

МОДЕЛЬ МАРКОВСКОГО СЛУЧАЙНОГО ПОЛЯ В ЗАДАЧАХ СИНТЕЗА И АНАЛИЗА ТЕКСТУРНЫХ ИЗОБРАЖЕНИЙ

© 2008 А.И. Пластинин, А.В. Куприянов

Институт систем обработки изображений РАН
Самарский государственный аэрокосмический университет

В работе предлагается алгоритм имитационного моделирования текстурных изображений по образцу на основе модели марковских случайных полей. Представлены результаты экспериментальных исследований качества генерации текстур. Анализ проведенных экспериментальных исследований свидетельствует о том, что разработанный алгоритм может эффективно применяться в исследованиях связанных с текстурным анализом

Текстурные изображения, марковские случайные поля, имитационное моделирование, генерация текстур, текстурный анализ

Введение

При проведении экспериментов по решению прикладных задач анализа текстурных изображений часто на практике доступно лишь небольшое количество натуральных изображений. Возникает необходимость в синтезировании изображений, повторяющих структуру заданного оригинала. Существуют различные классы алгоритмов для синтеза. Как правило, для синтеза одного типа изображений подбирается своя модель, по которой и происходит построение нового изображения. Один из наиболее распространенных – метод ЛИС-систем. Метод позволяет получить изображение с заданной корреляционной функций. Есть возможность его эффективной реализации на базе преобразования Фурье. Однако этот метод обладает рядом ограничений и не позволяет воспроизвести особенности сложных, нестационарных изображений.

В настоящей работе рассматривается универсальный метод и эффективный алгоритм синтезирования изображения по образцу на основе модели марковских случайных полей (МСП). Предлагаемый алгоритм является развитием идеи, представленной в [1].

1. Модель марковского случайного поля

Пусть S – конечное множество индексов, множество узлов. Тогда для любого $s \in S$, X_s определяется как конечное пространство состояний x_s . Произведение

$X = \prod_{s \in S} X_s$ – пространство конфигураций $x = (x_s)_{s \in S}$. Рассмотрим вероятностную меру или распределение P на X , т.е. $P(x) \geq 0$ и $\sum_{x \in X} P(x) = 1$. Подмножества $E \subset X$ называются событиями, вероятность события определяется как $P(E) = \sum_{x \in E} P(x)$.

Если $P(x) > 0$, тогда случайный вектор X над вероятностным пространством (X, P) будем называть случайным полем.

Для событий E и F условная вероятность F при условии E определяется как $P(F|E) = P(F \cap E)/P(E)$. Условные вероятности вида $P(X_A = x_A | X_{S \setminus A} = x_{S \setminus A})$, где $A \subset S$, $x_A \in X_A$, $x_{S \setminus A} \in X_{S \setminus A}$, называются локальными характеристиками. Локальные характеристики всегда определены, т.к. распределение P строго положительно. Далее будем использовать короткую запись $P(x_A | x_{S \setminus A})$.

Можно легко показать [2], что распределение (вероятностная мера) $P(x)$ однозначно определяется локальными характеристиками. Совокупность $\mathbf{N} = \{N_s : s \in S\}$ подмножеств множества S называется системой окрестностей, если $s \notin N_s$. Узел $s \in N_t$ называется соседом узла t . Окрестность называется симметричной, если выполнено условие: $s \in N_t$, тогда и только тогда, когда $t \in N_s$.

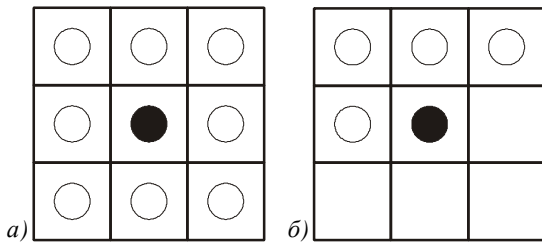


Рис. 1. Примеры окрестностей: некаузальная 3×3 (а); каузальная 3×3 (б)

Случайное поле X над пространством (X, P) называется марковским случайным полем (МСП) по отношению к системе окрестности \mathbf{N} , если для любого $x \in X$ выполнено:

