



Рис 2. фрагмент диалогового дерева, использованного для реализации мобильной игры “DON’T MAKE ME AN ORPHAN, MOM”.

#### ЛИТЕРАТУРА

1. EDTree: Emotional Dialogue Trees for Game Based Training Jay Collins, William Hisrt, Wen Tang, Colin Luu, Peter Smith, Andrew Watson, and Reza Sahandi Department of Creative Technology, Faculty of Science and Technology, Bournemouth University, Poole, Dorset BH12 5BB, UK,
2. Emotional Dialogue in a Virtual Tutor for Educational Software María Lucila Morales-Rodríguez, Juan J. González B., Rogelio Florencia-Juárez, and Julia Patricia Sánchez-Solís División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Tamaulipas, México
3. Aleynikov V.V., Shabalina, O., (2017). “DON’T MAKE ME AN ORPHAN, MOM”: A MOBILE GAME TO SUPPORT SMOKING CESSATION Paper presented at the Proceedings of the International Conferences on ICT, Society and Human. Beings 2017, Web Based Communities and Social Media 2017, e-Commerce 2017, Information Systems Post-Implementation and Change Management 2017 and e-Health 2017 - Part of the Multi

#### СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ АДАПТИРОВАННЫМ АЛГОРИТМОМ ОБЕЗЬЯН НА ОСНОВЕ К-СРЕДНИХ

*Штучный А.М. Курейчик В.М.*

*(г. Таганрог, Институт компьютерных технологий и информационной безопасности)  
zdumbz@yandex.ru*

#### SEGMENTATION OF IMAGES BY AN ADAPTED MONKEY ALGORITHM BASED ON K-MEANS

*Shtuchny AM Kureichik V.M.*

*(Taganrog, Institute of Computer Technologies and Information Security)*

**Annotation.** The article presents theoretical development of adaptation of the monkey algorithm based on k - means. Segmentation of images and the work of genetic algorithms are considered. The analysis

of the shortcomings of the k-means algorithm is carried out. The advantages and disadvantages of using this approach to image segmentation are revealed. The novelty is the use of the monkey algorithm for image segmentation. The article draws conclusions about the prospects of research in this direction and suggests the possibilities of using the results of the work.

**Key words:** visual and measuring control, design automation systems, image segmentation, monkey algorithm, clustering.

**Введение.** На текущий момент существует множество систем автоматизирующих процесс визуального и измерительного контроля печатных плат. Многие такие системы используют предварительную обработку изображений, которая включает в себя сегментацию изображений. Под сегментацией понимают процесс разделения оцифрованного изображения на несколько сегментов. Целью является упрощение изображения, или изменение его представления, для упрощения его дальнейшего анализа. Большинство алгоритмов сегментации работают только с изображениями в оттенках серого, для сегментации цветного изображения необходимо разложить изображение на цветовые компоненты и работать с ними как с яркостными. (например *RGB* изображение нужно разложить на три – красный, синий и зеленый).

**Постановка проблемы.** Алгоритм кластеризации – это функция  $a: X \rightarrow Y$ , которая любому объекту  $x \in X$  ставит в соответствие номер кластера  $y \in Y$ . Множество  $Y$  в некоторых случаях известно заранее, однако чаще ставится задача определения оптимального числа кластеров с точки зрения того или иного критерия качества кластеризации. При решении задачи кластеризации метки исходных объектов  $y_i$  изначально не заданы, и даже само множество  $Y$  может быть неизвестно.[1] Одним из простейших алгоритмов кластеризации данных является алгоритм *K-средних* (*k-means*). Он предполагает быстрый кластерный анализ путем выделения  $K$  сегментов (кластеров), которые располагаются на максимальном расстоянии друг от друга, где центры кластеров соответствуют локальным максимумам плотности распределения данных. Базовый алгоритм *K-средних* предполагает случайный или эвристический выбор  $K$  центров кластеров, размещение каждого пикселя изображения в кластер с ближайшим центром к этому пикселю, после чего заново пересчитываются центры кластеров до сходимости процесса[2].

Недостатком такого метода для задачи сегментации изображений САПР, можно назвать: необходимость точно знать количество кластеров, что сложно выполнимо при сегментации изображения; точность сегментации зависима от начального выбора множества центров сегментов. В качестве возможного решения этих проблем по пробуем использовать алгоритм обезьян (АО).

