

Н.И. Кокряцкая

Методологические аспекты формирования параллельно-иерархической сети для быстрой обработки динамических изображений

Рассмотрены структуры параллельно-иерархических сетей. Материал имеет прикладное и познавательное значение. Приведены исследования изображений пятен лазерных трасс с использованием параллельно-иерархической сети.

The structures of the parallel-hierarchic network are considered. The material has applied and cognitive nature. The studies of the laser track spot image processing with the use of the parallel-hierarchic network are presented.

Розглянуто структури паралельно-ієрархічних мереж. Матеріал має прикладне та пізнавальне значення. Подано дослідження зображень плям лазерних трас з використанням паралельно-ієрархічної мережі.

Введение. Новейшие нейрофизиологические исследования, открывающие новые научные взгляды на природу взаимодействия сигналов в мозге, проводились учеными Университета Южной Калифорнии в Лос-Анджелесе [1]. У подопытного животного (крысы) изолировали определенный участок мозга, в мозговую ткань внедряли молекулы-датчики, которые в результате облучения видимым светом были заметны под микроскопом. Датчики устанавливали так, чтобы можно было определить направление сигналов: откуда они поступали и куда направлялись.

В результате удалось установить, что сигналы поступали не от различных частей мозга к единому центру, как считалось долгое время, а петлеобразно, связывая *в единую сеть* различные участки мозга. Предположения о такой его сложной организации высказывались и ранее, но впервые их подтверждение получено экспериментальным путем. Последовательное изучение различных разделов мозга позволит в итоге составить полную схему нервной системы животных и человека, а также поможет изучить процесс обработки и анализа информации.

Эта идея используется в данной статье для моделирования параллельно-иерархической (ПИ) сети применительно к обработке изображений.

Анализ публикаций [1, 2] показывает, что в последние десятилетия получили развитие исследования, связанные с имитацией нейробиологических аналогов принципов работы ней-

роподобных кибернетических устройств. Ярким примером подобных аналогов есть работа, описывающая биологически-ориентированную иерархическую модель [3]. Появляется реальная возможность получения максимальной интеллектуальности в работе таких моделей. Данные исследования показывают, что, кроме формирования временного сдвига, в сенсорных и особенно ассоциативных зонах коры мозга наряду с вертикальными нервными путями (кора-подкорка-кора) существуют также многочисленные горизонтальные нервные пути, соединяющие расположенные рядом нервные клетки [4]. Учитывая эту особенность, т.е. наличие вертикальных и горизонтальных путей и, развивая идею Q преобразования применительно к сетевому представлению, предлагается в иерархической структуре нейронной сети учитывать также горизонтальные пути, благодаря чему образуется структура трехмерной сети [2].

При таком смешанном использовании нейронных путей (горизонтального – вертикального – горизонтального и т.д.) между ветвями каждого уровня сети формируется временной сдвиг, впервые в начале 70-х годов описанный в структурах одномерной N сети, а затем в 80-х годах предложен на двумерных оптоэлектронных средах и в настоящее время развивается на трехмерных структурах [5].

Эти нейробиологические подходы мотивирует исследование аппаратных средств, таких как графические обрабатывающие модули, а также алгоритмы с низкоуровневой сложно-

Ключевые слова: параллельно-иерархическая сеть, единая иерархическая сеть, GPU-технологии.

стью, например используемые в *адаптивной резонансной теории* (АРТ), эффективно реализующие параллелизм в обработке данных. Несмотря на то, что многие исследования АРТ выполнены на процессорах *Graphics Processing Units (GPU)*, иерархические сети применялись в технологиях АРТ *GPU*. Каждый АРТ-слой распространяется в мультипроцессорах *GPU* и одновременно обучается. Экспериментальные данные свидетельствуют о том, что для больших деревьев целесообразно использовать графические процессоры, поскольку преимущество *GPU* достаточно существенно.

Одним из самых эффективных способов обработки больших массивов данных есть их параллельная обработка на основе специализированных системных решений, в частности нейроподобных параллельно-иерархических 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.

Программирование графических процессоров *GPU* на основе технологии *NVIDIA CUDA (Compute Unified Device Architecture)* целесообразно в вычислительных системах, особенно для популяционных алгоритмов [6–8] и, как показали исследования, изложенные в настоящей статье, для *быстрой обработки изображений*. Это существенно при использовании программирования *GPU* для достижения преимуществ в скорости обработки алгоритмов иерархического преобразования [9, 10].

АРТ привлекательна для иерархического преобразования скоростью и масштабируемостью к параллельному преобразованию [11]. В частности, иерархическая АРТ, основанная на процессорах *GPU*, не была выявлена ранее. Единственное главное ограничение в *CUDA* – жесткость памяти внутри ядерного значения, если множественное число массивов с переменными пределами ограничиваются только центральным процессором *CPU*. Типичные алгоритмы древовидной структуры обеспечивают указатели как для создания узла, так и ссылки [12], неэффективной при использовании техно-

логии *CUDA*. В работах [13–15] выполнено программирование *GPU*, которое фокусировалось на поведении единственной АРТ-единицы, однако, иерархическое преобразование нуждается в несколько другом подходе. Архитектура такого преобразования относится к структуре конвейерной обработки [16].

Следует отметить, что неразрешенной частью общей проблемы в структурах трехмерных сетей при смешанном использовании упомянутых нейронных путей остается разработка общей методологии формирования параллельно-иерархической сети применительно к обработке динамических изображений, чему и посвящена предлагаемая статья.

Цель статьи – разработка общей методологии формирования нейроподобной параллельно-иерархической сети и ее применение для обработки динамических изображений.

