

**Проективные морфологии изображений  
на базе моделей, описываемых структурирующими функционалами**

Предложены проективные морфологии на базе операторов сопоставления одномерных и двумерных функций, вычисляемых методом динамического программирования. В качестве обобщения ранее построенных частных проективных морфологий предложен обобщающий формализм построения проективных морфологий изображений на базе моделей, описываемых структурирующими функционалами. Исследованы и доказаны свойства предложенных морфологических операторов.

***Keywords: Mathematical Morphology; Image Matching; Dynamic Programming.***

**Введение**

В статьях [1,2] была предложена т.н. *проективная морфология*, представляющая собой алгебру структур с однородной связью, позволяющую формально описывать и сравнивать структурные модели и структурные объекты вне зависимости от конкретной прикладной области. Было также показано, что введенные таким образом однородные структуры всегда могут быть однозначно охарактеризованы массивами или векторами признаков, которые в семантическом плане можно рассматривать как содержательные структурные описания соответствующих объектов. При этом в пространстве морфологических разложений стандартным образом определяются удобные нормированные коэффициенты морфологической корреляции, позволяющие оценивать степень сходства образов с точностью до некоторых классов эквивалентности, которые, в свою очередь, соответствуют определенным семантическим таксонам предметной области («фигуры одной формы», «модели одного класса» и т.п.).

В статье [3] предложен еще один класс проективных морфологий и соответствующих им процедур анализа морфологических свидетельств, построенный на базе неоднородных структурных моделей, описываемых логическими предикатами. Такие проективные морфологии могут быть названы *морфологиями детектирования*, поскольку при описанном подходе к построению оператора проекции (морфологического фильтра) отфильтрованное изображение сохраняет неизменными все те элементы, которые проголосовали в пользу успешных гипотез об обнаружении (детектировании) объектов, заданных рассматриваемой структурной моделью. В то же время существует, очевидно, и иной класс неоднородных проективных морфологий, который условно можно назвать *морфологиями реконструкции* (или *морфологиями сегментации*), в

которых морфологическая проекция представляет собой реконструированный «идеальный образ» заданного класса.

В статье [4], продолжением и развитием которой является настоящая статья, описаны реконструктивные проективные морфологии на базе операторов фильтрации и сегментации изображений, вычисляемых методом динамического программирования (ДП-морфологии). В отличие от предложенных ранее проективных морфологических разложений, морфологические ДП-операторы могут быть названы «неразложимыми», поскольку проективные свойства таких операторов могут быть доказаны и без разложения. В то же время, между предложенной ДП-морфологией и построенной ранее проективной морфологией на базе однородных моделей имеется целый ряд существенных аналогий. Так, используемые в ДП-морфологиях структурирующие функционалы аналогичны структурирующим функциям морфологии Серра, а сглаживающий параметр играет роль масштаба (размера) структурирующих элементов.

В данной статье, во-первых, будут предложены проективные морфологии на базе операторов сопоставления (matching) одномерных и двумерных функций, вычисляемых методом динамического программирования. Во-вторых, в качестве обобщения всех ранее построенных в статьях [1-4] проективных морфологий будет предложен объемлющий формализм построения проективных морфологий изображений на базе моделей, описываемых структурирующими функционалами.

Необходимо отметить, что большинство описанных в данной статье процедур сравнения (отождествления) и анализа фрагментов изображений методом динамического программирования не являются сколько-нибудь новыми. Такие процедуры были неоднократно описаны в литературе по компьютерному и машинному зрению, начиная с 1970-х-80-х годов. Новизна предлагаемого подхода заключается в рассмотрении подобных процедур отождествления фрагментов в качестве базовых операторов (фильтров) специального класса проективных морфологий изображения.

