366 & 0.696 & 0.184 & 0.259 \\ 0.562 & 0.695 & 0.595 & 0.096 & 0.044 & 0.562 \\ 0.736 & 0.793 & 0.627 & 0.525 & 0.103 & 0.736 \\ 0.737 & 0.797 & 0.148 & 0.318 & 0.01 & 0.737 \end{bmatrix}$$

сгенерирована карта $F^{<n \times m>}$ с экстремальными значениями $\min(F) \approx 0.01$ и $\max(F) \approx 0.797$. Изображение получено по (13) цветовым смешиванием $\text{mix}2(c_{\text{ТК}}, c_{\text{СГ}}, F)$ от темно-красного цвета $c_{\text{ТК}} = [128 \ 0 \ 0]$ до светло-голубого $c_{\text{СГ}} = [128 \ 255 \ 255]$;

- б) на той же решетке по (7) создана нормализованная карта $H=norm(F)$ с $\min(H)=0$, $\max(H)=1$ и хорошо заметным повышением контраста;
- в) на той же решетке по (8) сгенерирована карта $B=bright(F,2)$ с $\min(B)=0.505$, $\max(B)=0.899$ и вдвое большей яркостью, чем у карты F ;
- г) на той же решетке по (9) выполнена гамма-коррекция $G=gamma(F,0.5)$ с $\min(G)=0.1$, $\max(G)=0.893$ и бóльшим контрастом, чем у карты B ;
- д) на той же решетке по (10) построен штамп $S^{<n,m>}=stamp(F^{<n,m>},0.5)$ с пороговым уровнем $f_1=0.5$;

