

---

УДК 658.012:681.32:621.38

**Л.И. Тимченко**, д-р. техн. наук,  
**В.В. Шпакович**, аспирант, **Н.И. Кокряцкая**, канд. техн. наук  
Государственный экономико-технологический университет транспорта,  
(Украина, 03049, г. Киев-49, ул. Н. Лукашевича, 19,  
тел. (044) 5915135, 0673550976; e-mail: timchen@list.ru)

## **Моделирование метода параллельно-иерархического преобразования с формированием нормирующего уравнения для быстрого распознавания динамических изображений**

Разработан метод и компьютерные средства параллельно-иерархического (ПИ) преобразования изображений с использованием высокопроизводительных GPU-адаптеров. Предложены математические модели и метод обучения ПИ сети для распознавания динамических образов.

Розроблено метод і комп'ютерні засоби паралельно-ієрархічного (ПІ) перетворення зображень із застосуванням високопродуктивних GPU-адаптерів. Запропоновано математичні моделі та метод навчання ПІ мережі для розпізнавання динамічних образів.

*К л ю ч е в ы е с л о в а:* параллельно-иерархическое преобразование, обучение ПИ сети, динамические изображения, нормирующее уравнение, классификация, лазерные трассы.

В настоящее время становится актуальным вопрос о реализации высокопроизводительных скоростных вычислений, что объясняется не только увеличением сверхбольших объемов информации, но и возрастанием их вычислительной сложности. Оперативная реакция и принятие решений в режиме реального времени, обработка видеоданных с большой разрешающей способностью, классификация и прогнозирование быстропеременных динамических данных являются наиболее актуальными задачами [1—3].

При реализации высокопроизводительных вычислительных компьютерных средств, которые достаточно активно развиваются, возникает много сложных прикладных задач, успешно решаемых с помощью быстрых вычислений. Кроме преимуществ по сравнению с обычными компьютерами Гарвардской архитектуры нейрокомпьютеры имеют существенный недостаток — высокую стоимость, что обуславливает необходимость искать определенную альтернативу.

© Л.И. Тимченко, В.В. Шпакович, Н.И. Кокряцкая, 2013

Одним из самых эффективных способов обработки больших массивов данных является их параллельная обработка на основе специализированных системных решений, в частности нейроподобных параллельно-иерархических (ПИ) систем. Однако при этом возникает задача программно-аппаратной реализации таких систем, а именно выбор адекватной платформы для быстрой и эффективной параллельной обработки больших массивов данных.

Иерархическое преобразование образа заключается в последовательном преобразовании совпадающих и выявлении (фильтрации) несовпадающих во времени составляющих образа при переходе элементов нейронной сети из текущих энергетических состояний с определенными пространственными координатами в состояния с другими пространственными координатами и с меньшей энергией.

Образ — это матрица, составленная из значений связностей пикселей изображения фрагмента лазерной трассы. Условием перехода составляющих образа на следующий уровень ПИ сети, т.е. детерминированного графа ПИ преобразования, является наличие динамики взаимного совпадения промежуточных результатов обработки во времени в одноименных каналах нижнего уровня. Результат анализа образа формируется из изолированных в пространственно-временной области составляющих образа.

Иерархическое преобразование является важной и мощной вычислительной операцией. Это мотивирует исследование параллельных подходов, таких как графические обрабатывающие модули, а также алгоритмы с низкоуровневой сложностью, например, используемые в адаптивной резонансной теории (АРТ). Несмотря на то что большое число исследований АРТ выполнено на процессорах Graphics Processing Units (GPU), иерархические сети применялись в технологиях АРТ GPU достаточно редко. В каждом слое сети информация распространяется в мультипроцессорах GPU с одновременным обучением. Экспериментальные данные свидетельствуют о том, что для больших деревьев целесообразно использовать графические процессоры, поскольку преимущество GPU является существенным.

Программирование графических процессоров GPU на основе технологии NVIDIA CUDA (Compute Unified Device Architecture) целесообразно в вычислительных системах, особенно для популяционных алгоритмов [4—6]. Это имеет большое значение при использовании программирования GPU для достижения преимуществ по скорости обработки в алгоритмах иерархического преобразования [7, 8].

Применение АРТ для иерархического преобразования представляется целесообразным вследствие скорости и масштабируемости к параллельному преобразованию [9]. В частности, иерархическая АРТ, основанная

на процессорах GPU, исследована недостаточно. Единственным ее недостатком является негибкость использования памяти для внутриядерного значения, если множество массивов с переменными пределами ограничены только центральным процессором CPU.

Формирование типичных алгоритмов древовидной структуры обеспечивают указатели при создании узла и ссылки, которая в случае использования технологии CUDA неэффективна [10].

В работах [11—13] выполнено программирование GPU, ограничиваемое поведением единственной структурной единицы АРТ, однако архитектура такого иерархического преобразования основана на использовании структуры конвейерной обработки [14].

В работе [15] описан метод организации многослойной древовидной структуры, составленной из модулей, выполненных на платформе технологии CUDA. Согласно представленным экспериментальным результатам происходит существенное увеличение производительности на различных наборах данных и улучшение параметров по сравнению с аналогичными, полученными на обычных центральных процессорах CPU.

В отличие от широко распространенных типов кодирования видеoinформации, например разностного [16], кодово-импульсного [17], дельта-модуляции [18], в основу которых положен принцип кодирования разностной информации, предлагается использовать подход, в основу которого положен принцип иерархического преобразования.

Одним из преимуществ использования нормирующего уравнения является тот факт, что с его помощью достаточно просто реализовать предварительную процедуру классификации образов, сформировав коэффициенты настройки и выполняя процедуру определения взвешивающих коэффициентов для каждого класса. Как показали результаты сравнительного анализа [19], для выполнения классификации в реальном времени, например, соседних кадров мультимедийного изображения, применение нормирующего уравнения более эффективно.