Методологические аспекты формирования параллельно-иерархической сети

Принцип построения параллельно-иерархической сети (ПИ) [2] можно определить как последовательность операций над множествами массивов данных, образующих множества информационных полей различных уровней иерархии, взаимодействие между которыми осуществляется посредством пирамидальной ПИ-структуры и реализуется на основе сетевой архитектуры.

Сетевая структура позволяет имитировать принцип действия распределенной ПИ-сети и при пространственном разделении во времени обрабатывает информацию в детерминированной ПИ-сети (рис. 1,а). Сеть, построенная на основе ПИ-преобразования [2], состоит из совокупности подсетей (рис. 1,б) формирования признаков о состояниях пространственно-временной среды (ПВС), структура которой однородна и состоит из ряда взаимозависимых иерархических уровней.

Алгоритм работы сети универсален и заключается в ПИ-формировании совокупностей общих и разных состояний ПВС. Обобщение всех видов сенсорной информации происходит на конечном этапе преобразования вне иерархической обработки каждого вида сенсорной информации. Итак, процесс обобщения между разными

видами сенсорной информации начинается лишь тогда, когда построение по определенной совокупности признаков заканчивается.

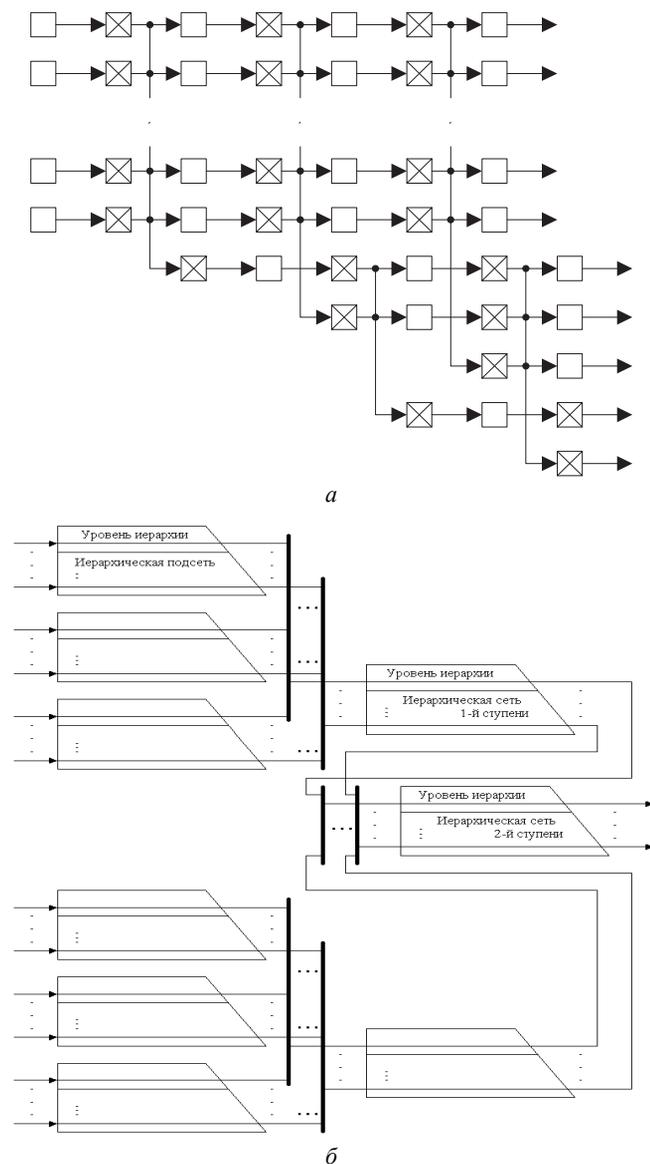


Рис. 1. Модели нейроподобных структур на основе ПИ-преобразования: *a* – модель сети, формирующейся при взаимодействии информационных потоков в горизонтальных и вертикальных путях; *б* – модель единой иерархической сети с пирамидальной схемой обработки – аналога природной единой сети петлеобразного связывания различных участков мозга [1]. \boxtimes – общие состояния ПВС; \square – множество разных состояний ПВС

Основой этой структуры (рис. 1) служат совокупности компонентов, передающие на соответствующем уровне иерархии образы в целом. Данные пирамидальные структуры [17] могут быть воссозданы модельно с помощью

математического аппарата – так называемых растущих пирамидальных сетей [18, 19].

В общей форме методологию формирования ПИ-сети [19] можно представить в формализованном виде, исходя из следующих положений. Пусть имеется множество информационных потоков входных данных. Возникает следующая проблема. Каким способом организовать в реальном времени параллельный вычислительный процесс, чтобы получить строго распределенную во времени и иерархии вычислительную сеть? Ответ на этот вопрос может дать принцип построения следующей параллельной вычислительной сети (рис. 2). Пусть имеются множества

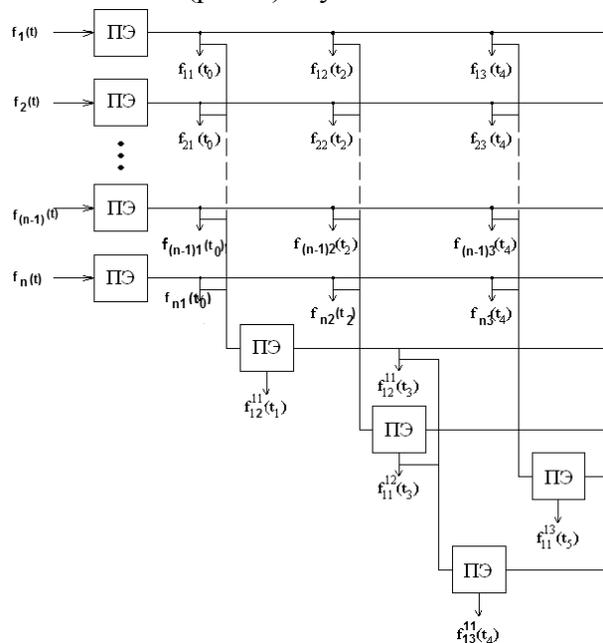


Рис. 2. Структурная схема взаимодействия информационных потоков в горизонтальных и вертикальных путях параллельно-иерархической сети

числовых потоков данных. Обработка этих данных организована так, что вначале информация обрабатывается в горизонтальных каналах, а затем со временем – пошагово в вертикальных каналах, и далее процесс обработки со сдвигом во времени и также пошагово переходит в чередующиеся горизонтальные и вертикальные каналы.