Следует также отметить, что метод динамического программирования является лишь одним из известных методов отыскания оптимального решения в некотором заданном классе, и использование этого метода далеко не всегда обеспечивает наилучшее решение с точки зрения использования вычислительных ресурсов. Однако с точки зрения теоретического анализа подобных задач метод динамического программирования имеет следующие два основных достоинства. Во-первых, это неитеративный метод оптимизации, то есть время обработки одного изображения заданного размера всегда конечно и заранее известно, что позволяет в принципе оценить *вычислительную разрешимость* задачи. Во-вторых, данный метод обеспечивает нахождение не локального, а глобального экстремума рассматриваемого критерия качества, что, в свою очередь, позволяет говорить об *оптимальности*, а не псевдо- или квази-оптимальности полученных результатов, а также трактовать полученный результат фильтрации или сегментации изображений как *проекцию* исходного изображения на некоторый класс заданных структур. Последнее свойство как раз и является основанием математического формализма проективной

морфологии на базе операторов, реализуемых методом динамического программирования (*ДП-морфологии*).

### Сопоставление (отождествление фрагментов) одномерных функций методом динамического программирования

Рассмотрим задачу сопоставления (отождествления фрагментов) двух одномерных функций  $f(x)$  и  $g(x)$ . При этом предполагается, что

$$g(x) = f(x+v(x)) + \xi, \quad (1)$$

где  $\xi$  - случайная аддитивная помеха,  $v(x)$  – неизвестная идеальная функция относительного сдвига или *диспаратности*. Пусть функции  $g(x)$ ,  $f(x)$  и  $v(x)$  принимают значения на множестве  $l=0..N-1$ . На самом деле диспаратность, конечно, может принимать и отрицательные значения, однако на практике всегда можно отсчитывать их от самого левого возможного значения  $l_{\min}$ , которое после этого без потери общности может считаться нулем.

При этом в зависимости от типа задачи отождествления, *функция-решение диспаратности*  $L(x)$  считается принадлежащей некоторому классу функций  $V$ : гладких, кусочно-постоянных или кусочно-гладких функций.

Следуя методике, описанной в статье [4], рассмотрим элементы метода динамического программирования применительно к данной задаче сравнения/отождествления одномерных функций.

*Критерий соответствия*  $J$  в данном случае принимает вид:

$$J(f(x),g(x),L(x))=J(f(x),g(x+L(x))). \quad (2)$$

Например, здесь может использоваться критерий минимума суммы квадратов отклонений, обеспечивающий решение, оптимальное в случае, когда  $\xi$  представляет собой Гауссов белый шум:

$$J(f(x),g(x),L(x))=\sum_x(f(x) - g(x+L(x)))^2, \quad (3)$$

$$L(x): J(f(x),g(x),L(x))\rightarrow\min.$$

Вид *функционала качества решения*  $Q(L(x))$ , как и в случае решения задачи фильтрации/сегментации функции, зависит от класса искомых решений. Например, в задаче *стерео отождествления без «затенений»* (разрывов диспаратности) может использоваться функционал качества (4), штрафующий негладкость и разрывы функций-решений.

$$Q(L(x))=\sum_x(L'(x))^2 \text{ или } Q(L(x))=\sum_x(L''(x))^2, \quad (4)$$

$$Q(L(x))\rightarrow\min.$$

Собственно задача отыскания решения поэлементного соответствия функций представляет собой задачу поиска экстремума комбинированного функционала

$$\Phi(f(x),L(x))= J(f(x),g(x),L(x)) + \alpha Q(L(x)), \quad (5)$$

$$L(x): \Phi(f(x),g(x),L(x))\rightarrow\min,$$

где  $\alpha$  - весовой коэффициент, определяющий целевую значимость качества решения по отношению к степени поэлементного сходства сравниваемых функций.

Для того, чтобы задача (5) могла быть решена методом динамического программирования, функционал соответствия и функционал качества решения должны быть представлены в *рекурсивно-кумулятивной* форме:

$$J(f(x),g(x),L(x))=J(x_{\max}): J(x_i) = J(x_{i-1}) + \Delta J(x_i,L(x_i)); J(0)=\Delta J(0,L(0)). \quad (6)$$

$$Q(L(x))=Q(x_{\max}): Q(x_i) = Q(x_{i-1}) + \Delta Q(L(x_{i-1}),L(x_i)); Q(0)=0, \quad (7)$$

где  $\{x_i\}$ ,  $i=1..x_{\max}$  – множество дискретных значений аргумента рассматриваемой функции в ее области определения.

