

УДК 621.397.331

В. А. Бобух, Н. В. Павлова

## РЕАЛИЗАЦИЯ НЕКОТОРЫХ КОЛОРИМЕТРИЧЕСКИХ ПРЕОБРАЗОВАНИЙ ИЗОБРАЖЕНИЙ В РЕАЛЬНОМ ВРЕМЕНИ

### 1. Введение

В настоящее время телевизионные системы используются практически во всех отраслях науки и техники. Без телевидения немислмы освоение космоса, решение технологических задач и автоматизация многих производственных процессов, создание уникальных инструментов научного исследования, развитие радио- и гидролокации, создание реалистичных тренажеров для обучения летчиков и космонавтов и многое другое. В силу этого обстоятельства особую актуальность приобрели вопросы цифровой обработки видеoinформации. Существует множество зарубежных систем, однако для отечественного пользователя доступными являются в основном системы нелинейного монтажа. Главными их недостатками являются потеря качества информации при сжатии, необходимом для ее хранения, и существенно большее по сравнению с длительностью материала время монтажа. Указанных недостатков лишены системы, работающие в реальном времени. Такие системы выполняют две группы преобразований изображений: геометрические (изменение положения и формы) и колориметрические (изменение цветовых характеристик, задание способов взаимодействия нескольких изображений) преобразования. В Харьковском национальном университете радиоэлектроники ведутся государственные и договорные работы, связанные с созданием алгоритмов, методов и структур для цифровой обработки изображений в реальном времени. В рамках этих работ проведены исследования и предложены методы для реализации некоторых колориметрических преобразований изображений, которые будут рассмотрены в данной статье.

### 2. Анализ состояния вопроса

К колориметрическим преобразованиям изображений относят цветовую коррекцию, пастеризацию, соларизацию, раскраску в условные цвета, обесцвечивание, зональное, аддитивное и неаддитивное микширование, электронную пир-проецию. Рассмотрим реализацию цветовой коррекции и микширования изображений в системе обработки видеoinформации.

#### 2.1. Реализация цветовой коррекции изображений

Нередки случаи, когда по независимым от оператора обстоятельствам необходимо проводить съемку в быстро изменяющихся условиях освещенности и цветовой гаммы. При этом современные автоматически настраиваемые видеокамеры зачастую, несмотря на их высокую цену, дают резкое ухудшение качества изображения, проявляющееся в виде за-

ливки изображения нехарактерным цветом. Такой дефект может быть устранен путем изменения передаточной характеристики обрабатывающего тракта. Этот же способ может быть применен для устранения искажений, вызванных нелинейностью регистрирующих и отображающих устройств. Это делает желательным использование цветовой коррекции как в телевизионных системах, так и в промышленных системах наблюдения и контроля.

Наиболее простой способ коррекции искажений при передаче изображений используется в аналоговых бытовых телевизионных приемниках. Он получил название коррекции нелинейных искажений [1]. Этот способ основан на знании того, что передающая и индицирующая трубки имеют степенные зависимости между напряжением на модуляторе и яркостью точек на экране, а именно (1) — для передающей и (2) — для индицирующей трубки:

$$U_T = m_T Y_I^{Y_T}, \quad (1)$$

$$Y_O = m_R U_R^{Y_R}, \quad (2)$$

где  $Y_I, Y_O$  — яркости точек исходного объекта и соответствующих точек на отображающем экране соответственно,  $U_T$  и  $U_R$  — напряжения на выходе регистрирующей аппаратуры и на входе отображающей аппаратуры соответственно.

Корректор нелинейных искажений связывает  $U_T$  и  $U_R$  аналогичной зависимостью:

$$U_R = m_K U_T^{Y_K}. \quad (3)$$

При этом получим зависимость исходной и видимой яркостей вида:

$$Y_O = m_R m_K^{Y_K} m_T^{Y_T Y_K} Y_I^{Y_T Y_K Y_R}, \quad (4)$$

где  $m_T, m_R, m_K$  — коэффициенты пропорциональности;  $Y_T, Y_R, Y_K$  — коэффициенты нелинейности передаточных характеристик передающей, приемной аппаратуры и корректирующего блока соответственно.

Из зависимости (4) видно, что для устранения нелинейных искажений необходимо, чтобы

$$Y_T Y_K Y_R = 1 \quad (5)$$

Условие (5) реализуется путем выбора соответствующего значения  $Y_K$ . При необходимости подчеркнуть контраст преимущественно светлых мест это значение следует увеличить, если же требуется подчеркнуть контраст темных мест, например при «вялых» изображениях, значение  $Y_K$  следует уменьшить.

Устройства, реализующие зависимость (3), называются гамма-корректорами. Наиболее распространенные схемы таких устройств основаны на кусочно-линейной аппроксимации. В основе этих схем

лежит аналоговая коммутация резисторов в обратной связи усилительного тракта, благодаря которой реализуется нелинейная передаточная характеристика. Качество аппроксимации зависит от количества сравнивающих и коммутирующих элементов.

Существенным недостатком такой реализации является невозможность подстройки передаточной характеристики и ее реализация исключительно для яркостной составляющей видеосигнала.

Современные высококачественные системы обработки регистрации и обработки видеoinформации для представления цветного изображения используют колориметрическую систему RGB (RED, GREEN, BLUE), в которой цвет любой точки изображения формируется путем комбинации трех базовых цветов: красного, зеленого и синего. Используются и другие форматы, такие как YUV, в которых изображение представлено в виде яркости и цветоразностных сигналов. При этом для сокращения объема передаваемой и хранимой информации используют одинаковые значения цветоразностных сигналов для соседних точек в строке (стандарт 4:2:2) либо в строке и столбцах (стандарт 4:1:1). Однако, по мнению экспертов, это приводит к видимому ухудшению качества изображений.