Здесь обработке подвергается множество входных потоков данных на различных (k) иерархических уровнях. Каждый уровень представляет собой совокупность процессорных элементов (ПЭ), функционирующих в строго фик-

сированные моменты времени (t_j) ". На каждом уровне ПИ-сети в ее ветвях происходит обобщение сенсорной информации. Чем больше степень обобщения сенсорной информации, при ее прохождении по ветвям сети, тем быстрее сходится ПИ-сеть, т.е. заканчивается вычислительный процесс в ее ветвях.

Пирамидальная вычислительная структура на основе ПИ-преобразования образует сеть в виде ПИ-пирамиды. Здесь для каждой пирамиды используется свой ПЭ, а число ПЭ определяется суммарным числом ветвей ПИ-сети (см. рис. 2).

Пусть заданы n функций $f_1(t), f_2(t), \dots, f_n(t)$. Данные функции опишем на разных уровнях иерархии их представления от первого до j -го

$$(j = 2l, l = 1, \overbrace{\sum_{j=2}^{n^2} \max_{j-1} n_G} \text{ и } j = 2l + 3, \\ l = 0, \overbrace{\sum_{j=3}^{n^2} \max_{j-1} n_G}).$$

$$\begin{aligned} \sum_{j=1}^{n_1} \sum_{i=1}^{n_{j1}} f_{j1}(t - i\tau) &= \sum_{j=1}^{n_2} \sum_{i=1}^{n_{j2}} f_{j2}(t - 2ij\tau) = \\ &= \sum_{i=1}^{n_{13}} f_{13}(t - (2i + 3)\tau) + \sum_{i=2}^{n_{23}} f_{23}(t - (2i + 3)\tau) + \\ &+ \sum_{i=3}^{n_{33}} f_{33}(t - (2i + 3)\tau) + \dots + \sum_{i=n}^{n_{n3}} f_{n3}(t - (2i + 3)\tau) = \\ &= \sum_{i=1}^{n_{14}} f_{14}(t - (2i + 6)\tau) + \sum_{i=2}^{n_{24}} f_{24}(t - (2i + 6)\tau) + \dots + \\ &+ \sum_{i=n-1}^{n_{(n-1)4}} f_{(n-1)4}(t - (2i + 6)\tau) + \sum_{i=n}^{n_{n4}} f_{n4}(t - (2i + 6)\tau) = \dots = \\ &= \sum_{i=1}^{n_{12l}} f_{12l}(t - (2i + 6l)\tau) + \sum_{i=2}^{n_{22l}} f_{22l}(t - (2i + 6l)\tau) + \dots + \\ &+ \sum_{i=n-1}^{n_{(n-1)2l}} f_{(n-1)2l}(t - (2i + 6l)\tau) + \sum_{i=n}^{n_{n2l}} f_{n2l}(t - (2i + 6l)\tau) = \\ &= \sum_{i=1}^{n_{2(2l+3)}} f_{1(2l+3)}(t - (2i + 6l + 3)\tau) + \\ &+ \sum_{i=2}^{n_{2(2l+3)}} f_{2(2l+3)}(t - (2i + 6l + 3)\tau) + \dots + \\ &+ \sum_{i=n-1}^{n_{(n-1)(2l+3)}} f_{(n-1)(2l+3)}(t - (2i + 6l + 3)\tau) + \\ &+ \sum_{i=n}^{n_{n(2l+3)}} f_{n(2l+3)}(t - (2i + 6l + 3)\tau), \end{aligned} \quad (1)$$

где τ – задержка формирования последующей функции по отношению к предыдущей.

Функции первого уровня можно также представить как функцию хвостовых элементов:

$$\sum_{j=1}^{n_1} \sum_{i=1}^{n_{j1}} f_{j1}(t - i\tau) = \sum_{j=2}^{n^2} f_{1j}^{11}(t - (3j - 4)\tau). \quad (2)$$

Правая часть выражения (2) формирует хвостовые функции $f_{1j}^{11}(\cdot)$, получаемые в результате функционального преобразования (1).

Анализируя сетевое преобразование (1), можно сделать вывод о том, что в процессе образования каждого уровня формируется временной сдвиг (τ), наличие которого приводит к получению хвостовых функций (2).

На первом уровне входные потоки данных обрабатываются в независимых параллельных ветвях.

