

А.А. Дунаев, Т.Ф. Валеев, Е.А. Тарасов

## ИССЛЕДОВАНИЕ МЕТОДОВ ОРГАНИЗАЦИИ ВИЗУАЛЬНОЙ ОБРАТНОЙ СВЯЗИ В АППАРАТНО-ПРОГРАММНОМ КОМПЛЕКСЕ «БОСЛАБ»

### ВВЕДЕНИЕ

Специалистами НИИ молекулярной биологии и биофизики СО РАМН<sup>1</sup> разработан аппаратно-программный комплекс «Бослаб», позволяющий регистрировать физиологические параметры испытуемого человека, такие как температура кожи, мышечная активность (миограммы), кардиоинтервалы, биоэлектрическая активность головного мозга (электроэнцефалограммы) и т. п. «Бослаб» организует работу с испытуемым в виде сеанса, состоящего из сессий — различных задач (тестов, лечебных процедур), во время выполнения которых испытуемым программа непрерывно регистрирует состояние параметров, интересующих экспериментатора. После окончания сеанса экспериментатор имеет возможность просмотреть ход работы и сопоставить с ним изменения зарегистрированных параметров. Параметры процедур в «Бослабе», заданные до начала сеанса, лишь частично могут быть изменены в ходе выполнения. Например, пороговые значения, относительно которых в тестах необходимо изменить физиологические параметры (повышать температуру, снижать мышечное напряжение) для эффективного выполнения задания представлены, в основном, в простейшем графическом варианте — в виде линий, столбиков. Они могут быть изменены лишь в ручном режиме в процессе тестирования. К сожалению, такая форма представления сигналов не способствует повышению мотивации испытуемого и не использует современные возможности обработки изображений. С другой стороны, существующие в «Бослабе» игровые представления тестов трудоемки в создании, и их настройки не могут быть изменены в ходе выполнения. Данное обстоятельство существенно ограничивает свободу действий экспериментатора, вынуждая его прерывать работу с испытуемым для корректировки параметров.

В рамках данной работы создано приложение для проведения тестов, обеспечивающее визуальную и звуковую обратную связь с испытуемым в

---

<sup>1</sup> Ранее — Институт медицинской и биологической кибернетики СО РАМН.

комплексе «Бослаб» и позволяющее изменять параметры теста непосредственно во время проведения сеанса.

Программа представляет собой приложение для ОС Microsoft Windows 2000/XP и запускается «Бослабом» в качестве внешнего приложения для игрового представления рабочей сессии. Предусмотрены два режима работы программы — настройка и воспроизведение. Используя совместно функции «Бослаба» и описываемой программы, экспериментатор имеет возможность создать набор готовых рабочих сессий и использовать их в дальнейшей работе.

## 1. ОБЩЕЕ УСТРОЙСТВО СИСТЕМЫ

### 1.1. Принцип работы

«Бослаб» регистрирует одновременно до 14 различных типов сигналов: ЭЭГ, ЭМГ, температура, альфа-, бета- и тета-ритмы и т. п., причём для некоторых сигналов предусмотрены два канала. Общее количество регистрируемых параметров — 22. Один из параметров может быть выбран для передачи внешнему приложению, в роли которого может выступать описываемая программа.

Приложение получает от «Бослаба» данные о текущем состоянии выбранного сигнала, используя функции, экспортируемые специальной динамической библиотекой. Для каждого вида сигнала существует характерный диапазон значений (например, значение сигнала ЭЭГ лежит в диапазоне  $-100 \div 100$  мкВ), в пределах которого экспериментатором задаётся эталонное значение параметра. Вычисляемое значение отклонения регистрируемого значения от эталона используется в качестве параметра обратной связи с испытуемым.

Обратная связь организуется следующим образом. Испытуемому на дисплее ПК предлагается изображение, в которое программа вносит помехи или искажения. Сила помех или степень искажений зависит от параметра обратной связи. Разработаны разнообразные методы генерации помех и искажений (далее — «эффекты»), и в зависимости от особенностей теста и индивидуальных качеств испытуемого может быть выбран наиболее подходящий эффект. Применение эффекта может также сопровождаться звуком.

## 1.2. Интеграция с «Бослабом»

«Бослаб» даёт возможность вызвать внешнее приложение в двух режимах: настройки и воспроизведения. Требуемый режим устанавливается передачей программе ключа через командную строку.