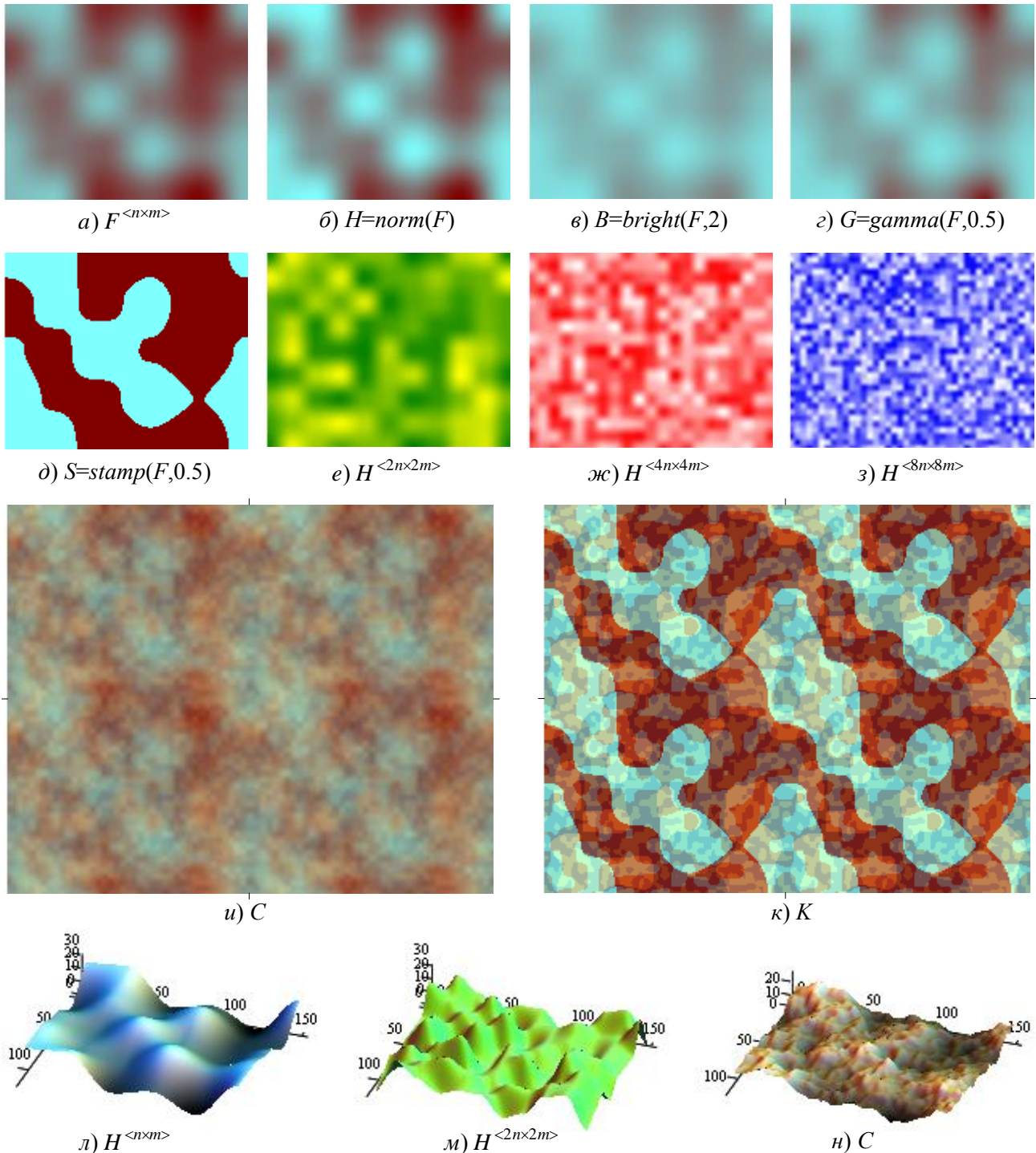


Рис. 3. Численный шум

- е) изображение карты первой октавы $H^{<2n \times 2m>}$ получено по (13) линейным смешиванием $\text{mix}2(c_{\text{тз}}, c_{\text{ж}}, H^{<8 \times 10>})$ от темно-зеленого цвета $c_{\text{тз}} = [0 \ 128 \ 0]$ до желтого $c_{\text{ж}} = [255 \ 255 \ 0]$;
- ж) карта второй октавы $H^{<4n \times 4m>}$ получена линейным смешиванием $\text{mix}2(c_{\text{б}}, c_{\text{к}}, H^{<16 \times 20>})$ от белого цвета $c_{\text{б}} = [255 \ 255 \ 255]$ до красного $c_{\text{к}} = [255 \ 0 \ 0]$;
- з) карта третьей октавы $H^{<8n \times 8m>}$ получена линейным смешиванием $\text{mix}2(c_{\text{с}}, c_{\text{б}}, H^{<32 \times 40>})$ от синего цвета $c_{\text{с}} = [0 \ 0 \ 255]$ до белого $c_{\text{б}}$;
- и) разноцветный шум $C = 0.5H^{<n, m>} + 0.2H^{<2n, 2m>} + 0.2H^{<4n, 4m>} + 0.1H^{<8n, 8m>}$ получен по (15) покомпонентным смешиванием четырех октавных карт и бесшовно размножен в два ряда и две колонки;
- к) разноцветный камуфляж $K = 0.5S^{<n, m>} + 0.2S^{<2n, 2m>} + 0.2S^{<4n, 4m>} + 0.1S^{<8n, 8m>}$ получен покомпонентным смешиванием четырех октавных штампов и бесшовно размножен в два ряда и две колонки.

Процедурный шум, и не только численный, имеет полезное применение как *генератор 3D-ландшафта* [2], в котором значения шумовой функции рассматриваются не как цвета текселей плоского изображения, а как *карта высот* рельефа на плане $v \times u$, по которой строится полигональная сеть. На рис. 3, л – н показаны поверхности карт $H^{<n \times m>}$ (см. рис. 3, б), $H^{<2n \times 2m>}$ (см. рис. 3, е) и одной секции тайловой текстуры на рис. 3, и. Для получения хорошей выпуклости рельефа значения карт увеличены в 30 раз.