На втором уровне ($j = 2$) и переход на третий уровень процесс обработки на структурно-функциональном уровне можно представить в виде

$$f_{p_2 n_2}^{j p_1}(t) = F \left(\bigcup_{p=1}^{p_1} f_{(p_1 - p + 2)l}^{(j-1)j}(t_{p+2}) \right), \quad (3)$$

где n_1, n_2 – номер элемента в ветви и номер сети ($n_2 = 1$), $n_1 = 1, n_G, n_G$ – число групп одинаковых элементов в преобразовании G , применяемого в каждой ветви j -го уровня, p_1 – номер ветви j -го уровня, $p_1 = 1, \max_{j-1} n_G, \max_{j-1} n_G$ – максимальное число групп одинаковых элементов в преобразовании G , применяемого в ветвях $(j - 1)$ -го уровня, F – функциональный оператор, определяющий критерий выбора общей части, применяемого в преобразовании G при обработке потока данных $f(t)$. На всех иерархических уровнях ($j = 2, 3, 4 \dots$) данный процесс обработки можно представить в виде

$$f_{p_2 n_2}^{j p_1}(t) = F \left(\bigcup_{p=1}^{p_1} f_{(p_1 - p + 2)l}^{(j-1)j} \left(\left[\frac{p_1}{p} \right]_l \right) (t_{p+2(j-1)}) \right), \quad (4)$$

Последовательный процесс перехода информационного потока данных из предыдущего уровня ПИ-сети на последующий можно отразить в виде

$$M(j) = \bigcup_{p=1}^j M] \frac{j}{p} [, \quad (5)$$

где $] \frac{j}{p} [$ указывает на размерность множества M на j -м уровне, $j = \overline{1, n^2}$, n – число исходных множеств, $|j| \leq n^2$.

Параллельно-иерархическая сеть

Для более полного понимания особенностей структурно-функционального построения ПИ-сети применительно к распознаванию изображений рассмотрим преобразование $G(M) = \{a_s \mid s = 1, 2, \dots, m_i\}$, применяемое в каждом канале ее обработки. В экспериментах каждое числовое множество (например, информация о яркости пикселей изображения) вводилось в отдельную ПИ-сеть так, чтобы каждая строка изображения обрабатывалась в отдельном канале. В каждом канале сети применяется одиночное преобразование G . Первая стадия G -преобразования – кластеризация [20] входных компонентов, т.е. обнаружение корреляций между соседями. Наиболее коррелированные компоненты объединяются, а значения новых компонентов вновь формируются путем структурирования этой информации. После фазы кластеризации выполняется вторая стадия G -преобразования, сортирующая компоненты. Цель сортировки состоит в том, чтобы разные компоненты обрабатывались в разное время.

Преобразованные компоненты передаются на второй уровень, где цель обработки остается прежней – найти наиболее коррелированные компоненты и выполнить их сортировку. То же самое преобразование проводится на каждом последующем уровне. Некоррелированные несвязанные компоненты не будут воздействовать на выходной результат на первых уровнях и, следовательно, первые извлеченные выходные компоненты будут иметь в результате более весомый приоритет, чем последующие. Процесс анализа завершается с последней извлеченной компонентой.

Исходное изображение M представим n сегментами, над которыми преобразование G выполняется в каждом канале обработки i , $i = 1, \dots, n$ одновременно. Тогда анализируемый i -й сегмент изображения – $M_i^1 = \{a_{is}^1 \mid j=1, 2, \dots, m_i^1\}$, где m_i^1 – число компонентов в i -м канале первого уровня, а все анализируемые сегменты могут быть представлены в виде матрицы A_1 :

$$A_1 = \begin{bmatrix} M_1^1 \\ M_2^1 \\ \dots \\ M_i^1 \\ \dots \\ M_n^1 \end{bmatrix} = \begin{bmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1m_1^1}^1 & x & x & \dots & x \\ a_{21}^1 & a_{22}^1 & \dots & \dots & a_{2m_2^1}^1 & x & \dots & x \\ \dots & x \\ a_{i1}^1 & a_{i2}^1 & \dots & \dots & \dots & \dots & \dots & a_{im_i^1}^1 \\ \dots & x \\ a_{n1}^1 & a_{n2}^1 & \dots & a_{nm_n^1}^1 & x & x & \dots & x \end{bmatrix},$$

где x обозначает отсутствие компоненты. Индекс столбца указывает на шаг извлечения компоненты, а индекс строки указывает канал, из которого этот компонент извлечен. В операторной форме анализ данных, заданных матрицей A_1 , на втором уровне отобразим в операции транспонирования – T , т.е. исходная матрица на втором уровне будет представлена как $A_1^T = T(A_1)$. После выполнения операции транспонирования индекс строки указывает на номер шага для извлечения компоненты.

На втором уровне обработки преобразование G организовано так, что возможно исследование корреляций между компонентами, одновременно извлеченными из различных каналов первого уровня. Эта процедура применяется к строкам матрицы A_1^T .

$$A_2 = G(A_1^T) = G(T(G(M))) = \begin{bmatrix} a_{11}^2 & a_{12}^2 & \dots & \dots & a_{1m_1^2}^2 & x & \dots & x \\ a_{21}^2 & a_{22}^2 & \dots & a_{2m_2^2}^2 & x & x & \dots & x \\ \dots & x \\ a_{i1}^2 & a_{i2}^2 & \dots & \dots & \dots & \dots & \dots & a_{im_i^2}^2 \\ \dots & x \\ a_{n^2 1}^2 & a_{n^2 2}^2 & \dots & a_{n^2 m_n^2}^2 & x & x & \dots & x \end{bmatrix},$$

где $n^2 = m^1, m^2$ соответственно обозначает число параллельных каналов и номер шага обработки на втором уровне.