В режиме настройки «Бослаб» создаёт в рабочей директории программы специальный инициализационный файл, после чего запускает приложение. Далее экспериментатор формирует будущую рабочую сессию, для чего используется весь пользовательский интерфейс программы. Сессия содержит последовательность изображений, каждое из которых хранится в отдельном файле в одном из общепринятых форматов<sup>2</sup>.

Для каждого элемента последовательности задаётся длительность интервала времени, в течение которого этот элемент будет работать в тесте, и устанавливается один из доступных эффектов. Помимо последовательности изображений, экспериментатор задаёт диапазон значений, который будет использоваться при тестировании.

Тест сохраняется во внешнем файле, имя которого записывается в инициализационный файл, копия которого затем сохраняется «Бослабом» в его внутренней базе данных.

В режиме воспроизведения «Бослаб» извлекает из базы данных требуемый инициализационный файл и копирует его в рабочую директорию программы. После этого запускается в режиме воспроизведения описываемое приложение, которое автоматически открывает указанный в инициализационном файле тест.

В режиме воспроизведения доступны только элементы пользовательского интерфейса, управляющие граничными и пороговыми значениями сигнала. Экспериментатор имеет возможность подстроить границы диапазона и пороговое значение, после чего вручную запустить тестирование, в ходе которого изображения, заданные в тесте, будут показаны на дисплее ПК с применением соответствующих эффектов в течение заданных интервалов времени.

---

<sup>2</sup> Для загрузки изображений используется сторонняя библиотека NexgenIPL, свободно распространяемая компанией Binary Technologies. Эта библиотека распознаёт более 20 различных графических форматов, включая форматы семейства JPEG, а также такие популярные форматы, как PNG, GIF, TIFF, TGA и др.

## 2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ АЛГОРИТМОВ

### 2.1. Постановка задачи

Поскольку элементы теста полностью однотипны, без ограничения общности можно рассматривать один отдельно взятый элемент, а точнее — заданные в этом элементе файл изображения, эффект и длительность воспроизведения.

Задача программы состоит в применении выбранного эффекта к изображению в течение всего периода воспроизведения. Результатом применения эффекта является другое изображение, которое отображается на дисплее. Параметр обратной связи для эффекта постоянно запрашивается от «Бослаба», причём для эффектов, учитывающих случайные факторы, результат может пересчитываться постоянно.

Важным условием, которому должны удовлетворять алгоритмы обработки, является минимизация задержки, возникающей между моментом изменения сигнала и появлением результата. Для испытуемого реакция программы на его действия должна быть субъективно мгновенной. Известно, что типичное время осознанной реакции человека на визуальные раздражители составляет не менее 100 мсек. Следовательно, чтобы испытуемый не замечал задержки, время полной обработки изображения не должно превышать 100 мсек.

Другое важное качество, которым должен обладать эффект, можно сформулировать так: испытуемый должен уверенно идентифицировать изменения изображения. Иными словами, не должно возникать сомнений в том, проявился ли эффект. С другой стороны, визуальное восприятие разных людей существенно отличается, поэтому универсального эффекта, одинаково хорошо воспринимаемого всеми людьми, может не быть.

Кроме того, эффекты должны удовлетворять ряду простых условий. При значении параметра обратной связи, равном нулю, в изображение не должно вноситься никаких искажений. При возрастании или убывании значения параметра сила эффекта не должна, соответственно, убывать или возрастать (хотя допускается ступенчатое изменение).

### 2.2. Обозначения

Будем считать изображением прямоугольный растр точек, каждая из которых может иметь свой цвет. Как уже отмечалось выше, эффект является,

по сути, преобразованием исходного изображения, в результате которого получается другое изображение.

При обработке изображений в программе используется система цветowych координат RGB. Цветное изображение размером  $m \times n$  может быть представлено матрицей  $S[m, n]$ , в которой каждый элемент представляет одну точку изображения и является трёхкомпонентным вектором  $(r, g, b)$ , в котором каждая компонента соответствует одной из компонент цвета точки и может принимать значения в интервале  $[0, 255]$ .