При создании тайлового шума Перлина уравниваются граничные векторы

$$\mathbf{G}_{n0} = \mathbf{G}_{00}, \mathbf{G}_{n1} = \mathbf{G}_{01}, \dots, \mathbf{G}_{nm} = \mathbf{G}_{0m};$$

$$\mathbf{G}_{0m} = \mathbf{G}_{00}, \mathbf{G}_{1m} = \mathbf{G}_{10}, \dots, \mathbf{G}_{nm} = \mathbf{G}_{n0},$$

тогда в (5) автоматически получатся требуемые граничные элементы (16), (17).

Шум Перлина обладает следующими свойствами:

- благодаря нормализованной параметризации (3), все клетки решетки параметрически квадратны при любых геометрических размерах (1);
- во всех узлах решетки значения карты $F_{ih, jd}$ равны нулю ввиду обнуления соответствующих векторов в (4). Это объясняет появление в Перлин-текстурах горизонтально и вертикально ориентированных прямоугольных деталей;
- на одноклеточной тайловой решетке с $n=m=1$ и равными узловыми векторами $\mathbf{G}_{00} = \mathbf{G}_{01} = \mathbf{G}_{10} = \mathbf{G}_{11}$ значения карты кососимметричны: $F_{\tau\tau} = -F_{1-t, 1-\tau}$;
- все значения карты гарантированно лежат в интервале $[-\sqrt{0.5}, \sqrt{0.5}]$ (реально в гораздо более узком), поэтому она нуждается в нормализации.

Возможности шума Перлина проиллюстрированы на рис. 4, где показано несколько его тайловых реализаций с размерами карты $N=120$ и $M=150$, с разными размерами решетки $n \times m$, фильтрами (7)–(12) и цветовыми / фрактальными смесями (13)–(15):

- а) карта $F \in [-0.197, 0.197]$ создана на одноклеточной решетке и автоматически промодулирована видеоадаптером так, что отрицательные значения $F_{vu} < 0$ заменились на положительные $F_{vu} + 1$;
- б) та же карта F нормализована по (7);
- в) та же карта преобразована абсолютным фильтром (11) и нормализована;
- г) типичный нормализованный Перлин-шум на решетке из 12×15 квадратных клеток;
- д) волноподобный шум построен на решетке из 10×3 горизонтально вытянутых клеток;
- е) шум построен на решетке 3×3 , усилен в 10 раз и модулирован по (12);
- ж) шум, построенный на решетке 3×10 , абсолютизирован по (11) и отштампован по (10) с порогом $f_1 = 0.1$, влияющим на толщину светлых линий;

- з) на решетке из 6×5 клеток созданы и соединены в конечное изображение независимые цветовые Перлин-карты $stamp(R,0)$, $stamp(G,0)$ и $stamp(B,0)$;
- и) – м) карта $H^{<4 \times 2>}$ и три октавные карты $H^{<8 \times 4>}$, $H^{<16 \times 8>}$ и $H^{<32 \times 16>}$;
- н) фрактальная смесь $C = 0.5H^{<4 \times 2>} + 0.2H^{<8 \times 4>} + 0.2H^{<16 \times 8>} + 0.1H^{<32 \times 16>}$, размноженная в два ряда и две колонки.