Матрица A_2 формируется на втором уровне обработки после выполнения временной декомпозиции, при которой каждая компонента a_{is}^2 извлекается на шаге $t = i + s - 1$. Если матрица A_2 сформирована так, что индекс столбца указывает шаг извлечения компоненты, то та же самая процедура обработки может быть применена на каждом последующем уровне иерархии. Эта процедура реализована как горизонтальное выравнивание строк матрицы A_2 и располагает первый непустой элемент в каждой строке на главной диагонали. Результирующая матрица обозначена как A_2^* , и $A_2^* = P(A_2)$, где P – оператор сдвига строки.

$$A_2^* = \begin{bmatrix} a_{11}^2 & a_{12}^2 & \dots & \dots & \dots & a_{1m_1}^2 & x & \dots & \dots & \dots & \dots & x \\ x & a_{21}^2 & a_{22}^2 & \dots & a_{2m_2}^2 & x & x & \dots & \dots & \dots & \dots & x \\ \dots & x \\ x & x & x & a_{i1}^2 & a_{i2}^2 & \dots & \dots & \dots & \dots & \dots & \dots & a_{im}^2 \\ \dots & x \\ x & x & x & x & x & a_{n^2_1}^2 & a_{n^2_2}^2 & \dots & a_{n^2_{m_1}}^2 & x & \dots & x \end{bmatrix}.$$

Компонента a_{11}^2 – единственная и поэтому может быть извлечена на первом шаге обработки второго уровня, т.е. формируется первый хвостовой элемент ПИ-сети. Эта компонента не связана во времени со всеми другими компонентами второго уровня и поэтому будет выходной. Она есть также промежуточным результатом сетевой обработки, и – единственной компонентой, извлеченной в результате многоступенчатой обработки на втором уровне. Эта компонента получается из матрицы путем удаления первого ее столбца, а остающиеся компоненты сохраняются в матрице A_2^{**} . Эта процедура выполняется путем применения оператора L , т.е. $L(A_2^*) = a_{11}^2 + A_2^{**}$. Матрица третьего уровня A_2^T получается после выполнения операции транспонирования T матрицы A_2^{**} , которая может быть записана в следующем виде: $A_2^T = T(L(P(A_2)))$.

Описанная процедура выполняется на каждом последующем иерархическом уровне до последнего уровня – $j = k_{\max}$, в котором матрица $A_{k_{\max}}$ содержит одиночный элемент. То-

гда $T(L(P(A_{k_{\max}}))) = \emptyset$. Последовательное использование операторов G, P, L, T может быть представлено как выполнение оператора F : $F(A_j^T) = T(L(P(G(A_j^T)))) = A_{j+1}^T$, а последовательное применение оператора F – как $F^j(A_1^T) = F^{j-1}[F(A_1^T)]$. Тогда весь многоуровневый процесс можно представить в следующей операторной форме:

$$F^{k_{\max}-1} \left[T \left(G \left(\bigcup_{i=1}^n M_i \right) \right) \right] = \{ a_{11}^j \mid j = 2, 3, \dots, k_{\max} \}.$$

Таким образом, результат обработки в ПИ-сети представляется как вектор образа в виде формирования на каждом уровне этой сети, начиная со второго уровня, хвостовых ее элементов.

Компьютерное и физическое моделирование, исследования параллельно-иерархической сети для быстрой обработки динамических изображений

В настоящее время в полиграфии, локации, оптической связи ощущается необходимость более широкого внедрения оптико-электронных систем с автоматической коррекцией искажений сформированного светового излучения. Причинами искажений могут быть дестабилизирующее действие механических или климатических факторов, нестабильность характеристик источника излучения, разъюстировка оптических элементов и пр. Обеспечение желаемого качества коррекции нуждается в непрерывном динамическом контроле характеристик светового излучения, например пространственного распределения его интенсивности, в частности оценки отклонения указанного распределения от начального или эталонного распределения.

Проверка эффективности ПИ-сети в качестве модели при реализации контролируемого обучения для задачи распознавания осуществлялась на примере обработки изображений пятен лазерных пучков.

При этом по аналогии с *RBF*-сетями [21] задача обучения ПИ-сети практически сводилась к идее контролируемого обучения элементов

ее выходного слоя. Усредненные значения весовых коэффициентов $\overline{w}_1 \div \overline{w}_{k-1}$ определяются

так: $w_t = \frac{\sum_{p=1}^N w_t^{(p)}}{N}$, где N – размерность обучаемой выборки P , $t = \overline{1, k-1}$. Используя свойство равенства суммы входных элементов ($\sum_i a_i$) ПИ-сети и суммы хвостовых элементов ($\sum_{t=2}^k a_{11}^t$) [22], несложно составить систему уравнений для получения таких настроечных коэффициентов $w_1 \div w_{k-1}$ в виде (6), чтобы сформировать нормирующее уравнение (7):

т.е. выходного слоя. Усредненные значения весовых коэффициентов $\overline{w}_1 \div \overline{w}_{k-1}$ определяются так: $w_t = \frac{\sum_{p=1}^N w_t^{(p)}}{N}$, где N – размерность обучаемой выборки P , $t = \overline{1, k-1}$. Используя свойство равенства суммы входных элементов ($\sum_i a_i$) ПИ-сети и суммы хвостовых элементов ($\sum_{t=2}^k a_{11}^t$) [22], несложно составить систему уравнений для получения таких настроечных коэффициентов $w_1 \div w_{k-1}$ в виде (6), чтобы сформировать нормирующее уравнение (7):

$$\left. \begin{aligned} w_1 &= \frac{\sum_{t=2}^k a_{11}^t}{(a_{11}^2 + \sum_i a_i^2)}; \\ w_2 &= \frac{\sum_{t=2}^k a_{11}^t}{(a_{11}^3 + \sum_i a_i^3)} - \frac{w_1 a_{11}^2}{(a_{11}^3 + \sum_i a_i^3)}; \\ &\dots \\ w_{k-2} &= \frac{\sum_{t=2}^k a_{11}^t}{(a_{11}^{k-1} + \sum_i a_i^{k-1})} - \frac{w_1 a_{11}^2 + w_2 a_{11}^3 + \dots + w_{k-3} a_{11}^{k-2}}{(a_{11}^{k-1} + \sum_i a_i^{k-1})}; \\ w_{k-1} &= \frac{\sum_{t=2}^k a_{11}^t}{(a_{11}^k + \sum_i a_i^k)} - \frac{w_1 a_{11}^2 + w_2 a_{11}^3 + \dots + w_{k-2} a_{11}^{k-1}}{(a_{11}^k + \sum_i a_i^k)}, \end{aligned} \right\} \quad (6)$$