В [2] рассмотрены математические основы цветовой коррекции изображений в колориметрической системе RGB. Пусть имеется некоторый объект, изображение которого в плоскости OXY можно представить в виде системы функций, определяющих яркость и цветность объекта в точке  $(x, y)$ .

$$\begin{cases} R_I(x, y); \\ G_I(x, y); \\ B_I(x, y). \end{cases} \quad (6)$$

В результате обработки изображение, предлагаемое зрителю, описывается системой функций

$$\begin{cases} R_O(x, y); \\ G_O(x, y); \\ B_O(x, y). \end{cases} \quad (7)$$

связанных с исходной системой (6) соотношением:

$$\begin{cases} R_O(x, y) = H_{RR}(H_{KR}(H_{TR}(R_I(x, y))))); \\ G_O(x, y) = H_{RG}(H_{KG}(H_{TG}(G_I(x, y))))); \\ B_O(x, y) = H_{RB}(H_{KB}(H_{TB}(B_I(x, y))))). \end{cases} \quad (8)$$

где  $H_{TR}, H_{TG}, H_{TB}$  — передаточные характеристики цветных каналов регистрирующей аппаратуры;  $H_{KR}, H_{KG}, H_{KB}$  и  $H_{RR}, H_{RG}, H_{RB}$  — передаточные характеристики обрабатывающей (корректирующей) и отображающей аппаратуры.

Передаточные характеристики регистрирующей и отображающей аппаратуры считаются известными и не подлежащими изменению. Тогда для достижения точности передачи изображения наблюдателю, то есть для обеспечения тождественности систем (6) и (7), можно воспользоваться только изменени-

ем передаточных характеристик обрабатывающей аппаратуры. При этом

$$\begin{cases} H_{KR}(H_{TR}^{-1}) = H_{RR}^{-1}; \\ H_{KG}(H_{TG}^{-1}) = H_{RG}^{-1}; \\ H_{KB}(H_{TB}^{-1}) = H_{RB}^{-1}. \end{cases} \quad (9)$$

где  $H^{-1}$  — обратные функции.

Жесткая реализация соотношения (9) не является желательной. Причина — в отсутствии точных аналитических зависимостей для передаточных характеристик передающей и отображающей аппаратуры вследствие их отличия для различных ее типов и условий применения (как в приведенном примере с видеокамерой). Поэтому задача воспроизведения соотношения (9) расширяется до реализации переформируемых передаточных характеристик корректирующего блока. В таком случае корректирующий блок процессора обработки изображений выполняет преобразование согласно соотношению:

$$\begin{cases} R_O(x, y) = h_R(R_I(x, y)); \\ G_O(x, y) = h_G(G_I(x, y)); \\ B_O(x, y) = h_B(B_I(x, y)). \end{cases} \quad (10)$$

где  $h$  — передаточная характеристика соответствующего цветового канала корректора.

Передаточные характеристики задаются в управляющей программе.

## 2.2. Реализация микширования изображений

Другой базовой операцией колориметрической обработки изображений является микширование. При зональном микшировании граница раздела двух изображений может быть замкнутой (фигура) и разомкнутой (шторка). Во время действия эффекта фигура или шторка может быть подвижной или статичной, может вращаться, перемещаться, быть плавной или резкой, окаймленной или модулированной и т. д. Окаймление (бордер) может иметь разную ширину и цвет. Закон модуляции границы может быть неизменным, а также может меняться во время видеоэффекта. Фигуры в процессе выполнения эффекта могут трансформироваться одна в другую, переходить от фигуры к шторке и обратно.

Аддитивное микширование заключается в суммировании видеосигналов нескольких изображений по формуле  $(K_A \cdot A) + (K_B \cdot B) + \dots$ . Составляющие комбинированного изображения при этом как бы просвечивают одна через другую. Для широко применяемого режима X-образного микширования регулируется только один коэффициент  $K_A$  (в пределах от 0 до 1), а второй вычисляется по формуле  $K_B = K_A - 1$ . Контраст составляющих в процессе микширования всегда понижен. В режиме V-образного микширования (переход через «черное») сначала при  $K_B = 0$  коэффициент  $K_A$  уменьшается от 1 до 0, а затем при  $K_A = 0$  коэффициент  $K_B$  возрастает от 0 до 1.

Неаддитивное микширование представляет собой объединение двух видеосигналов  $A$  и  $B$  в соответствии с формулой  $\max(A, B)$ . Этот видеоэффект применяется очень редко.

Наиболее часто используемое аддитивное и зональное микширование математически можно представить в виде линейной комбинации составляющих микшируемых изображений, независимо от используемой колориметрической системы. В цифровой системе обработки видеoinформации, использующей колориметрическую систему RGB, операция микширования может быть представлена в виде:

$$\begin{cases} R_O(x, y) = K_{R1}(x, y)R_{I1}(x, y) + K_{R2}(x, y)R_{I2}(x, y); \\ G_O(x, y) = K_{G1}(x, y)G_{I1}(x, y) + K_{G2}(x, y)G_{I2}(x, y); \\ B_O(x, y) = K_{B1}(x, y)B_{I1}(x, y) + K_{B2}(x, y)B_{I2}(x, y). \end{cases} \quad (11)$$

где буквенный индекс определяет принадлежность к цветовому каналу, а цифровой индекс — к шине данных.

Для задания зоны микширования вводится понятие коэффициента присутствия. Этот коэффициент выражает долю интенсивности цветовой составляющей в данной точке результирующего изображения, взятую от интенсивности этой цветовой составляющей в той же точке исходного изображения. Вводится также функция присутствия  $K = K(x, y)$ , выражающая зависимость значения коэффициента присутствия  $K$  от координат  $(x, y)$  точки на экране.

Функция присутствия в общем случае может быть различной для различных цветовых составляющих изображения. Однако тогда произойдет изменение не только яркости, но и цветности исходного изображения в результирующем. Это, в свою очередь, приведет к искажению информации об объекте. В случае тождественности функций присутствия для различных цветовых составляющих одного и того же изображения происходит только изменение яркости исходного изображения. В этом случае информация о цвете объекта не искажается. В дальнейших рассуждениях будем считать, что  $K_{R1} = K_{G1} = K_{B1}$  и  $K_{R2} = K_{G2} = K_{B2}$ .