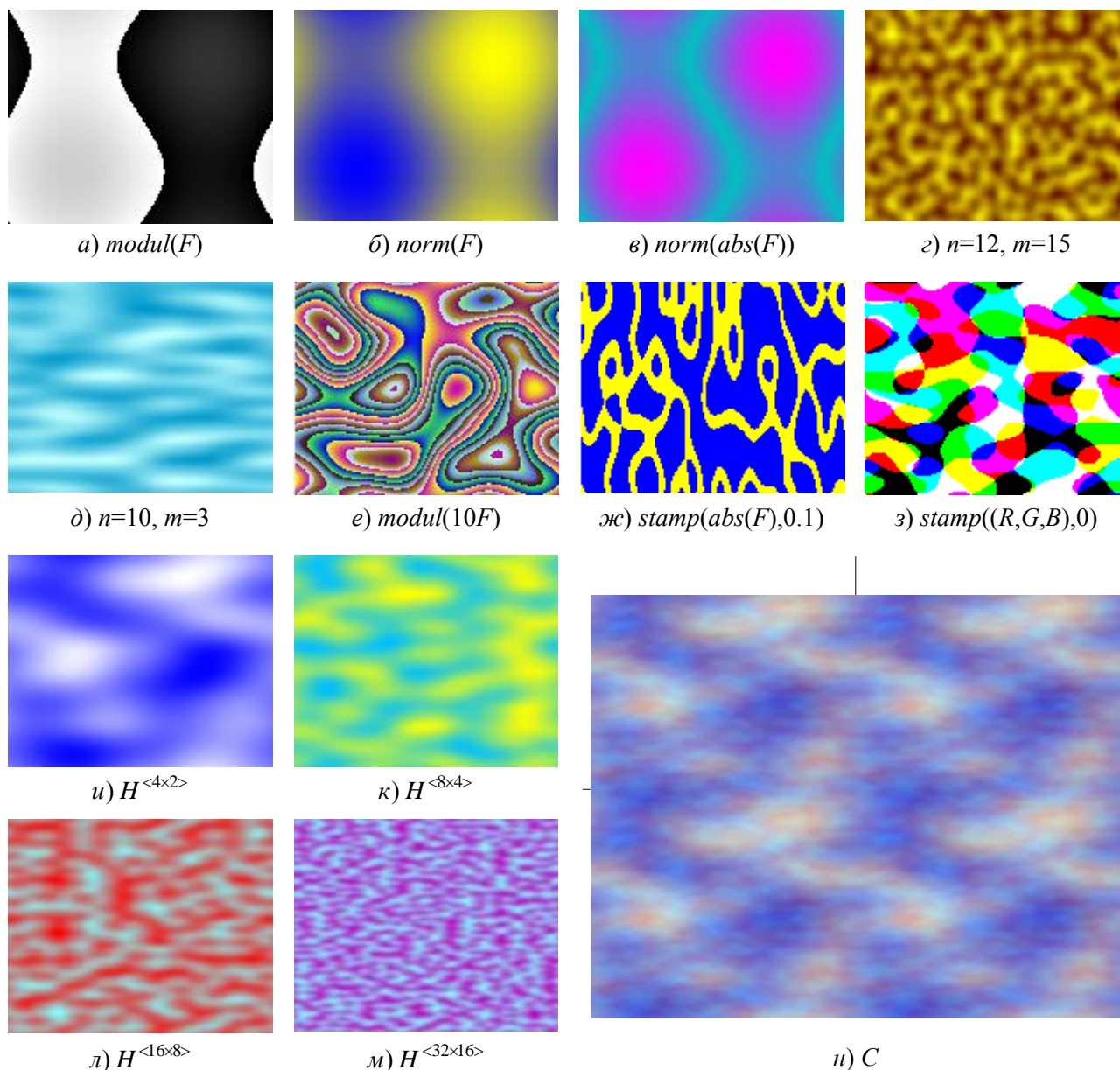


Рис. 4. Шум Перлина

Для того чтобы тайловая мультитекстура, собранная из одинаковых плиток, не утомляла назойливой периодичностью, нужно генерировать все «плитки» с одинаковыми граничными элементами (16), (17), но с *разными* внутренними элементами $e_{11} \div e_{n-1, m-1}$. На рис. 5 показаны слева периодическая 2×3 -укладка из шести одинаковых плиток, а справа – мультивариантная тайловая текстура, в которой все плитки отличаются друг от друга внутри, но совпадают по обозначенным границам сшивки. Заметим, что только левые верхние плитки в обеих текстурах одинаковые. Цветовые переходы изображений получены с помощью сплайнов Безье: левого – линейным смешиванием (13) от темно-коричневого цвета до желтого, а правого – квадратичным смешиванием (14) от коричневого цвета через красный до желтого.