Поместим начало координат в верхний левый угол изображения. Оси координат направим влево и вниз. Будем обозначать произвольную точку изображения  $P$  так:  $P_{xy}$ , где  $x$  — смещение точки влево от начала координат (номер столбца), а  $y$  — вниз (номер ряда).

### 2.3. Эффект «случайный шум»

Эффект «случайный шум» изменяет цвет каждой точки, добавляя к каждой компоненте её цвета значение, вычисляемое по следующей формуле:

$$a = (rand(0, 512) - 256) \cdot \frac{p}{100},$$

где  $rand(0, 512)$  обозначает псевдослучайное целое число в интервале  $[0, 512]$ , а  $p$  является параметром обратной связи эффекта и принимает только целые значения от 0 до 100.

Для каждой точки добавка вычисляется один раз и затем суммируется к каждой из компонент. В случае, если результат оказывается за пределами интервала  $[0, 255]$ , выполняется корректировка, и компонента принимает минимальное или максимальное значение в допустимом интервале.

Несмотря на простоту алгоритма, эффект «случайный шум» имеет важное свойство: шум становится заметен уже при значениях параметра обратной связи, равных единицам (то есть, единицы процентов от максимальной силы эффекта).

### 2.4. Эффект размытия («Blur»)

Эффект размытия достигается применением линейного однородного фильтра [1] с размером окрестности  $11 \times 11$  пикселей и весовыми коэффициентами, равными значениям функции Гаусса. Результирующий цвет точки определяется следующим образом:

$$P_{xy}^* = \frac{\sum_{i=-5}^5 \sum_{j=-5}^5 \alpha_{ij} P_{x+i, y+j}}{\sum_{i=-5}^5 \sum_{j=-5}^5 \alpha_{ij}}, \quad (1)$$

где  $P_{xy}$  — цвет точки до обработки. Красная, зелёная и синяя компоненты цвета обрабатываются независимо по формуле (1). Для корректной обработки краёв исходное изображение увеличивается во всех направлениях на 5 пикселей, которые закрашиваются чёрным цветом. Коэффициенты  $\alpha_{ij}$  определяются по формуле:

$$\alpha_{ij} = \exp\left(-\frac{r(i, j)^2}{2\sigma^2}\right), \quad (2)$$

где  $r(i, j) = \sqrt{i^2 + j^2}$  — расстояние до центра, а  $\sigma > 0$  — степень размытия, переменный параметр фильтрации. Из (2) видно, что чем меньше  $\sigma$ , тем сильнее влияние центрального элемента окрестности. Если  $\sigma$  приближается к нулю, то полученное после фильтрации изображение практически не отличается от исходного. При увеличении  $\sigma$  веса всех точек окрестности становятся более близкими, и изображение размывается сильнее.

Подставим (2) в (1) и преобразуем:

$$\begin{aligned} P_{xy}^* &= \frac{\sum_{i=-5}^5 \sum_{j=-5}^5 e^{-(i^2+j^2)/2\sigma^2} P_{x+i, y+j}}{\sum_{i=-5}^5 \sum_{j=-5}^5 e^{-(i^2+j^2)/2\sigma^2}} = \frac{\sum_{i=-5}^5 \sum_{j=-5}^5 e^{-i^2/2\sigma^2} e^{-j^2/2\sigma^2} P_{x+i, y+j}}{\sum_{i=-5}^5 \sum_{j=-5}^5 e^{-i^2/2\sigma^2} e^{-j^2/2\sigma^2}} \\ &= \frac{\sum_{i=-5}^5 \left( e^{-i^2/2\sigma^2} \sum_{j=-5}^5 e^{-j^2/2\sigma^2} P_{x+i, y+j} \right)}{\left( \sum_{i=-5}^5 e^{-i^2/2\sigma^2} \right) \left( \sum_{j=-5}^5 e^{-j^2/2\sigma^2} \right)}. \end{aligned}$$

Обозначим

$$\beta_i = \frac{e^{-i^2/2\sigma^2}}{\sum_{j=-5}^5 e^{-j^2/2\sigma^2}}, \quad (3)$$

тогда

$$P_{xy}^* = \sum_{i=-5}^5 \left( \beta_i \sum_{j=-5}^5 \beta_j P_{x+i, y+j} \right). \quad (4)$$

Используя формулу (4), фильтрацию можно проводить в два этапа. Сначала вычисляется промежуточное изображение

$$Q_{xy} = \sum_{j=-5}^5 \beta_j P_{x, y+j}, \quad (5)$$

а затем конечное

$$P_{xy}^* = \sum_{i=-5}^5 \beta_i Q_{x+i, y}. \quad (6)$$

Таким образом, для обработки одной точки по (5) и (6) требуется 22 умножения и сложения, тогда как по формуле (1) их требовалось не менее 121.