**Метод ПИ преобразования с формированием нормирующего уравнения для быстрого распознавания изображений.** Задача обучения предложенной ПИ сети по аналогии с обучением в радиально-базисных сетях практически сводится к контролируемому обучению элементов исходного слоя сети. Используя общую идею структурной организации искусственных нейронных сетей по схеме входной слой — скрытый слой — выходной слой, можно синтезировать обучаемую многослойную ПИ сеть (рис. 1). В качестве входного слоя необходимо использовать первый уровень сети, в качестве скрытого слоя  $k$  следует использовать уровни от 2 до  $k$ , а в ка-

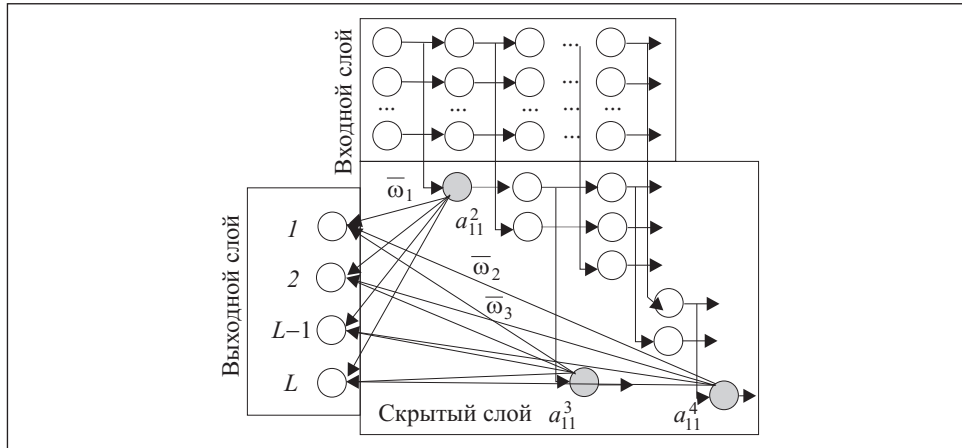


Рис. 1. Структура ПИ сети: 1 — первый класс — недеформированные атмосферой хорошие изображения пятен лазерных пучков; 2 — второй класс — деформированные атмосферой плохие изображения

честве результирующего — выходной слой, традиционно применяемый в искусственных нейронных сетях.

Алгоритм работы ПИ сети сводится к выполнению в каждой ветви ее произвольного уровня операции  $G$  преобразования информации, определению коэффициентов настройки  $w_1 \dots w_{k-1}$ , хвостовых элементов  $a_{11}^2 \dots a_{11}^k$  и нормирующей меры  $d$  с дальнейшим формированием нормирующего уравнения.

Число элементов скрытого слоя можно вычислить исходя из длины сетевого алгоритма и тем самым формализовать процедуру определения числа элементов скрытого слоя. Значения взвешивающих коэффициентов определяем по формуле

$$\bar{w}_t = \frac{\sum_{p=1}^N w_t^{(p)}}{N}, \quad t = \overline{1, k-1},$$

где  $N$  — размерность обучаемой выборки  $P$ .

Составим систему уравнений для определения коэффициентов настройки  $w_1$  от  $w_{k-1}$ :

$$w_1 = \frac{\sum_{t=2}^k a_{11}^t}{\left( a_{11}^2 + \sum_i a_i^2 \right)},$$

$$\begin{aligned}
 w_2 &= \frac{\sum_{t=2}^k a_{11}^t}{\left(a_{11}^3 + \sum_i a_i^3\right)} - \frac{w_1 a_{11}^2}{\left(a_{11}^3 + \sum_i a_i^3\right)}, \\
 &\dots\dots\dots \\
 w_{k-2} &= \frac{\sum_{t=2}^k a_{11}^t}{\left(a_{11}^{k-1} + \sum_i a_i^{k-1}\right)} - \frac{w_1 a_{11}^2 + w_2 a_{11}^3 + \dots + w_{k-3} a_{11}^{k-2}}{\left(a_{11}^{k-1} + \sum_i a_i^{k-1}\right)}, \\
 w_{k-1} &= \frac{\sum_{t=2}^k a_{11}^t}{\left(a_{11}^k + \sum_i a_i^k\right)} - \frac{w_1 a_{11}^2 + w_2 a_{11}^3 + \dots + w_{k-2} a_{11}^{k-1}}{\left(a_{11}^k + \sum_i a_i^k\right)},
 \end{aligned} \tag{1}$$

где  $\sum_{t=2}^k a_{11}^t$  — сумма хвостовых элементов для эталонного изображения распознаваемой протяженной трассы;  $\sum_i a_i^k$  и  $a_{11}^2 \dots a_{11}^k$  — сумма промежуточных элементов для текущего изображения и хвостовые элементы распознаваемой протяженной трассы.

Теперь сформируем нормирующее уравнение:

$$d = \frac{\bar{w}_1 a_{11}^2}{\sum_{t=2}^k a_{11}^t} + \frac{\bar{w}_2 a_{11}^3}{\sum_{t=2}^k a_{11}^t} + \dots + \frac{\bar{w}_{k-2} a_{11}^{k-1}}{\sum_{t=2}^k a_{11}^t} + \frac{\bar{w}_{k-1} a_{11}^k}{\sum_{t=2}^k a_{11}^t} = \frac{\sum_{t=2}^k \bar{w}_{t-1} a_{11}^t}{\sum_{t=2}^k a_{11}^t}. \tag{2}$$

Для нормирования результатов ПИ сети с коэффициентами настройки (1) используем основное свойство ПИ сети:  $\sum_{t=2}^k a_{11}^t = \sum_i a_i$  [19]. При этом правая часть нормирующего уравнения (2) представляет собой отношение суммы произведений усредненных значений коэффициентов настройки и хвостовых элементов к сумме хвостовых элементов сети.

Особенностью объекта исследования является анализ последовательности кадров с изображением пятен лазерных пучков, формирующих отдельные трассы. Каждая трасса формируется из фрагментов (как правило,

от двух до пяти тысяч), полученных от приемного устройства с периодичностью 40 мс. Каждое изображение образа имеет размерность  $128 \times 128$  элементов разложения (э.р). При корректном распознавании изображений пятен лазерных пучков нормирующая мера  $d$  стремится к единице. Степень близости к единице и является мерой подобия распознаваемых изображений.