Для случая критериев (3),(4) это будет соответственно

$$\Delta J(x_i,L(x_i))=(f(x_i)-g(x_i+L(x_i)))^2; \quad (8)$$

$$\Delta Q(L(x_{i-1}),L(x_i))=(L(x_{i-1})-L(x_i))^2. \quad (9)$$

С учетом сделанных обозначений, решение задачи сравнения/отождествления одномерных функций методом динамического программирования оказывается полностью аналогичным решению задачи фильтрации/сегментации одномерных функций [4]. Оно основано на использовании двумерного *аккумулятора*  $A(x,l)$  размера  $(1+x_{\max}) \times N$  и достигается при помощи следующего Алгоритма 1.

### Алгоритм 1.

Прямой проход динамического программирования

Шаг 0. Инициализировать значения крайнего левого столбца аккумулятора:

$$A(0,l) = \Delta J(0,l), l=0..N-1. \quad (10)$$

Шаги  $i=1..x_{\max}$ . Определить значение каждого следующего столбца аккумулятора по формуле:

$$A(x_i,l) = \Delta J(x_i,l) + \min_t \{ A(x_{i-1},t) + \alpha \Delta Q(t,l) \}, t,l=0..N-1. \quad (11)$$

Обратный проход динамического программирования

Шаг 0. Инициализировать значение крайнего правого элемента решения:

$$L(x_{\max}) = \operatorname{argmin}_l \{ A(x_{\max},l) \}, l=0..N-1. \quad (12)$$

Шаги  $i=x_{\max}-1..0$ . Справа налево определить значение всех следующих элементов решения по формуле:

$$L(x_i) = \operatorname{argmin}_l \{ A(x_i,l) + \alpha \Delta Q(l,L(x_{i+1})) \}, l=0..N-1. \quad (13)$$

Конец алгоритма. ♦

В случае поиска функции-решения в классе кусочно-постоянных функций может использоваться критерий качества (14):

$$\Delta Q(L(x_{i-1}),L(x_i)) = \begin{cases} 0, & \text{если } L(x_{i-1})=L(x_i), \\ 1, & \text{если } L(x_{i-1}) \neq L(x_i). \end{cases} \quad (14)$$

Штрафной функционал качества решения  $Q(L(x))$  в этом случае будет иметь значение, равное числу переходов от одной области постоянного значения к другой. Если при этом выбрать весовой коэффициент  $\alpha$  достаточно большим, то любое изменение значения  $L(x)$  будет оплачиваться столь дорогим «штрафом», что оно окажется оправданным лишь при достаточно сильном

соответствующем уменьшении критерия соответствия  $J(f(x),g(x),L(x))$ . Такая модель «независимых сдвигов» фрагментов функций менее характерна для стереопар естественных сцен, однако именно эта модель наиболее естественна при сравнении пар последовательных кадров динамических видеопоследовательностей. Фрагменты, обладающие разными межкадровыми сдвигами друг относительно друга и в то же время одинаковыми межкадровыми сдвигами для всех точек внутри фрагмента, как раз и являются искомыми движущимися объектами в задаче анализа движения.

В случае поиска функции-решения в классе кусочно-гладких функций, также могут использоваться критерии качества типа (15).

$$\Delta Q(L(x_{i-1}),L(x_i)) = \begin{cases} (L(x_{i-1}) - L(x_i))^2, & \text{если } |L(x_{i-1}) - L(x_i)| \leq \Delta Q_{\max}, \\ (\Delta Q_{\max})^2, & \text{если } |L(x_{i-1}) - L(x_i)| > \Delta Q_{\max}. \end{cases} \quad (15)$$

При этом штрафной функционал качества решения  $Q(L(x))$  на относительно гладких областях штрафует негладкость функции решения, и в то же время допускает существование в решении разрывов («скачков»), если их величина больше некоторой заданной. Такая модель характерна либо для задач стереотождествления при наличии перекрытия и загораживания объектов, либо для задач анализа движения, в которых допускается приближение/удаление объектов.