Величина  $\sigma$  определяется из параметра  $p$ , передаваемого фильтру, как  $\sigma = p/20$ . Параметр  $p$  является параметром обратной связи эффекта и принимает только целые значения от 0 до 100. Поэтому величины  $\beta_i$  вычисляются для всех возможных  $\sigma$  по формуле (3) при инициализации фильтра. Во время вычисления очередного кадра используется набор значений, соответствующий текущему  $p$ . Если  $p = 0$ , то фильтрация не производится, т.е.  $P_{xy}^* = Q_{xy} = P_{xy}$ . Это достигается, если положить при  $\sigma = 0$

$$\beta_i = \begin{cases} 1, & \text{при } i = 0 \\ 0, & \text{иначе} \end{cases}.$$

Чтобы процесс фильтрации был достаточно быстрым, вычисление  $Q_{xy}$  и  $P_{xy}^*$  реализовано на ассемблере процессора Intel Pentium с использованием технологии MMX. Для ускорения вычислений полученные при инициализации фильтра значения  $\beta_i$  умножаются на 256 и округляются до целого значения. Таким образом, вычисления по формулам (5) и (6) сводятся к умножению однобайтовых целых и суммированию результатов, которые затем сдвигаются на 8 бит вправо (таким сдвигом достигается деление на 256). С помощью инструкций MMX все компоненты цвета одной точки умножаются, складываются и сдвигаются параллельно.

## 2.5. Препарирование

Препарированием называют класс поэлементных преобразований полутонового изображения, выделяющих фрагменты изображения с определёнными яркостными характеристиками. К классу препарировующих преобразований относятся пороговая обработка, выделение яркостного среза, линейное контрастирование и другие методы [1].

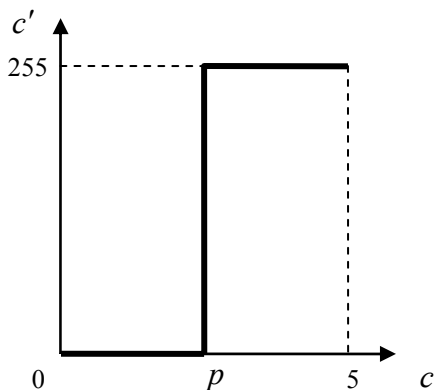


Рис. 1. Пороговое преобразование

Пример препарировующего преобразования — порогового преобразования — показан на рис. 1. Жирной линией показана зависимость яркости  $c'$  точки результирующего изображения от яркости  $c$  точки исходного изображения. Пунктирными линиями выделены максимальные значения яркости, равные  $255^3$ . Пороговое значение  $p$  прямо пропорционально параметру обратной связи.

Таким образом, пороговая обработка может быть описана следующей формулой:

$$c' = \begin{cases} 0, & 0 \leq c < p \\ 255, & p \leq c \leq 255 \end{cases}, \quad 0 \leq p \leq 255.$$

В описываемой программе используются цветные изображения, что даёт возможность препарирования двумя способами: покомпонентно, с пре-

<sup>3</sup> Из соображений удобства обработки для хранения значения яркости одной точки выделяется один байт.

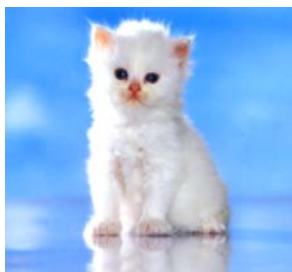


парированием каждой компоненты цвета независимо, и с препарированием значения яркости, вычисленного по значениям компонент. В первом случае результатом преобразования является цветное изображение, во втором — полутоновое. Для второго случая возможны также различные способы вычисления яркости.

Все запрограммированные препарировующие преобразования описаны в приложении 1.