Используя нормирующее уравнение (2), в котором  $\bar{w}_1 = \bar{w}_2 = \dots = \bar{w}_{k-2} = \bar{w}_{k-1} = 1$ , можно по принятым значениям  $d$  достаточно просто сформировать предварительную процедуру классификации образов, а затем в соответствии с системой (1) определить коэффициенты настройки  $\bar{w}_1 \dots \bar{w}_{k-1}$ , выполняя процедуру определения взвешивающих коэффициентов для каждого класса. В частности, если выполнять классификацию в реальном времени, т.е. разделение изображений на классы осуществлять в темпе поступления входной видеоинформации, и анализировать соседние кадры протяженных трасс лазерных пучков, то нормирующее уравнение (2) принимает следующий вид:

$$d = \frac{(a_{11}^2)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \frac{(a_{11}^3)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \dots + \frac{(a_{11}^{k-1})^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \frac{(a_{11}^k)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j},$$

где  $\left(\sum_{t=2}^k a_{11}^t\right)^j$  — сумма  $k-1$  хвостовых элементов;  $j$  — номер кадра,  $j = \overline{1, m-1}$ ;  $(a_{11}^2)^{j+1} \dots (a_{11}^k)^{j+1}$  — значения хвостовых элементов изображений предыдущего ( $j$ -го) и последующего ( $j+1$ -го) кадров.

С учетом указанного выше свойства ПИ сети уравнение (2) принимает вид

$$d = \frac{(a_1)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \frac{(a_2)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \dots + \frac{(a_{N-1})^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j} + \frac{(a_N)^{j+1}}{\left(\sum_{t=2}^k a_{11}^t\right)^j}. \quad (3)$$

Поскольку при вычислении (3) нет необходимости определять хвостовые элементы от  $(a_{11}^2)^{j+1}$  до  $(a_{11}^k)^{j+1}$  во время обработки  $j+1$ -го кадра, в случае нормирования результатов обработки входных данных от  $a_1$  до  $a_N$  время процедуры распознавания существенно сокращается. В отличие от известных структур нейронных сетей, для которых в процедуре распознавания в исходном слое  $j+1$ -го кадра невозможно использовать входные данные  $j$ -го кадра, в предложенной многослойной ПИ сети это ограничение отсутствует.

При реализации ПИ преобразования для быстрого распознавания изображений пятен лазерных пучков наиболее приемлемым является выбор из имеющейся числовой информации среднего значения  $\bar{a}_i^t$  с округлением до ближайшего целого и последующего формирования двух соединенных ПИ сетей для обработки положительных и отрицательных разностных составляющих. В этом случае для каждой из ПИ сетей можно составить систему уравнений, в которой корнями для всех уровней, кроме первого, являются пары коэффициентов настройки вида (1): для положительных разностных составляющих — от  $w_1^{(+)}$  до  $w_{k-1}^{(+)}$ , для отрицательных разностных составляющих — от  $w_1^{(-)}$  до  $w_{k-1}^{(-)}$ . При этом для первого уровня ПИ сети вычисляется только коэффициент настройки  $w_1^{(+)}$ , так как в этом случае имеются лишь положительные разностные составляющие.

При обработке последовательности изображений протяженных лазерных трасс более целесообразно использование нормирующего уравнения для двух соседних кадров изображений, в которых первое изображение принимается в качестве эталонного. Тогда нормирующее уравнение принимает вид

$$d = \frac{\bar{w}_1(a_{11}^2)^j}{\sum_{t=2}^k (a_{11}^t)^{j-1}} + \frac{\bar{w}_2(a_{11}^3)^j}{\sum_{t=2}^k (a_{11}^t)^{j-1}} + \dots + \frac{\bar{w}_{k-1}(a_{11}^{k-1})^j}{\sum_{t=2}^k (a_{11}^t)^{j-1}} + \frac{\bar{w}_k(a_{11}^k)^j}{\sum_{t=2}^k (a_{11}^t)^{j-1}} = \frac{\sum_{t=1}^k \bar{w}_t (a_{11}^t)^j}{\sum_{t=2}^k (a_{11}^t)^{j-1}}, \quad (4)$$

где  $\bar{w}_1 \dots \bar{w}_k$  — коэффициенты настройки, полученные при предварительной обработке на первых изображениях набора;  $(a_{11}^t)^j$  и  $(a_{11}^t)^{j-1}$  — хвостовые элементы текущего и предыдущих изображений.

При обработке последовательности изображений пятен лазерных пучков необходимо обнаруживать наиболее деформированные под влиянием атмосферы плохие изображения и исключать их из анализа трассы. Поскольку нормирующий критерий вида (4) можно находить как для одного кадра лазерной трассы, так и для всей трассы в целом, сформируем множество  $A$  из коэффициентов  $d_i$ :

$$A = (d_1, d_2, \dots, d_N), \quad (5)$$

где  $N$  — число кадров лазерной трассы. Коэффициент  $d$  — это мера соответствия настройки эталонной ПИ сети и настройки текущей сети, поэтому  $i$ -е элементы множества (5), имеющие малые значения, можно классифицировать как плохие изображения:

$$(a_{i,j}^{k(\dots)})_e \cap (a_{i,j}^{k(\dots)})_n = \begin{cases} \text{изображение хорошее, если } d_i \geq \chi, \\ \text{изображение плохое в противном случае,} \end{cases} \quad (6)$$

где  $\chi$  — определенный пороговый критерий.

Поскольку  $0 \leq d \leq 1$ , то  $0 \leq \chi \leq 1$ . Для качественной классификации и обработки последовательности изображений протяженных лазерных трасс можно использовать более жесткий пороговый критерий:  $0,8 \leq \chi \leq 1$ . Для лазерных трасс, имеющих значительные искажения изображений в результате действия атмосферных явлений, рекомендуется устанавливать менее жесткие рамки порогового значения:

$$0,4 \leq \chi \leq 1. \quad (7)$$

**Сравнительный анализ вычислительной эффективности метода ПИ преобразования и существующих методов.** Сравнивая эффективность ПИ преобразования с известными преобразованиями по числу используемых вычислительных операций, видим, что число операций для ПИ преобразования равняется  $N(N+1)$ , где  $N$  — общее число обрабатываемых элементов. Для широко применяемых на практике, например, ортогональных преобразований, число используемых вычислительных операций следующее:

быстрого преобразования Фурье —  $4N^2 \log_2 N$ ;

преобразования Адамара —  $2N^2 \log_2 N^2$ ;

преобразования Хаара —  $4N(N+1)$ .