При выполнении аддитивного микширования в общем случае значения коэффициентов  $K_1$  и  $K_2$  являются независимыми. При сложении (соотношение (11)) также возможно переполнение разрядной сетки, устраняемое использованием арифметики с насыщением. Однако это приводит к частичной потере информации об объектах на микшируемых изображениях. Для устранения этого недостатка широко используется X-образное микширование. Оно заключается в установлении связи между функциями  $K_1$  и  $K_2$ :

$$K_2 = 1 - K_1. \quad (12)$$

Управление значениями функции присутствия позволяет формировать фигуры и шторки различных видов.

Как правило, фигуры, используемые при зональном микшировании, симметричны относительно двух осей (ромбы, прямоугольники, кресты, овалы, правильные многогранники с четным числом граней и др.). Это позволяет записывать и хранить в памяти только  $\frac{1}{4}$  часть фигуры. С помощью специального алгоритма чтения такой памяти, возможен синтез всей фигуры. Рассмотрим этот алгоритм на примере фигуры типа «ромб», представленной на рис. 1.

На рис. 1 представлена фигура, для изображения которой используется блок памяти размером  $20 \times 10$  ячеек. Геометрический процессор формирует адреса чтения данной памяти. Для простоты рассуждений считаем, что геометрические преобразования не производятся. Это означает, что точке с координатами  $(x, y)$  экрана соответствует точка с теми же координатами в памяти. Анализ точек изображения фигуры в пределах любой из строк показывает, что чтение точки с координатой  $x = 10$  в точку экрана с той же координатой возможно заменить чтением точки с координатой  $x = 9$  вследствие симметрии изображения фигуры относительно вертикальной оси  $O_1$ . Аналогично можно рассуждать для остальных точек, находящихся справа относительно оси  $O_1$ , а также относительно точек в пределах любого из столбцов относительно оси  $O_2$ . Таким образом, в память достаточно записать изображение  $\frac{1}{4}$  части фигуры. Рассмотренные значения координат являются взаимно дополнительными до максимального значения этой координаты ( $10 + 9 = 19$ ,  $11 + 8 = 19$  и т. д.). Отсюда делаем вывод о том, что для реализации синтеза симметричной фигуры по  $\frac{1}{4}$  ее изображения необходимо вычислить дополнение для адресов  $A_X$  и  $A_Y$  по осям  $X$  и  $Y$  соответственно, поступающим от геометрического процессора:

$$\begin{aligned} DA_X &= \max(A_X) - A_X, \\ DA_Y &= \max(A_Y) - A_Y. \end{aligned} \quad (13)$$

Адреса  $ARd_X$ ,  $ARd_Y$  чтения памяти, хранящей изображение фигуры, формируем согласно соотношению:

$$\begin{aligned} ARd_X &= \begin{cases} A_X, & A_X < DA_X; \\ DA_X, & A_X > DA_X; \end{cases} \\ ARd_Y &= \begin{cases} A_Y, & A_Y < DA_Y; \\ DA_Y, & A_Y > DA_Y. \end{cases} \end{aligned} \quad (14)$$

Этот алгоритм применим и для фигур, симметричных только относительно одной оси (например,

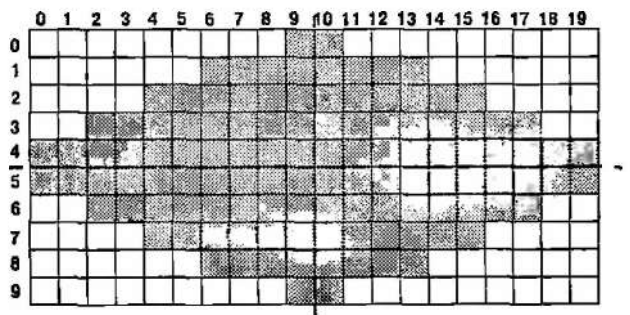


Рис. 1. Пример задания фигуры

цятиконечная звезда). В этом случае в память необходимо прописывать  $\frac{1}{2}$  часть изображения фигуры. Недостаток алгоритма в том, что он неприменим для несимметричных фигур.

Различные типы шторок могут быть получены аналогично фигурам, а также с помощью масштабирования и (или) размножения малой, прописываемой части памяти на всю длину или ширину экрана. Масштабирование, а также перемещение и поворот памяти осуществляется геометрическим процессором, а размножение в простейшем случае — путем маскирования части разрядов адресов чтения памяти. Диагональные шторки формируются аналогично вертикальным и горизонтальным, однако совместно с масштабированием и размножением применяется поворот памяти.

Еще одним недостатком такого способа формирования функций присутствия является необходимость перезаписи памяти для изменения вида фигуры или шторки.

Таким образом, и для реализации цветовой коррекции, и для синтеза функций присутствия следует выполнять аппроксимацию функций от нескольких переменных. Такая задача может быть решена множеством способов. При выборе одного из них главным критерием является возможность аппаратной реализации в реальном времени. При этом временные затраты на ее реализацию должны быть минимальны.

### 3. Использование нейронной сети СМАС для решения поставленных задач

Для решения поставленной задачи оказалось эффективным применение предложенной Дж. Альбусом модели, описывающей процессы управления движением, происходящие в мозжечке, и названной им СМАС — Cerebellar Model Articulation Controller [3, 4]. Эта нейронная сеть работает как ассоциативная память. При этом объем физической памяти для хранения настраиваемых параметров сети, требуемый для аппроксимации функции от нескольких переменных, существенно меньше соответствующего объема при табличном задании функции. Принцип работы данной сети состоит в следующем. Запоминаемая (аппроксимруемая) функция  $y = f(x)$  задается на ограниченном числе точек (значений аргументов)  $x$ , образующих  $N$ -мерное пространство входных сигналов. Данное пространство разбивается на  $M$  подпространств, образуемых входными сигналами  $x(i)$  ( $i = \overline{1, M}$ ). Для хранения значений функции  $y(i)$  (выходных сигналов сети), соответствующих  $x(i)$  ( $i = \overline{1, M}$ ), используется  $p$  ячеек памяти, число которых является постоянным для всех векторов входных сигналов данной сети. При поступлении на вход сети некоторого сигнала  $x(i)$  на ее выходе появляется сигнал  $y(i)$ , представляющий собой сумму содержимого  $p$  адресуемых ячеек.