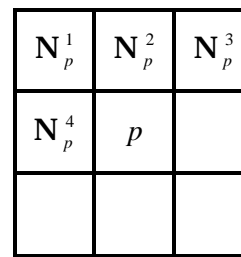
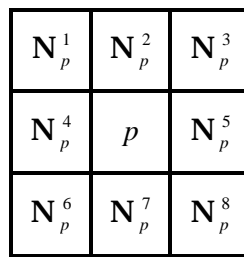


Рис. 2. Порядок пикселей в окрестности: некаузальная (а); каузальная (б)

Алгоритм состоит в построении последовательности изображений: I^0, I^1, \dots, I^L , где L – параметр алгоритма. I^0 инициализируется белым шумом. Одна итерация заключается в проходе по всем точкам изображения I^i , полученного на предыдущем шаге: $p \in D(I^i)$. Новое изображение формируется по следующему правилу: $I^{i+1}(p) = I(p^*)$, $p^* = \arg \min_{p \in D(I)} Q(N_p(I), N_p(I^i))$. Здесь

$Q(N_1, N_2)$ – критерий сходства двух векторов N_1 и N_2 . В качестве такого критерия можно взять евклидово расстояние между векторами: $Q(N_1, N_2) = \|N_1 - N_2\|$.

Окрестность для генерации может быть выбрана произвольная. В данном случае мы будем рассматривать два типа окрестностей: каузальную и некаузальную. Следует отметить, что использование каузальной окрестности дает лучший результат, т.к. при генерации используются только те пиксели, которые были получены на предыдущих шагах алгоритма. В случае некаузальной окрестности при синтезе текущего пикселя используются пиксели, еще не обработанные алгоритмом.

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathbf{N}_s).$$

2. Алгоритм синтеза изображений

Введем следующие обозначения: \mathbf{N}_p – окрестность пикселя p ; \mathbf{N}_p^k – k -ый элемент окрестности; $N_p(I) = (I(\mathbf{N}_p^1), \mathbf{K}, I(\mathbf{N}_p^K))$ – вектор значений.

Пример возможной нумерации пикселей приведен на рис. 2.

В общем случае пусть задано множество S из k векторов пространства \mathbb{R}^n . Необходимо для заданного вектора $x \in \mathbb{R}^n$ найти вектор $y \in S$, наиболее близкий к x : $y = \arg \min_{y \in S} d(x, y)$. В качестве меры сходства можно использовать евклидово расстояние $d(x, y) = \|x - y\|$.

3. Исследования алгоритма имитационного моделирования

Алгоритм исследовался на различных типах окрестностей (каузальной и некаузальной) и размерах (5×5, 7×7, 9×9, 11×11), использовались различные цветовые пространства (RGB и Ruderman Lab [4]), каналы генерировались совместно и по отдельности. Во всех случаях рассматривался алгоритм с кластеризацией, т.к. он обладает большей скоростью работы.

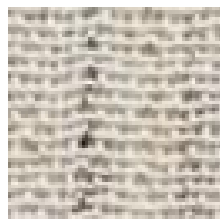
На рис. 3 и 4 показаны образцы и синтезированные по ним изображения размера 128×128 соответственно.

Видно, что алгоритм воспроизвел основные структурные элементы образца с сохранением взаимного расположения.

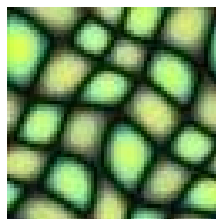
На рис. 5 представлены результаты генерирования с некаузальной окрестностью.

Видно, что некаузальная окрестность дает плохой результат. Это связано с тем, что при синтезе в окрестность наряду с уже об-

работанными пикселями в текущей итерации попадают отсчеты, которые не были обработаны.



а)

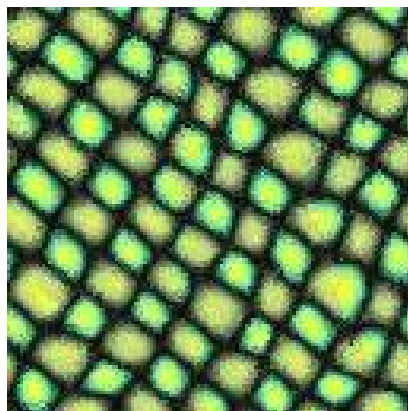


б)

Рис. 3. Примеры тестовых изображений: «тип 1» (а); «тип 2» (б)