Отсутствие трудоемких операций свидетельствует о достаточной простоте алгоритма вычислений, реализующего ПИ преобразование, и обеспечивает его эффективность при применении в различных прикладных областях, где необходима высокая степень параллелизма и компактная форма представления данных.

На основе предложенного метода ПИ преобразования для обработки динамических изображений разработан общий алгоритм быстрого распознавания изображений пятен лазерных пучков (рис. 2, а), предусматривающий вызов подпрограммы NetForm, с помощью которой выполняется построение структур сетей для изображений рис. 2, б. Подпрограмма выполняет функцию введения начальной информации в ПИ сеть. Далее происходит выполнение  $G$  преобразования информации [19] во всех уровнях ПИ сети.

Суть  $G$  преобразования информации заключается в следующем. На вход ПИ сети (см. рис. 1, входные элементы ветвей первого уровня) подается вектор входного образа  $\mu = \{a_i\}$ ,  $i = \overline{1, n}$  [19], применяемого в каждой ветви ПИ сети. Тогда

$$\sum_{i=1}^n a_i = \sum_{j^*=1}^R \left( n - \sum_{k=0}^{j^*-1} n_k \right) (a^{j^*} - a^{j^*-1}),$$



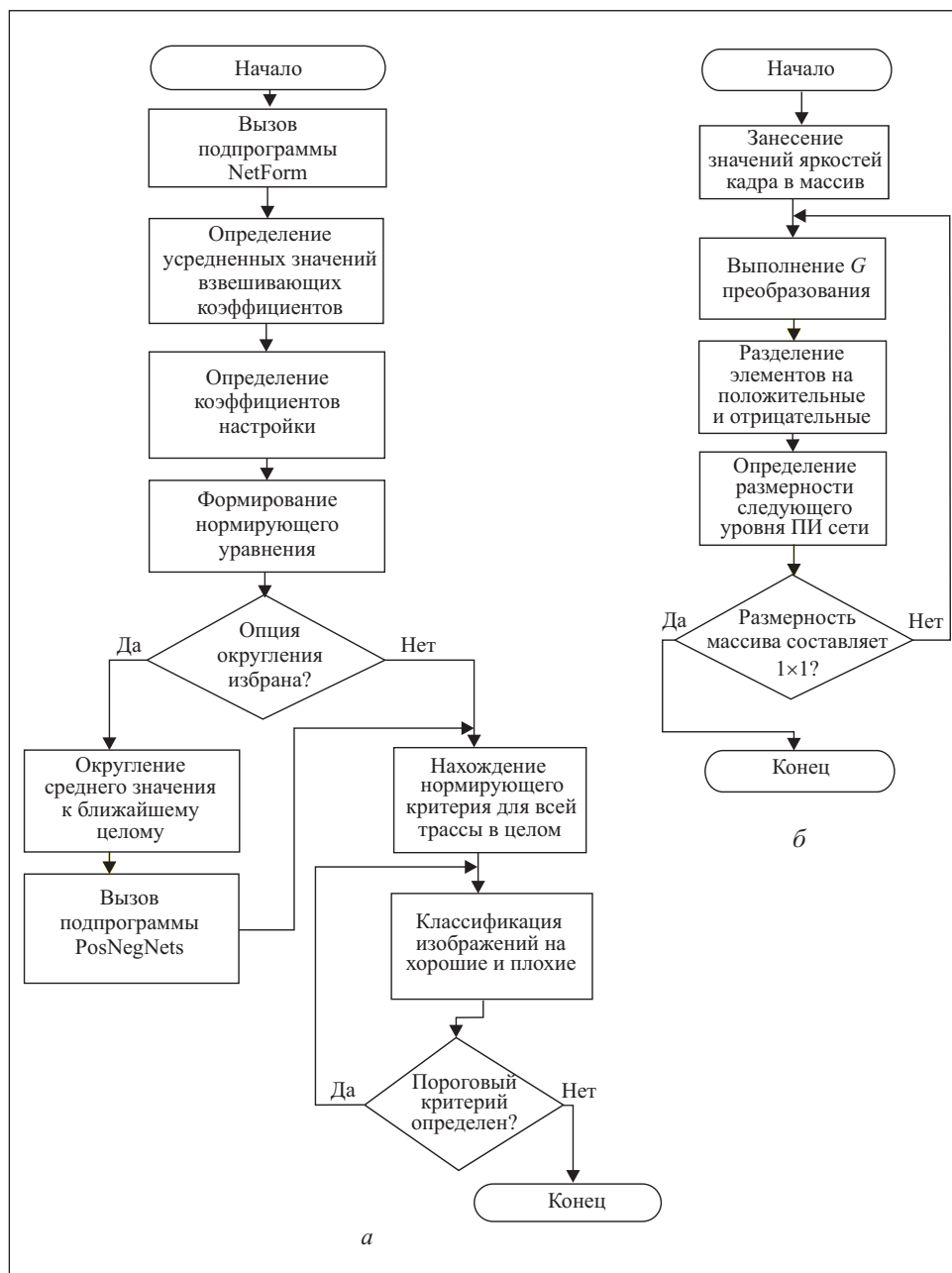


Рис. 2. Структурные схемы общего алгоритма распознавания изображений пятен лазерных пучков (а) и алгоритма подпрограммы создания структуры сетей изображений (б)

где  $a_i \neq 0$ ;  $R$  — размерность данного множества (число его элементов) или размерность ветви каждого уровня ПИ сети (определяется размерностью входного изображения);  $a^k$ ,  $k = \overline{1, R}$ , — элементы подмножеств, состоящих из одинаковых элементов;  $n_k$  — число элементов в  $k$ -м подмножестве (т.е. кратность числа  $a_i$ );  $a^{j^*}$  — элемент множества  $\{a^k\}$ , выбранный на  $j^*$ -м шаге,  $j^* = \overline{1, R}$ ;  $a^0 = 0$ ;  $n_0 = 0$ .