Отличительной особенностью СМАС является использование в ней специального кодирования, позволяющего резко уменьшить объем памяти, необходимый для хранения обрабатываемой информации. Кодирование информации заключается в том, что каждому  $N$ -мерному входному вектору  $x(i)$  ставится в соответствие  $n$ -мерный ( $n < N$ ) вектор ассоциаций  $a(i)$ , элементы которого могут принимать значения из интервала  $[0, 1]$ . При этом только  $r \ll n$  элементов данного вектора имеют отличные от нуля значения, т. е. только  $r$  элементов памяти являются активными. Дополнительное уменьшение памяти достигается применением хеширования информации.

Структура сети СМАС приведена на рис. 2. В общем случае сеть осуществляет следующие преобразования:

$$S: X \Rightarrow A, H: A \Rightarrow A', P: A' \Rightarrow y, \quad (15)$$

где  $X$  —  $N$ -мерное пространство непрерывных входных сигналов;  $A$  —  $n$ -мерное пространство ассоциаций;  $A'$  — преобразованное алгоритмом хеширования пространство ассоциаций;  $y$  — вектор выходных сигналов.

Преобразования (15) соответствуют кодированию, хешированию информации и вычислению выходного сигнала.

Важным вопросом при реализации сети является выбор базисных функций входного слоя нейронов. В традиционной сети СМАС используются прямоугольные базисные функции, что позволяет осуществлять кусочно-постоянную аппроксимацию. В этом случае вычислительные затраты будут минимальными, вследствие чего значительно сокращается время реакции сети на поступивший входной сигнал. Компоненты вектора ассоциаций при этом могут принимать значения «0» или «1». Скорость обучения сети при выборе базисных функций прямоугольной формы будет максимальной.

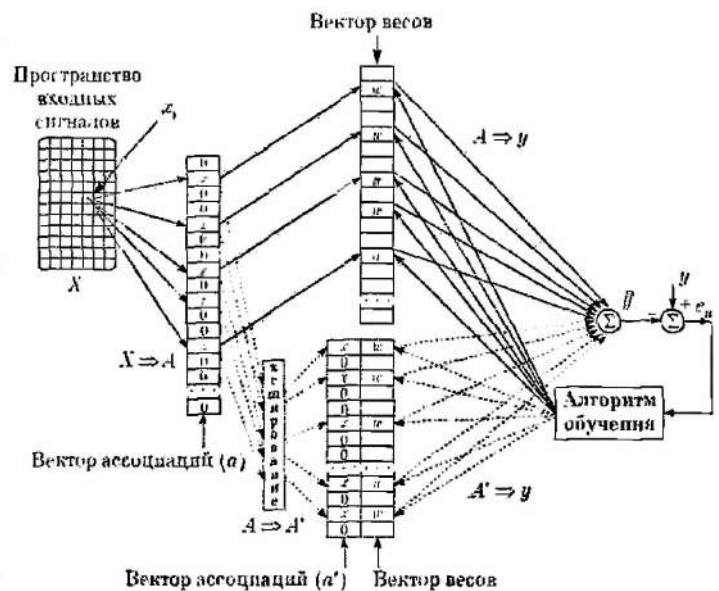


Рис. 2. Структура сети СМАС

Однако в последнее время в нейронной сети СМАС все чаще применяются базисные функции более сложного вида, такие как В-сплайны различных порядков, гауссовские и модифицированные гауссовские функции, косинусоидальные функции. Выбор базисной функции существенно влияет на точность аппроксимации.

Выходной сигнал сети определяется формулой (16).

Обучение сети СМАС состоит в настройке вектора ее весовых параметров  $w$ . Оно происходит на основании предъявления ей обучающих пар  $\{x(k), y(k)\}$  и сравнения реакции сети  $\hat{y}(k)$  с требуемой  $y(k)$ . Алгоритм обучения может быть записан в виде (17).

$$\hat{y} = H(a^T \Phi(x))w, \quad (16)$$

$$\text{где } \Phi(x) = \begin{bmatrix} \Phi_1(x) & 0 & \dots & 0 \\ 0 & \Phi_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Phi_n(x) \end{bmatrix}, \quad \Phi_i(x) = \prod_{j=1}^N \Phi_{ij}(x_j);$$

$\Phi_{ij}(x_j)$  — значение выбранной базисной функции в точке  $x_j$ .

$$w(k+1) = w(k) + \frac{y(k) - \hat{y}(k)}{\|\Phi(x(k))a(k)\|^2} \Phi(x(k))a(k). \quad (17)$$

Сложной является задача выбора оптимального значения параметра  $\rho$ , однако существуют некоторые рекомендации, позволяющие подобрать значение данного параметра близким к оптимальному. Длину вектора ассоциаций и число физических ячеек памяти можно определить по формуле

$$n \approx \left\lceil \rho \left( \frac{R-1}{\rho} + 1 \right)^N \right\rceil, \quad (18)$$

где  $\lceil \bullet \rceil$  означает округление в сторону ближайшего большего целого числа,  $R$  — используемое число уровней квантования входных сигналов,  $\rho$  — число областей квантования.

Из (18) следует, что при выборе  $\rho = 1$  длина вектора ассоциаций составит  $R^N$ , что соответствует табличному способу задания функции. В данном случае обучение сети не проводится и в каждую ячейку памяти записываются соответствующие значения функции. Заметим, что в этом случае в качестве базисных возможен выбор функций только прямоугольной формы и данная сеть будет производить кусочно-постоянную аппроксимацию функции.