а)

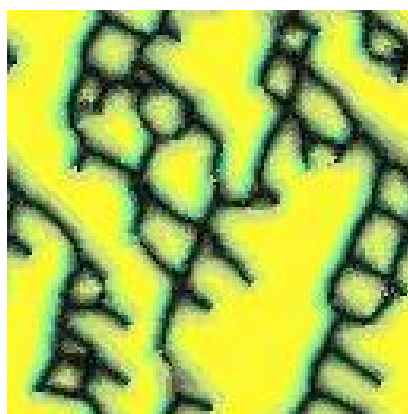


б)

Рис. 4. Синтезированные изображения, каузальная окрестность 11×11 , цветовые компоненты RGB генерировались совместно, 2 итерации, алгоритм с кластеризацией: «тип 1» (а); «тип 2» (б)



а)



б)

Рис. 5. Синтезированные изображения, некаузальная окрестность 11×11 , цветовые компоненты RGB генерировались совместно, 2 итерации, алгоритм с кластеризацией: «тип 1» (а); «тип 2» (б)

Далее на рис. 6 и 7 показаны результаты генерирования изображений в разных цветовых пространствах, причем цветовые каналы генерировались по отдельности.

На рис. 8 показаны результаты генерирования с различными размерами каузальных окрестностей.

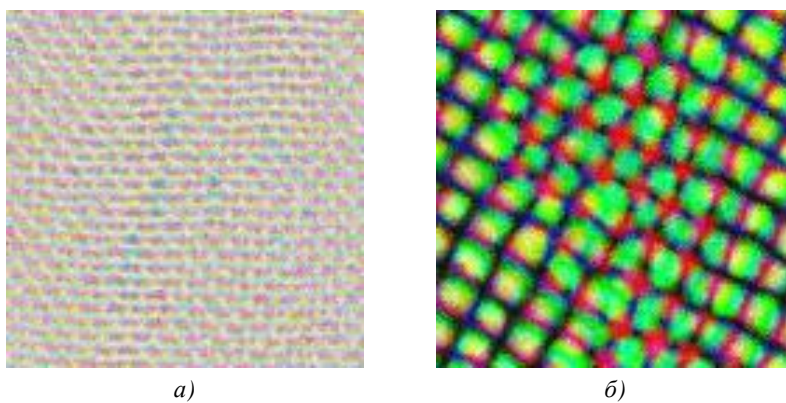


Рис. 6. Синтезированные изображения, каузальная окрестность 11×11 , цветовые компоненты RGB генерировались по отдельности, 2 итерации, алгоритм с кластеризацией: «тип 1» (а); «тип 2» (б)

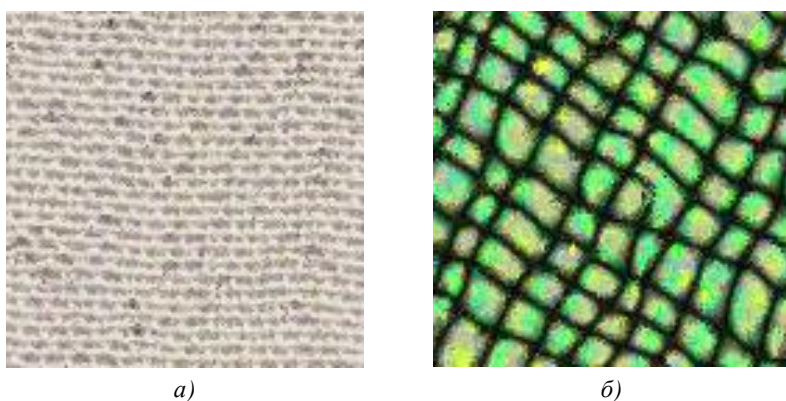


Рис. 7. Синтезированные изображения, каузальная окрестность 11×11 , цветовые компоненты Ruderman Lab генерировались по отдельности, 2 итерации, алгоритм с кластеризацией: «тип 1» (а); «тип 2» (б)