где $\sum_{t=2}^k a_{11}^t$ – эталонные, $\sum_i a_i^k$, $a_{11}^2 \div a_{11}^k$ – текущие составляющие (признаки) распознаваемого изображения.

$$\begin{aligned} d &= \frac{\overline{w}_1 a_{11}^2}{\sum_{t=2}^k a_{11}^t} + \frac{\overline{w}_2 a_{11}^3}{\sum_{t=2}^k a_{11}^t} + \dots + \\ &+ \frac{\overline{w}_{k-2} a_{11}^{k-1}}{\sum_{t=2}^k a_{11}^t} + \frac{\overline{w}_{k-1} a_{11}^k}{\sum_{t=2}^k a_{11}^t} = \frac{\sum_{t=2}^k \overline{w}_{t-1} a_{11}^t}{\sum_{t=2}^k a_{11}^t}. \end{aligned} \quad (7)$$

Левая часть нормирующего уравнения (7) представляет собой отношения суммы произведений усредненных значений настроечных

коэффициентов и хвостовых элементов к сумме хвостовых элементов сети, а правая при правильном распознавании приближается к единице, т.е. $d \rightarrow 1$. Степень этой близости к единице отражает меру сходства распознаваемых изображений пятен лазерных пучков и в идеальном случае при правильном распознавании $d \approx 1$. В экспериментах использовались 15 лазерных трасс с количеством изображений – 1000 в каждой трассе [22].

Для компьютерного моделирования ПИ-преобразования разработано программное средство, предназначенное для обработки и распознавания изображений пятен лазерных пучков (размерность изображения 128×128 э.р.) (рис. 3).

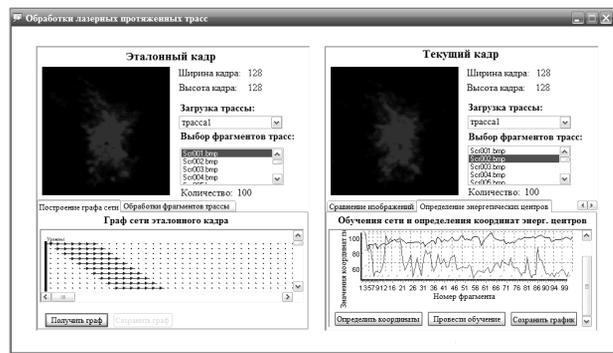


Рис. 3. Экранная форма средства параллельно-иерархического преобразования для распознавания изображений пятен лазерных пучков

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 обработки текущего кадра этой трассы.

В нижней части приведена информация о параметрах кадра, о трассе, в которой содержится кадр, название файла кадра, число кадров в трассе и расположены вкладки:

- построение графа сети, который можно получить и сохранить в виде файла графа ПИ-сети;
- обработка фрагментов трассы – можно обработать любой фрагмент трассы;
- сравнение изображений и распознавание изображений пятен лазерных пучков;
- определение энергетических центров изображений пятен лазерных пучков и обучение ПИ-сети.

Физическое моделирование приведено также на примере *быстрого распознавания изображений* для задачи определения координат изображений протяженных лазерных трасс [22] с использованием графического адаптера *Radeon X1300*. Исследования показали, что для обработки одного изображения требуется приблизительно 0,6 с.

Обработка изображений происходит на *GPU*, ядра которых выполняются на всех элементах, покрытых исходным регионом. Единственный очевидный путь расчета скаляра от входного вектора – это предоставление 1×1 исходного элемента и использование ядра, которое читается во всех значениях от входной текстуры. Этот подход имеет определенные недостатки. Во-первых, только один из параллельных элементарных процессоров был бы занят. Во-вторых, это, возможно, превысило бы максимум, разрешенный шейдерной длиной и статической инструкцией расчета на некоторые технические средства. Поэтому будем выполнять параллельное действие сокращения, основанное на глобальных методах коммуникации на параллельных компьютерах.

На высшем уровне параллельное вычисление на *GPU* [8] – это коррекция размеров входной текстуры и элементов индексов. Для предоставленного вектора M длины M выход из первого шага – $M/2$ с текстурой $M/2$. Для каждого из его элементов координаты для входной текстуры скорректированы таким образом, что они соответствуют разьединенным 2×2 субобластям. Значения в этих субобластях затем сравниваются. Это рекурсивно повторяется, пока текстура 2×2 будет уменьшена к финалу 1×1 «скалярной» текстурой в результате логарифмического ряда повторений.

Следующие серии изображений подытоживают первый шаг сокращения алгоритма 8×8 входной текстуры (рис. 4). Изображение с левой стороны показывает входную текстуру. Серым цветом (рис. 4,а) обозначены исходные элементы. Изображение справа – это результат первого прохода сокращения. Каждый исходный элемент содержит местный максимум передачи 2×2 субобласти во входной текстуре.

Это отношение, кроме того, выделяется во втором ряду изображений (рис. 4,б).

Эксперименты показывают, что для изображений пятен лазерных пучков в результате действия различных дестабилизовавших факторов невозможно точно измерять координаты их энергетических центров, однако точность можно существенно повысить, используя обучение ПИ-сети с их последующей классификацией на «хорошие» и «плохие» изображения.