Требуемый объем памяти будет минимален в случае выбора  $\rho = R$ . Но в связи с сокращением объема памяти это существенно скажется на точности аппроксимации. Кроме того, при таком выборе значения параметра  $\rho$  значительно увеличится время, необходимое для кодирования поступившего входного сигнала. Это связано с необходимостью вычисления  $N \cdot R$  значений базисных функций.

Следует заметить, что на некотором интервале  $[1, \rho_{\text{opt}}]$  увеличение значения параметра  $\rho$  приводит к увеличению точности аппроксимации, а в интер-

вале  $[\rho_{\text{opt}}, R]$  с ростом  $\rho$  происходит заметное ухудшение аппроксимирующих свойств сети в воз- растающие времени реакции на входной сигнал.

Таким образом, увеличение значения параметра  $\rho$ , с одной стороны, приводит к уменьшению требуемого объема памяти, а с другой — к увеличению времени кодирования входного сигнала, а начиная с некоторого  $\rho_{\text{opt}}$  — к ухудшению аппроксимирующих свойств сети.

При практической реализации сети СМАС рекомендуется вначале выбрать значение параметра  $\rho$  равным  $\left\lfloor \frac{R}{2} \right\rfloor$ , а затем, в случае получения неудовлетворительных результатов, производить его увеличение либо уменьшение, оценивая изменение времени кодирования входного сигнала, объема требуемой памяти и точности аппроксимации.

#### 4. Модель синтеза функций присутствия при микшировании изображений в реальном времени на основе нейронной сети СМАС

Аппаратная реализация сети СМАС рассмотрена в [5]. Там предлагается использовать значения числа ступеней квантования, равные степеням числа 2. Это позволяет существенно снизить аппаратные затраты на реализацию кодирования переменных.

В случае использования сети СМАС для синтеза функций присутствия при микшировании входными переменными являются координаты точки на экране, которые при построчной развертке изображения изменяются последовательно. В этом случае упрощается реализация блока кодирования переменных для произвольного числа ступеней. В результате анализа затрат в качестве переменной, для которой номера ступеней кодирования изменяются последовательно, выбрана координата  $y$ . Структура блока кодирования переменных для этого случая приведена на рис. 3.

Частота, с которой осуществляется кодирование, должна быть в  $\rho$  раз больше частоты следования точек. Для ее синтеза используется система автоматической подстройки частоты PLL. Вследствие того, что для любой строки переменная  $y$  постоянна, для ее кодирования используется счетчик с модулем  $\rho$ , который формирует аргумент базисной функции  $u_y^j$  и значение переноса, используемое счетчиком формирования номера нейрона на ступени кодирования  $NNS_y^j$  переменной  $y$ . Для устранения диагональной структуры ассоциативного поля номера ступеней для кодирования переменной  $x$  не должны изменяться последовательно. Они формируются блоком формирования  $j_x$  и, поступая на сумматоры совместно со значениями аргумента базисной функции  $u_x^0$  и номера нейрона  $NNS_x^0$  на 0-й ступени кодирования, формируют соответствующие значения на ступени  $j$ .

Микширующий блок для работы с видеосигналами в реальном времени, использующий для синтеза

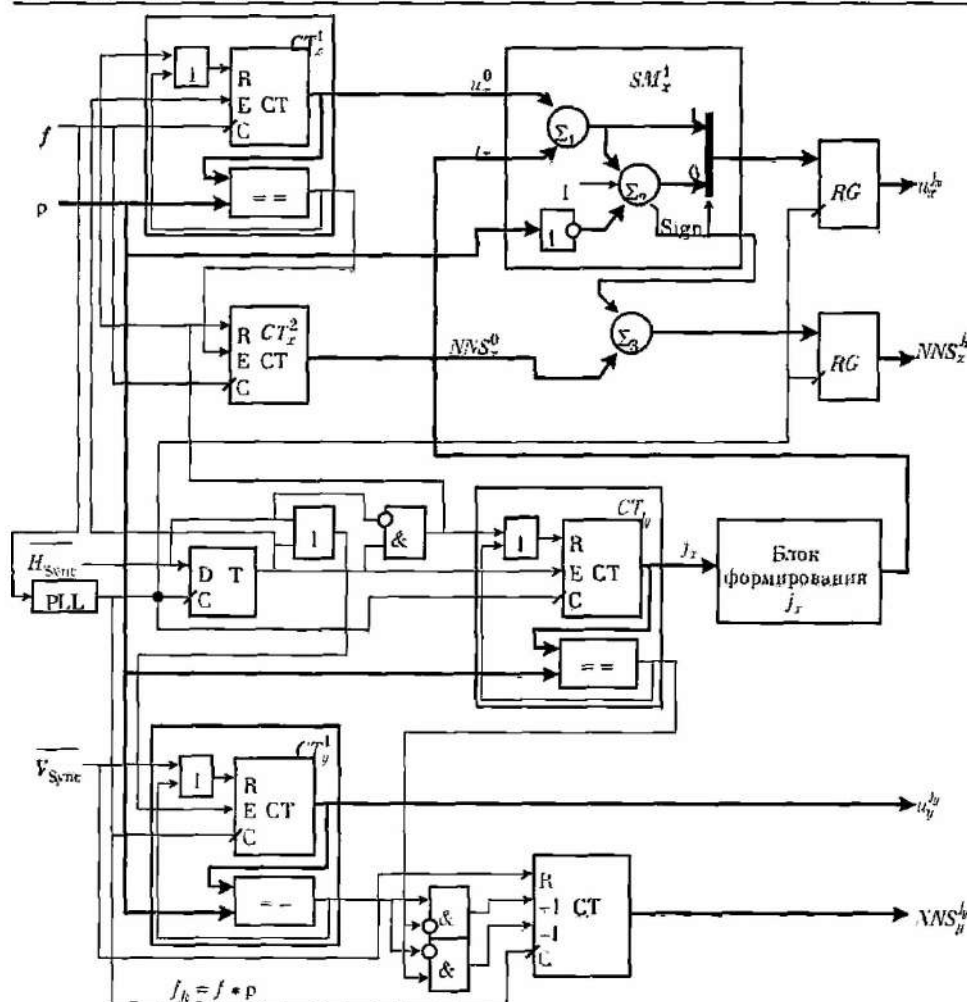


Рис. 3. Структура блока кодирования последовательно изменяющихся переменных

функций присутствия сеть СМАС, представлен на рис. 4.