После построения структуры двух сетей определяются усредненные значения взвешивающих коэффициентов и коэффициенты настройки (1), на основании которых формируется нормирующее уравнение (4). Если среднее значение  $\bar{a}_i^j$  округляется до ближайшего целого, то выполняется вызов подпрограммы PosNegNets, с помощью которой происходит формирование двух соединенных ПИ сетей для обработки положительных и отрицательных разностных составляющих. Алгоритм их формирования аналогичен алгоритму формирования структуры сетей для изображений. Затем с помощью (5) определяется нормирующий критерий для всей трассы в целом с последующей классификацией изображений на хорошие и плохие в соответствии с (6). В случае, если пороговый критерий определен, выполняется повторная классификация.

Рассмотрим алгоритм предварительной обработки фрагментов трассы (рис. 3). Изображение фрагмента лазерной трассы, попавшее на вход системы, проходит следующие этапы обработки:

1. Определяем максимум яркости фрагмента для процесса формирования  $n$  краевых линий.
2. После определения необходимого числа контуров определяем их геометрические центры и находим средние параметры для всего кадра.
3. Используя обучаемую выборку и усредненные параметры кадра, формируем пределы туннеля.
4. Проверяем, входят ли рассчитанные параметры кадра в туннель. На основе данной проверки выполняем классификацию изображений кадра. Если изображение классифицировалось как плохое, проводим коррекцию параметров изображения, заносим их в обучаемую выборку и выполняем дополнительное обучение сети.

**Программное моделирование и эффективность применения ПИ преобразования для быстрого распознавания изображений пятен лазерных пучков.** Для моделирования ПИ преобразования разработано программное средство, предназначенное для обработки и классификации изображений пятен лазерных пучков (рис. 4, см. вклейку). Представленная экранная форма созданного программного средства разделена на две части: в левой части расположена панель обработки эталонного кадра трассы изображений пятен лазерных пучков, в правой — панель обра-

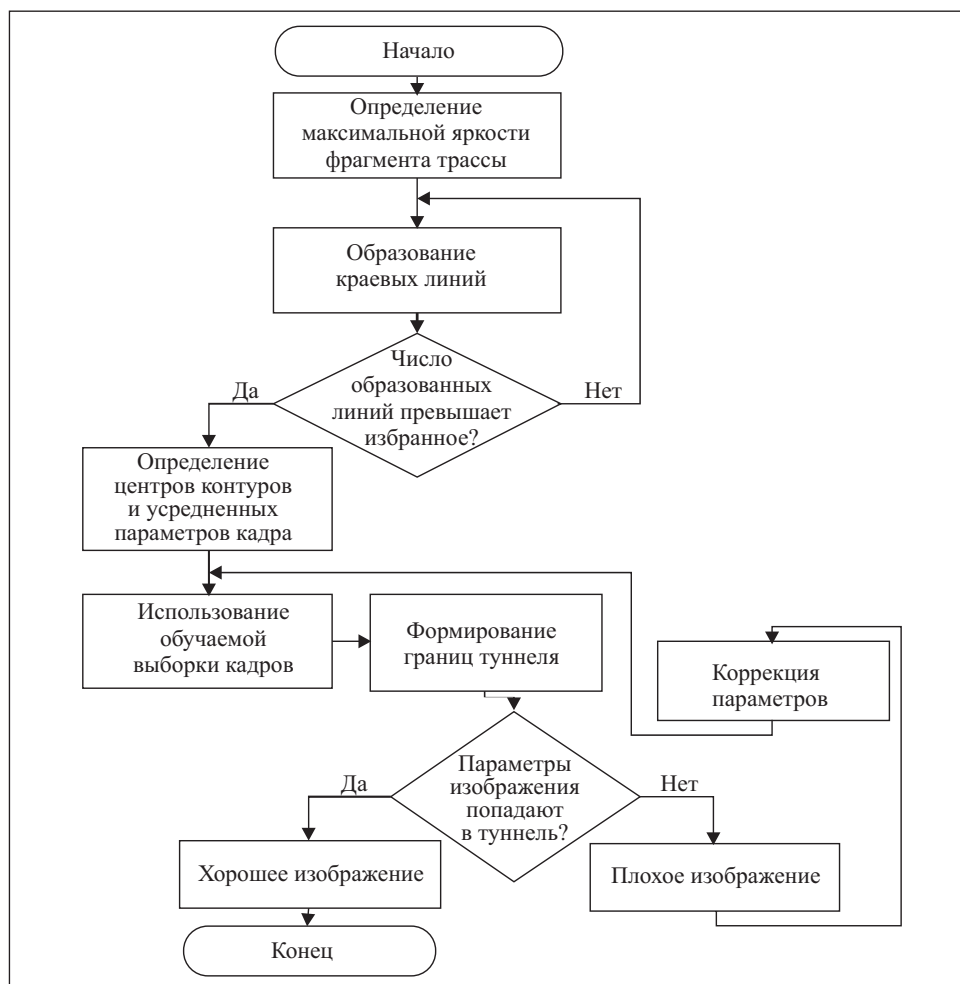


Рис. 3. Алгоритм предварительной обработки фрагментов трассы

ботки текущего кадра трассы изображений пятен лазерных пучков. В нижней части находится информация о параметрах кадра, информация о трассе, в которой содержится кадр, название файла кадра, число кадров в трассе и расположены следующие вкладки:

построение графа сети, который можно получить и сохранить в виде файла графа ПИ сети;

обработка фрагментов трассы (выполняется обработка любого фрагмента трассы);

сравнение изображений (выполняется сравнение и классификация изображений пятен лазерных пучков);

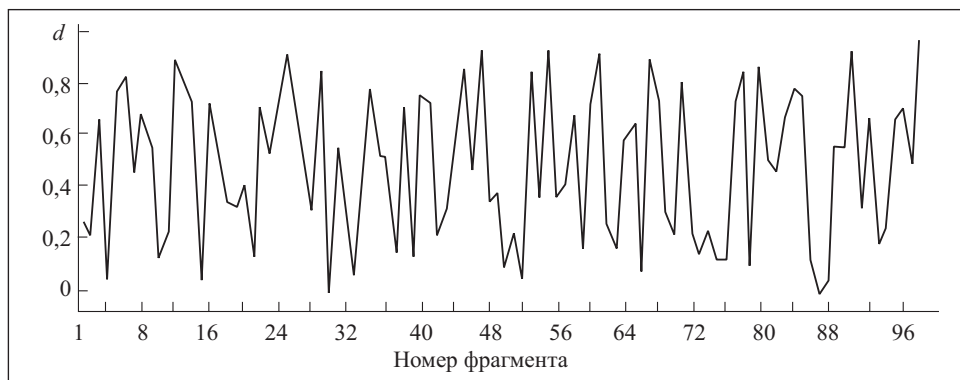


Рис. 5. График определения нормирующей меры для дальнейшего проведения классификации фрагментов трассы № 1

определение энергетических центров (определяются энергетические центры изображений пятен лазерных пучков, а также выполняется обучение ПИ сети).