В заключение данного раздела необходимо отметить принципиальное отличие типовой задачи сравнения функций от задачи анализа формы одной функции, рассмотренной ранее в статье [4]: при отождествлении функций со сдвигами (разрывами) некоторые точки обеих функций окажутся вообще неотожествленными. На практике соответствующий алгоритм должен учитывать специальные штрафы от этих точек определенным специальным образом. Кроме того, несколько меняется и структура алгоритма ДП-отождествления функций. Однако подробное рассмотрение данного вопроса выходит за рамки темы данной статьи, как и целый ряд других практических вопросов, связанных с построением робастных и эффективных алгоритмов стереотождествления и анализа движения.

Рассмотрим теперь обобщение описанной методики отождествления функций методом динамического программирования на двумерный случай.

### **Сопоставление (отождествление фрагментов) двумерных функций методом динамического программирования**

В задаче сравнения/отождествления фрагментов двумерных функций вместо модели (1) рассматривается аналогичная двумерная модель.

$$g(x,y) = f(x+v_x(x,y), y+v_y(x,y)) + \xi, \quad (16)$$

При этом в отличие от рассмотренной ранее задачи сегментации полутоновой функции [4], функция-решение по диспаратности становится векторной функцией двумерного аргумента:

$$\mathbf{L}(x,y) = \{L_x(x,y), L_y(x,y)\} \quad (17)$$

В остальном задача сравнения/отождествления фрагментов двумерных функций имеет аналогичный вид:

$$\Phi(f(x,y),g(x,y),\mathbf{L}(x,y))=J(f(x,y),g(x,y),\mathbf{L}(x,y))+\alpha Q(\mathbf{L}(x,y)), \quad (18)$$

$$\mathbf{L}(x,y): \Phi(f(x,y),\mathbf{L}(x,y)) \rightarrow \min,$$

где функционалы *соответствия*  $J$  и *качества*  $Q$  описываются аналогичными выражениями.

Например:

$$J(f(x,y),g(x,y),\mathbf{L}(x,y))= \sum_x \sum_y (f(x_i,y_i)-g(x_i+L_x(x,y),y_i+L_y(x,y)))^2, \quad (19)$$

$$Q(\mathbf{L}(x,y))= \sum_x \sum_y [\Delta Q(\mathbf{L}(x_{i-1},y_i),\mathbf{L}(x_i,y_i))+\Delta Q(\mathbf{L}(x_i,y_{i-1}),\mathbf{L}(x_i,y_i))], \quad (20)$$

где

$$\Delta Q(\mathbf{t},\mathbf{l})=(t_x-l_x)^2+(t_y-l_y)^2. \quad (21a)$$

в задаче стереотождествления без разрывов (для гладкой 3D поверхности) и

$$\Delta Q(\mathbf{t},\mathbf{l}) = \begin{cases} 0, & \text{если } \mathbf{t}=\mathbf{l}, \\ 1, & \text{если } \mathbf{t} \neq \mathbf{l}. \end{cases} \quad (21b)$$

в задаче сегментации областей межкадрового движения.

Используемый аккумулятор является четырехмерным  $A(x,y,k,l)$  размерности  $x_{\max} \times y_{\max} \times \Delta x_{\max} \times \Delta y_{\max}$ , где  $\Delta x_{\max} \times \Delta y_{\max}$  – максимально возможные значения межкадрового сдвига фрагмента изображения (размеры области поиска соответствия).

Соответствующий алгоритм динамического программирования принимает следующий вид.

## Алгоритм 2

Прямой проход динамического программирования

Шаг 0. Инициализировать значения аккумулятора для  $(x=0)$  и  $(y=0)$ :

$$A(i,0,k,l)= \Delta J(i,0,k,l), \quad k=0..\Delta x_{\max}, \quad l=0..\Delta y_{\max}, \quad i=0..x_{\max} \quad (22)$$

$$A(0,j,k,l)= \Delta J(0,j,k,l), \quad k=0..\Delta x_{\max}, \quad l=0..\Delta y_{\max}, \quad j=0..y_{\max} \quad (23)$$

Шаги  $j=1..y_{\max}; \quad i=1..x_{\max}$ . Определить значение каждого следующего  $k,l$ -столбца аккумулятора по формуле:

$$A(x_i,y_i,k,l) = \min \{ \min_{t,r} \{ A(x_{i-1},y_i,r,t) + \alpha \Delta Q(r,t,k,l) \}, \\ \min_{s,p} \{ A(x_i,y_{i-1},p,s) + \alpha \Delta Q(p,s,k,l) \} \} + \Delta J(x_i,y_i,k,l), \\ p,r,k=0..\Delta x_{\max}; \quad s,t,l=0..\Delta y_{\max}. \quad (24)$$