Структуру сети СМАС образуют блок кодирования переменных (БКП), оперативные запоминающие устройства для хранения значений произведений базисных функций (ОЗУ ПФФ) и значений весов (ОЗУ весов), умножитель M1 и накапливающий сумматор Σ1 — Rg1. Мультиплексоры MX1 и MX2, а также буферные элементы B1 и B2 необходимы для инициализации ОЗУ. БКП в качестве входных переменных использует сигналы тактовой частоты  $f_t$  и строчной синхронизации  $H_{Sync}$ . Эти сигналы соответствуют последовательно перебираемым координатам  $x$  и  $y$  текущей точки в микшируемых изображениях. Структура БКП представлена на рис. 3.

Вычисленное сетью СМАС значение функции присутствия для данной точки через мультиплексор MX3 поступает на регистр Rg2, в котором сохраняется в течение длительности точки. Мультиплексор MX3 предназначен для обеспечения возможности микширования изображений с использованием постоянного для всего кадра значения функции присутствия либо с использованием  $\alpha$ -канала, формируемого на шине АК. Последний удобен для вывода на изображение логотипов или титров различной степени прозрачности и сложности.

Операция микширования выполняется на умножителе M2 и сумматоре Σ2. Для сокращения аппаратных затрат в структуре предложена последовательная реализация соотношения (11). Первую половину периода тактовой частоты через мультиплексор MX4 на умножитель M2 поступает значение интенсивности цветовой составляющей с шины данных ШД1. На второй вход умножителя через управляемый инвертор =1 (по своей сути это схема «исключающее ИЛИ», на один вход которой подано значение функции присутствия, а на второй — тактовая частота следования точек) поступает значение функции присутствия в прямом коде. По истечении первой половины периода вычисленное умножителем значение произведения  $K \cdot I_1$  запоминается в регистре

Rg3. В течение второй половины периода тактовой частоты на умножитель M2 поступают значение интенсивности с шины данных ШД2 и значение функции присутствия в обратном коде. Вычисляется произведение  $(1-K) \cdot I_2$  и поступает на сумматор Σ2, на который также поступает сохраненное в регистре Rg3 значение произведения  $K \cdot I_1$ . Вычисленное значение суммы произведений запоминается в регистре Rg4.

Теоретически область допустимых значений функции присутствия определяется интервалом [0, 1]. Однако на практике использование дробных значений приводит к усложнению вычислений и потере их наглядности. Поэтому на практике предлагается считать областью определения функции присутствия интервал  $K = [0, 255]$ , а результирующее значение интенсивности определять по соотношения:

$$I(x, y) = \frac{K(x, y)I_1(x, y)}{256} + \frac{(255 - K(x, y))I_2(x, y)}{256},$$

$$K(x, y) \in [0, 255]. \tag{19}$$

Операция деления на 256 выполняется путем сдвига результата умножения на 8 разрядов влево или, что то же самое, путем взятия в качестве

результата перемножения старших 8-ми разрядов результата.

Для замены выражения (1-K) в (12) на (255-K) в (19) сделаем следующее допущение: за 1 принимается значение  $K = 255$ , а не 256, как этого следовало ожидать. Такое допущение не приводит к видимым изменениям в результирующем изображении и может быть выполнено путем инвертирования значения  $K$ . Это позволяет несколько сократить аппаратные затраты на перевод обратного кода в дополнительный.

В целом реализация эффектов микширования с использованием структуры, приведенной на рис. 5, состоит в следующем. Оператор, работающий за управляющей ЭВМ, создает изображение шторки или фигуры путем ее построения в каком-либо графическом редакторе. Возможна также генерация шторки или фигуры путем программной реализации некоторой аналитической зависимости. Сформированное изображение фигуры или шторки представляется программной модели сети СМАС, которая вычисляет значения весов, необходимые для аппроксимации этого изображения. Полученный массив весов передается по интерфейсу «ЭВМ — блок связи видеопроцессора». Далее адреса и значения весов по шине команд и данных (ШКД) поступают на микширующий блок, где после дешифрации команды в блоке УУДК формируются соответствующие адреса и данные для инициализации ОЗУ весов. После инициализации ОЗУ весов микширующий блок готов к работе.

Предусмотренное изменение значения  $\rho$  и вида базисных функций обеспечивает возможность изменять соотношение качество/скорость эффекта микширования. Так, при больших значениях  $\rho$  ухудшается качество аппроксимации, но уменьшается объем передаваемых параметров. Качество аппроксимации зависит также и от того, какой вид базисных функций используется для той или иной функции присутствия. Выбор вида базисных функций является предметом исследований.

Главным преимуществом предлагаемой реализации по сравнению с табличным методом задания функций присутствия является меньший объем передаваемых параметров, а по сравнению с алгоритмическим методом — высокая универсальность, то есть применимость для несимметричных функций присутствия произвольного вида.