Эффективность разработанных алгоритмов и созданных на их основе аппаратно-программных средств подтверждают данные, представленные в таблице, на основании которых можно сделать вывод о том, что предложенный метод, алгоритмы и программно-аппаратные средства, позволяют измерять координаты энергетических центров изображений фрагментов лазерных трасс с помощью нормирующего уравнения с точностью не более чем 0,01 э.р. Как видно из таблицы, известные программно-аппаратные средства (например, на основе определения центра тяжести методом моментных признаков [16, 17]) эффективнее приблизительно в 1,5 раза. При этом следует заметить, что время, необходимое для предварительной и сетевой обработки изображений фрагментов лазерных трасс, сокращается.

**Результаты экспериментальных исследований и физическое моделирование ПИ преобразования для быстрого распознавания изображений пятен лазерных пучков.** При проведении экспериментальных исследований использовано четыре трассы, каждая из которых содержит 100 кадров. С помощью уравнения (4) определены нормирующие меры для трассы № 1 (рис. 5) и выполнена классификация ее фрагментов (рис. 6, см. вклейку). Как видно из рис. 6, *a*, хорошие изображения составляют 18 %. В реальных условиях, как правило, невозможно абсолютно точно определить положение энергетических центров. Поэтому для порогового критерия устанавливаем менее жесткие рамки, используя соотношение (7). В этом случае среднее значение хороших изображений составляет 40 и 59 %

(см. соответственно рис. 6, б и в). Далее выполняем обучение ПИ сети для повторной обработки плохих фрагментов трассы.

При физическом моделировании метода ПИ преобразования и определении координат изображений протяженных лазерных трасс с использованием графического адаптера Radeon X1300 требуется приблизительно 0,6 с для обработки одного изображения. Обработка изображений происходит на графических процессорах GPU, ядра которых выполняются на всех элементах, покрытых исходным регионом. Единственным очевидным способом расчета скаляра от входного вектора является использование  $1 \times 1$  исходных элементов и ядра, прочитываемого во всех значениях от входной текстуры.

Однако этот подход имеет недостатки. Во-первых, только один из параллельных элементарных процессоров будет занят. Во-вторых, возможно превышение максимума, разрешенного шейдерной длиной и статической инструкцией расчета на некоторые технические средства. Поэтому будем выполнять параллельное действие сокращения, основанное на глобальных методах коммуникации на параллельных компьютерах.

Параллельное вычисление на GPU обеспечивает коррекцию размеров входной и исходной текстур и элементов индексов. Для рассматриваемого вектора  $\mathbf{M}$  длины  $M$  выход из первого шага составляет  $M/2$  с текстурой  $M/2$ . Для каждого из элементов вектора  $\mathbf{M}$  координаты для входной текстуры скорректированы так, что они соответствуют розьединенным  $2 \times 2$

#### Сравнительная характеристика средств ПИ преобразования

Параметр	Показатели программно-аппаратного средства	
	известного [19] <sup>1</sup>	предлагаемого <sup>2</sup>
Средняя величина хороших фрагментов трасс (%)	50	18
Средняя величина корректного распознавания (%)	84,8	94
Точность определения энергетического центра и элементов разложения (э.р)	1,2	0,01
Среднее время распознавания сетевой обработки (с)	8,4	1,52
Среднее время предварительной обработки фрагментов трассы (с)	3,32	0,6

Примечания: <sup>1</sup> — центральный процессор, метод ПИ преобразования на основе  $Q$ -разложения; <sup>2</sup> — графический процессор, метод обучения ПИ сети на основе нормирующего уравнения

субобластям, значения в которых затем сравниваются. Процедура рекурсивно повторяется, пока текстура  $2 \times 2$  не будет уменьшена до финального значения  $1 \times 1$  скалярной текстурой.

Следующие серии изображений подытоживают первый шаг сокращения алгоритма  $8 \times 8$  входной текстуры (рис. 7, см. вклейку). Каждый исходный элемент, представленный на рис. 7, а, содержит местный максимум передачи  $2 \times 2$  субобласти во входной текстуре.

При определении координат энергетических центров фрагментов трассы № 1 выявлены следующие статистические закономерности. При обработке 100 изображений лазерной трассы 18 % составляют хорошие изображения и 82 % — плохие. При этом среднее значение  $d$  для хороших изображений составило 80 %, а для плохих — 42 %.

Графическая интерпретация определения координат энергетических центров представлена на рис. 8 (см. вклейку) из которого видно, что хорошие изображения составили 82 % (по сравнению с 18 %).

## Выводы

В результате экспериментальных исследований установлено, что при использовании предложенного метода число хороших фрагментов трасс составляет в среднем 18 %. В результате аналогичных экспериментов с использованием известных методов [19] это число составило 50 %. Однако после обучения ПИ сети полученные результаты составили соответственно 79 и 65 %. Установлено также, что корректное распознавание составляет соответственно 94 % и 84,8 %.

Предлагаемый метод может быть применен в оптических системах связи и навигации, а также устройствах обработки изображений средств ПИ преобразования с формированием нормирующего уравнения для быстрого распознавания динамических изображений, а именно изображений пятен лазерных пучков.

The authors of the article consider conditions, necessary for development of the method and computer facilities for parallel-hierarchical image transformation, using highly productive GPU-adapters. The mathematical models for the parallel-hierarchical (PH) network and a method for PH network training to recognize dynamic patterns have been developed.