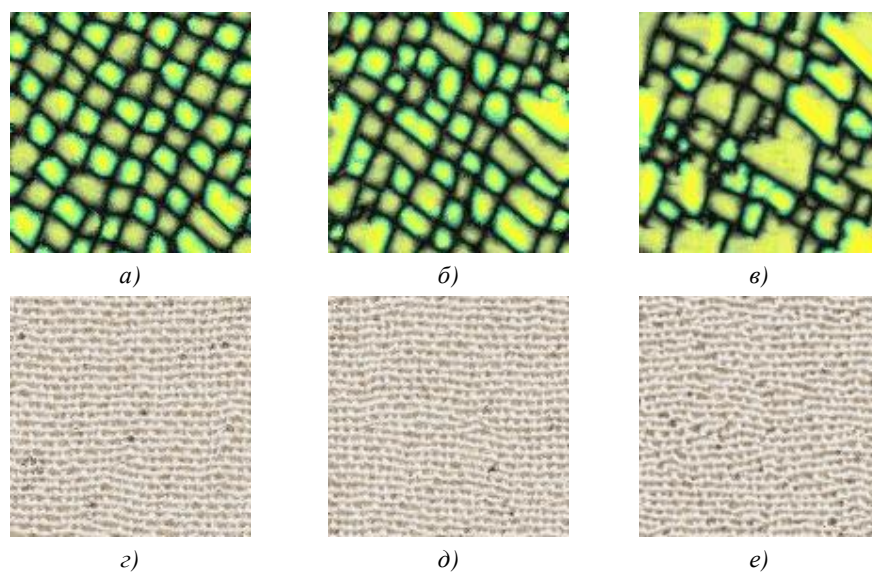


Рис. 8. Синтезированные изображения, каузальная окрестность, цветовые компоненты RGB генерировались совместно, 2 итерации, алгоритм с кластеризацией: «тип 2» 9×9 (а); «тип 2» 7×7 (б); «тип 2» 5×5 (в); «тип 1» 9×9 (г); «тип 1» 7×7 (д); «тип 1» 5×5 (е)

Заключение

В работе предложен метод имитационного моделирования текстурных изображений, проведены экспериментальные исследования разработанного алгоритма.

Полученные результаты позволяют сделать следующие выводы:

1. Необходимо использовать одновременно все цветовые компоненты. Декорреляция цветового пространства не дает желаемого результата.

2. Необходимо использовать каузальную окрестность для повышения эффективности моделирования.

Анализ проведенных экспериментальных исследований свидетельствует о том, что разработанный алгоритм может эффективно применяться в исследованиях связанных с текстурным анализом.

Благодарность

Работа выполнена при поддержке российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE) и программы Президиума РАН «Фундаментальные науки – медицине», гранта РФФИ № 06-07-08006-офи, гранта РФФИ № 07-08-96611.

Библиографический список

1. **Li-Yi, Wei.** Deterministic Texture Analysis and Synthesis using Tree Structure Vector Quantization [текст].
2. **Winkler, G.** Image Analysis, Random Fields and Dynamic Monte Carlo Methods [текст] / G. Winkler– Springer-Verlag, 1995.
3. **Ту, Дж.** Принципы распознавания образов [текст] / Дж. Ту, Р. Гонсалес – М.: МИР, 1978. – 412 с.
4. **Ruderman, D.L.** Statistics of Cone Responses to Natural Images: Implications for Visual Coding [текст] / D.L. Ruderman, T.W. Cronin, and C.C. Chiao, // J. Optical Soc. of America. – 1998. – Vol. 15, no. 8. – P. 2036-2045.

Сведения об авторах:

Пластинин Анатолий Игоревич, Самарский государственный аэрокосмический университет имени академика С.П. Королева, аспирант. Учреждение Российской академии наук Институт систем обработки изображений РАН, Самара, Россия. Стажер-исследователь лаборатории лазерных измерений. Опубликовано 6 работ в отечественных и зарубежных изданиях. Область научных интересов: pattern recognition, machine learning, intelligent data analysis, image processing, texture analysis, random process generation.

Куприянов Александр Викторович, Учреждение Российской академии наук Институт систем обработки изображений РАН, Самара, Россия. Кандидат технических наук. Старший научный сотрудник лаборатории лазерных измерений. Опубликовано свыше 50 работ в отечественных и зарубежных изданиях. Область научных интересов - обработка биомедицинских изображений, текстурный анализ, локальные и спектральные преобразования, биометрическая идентификация.