### 2.6. Эффект «чёрная дыра»

Результат применения эффекта «чёрная дыра» показан на рис. 2. Яркость точки результирующего изображения изменяется пропорционально расстоянию от центра и параметру обратной связи.



### 2.7. Геометрические эффекты

Реализованы несколько вариантов геометрических искажений, основанных на сдвиге строк исходного изображения.

Первый способ сдвигает строки изображения поочередно влево и вправо на величину, пропорциональную параметру обратной связи. Другие алгоритмы этой группы сдвигают строки со смещением, изменяющимся в зависимости от номера строки по определённому закону (синусоидальная зависимость, пилообразное изменение и т. п.).

Кроме того, создан эффект «вихрь», закручивающий исходное изображение вокруг центра (рис. 2).

Цвет точки результирующего изображения при применении «вихря» рассчитывается следующим образом. Координаты результирующей точки  $(x_r, y_r)$  переводятся из декартовых в



Рис. 2. Эффекты «чёрная дыра» и «вихрь». Вверху показано исходное изображение, эффекты показаны при максимальной силе

полярные  $(\rho_r, \varphi_r)$ , причём начало координат устанавливается в центр изображения. Затем выполняется поворот на угол, зависящий от параметра обратной связи, после чего координаты переводятся назад в декартову систему. Полученные координаты  $(x_s, y_s)$  идентифицируют точку исходного изображения, цвет которой копируется в точку результирующего изображения. В случае, если точка  $(x_s, y_s)$  оказывается за границами исходного изображения, выбирается цвет точки, лежащей на границе.

## 2.8. Преобразования цветового пространства

В данной группе запрограммированы два эффекта: сведение к полутонному изображению и «поворот цвета».

Алгоритм первого преобразования сводится к вычислению среднего значения компонент цвета точки. Полученное значение затем присваивается компонентам точки результирующего изображения.

«Поворот цвета» работает следующим образом. Цвет точки переводится в систему цветовых координат HSV<sup>4</sup>. После этого значение тона изменяется пропорционально параметру обратной связи, и выполняется обратное преобразование в координаты RGB цвета результирующей точки. Тон в системе координат HSV можно интерпретировать как угол, и всё преобразование в целом тогда будет соответствовать повороту на угол, пропорциональный параметру обратной связи.

## 3. РЕЗУЛЬТАТЫ ПРЕДВАРИТЕЛЬНЫХ ИСПЫТАНИЙ ПРОГРАММЫ

### 3.1. Цели и методика испытаний

Предварительные испытания проводились для того, чтобы выбрать из всего множества запрограммированных преобразований те, которые лучше всего подходят для визуальной обратной связи (см. п. 2.1.).

Скорость обработки изображения измерялась автоматически. Эффекты, применение которых занимало более 50 мсек, отбраковывались. (Очевидно, время обработки изображения зависит от нескольких параметров, таких как размер изображения, вычислительная сложность эффекта и общее быстродействие ЭВМ; эффекты, единственным недостатком которых является

---

<sup>4</sup> HSV — Hue, Saturation, Value — тон, насыщенность и яркость, соответственно.

высокая вычислительная сложность, могут с успехом применяться на более мощных ЭВМ.)

Другие критерии, такие как уверенная идентификация изменений, оценивались субъективно группой из трёх человек<sup>5</sup>.

### 3.2. Результаты

Первый важный вывод, который можно сделать после предварительного исследования, — для большинства эффектов следует подбирать изображения со специальными особенностями. Так, препарирующие эффекты лучше выглядят на изображениях с плавными изменениями яркости, для гауссова размытия нужны, наоборот, чёткие контуры, а «вихрь» хорошо работает только на таких изображениях, изменение геометрии которых вызывает ощущение «неправильности» у испытуемого.

С уверенностью можно сказать, что в комплексе наилучшие показатели имеют препарирующие эффекты. Простота вычислительного алгоритма обуславливает высокую скорость обработки, а для того, чтобы эффект уверенно идентифицировался, подходят многие обыкновенные фотографические изображения (например, живой природы).

Эффекты, основанные на преобразовании цветового пространства изображения, малопригодны для организации обратной связи, поскольку плавные изменения цветовой гаммы всего изображения крайне трудно отследить визуально, особенно при медленном изменении.