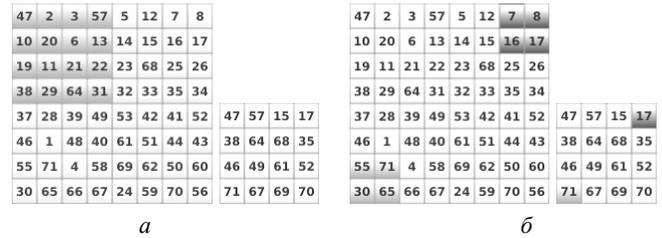


Рис. 4. Схема обработки пикселей с использованием графического адаптера *Radeon X1300*

Определим координаты энергетических центров фрагментов лазерной трассы на основе вычисления моментных признаков [22].

Определение «хороших» и «плохих» изображений показано на рис. 5.

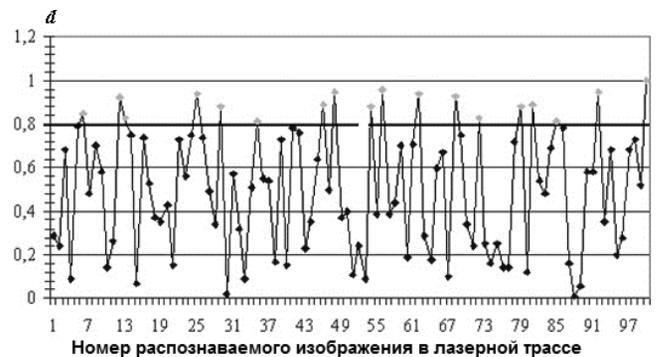


Рис. 5. Определение «хороших» и «плохих» изображений

Приведем графическую интерпретацию определения координат энергетических центров (рис. 6).

После обучения ПИ сети так, как это выше изложено, процент «хороших» изображений составил 83% (в работе [23] – 18%). «Хорошие» недеформированные атмосферой изображения используются для определения координат энергетических центров.

Графическая интерпретация определения координат энергетических центров после обучения ПИ сети приведена на рис. 7:

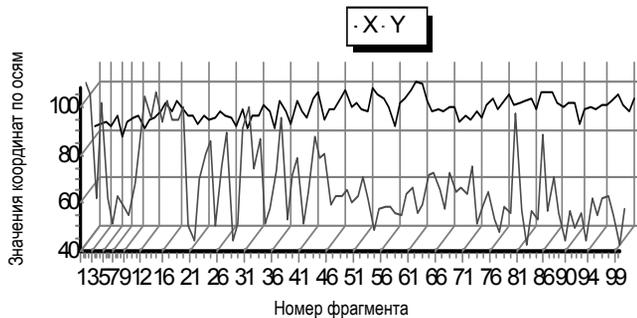


Рис. 6. Определение координат энергетических центров фрагментов лазерной трассы

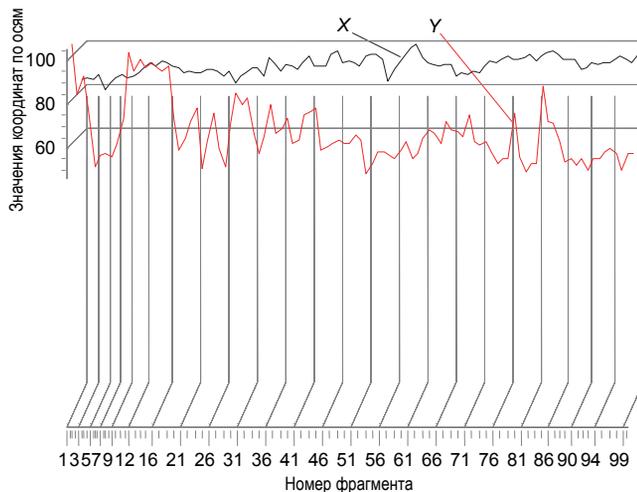


Рис. 7. Определение координат энергетических центров фрагментов трассы после обучения параллельно-иерархической сети

Заключение. Таким образом, в статье предлагается методология формирования параллельно-иерархической сети для быстрой обработки изображений, используя интерпретацию нейробиологического процесса в виде одновременно распространяющихся возбуждений, что-то вроде цепной реакции, в которой действие одной нейронной сети возбуждает соответствующий отклик в другой. Тогда восприятие времени и причинно-следственных связей, столь сложные для традиционного программирования, закладывается в архитектуру таких нейронных систем.

В работе проводилось физическое моделирование ПИ сети, т.е. процесса распараллеливания вычислительного процесса с использованием компьютерных GPU технологий.

Исследования показывают, что средний процент «хороших» фрагментов трасс составил 17 процентов в сравнении с 50 процентами в работе

[23]. Однако после обучения он уже составил 79,25 (в сравнении с 65 процентами в работе [23]). Также установлено, что средний процент корректного распознавания составил 94 процента (84,8 процента), а точность определения энергетического центра, элементов разложения — не более 0,01 е.р. (в [23] 1,2 е.р.), которые по точности превышают известные, например на основе определения центра тяжести по методу моментных признаков, в среднем в 1,5 раза. Физическое моделирование метода параллельно-иерархического преобразования для распознавания изображений с использованием графического адаптера Radeon X1300 показывает, что для обработки одного изображения нужно приблизительно 0,6 с (3,32 с), а для сетевой обработки — 1,52 с (8,4 с), что позволяет говорить о быстрой обработке изображений.

Перспективы дальнейших исследований в данном направлении связаны с разработкой обучающего алгоритма на основе структуры параллельной многоуровневой (иерархической) сети, формирующейся из цепочек последовательностей структур ПИ-сетей. В этом случае алгоритм обработки изображения с использованием многоуровневой ПИ-сети позволит обрабатывать изображения частями, т.е. окнами определенной размерности. При этом окна обрабатываются независимо, что позволит использовать таких многоуровневых ПИ-сетей для разных окон параллельно в соответствии с пирамидальной схемой. Это даст возможность для повышения производительности еще более полного использования перспективных технологий параллельных гетерогенных вычислений на основе *General-Purpose Computing on Graphics Processing Units (GPGPU)* при обработке динамических изображений протяженных лазерных трасс и прогнозирования их характеристик в реальном времени.