СПИСОК ЛИТЕРАТУРЫ

1. *Адинец А., Воеводин Вл.* Графический вызов суперкомпьютерам // Открытые системы. — 2008. — № 4. — [Электронный ресурс] Режим доступа: <http://www.osp.ru/os/2008/04/5114497/>.
2. *Скрибцов П.В., Долгополов А.В.* Сравнение производительности графических ускорителей и центрального процессора при вычислениях для больших объемов обрабатываемых данных // Нейрокомпьютеры: разработка, применение. — 2007. — № 9. — С. 421—425.
3. *Acceleware* выпустила первый в мире коммерчески доступный кластер на базе GPU NVIDIA // Журнал iXBT.com. — 2009. — № 10. — [Электронный ресурс] Режим доступа: <http://www.ixbt.com/news/all/index.shtml?10/63/33>.
4. *Chitty D.M.* A data parallel approach to genetic programming using programmable graphics hardware «GECCO 07» // Proc. of the 9th annual conf. on genetic and evolutionary computation. — 1991. — Vol. 2. — P. 1566—1573.
5. *Luo Z., Liu H., Wu X.* Artificial neural network computation on graphic process unit «JCNN '05» // Proc. of the IEEE International Joint Conf. on Neural Networks. — 2005. — Vol. 1. — P. 622—626.
6. *Li J.M., Wan D.L., Chi Z.X., Hu X. P.* A parallel particle swarm optimization algorithm based on fine-grained model with GPU accelerating // J. of Harbin Institute of Technology. — 2006. — Vol. 38, N 12. — P. 2162—2166.
7. *Xu R., Wunsch II D.C.* Clustering // IEEE — Hoboken, NJ : Wiley Press, 2008.
8. *Everitt D., Landau S., Leese M.* Clustering analysis. 4th edition. — London, UK : Arnold, 2001.
9. *Wunsch II D.C.* ART properties of interest in engineering applications // Proc. IEEE/INNS International Joint Conf. on Neural Networks. — Atlanta, GA, 2009.
10. *Knuth D.* The Art of Computing Programming: Fundamental Algorithms. 3rd Edition. — 1997. — Vol. 1. Addison-Wesley.
11. *Martnez-Zarzuela M., Pernas F., de Pablos A. et al.* Adaptive Resonance Theory Fuzzy Networks Parallel Computation Using CUDA // Bio-Inspired Systems: Computational and Ambient Intelligence. — 2009. — Vol. 5517. — P. 149—156.
12. *Gorchetchnikov M., Ames H., Versace M.* Simulating Biologically Realistic Neural Models on Graphics Process Units // ICCNS, 2008. — Boston, MA.
13. *Meuth R.J.* GPUs surpass computers at repetitive calculations // Potentials, IEEE. — 2007. — Vol. 26, N 6. — P. 12—23.
14. *Martin A.J., Burns S.M., Lee T.K. et al.* The design of an asynchronous microprocessor // Advanced Res // VLSI: Proc. Decennial Caltech Conf. — MIT Press Cambridge, MA, 1986.
15. *Sejun Kim.* A GPU based Parallel Hierarchical Fuzzy ART Clustering // Advanced Res, 2003.
16. *Прэнтт У.* Цифровая обработка изображений. В 2-х кн. — М. : Мир, 1982. — Т. 1. — 310 с., Т. 2. — 790 с.
17. *Применение цифровой обработки сигналов / Под ред. Э. Опенгейма.* — М. : Мир, 1980. — 545 с.
18. *Погребной В.А.* Бортовые системы обработки сигналов. — К. : Наук. думка, 1984. — 216 с.
19. *Тимченко Л.И., Мельников В.В., Кокряцкая Н.И.* Метод организации параллельно-иерархической сети для распознавания образов. // Кибернетика и системный анализ. — 2011. — № 1. — С. 152—163.

Поступила 14.06.12;  
после доработки 26.10.12

*ТИМЧЕНКО Леонид Иванович, д-р техн. наук, профессор, зав. кафедрой телекоммуникационных технологий и автоматики Государственного экономико-технологического университета транспорта. В 1979 г. окончил Винницкий политехнический ин-т. Область научных исследований — системы искусственного интеллекта.*

*ШПАКОВИЧ Виктория Васильевна, аспирант кафедры телекоммуникационных технологий и автоматики Государственного экономико-технологического университета транспорта, который окончила в 2008 г. Область научных исследований — распознавание динамических изображений.*

*КОКРЯЦКАЯ Наталья Ивановна, канд. техн. наук, доцент кафедры телекоммуникационных технологий и автоматики Государственного экономико-технологического университета транспорта. В 1973 г. окончила Винницкий педагогический ин-т. Область научных исследований — математическое моделирование параллельных процессов.*



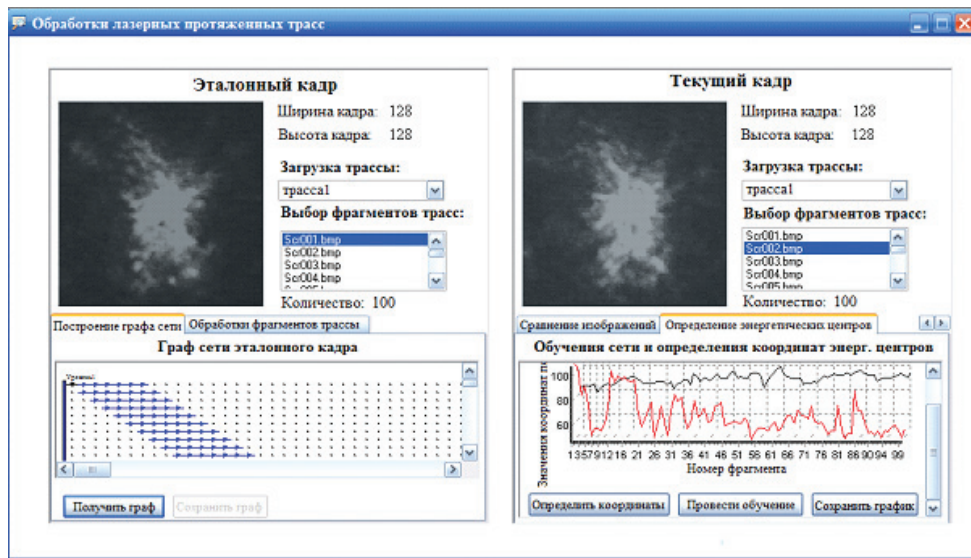


Рис. 4. Экранная форма средства ПИ преобразования для быстрого распознавания изображений пятен лазерных пучков