### 3.3. Перспективы

В настоящее время планируется эксперимент с участием группы добровольцев, в котором будет выявлена эффективность различных эффектов в бета- и релаксационных тренингах. В случае, если на практике будет показано, что описанные методы визуальной обратной связи в сравнении с существующими средствами качественно улучшают результаты тренингов, будет поставлена задача разработки ПО для широкого применения в комплексе «Бослаб».

Кроме того, прорабатывается возможность использования аудиального канала обратной связи с испытуемым. Звуковое сопровождение тренинга в описываемом образце ПО весьма примитивно: приложение оповещает испытуемого звуком о том, что значение регистрируемого сигнала превысило

---

<sup>5</sup> Вполне достаточно для того, чтобы получить бинарную оценку «подходит — не подходит» каждого отдельного эффекта.

установленный порог. Таким образом, возможность проведения тренинга «на слух» присутствует, но в очень простой форме. Дальнейшее развитие приложения будет продолжено после успешного испытания первого опытного образца.

## Приложение 1

## ПРЕПАРИРУЮЩИЕ ПРЕОБРАЗОВАНИЯ

№ п.п.	Формула преобразования	Примечание
1	$c' = \begin{cases} 0, c < p \\ 255, c \geq p \end{cases}, p \in [0, 255]$	Пороговая обработка.
2	$c' = \begin{cases} 0, c \in (0, 128 - p) \cup (128 + p, 255) \\ 255, c \in [128 - p, 128 + p] \end{cases}, p \in [0, 127]$	Яркостные срезы.
3	$c' = \begin{cases} c, c \in (0, 128 - p) \cup (128 + p, 255) \\ 255, c \in [128 - p, 128 + p] \end{cases}, p \in [0, 127]$	
4	$c' = \begin{cases} c, c < p \\ 255, c \geq p \end{cases}, p \in [0, 255]$	
5	$c' = \begin{cases} 0, c \in (0, 128 - p) \\ (c - p) \cdot \frac{255}{255 - 2p}, c \in [128 - p, 128 + p], p \in [0, 127] \\ 255, c \in (128 + p, 255) \end{cases}$	Линейное контрастирование с насыщением.
6	$c' = \begin{cases} 255, c \in (0, 128 - p) \\ 255 - (c - p) \cdot \frac{255}{255 - 2p}, c \in [128 - p, 128 + p], p \in [0, 127] \\ 0, c \in (128 + p, 255) \end{cases}$	Инверсия линейного контрастирования с насыщением.
7	$c' = \begin{cases} 0, c \in (0, 128 - p) \cup (128 + p, 255) \\ (c - p) \cdot \frac{255}{255 - 2p}, c \in [128 - p, 128 + p], p \in [0, 127] \end{cases}$	Линейное контрастирование в диапазоне. В зависимости от
8	$c' = \begin{cases} 255, c \in (0, 128 - p) \cup (128 + p, 255) \\ (c - p) \cdot \frac{255}{255 - 2p}, c \in [128 - p, 128 + p], p \in [0, 127] \end{cases}$	разновидности преобразования, вне диапазона

№ п.п.	Формула преобразования	Примечание
9	$c' = \begin{cases} 127, c \in (0, 128 - p) \cup (128 + p, 255) \\ (c - p) \cdot \frac{255}{255 - 2p}, c \in [128 - p, 128 + p], p \in [0, 127] \end{cases}$	выполняется подавление до нуля (7), насыщение до белого (8) и выравнивание до серого (9).
11	$c' = \begin{cases} 0, c \in (0, 128 - p) \cup (128 + p, 255) \\ c, c \in [128 - p, 128 + p] \end{cases}, p \in [0, 127]$	
12	$c' = \begin{cases} 255, c \in (0, 128 - p) \cup (128 + p, 255) \\ c, c \in [128 - p, 128 + p] \end{cases}, p \in [0, 127]$	

### СПИСОК ЛИТЕРАТУРЫ

1. **Грузман И.С., Киричук В.С., Косых В.П. и др.** Цифровая обработка изображений в информационных системах. Учебное пособие. — Новосибирск: Изд-во НГТУ, 2000. — 168 с.
2. **Binary Technologies** <http://www.binary-technologies.com/>
3. **Carey Bunks.** Grogking the GIMP. — New Riders Publishing, 2000. — ISBN 0-7357-0924-6