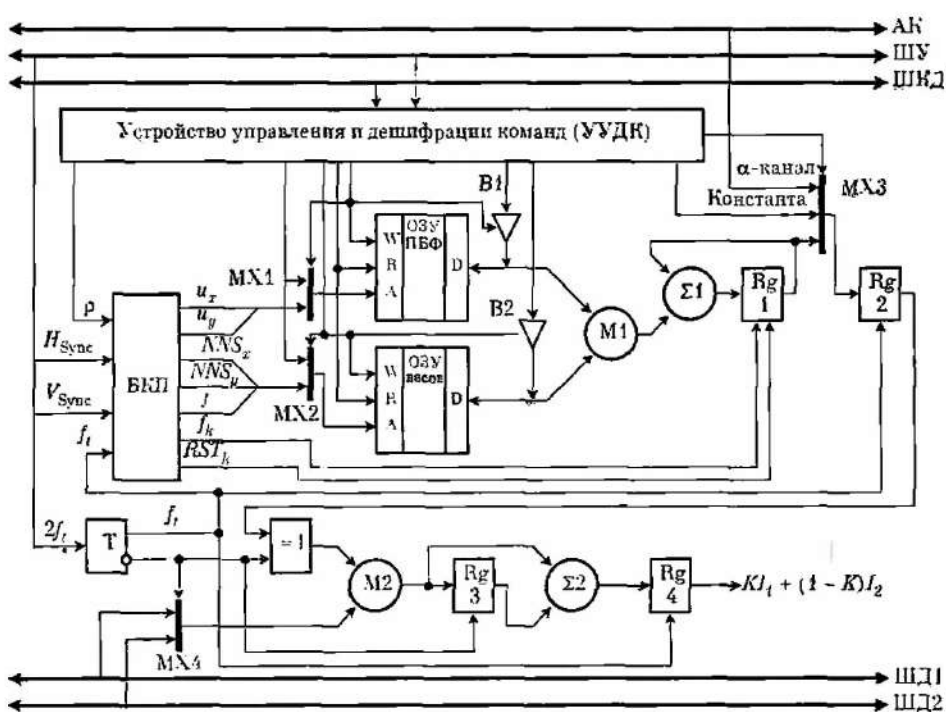


Рис. 4. Структура микширующего блока

В структуре на рис. 4 не используется хеширование вектора ассоциаций. Это связано с тем, что требуемые объемы памяти достаточно малы, а возникновение хеш-коллизии может привести к нежелательным артефактам на функциях присутствия, что не оправдывается еще большим сокращением объемов памяти.

### 5. Модель цветовой коррекции изображений в реальном времени на основе нейронной сети СМАС

Структура цветового корректора, в основе которого лежит аппроксимация передаточной характеристики сетью СМАС, представлена на рис. 5. В ее основе лежит сеть СМАС, имеющая три входные переменные и вычисляющая три функции.

Для реализации сети имеется три идентичных блока кодирования переменных R, G, B (соответственно БКП R, БКП G, БКП B). ОЗУ для хранения произведенных базисных функций (ОЗУ ПБФ), общее для трех реализуемых функций (ОЗУ для хранения весов для каждой цветовой составляющей (ОЗУ весов R, G и B), умножители M1 и накапливающие сумматоры Σ1, вычисляющие произведение векторов весов и коэффициентов. В рассматриваемой структуре используется колориметрическая система RGB как наиболее наглядная и обеспечивающая наилучшее качество изображения. Для системы YUV структура будет идентичной. Изменится лишь интерфейс задания передаточной характеристики. Подобно рассмотренной в структуре микширующего блока имеются элементы (мультиплексоры MX1, MX2R, MX2G, MX2B и буферные элементы B1, B2R, B2G, B2B) для инициализации используемых в структуре ОЗУ значениями от управляющей ЭВМ.

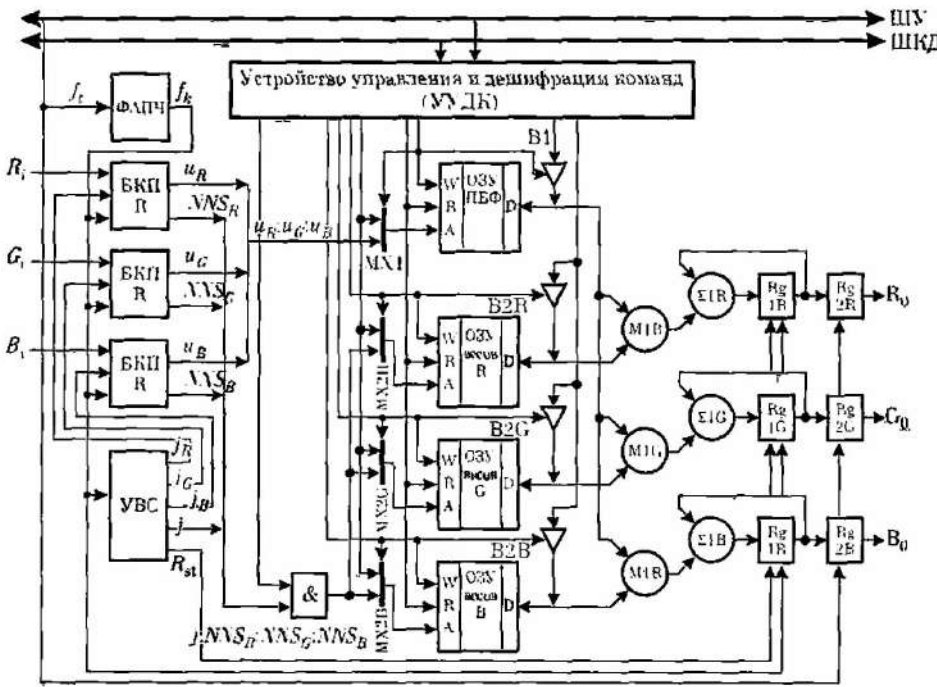


Рис. 5. Структура цветного корректора

В рассмотренной структуре имеется элемент «логическое П», осуществляющий управляемое маскирование старших разрядов формируемого адреса в ОЗУ весов. Так предлагается выполнять операцию хеширования вектора ассоциаций по модулю, равному степени числа 2. Хеширование применено для сокращения объема передаваемых из ЭВМ в видеосистему данных. К примеру, при задании передаточной характеристики оператор желает видеть изменения на реальном видеотоке. Для этого вводится большой коэффициент хеширования, чем обеспечивается сокращение объема настраиваемых и передаваемых весов. При этом качество аппроксимации будет снижено, но достаточно для визуальной оценки выполняемых корректировок. Когда будет получен требуемый результат, степень хеширования уменьшается либо вообще не используется. Настраиваются и передаются все веса, а качество аппроксимации будет максимальным.