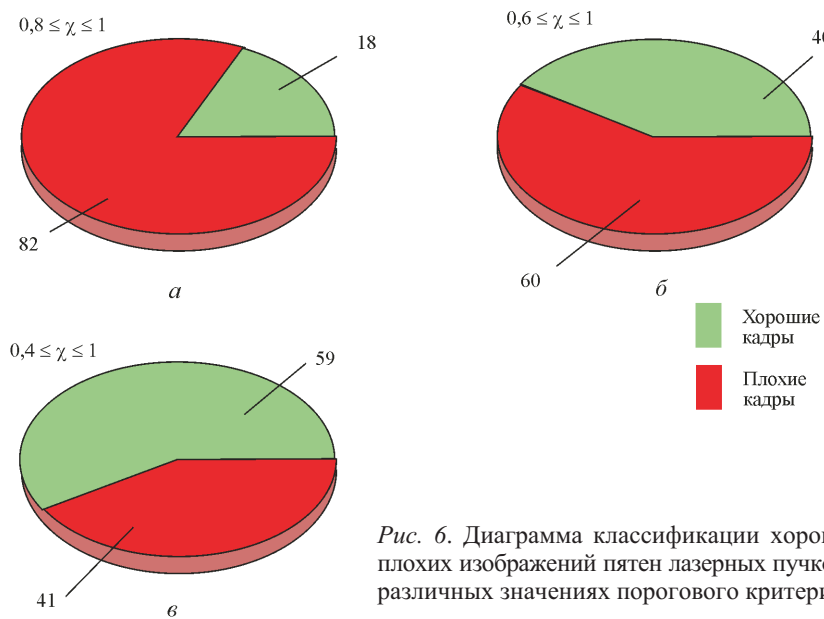


Рис. 6. Диаграмма классификации хороших и плохих изображений пятен лазерных пучков при различных значениях порогового критерия ( $\chi$ )

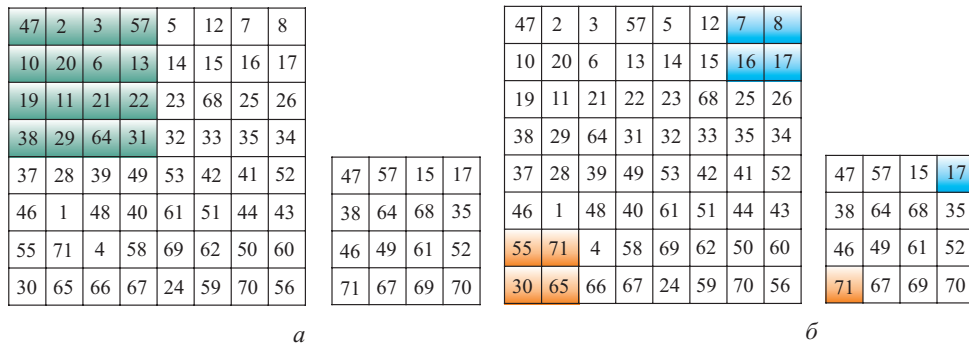


Рис. 7. Схема обработки пикселей с использованием графического адаптера Radeon X1300: слева — входная текстура, справа — результат первого сокращения; зеленым цветом обозначены исходные элементы (*a*), оранжевым цветом — максимум, формирующийся во втором ряду изображений (*б*)

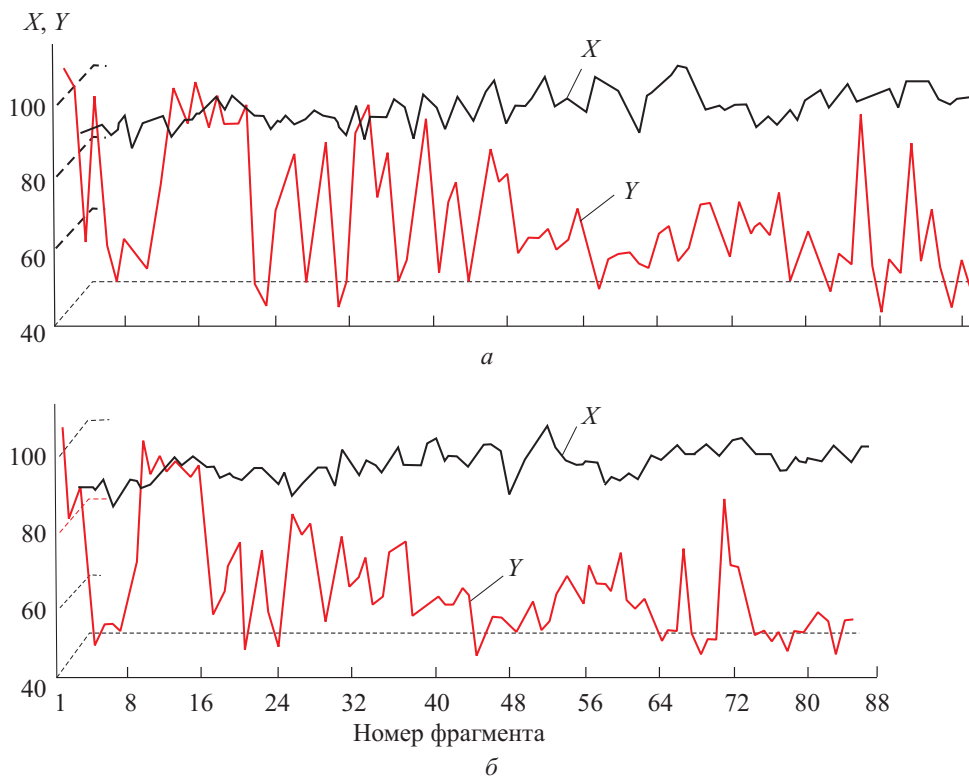


Рис. 8. Графики координат энергетических центров фрагментов трассы № 1, полученные до (*a*) и после обучения ПИ сети (*б*)