Результаты могут найти широкое применение в разных системах, для которых необходимо осуществлять автоматическую коррекцию искажений лазерного источника светового излучения. Это, например, системы оптической навигации с использованием лазеров в условиях сниженной видимости в атмосфере, системы точного ориентирования, которые позво-

ляют определить пространственное положение движущегося объекта при разных метеорологических условиях, лазерные навигационные системы. Это также лазерные системы связи и передачи информации, локации и дальнометрии, трассировки и другие системы, которые ощущают влияние атмосферы на параметры лазерного луча, несущего полезную информацию.

1. *Richard H. Thompson, Larry W. Swanson.* Hypothesis-driven structural connectivity analysis supports network over hierarchical model of brain architecture // Proc. of the National Acad. of Sci. USA. 2010 Aug. 24; 107(34): 15235–15239. Publ. online 2010 Aug. 9. doi: 10.1073/pnas.1009112107.
2. *Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідально-сітьового перетворення зображень / В.П. Кожем'яко, Л.І. Тимченко, Ю.Ф. Кутаєв та ін. – К.: УМК ВО, 1994. – 272 с.*
3. *What is the Best Multi-Stage Architecture for Object Recognition? / K. Jarrett, K. Kavukcuoglu, M.A. Ranzato et al. // Proc. Int. Conf. on Comp. Vision (ICCV'09), 2009.*
4. *Рабинович З.Л.* О думающих машинах и интеллектуальных ЭВМ // Кибернетика. – 2003. – № 5. – С. 83–88.
5. *Рабинович З.Л.* Пам'ять людини й мислення – образне й символічне (концептуальне модельне відтворення) // Доп. НАН України. – 2009. – № 3. – С. 61–65.
6. *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. – Jul. 1991. – 2. – P. 1566–1573.
7. *Luo Z., Liu H., Wu X.* Artificial neural network computation on graphic process unit, IJCNN'05 // Proc. of the IEEE Int. Joint Conf. on Neural Networks. – Jul. 2005. – 1. – P. 622–626.
8. *A parallel particle swarm optimization algorithm based on fine-grained model with GPUaccelerating / J.M. Li, D.L. Wan, Z.X. Chi et al. // J. of Harbin Inst. of Technology. – Dec. 2006. – 38, N 12. – P. 2162–2166.*
9. *Xu R., Wunsch D.C. II,* Clustering // IEEE / Wiley Press, Hoboken, NJ, 2008. – 368 p.
10. *Everitt D., Landau S., Leese M.* Clustering analysis. – London: Arnold, 2001. – 330 p.
11. *Wunsch D.C. II* ART properties of interest in engineering applications // Proc. IEEE/INNS Int. Joint Conf. on Neural Networks, Atlanta, GA, 2009. – P. 3556–3559.
12. *Knuth D.* The Art of Computing Programming: Fundamental Algorithms. – Addison-Wesley. – 1997. – 1. – 650 p.
13. *Adaptive Resonance Theory Fuzzy Networks Parallel Computation Using CUDA, Bio-Inspired Systems: Comput. and Ambient Intell. / M. Martinez-Zarzuela, F. Pernas, A. de Pablos et al. – 2009. – 5517. – P. 149–156.*
14. *Gorchetchnikov M., Ames H, Versace M.* Simulating Biologically Realistic Neural Models on Graphics Process Units, ICCNS 2008, Boston, MA, 2008.
15. *Meuth R.J.* GPUs surpass computers at repetitive calculations // Potentials, IEEE. – Nov.–Dec. 2007. – 26, N 6. – P. 12–23
16. *The design of an asynchronous microprocessor, Advanced Res. VLSI / A.J. Martin, S.M. Burns, T.K. Lee et al. // Proc. Decennial Caltech Conf. VLSI, MIT Press Cambridge, MA, Mar. 1986.*
17. *Кожем'яко В.П., Тимченко Л.І., Яровий А.А.* Паралельно-ієрархічні мережі як структурно-функціональний базис для побудови спеціалізованих моделей образного комп'ютера. – Вінниця: Універсум-Вінниця, 2005 – 161 с.
18. *Тимченко Л.І.* Паралельно-ієрархічне перетворення як системна модель ефективних засобів штучного інтелекту: Автореф. дис. ... д-ра техн. наук. – Львів, 2002. – 36 с.
19. *Gladun V.P.* Decision planning. – Kiev: Nauk. Dumka, 1987. – 167 p.
20. *Frucci Maria, Sannit di Baja, Gabriella From* Segmentation to Binarization of Gray-level Images // J. of Pattern Recognition Research. – 2008. – 3 (1). – P. 1–13.
21. *Haykin S.* Neural Networks. – New Jersey: Prentice Hall, 1999. – 1104 p.
22. *Паралельно-ієрархічні мережі / Л.І. Тимченко, С.В. Свечников, Н.І. Кокряцкая и др. – К.: Віпол, 2010. – 653 с.*
23. *Тимченко Л.І., Мельников В.В., Кокряцкая Н.І.* Метод организации параллельно-иерархической сети для распознавания образов // Кибернетика и системный анализ. – 2011. – № 1. – С. 152–163.

Поступила 16.08.2012

(после доработки 05.02.2014)

Тел. для справок: +38 043 253-1123, (044) 591-5135,

097 313-7633 (Киев)

E-mail: kokriatskaia@rambler.ru

© Н.И. Кокряцкая, 2014