АО - это новый вид эволюционного алгоритма, который может решить множество сложных проблем оптимизации с нелинейностью, недифференцируемостью и высокой размерностью. Отличие от других алгоритмов заключается в том, что время, затрачиваемое АО, заключается главным образом в использовании процесса набора высоты для поиска локальных оптимальных решений. Поэтому в соответствии с характеристиками проблемы сегментации предлагается новый алгоритм обезьян соединить с алгоритмом *K-средних* в качестве последнего шага заменив им процесс глобальных прыжков алгоритма обезьян. Далее опишем основные процессы алгоритма обезьян[3,4]:

1. Процесс инициализация популяции обезьян
2. Процесс движения вверх (climb process)
3. Процесс локальных прыжков (watch-jump process)
4. Процесс глобальных прыжков (somersault process)

**Представление решения.** Определяем  $M$  как размер популяции обезьян. Инициализируем обезьяну  $i$ , и её положение обозначается как вектор  $Xi = (x_{i1}, x_{i2}, \dots, x_{iK * p})$ , где  $K$  равно числу центров сегмента и каждый центр сегмента включает в себя компоненты. Это положение будет использовано для выражения решения проблемы оптимизации.

**Процесс инициализации популяции обезьян.** Процесс инициализации популяции хоть оказывает влияние на точность, но для задачи сегментации изображений САПР каждый компонент данных имеет разные интервалы, что несколько нивелирует оказываемое влияние выбора начальной популяции. Для  $i$  - агента ( $i$  обезьяны) произвольно выберем  $K$  выборки (каждый образец включает в себя  $p$  компоненты) из набора данных.

**Процесс движения вверх.** Процесс движения вверх - поэтапная процедура для изменения позиций обезьян от исходных позиций к новым, которые могут улучшить целевую функцию. Процесс движения вверх предназначен для использования идеи одновременной стохастической аппроксимации возмущения на основе псевдоградиента (SPSA) [5,6], своего рода рекурсивный алгоритм оптимизации. Для  $i$ -агента ее позиция  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,K * p})$ ,  $i = 1, 2, \dots, M$  соответственно.  $f(X_i)$  - соответствующее значение пригодности. Улучшенный процесс поднятия задается следующим образом:

- 1) Произвольно генерируем два вектора  $\Delta x_i = (\Delta x_{i,1}, \Delta x_{i,2}, \dots, \Delta x_{i,K * p})$ , где

$$\Delta x_{ij} = \begin{cases} a & \text{с вероятностью } \frac{1}{2} \\ -a & \text{с вероятностью } \frac{1}{2} \end{cases}$$

$j = 1, 2, \dots, K * p$  соответственно. Параметр  $a$  ( $a > 0$ ), называемый шагом процесса набора высоты, может быть определен конкретными ситуациями. Длина шага  $a$  играет решающую роль в точности аппроксимации локального решения в процессе набора высоты. Обычно, чем меньше параметр  $a$ , тем точнее решения.

- 2)  $f'_{ij}(X_i) = \frac{f(X_i + \Delta x_i) - f(X_i - \Delta x_i)}{2\Delta x_{ij}}$

Где  $j = 1, 2, \dots, K * p$  соответственно. Вектор  $f'_{ij}(X_i)$  называется псевдоградиентом целевой функции в точке  $X_i$ , и  $f'_{ij}(X_i) = (f'_{i,1}(X_i), f'_{i,2}(X_i), \dots, f'_{i,K * p}(X_i))$

- 3) Установим  $y_j = x_{ij} + a * \text{sign}(f'_{ij}(X_i))$ ,  $j = 1, 2, \dots, K * p$ , соответственно, и положим  $Y = (y_1, y_2, \dots, y_{K * p})$
- 4) Обновим множество значений  $X_i$  множеством  $Y$ , если множество  $Y$  возможно, в противном случае оставляем  $X_i$  без изменений
- 5) Повторим шаги (1) - (4), пока не будет достигнуто максимально допустимое количество итераций (называемое номером набора высоты, обозначаемое  $N_c$ ).

**Процесс локальных прыжков.** После процесса подъема каждый агент прибывает на свой собственный горный склон, локальный максимум. Тогда он должен осмотреться и определить, есть ли другие точки вокруг него выше, чем его текущая. Если да, он будет прыгать туда из текущей позиции, а затем повторять процесс набора высоты, пока не достигнет вершины горы. Для  $i$ -агента позиция равна  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,K * p})$ ,  $i = 1, 2, \dots, M$ . Процесс локальных прыжков задается следующим образом:

- 1) Произвольно выбрать числа для  $y_j$  из  $(x_{ij} - b, x_{ij} + b)$   $j = 1, 2, \dots, K * p$  соответственно, положим  $Y = (y_1, y_2, \dots, y_{K * p})$ , и параметр  $b$  называется зрением обезьян, которые можно определить по конкретным ситуациям. Обычно, чем больше допустимое пространство оптимальной задачи, тем значение  $b$  должно быть принято больше.