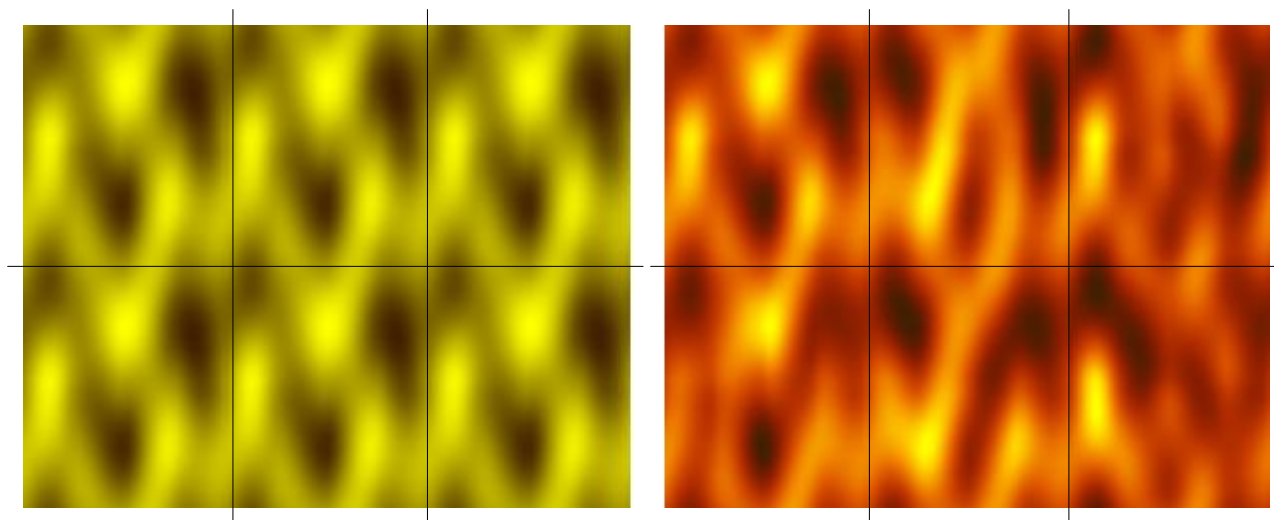


Рис. 5. Периодический и мультивариантный тайлинг

Метод мультивариантного тайлинга предназначен для бесшовного *последовательного* покрытия больших объемов пространства с малыми затратами памяти по сравнению с ресурсоемким одноразовым генерированием огромной текстуры.

Библиографический список

1. **Ebert, D.S.** Texturing and modeling. A procedural approach. Third edition / D.S. Ebert [et al.]. – San Francisco, CA: Morgan Kaufmann Publishers Inc., 2002. – 600 p.
2. **Снук, Г.** Создание 3D-ландшафтов в реальном времени с использованием C++ и DirectX 9: [пер. с англ.] / Г. Снук. – М.: КУДИЦ – ОБРАЗ, 2007. – 368 с.

*Дата поступления
в редакцию 25.01.16*

Е.А. Nikulin

PROCEDURAL NOISE AND MULTIVARIATE TILING

Nizhny Novgorod state technical university n.a. R.E. Alexeev

Subject: Creation of procedural seamless textures with non-repeating tiles.

Purpose: Evolution of methods of procedural seamless texturing.

Methodology: A special choice of node elements of the texture lattice.

Originality: Is provided non-periodical tiling of the large multitextures.

Findings: The method is designed for seamless coverage of large areas at low cost resources.

Key words: texture, cellular noise, tiling.