В отличие от синтезатора функций присутствия, входные переменные цветного корректора изменяются последовательно. Вследствие этого предлагается использовать блок кодирования переменных, работающий со значениями  $\epsilon$ , равными степени числа 2. Такой блок рассмотрен в [5]. Отличие состоит в том, что на выходе БКП идет не значение базисной функции, а ее аргумент  $u$ . Эти аргументы путем конкатенации разрядов объединяются и поступают на адресный вход ОЗУ хранения произведений базисных функций. Основной эффект от такой реализации состоит в сокращении аппаратных и временных затрат на реализацию вычисления произведения

трех переменных и упрощения схем инициализации ОЗУ. При значениях  $\rho = 4, 8, 16$  и 32 объем ОЗУ ПБФ составит соответственно 64, 512, 4096 и 32768 ячеек.

Система ФАПЧ обща для всех трех БКП и предназначена для синтеза частоты кодирования.

Устройство выборки ступени формирует номера ступеней кодирования для каждого БКП. Оно может быть реализовано как счетчик с непоследовательным счетом, либо как линейный рекуррентный регистр с добавленным комбинации «все нули».

Традиционный способ задания кривой коррекции,

основанный на задании опорных точек на передаточной характеристике и интерполяции кривой не дает возможности получить зависимость от всех входных цветовых составляющих. Можно предложить следующий способ, рассмотренный на примере красного цветного канала:

1. Строится передаточная характеристика  $R_{out} = f_R(R_{in}, 0, 0)$  путем графического задания дополнительных точек на графике в системе координат  $(R_{in}, R_{out})$  так, как в традиционном методе.

2. В системе координат  $(G_{in}, B_{in})$  задаем положение точек, в которых интенсивность красного цвета должна зависеть от значений интенсивности зеленого и синего цветов.

3. В системах координат  $(G_{in}, \Delta R_{out}), (B_{in}, \Delta R_{out})$  задаются величины и знак зависимости красного от зеленого и синего цветов соответственно.

4. Получаем несколько точек передаточной характеристики для канала красного цвета:

$$R_{out} = f_R(R_{in}, 0, 0) + \Delta R_{out}(G_{in}) + \Delta R_{out}(B_{in}).$$

5. Предъявляем полученные точки в качестве обучающих программной модели сети СМАС.

Главными достоинствами предлагаемой реализации являются сокращение объема памяти и передаваемых данных по сравнению с табличным методом при сохранении универсальности и широких возможностей цветокоррекции, которые обеспечиваются за счет синтеза передаточных характеристик, зависящих от всех цветовых составляющих изображения.

**6. Заключение**

Предлагаемые модели синтеза функций присутствия при микшировании изображений и реализации цветовой коррекции, основанные на аппроксимации функций от нескольких переменных нейронной сетью СМАС в реальном времени, в качестве преимущества по сравнению с существующими табличными методами позволяют снизить объем хранимых и передаваемых параметров. Это не только позволяет сократить аппаратные затраты, но и получить новые преобразования и видеоэффекты, связанные с микшированием, использующим несимметричные функции присутствия сложной формы, и с синтезом передаточных характеристик цветового корректора, зависящих от всех трех цветов разложения исходного изображения. Предложен способ задания таких передаточных характеристик в управляющей программе. Для оценки снижения объемов параметров приведем таблицу 1 для модели синтеза функций присутствия и таблицу 2 для модели цветовой коррекции.

Таблица 1

Табличный метод для фигур		Предлагаемый метод для произвольных фигур			
произвольных	симметричных	$\rho = 4$	$\rho = 8$	$\rho = 16$	$\rho = 32$
207 360	51 840	52 600	26 810	13 920	7488

Из таблицы 1 видно, что при  $\rho = 4$  предлагаемая модель для произвольных фигур практически не уступает табличному методу для симметричных относительно обеих осей фигур и почти в 4 раза сокращает объем памяти для произвольных фигур. Уже

при  $\rho = 8$  предлагаемая модель почти в 2 раза сокращает объем памяти даже по сравнению с табличным методом для симметричных фигур и почти в 8 раз для произвольных. Большие значения  $\rho$  позволяют сократить объем памяти до 27 раз.

Таблица 2

Табличный метод	Предлагаемый метод			
	$\rho = 4$	$\rho = 8$	$\rho = 16$	$\rho = 32$
3·16 777 216	3·1085 874	3·284 242	3·77 744	3·23 086

В приведенной таблице представлены объемы памяти для 8-битного представления цветовых составляющих. Для предлагаемой модели выполнения цветовой коррекции объемы указаны без выполнения кеширования. Как видно из приведенных данных, экономия объемов составляет от 15,5 до 726 раз. С применением кеширования эти показатели могут быть увеличены еще в несколько раз.

Список литературы: 1. Дюбругов Р. М. Телевидение: 2-е изд. — К.: Вища школа, 1988. — 215 с. 2. Основные задачи колориметрической обработки изображений и их решение в цифровом видеопроцессоре // Вестник НТУ ХПИ — 2003 — № 5. 3. Albus J. S. A new approach to manipulator control: the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control. — 1975. — Vol. 97, № 3. — P. 220-227. 4. Albus J. S. Data storage in cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control. — 1975. — Vol. 97, № 3. — P. 228-233. 5. Бессонов А. А., Бобух В. А., Руденко О. Г. Аппаратная реализация нечеткой сети СМАС и ее применение для задач сжатия изображений // Автоматика. Автоматизация. Электротехнические комплексы и системы. — 2005. — № 2(16). — С. 47-52.

Поступила в редколлегию 21.08.2006