- 2) Обновим  $X_i$  с помощью  $Y$  при условии, что и  $f(Y) \geq f(X_i)$  и  $Y$  возможны. В противном случае повторим шаг 1), пока не будет найдена соответствующая точка. Для задачи сегментации мы заменяем только  $X_i$  на  $Y$ , значение функции которого меньше или равно  $f(X_i)$ .

3) Повторим процесс набора высоты, используя  $Y$  в качестве начальной позиции.

**Процесс глобального прыжка на основе  $K$  - средних.** После повторений процесса набора высоты и процесса локальных прыжков каждая обезьяна найдет локальную максимальную вершину вокруг своей начальной точки. Чтобы найти гораздо более высокую вершину горы, для каждого агента естественно переходить в новый участок поиска. В оригинальном АО, агенты используют процесс глобального прыжка в направлении, указывающем на ось вращения, которая равна центру положений всех текущих положений агентов.

Агент может легко оставить интервал сегментации, и все агенты утратят разнообразие популяции из-за процесса глобального прыжка по направлению, указывающему на ось вращения после многих итераций. Здесь мы выберем центр пикселей, принадлежащих сегменту, чтобы заменить процесс кувырка АО алгоритмом  $i$  - средних. Для  $i$  агента позиция  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,K * p})$ ; измененный процесс глобального прыжка на алгоритм  $k$  - средних дается следующим образом.

(1) Назначаем каждый объект группе, имеющей ближайший центроид  $G_1, G_2, \dots, G_K$  в зависимости от местонахождения агента  $i$ .

(2) Произвольно создаем действительные числа  $\theta$  из интервала  $[c, d]$  (называемый интервалом глобального прыжка, который определяет максимальное расстояние, для процесса глобального прыжка агентов).

(3) Вычислим  $K$  позиций  $c^*_1, c^*_2, \dots, c^*_K$ , которые являются центрами целей, относящихся к центроиду  $G_1, G_2, \dots, G_K$  согласно формуле (1), соответственно.

Позиции  $K$  образуют вектор, представляющий центр для замены центра агентов. Пусть  $c = (c^*_1, c^*_2, \dots, c^*_K) = (c^*_1, c^*_2, \dots, c^*_{K * p})$ .

(4) Установим  $y_j = x_{ij} + \theta (c_j - x_{ij}), j = 1, 2, \dots, K * p$ , соответственно.

(5) Обновим  $X_i$  данными  $Y$  при условии, что и  $f(Y) \geq f(X_i)$  и  $Y$  возможны. В противном случае создаем новое решение для замены  $X_i$ .

**Выводы.** В этой статье предлагается адаптированный АО на основе  $k$ -средних, и оригинальной АО который позволяет проводить сегментацию изображений. Преимуществом которого является самостоятельное определение количества кластеров и их центров.

В последующих работа планируется провести сравнительный анализ алгоритмов сегментации с автоматическим определением количества кластеров в том числе и основанных на алгоритме  $k$  - средних, с предложенным в статье алгоритмом на изображениях САПР.

#### ЛИТЕРАТУРА

1. Курейчик В.М., А.С. Григораш - Программный комплекс решения задачи кластеризации, [Электронный ресурс], <https://elibrary.ru/item.asp?id=29877268>
2. Эль-Хатиб С.А. - Разработка и исследование методов сегментации изображений с применением бионических моделей, [Электронный ресурс], [http://hub.sfedu.ru/media/diss/04acfd27-750f-4644-83ba-105238eb0219/Автореферат\\_Эль-Хатиб\\_\(05.13.17\).pdf](http://hub.sfedu.ru/media/diss/04acfd27-750f-4644-83ba-105238eb0219/Автореферат_Эль-Хатиб_(05.13.17).pdf)
3. Карпенко А.П. – Современные алгоритмы оптимизации. Алгоритмы вдохновленные природой. -М:Издательство МГТУ им Н.Э Баумана, 2014.
4. Ruiqing Zhao Wansheng Tang - Monkey Algorithm for Global Numerical Optimization, Journal of Uncertain Systems Vol.2, No.3, pp.165-176, 2008
5. J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no 3, pp. 462–466, 1952.
6. J. C. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Technical Digest*, vol. 19, no. 4, pp. 482–492, 1998.