Обратный проход динамического программирования

Шаг 0. Инициализировать значение «правого нижнего» элемента решения:

$$L(x_{\max},y_{\max}) = \operatorname{argmin}_l \{ A(x_{\max},y_{\max},k,l) \}, \quad k=0..\Delta x_{\max}, \quad l=0..\Delta y_{\max}. \quad (25)$$

Затем получить одномерные решения  $\mathbf{L}(x_{\max},y)$  и  $\mathbf{L}(x,y_{\max})$  соответственно для правого столбца и нижней строки как описано в *Алгоритме 1*.

Шаги  $j=y_{\max}-1..0, \quad i=x_{\max}-1..0$ . Справа налево снизу вверх определить значение всех следующих элементов решения по формуле:

$$L(x_i,y_i) = \operatorname{argmin}_l \{ \min \{ A(x_i,y_i,k,l) + \alpha \Delta Q(k,l,\mathbf{L}(x_{i+1},y_i)) \},$$

$$A(x_i, y_i, k, l) + \alpha \Delta Q(k, l, \mathbf{L}(x_i, y_{i+1})) \}, k=0.. \Delta x_{\max}, l=0.. \Delta y_{\max}. \quad (26)$$

### Конец алгоритма. ♦

Заметим, что в данной статье с целью соблюдения простоты и общности изложения как для задач анализа одного изображения, так и для задач сравнения двух изображений мы рассматриваем только критерий соответствия изображений яркостного типа (3). Между тем, техника сравнения (matching) изображений позволяет использовать, помимо этого также ряд других критериев соответствия, каждый из которых имеет свои особенности и преимущества. Специальные критерии соответствия (и соответственно специальные процедуры ДП-сравнения и отождествления фрагментов изображений) могут быть построены на базе контурных, градиентных, признаковых, текстурных, топологических и ряда других мер сходства. Это, в свою очередь, влияет и на саму структуру алгоритма отождествления. Например, в случае отождествления признаковых статистик на множествах точек интереса ДП-аккумулятор может быть представлен не в виде регулярного массива, а в виде списка возможных соответствий особых точек, которые определяются, исходя из геометрических ограничений на область поиска и требуемую степень сходства между векторами признаков. В вычислительном смысле такой алгоритм может быть весьма эффективен, так как число особых точек с высокой информативностью обычно оказывается на несколько порядков меньше размера исходного множества точек. Однако рассмотрение этих специальных модификаций процедуры отождествления также выходит за рамки тематики данной статьи.

### **Морфологическое проектирование функции на функцию на базе сравнения/отождествления фрагментов двумерных функций методом динамического программирования**

Дадим теперь соответствующие морфологические определения в соответствии с методикой, использованной в статье [4].

**Определение 1.** Пусть определена задача сравнения/отождествления функции  $f(x, y)$  с функцией  $g(x, y)$  следующего вида:

$$\Phi(f(x, y), g(x, y), \mathbf{L}(x, y)) = J(f(x, y), g(x, y), \mathbf{L}(x, y)) + \alpha Q(\mathbf{L}(x, y)), \quad (27)$$

$$\mathbf{L}(x, y) = \{L_x(x, y), L_y(x, y)\},$$

$$J(f(x, y), g(x, y), \mathbf{L}(x, y)) = J(f(x, y), g(x+L_x(x, y), y+L_y(x, y))),$$

$$\mathbf{L}(x, y): \Phi(f(x, y), g(x, y), \mathbf{L}(x, y)) \rightarrow \min(\mathbf{L}(x, y)),$$

где  $\mathbf{L}(x, y)$  – функция диспаратности;  $g(x+L_x(x, y), y+L_y(x, y))$  – функция-решение;  $\Phi(f(x, y), g(x, y), \mathbf{L}(x, y))$  – комплексный критерий оптимизации;  $J(f(x, y), g(x, y), \mathbf{L}(x, y))$  – критерий соответствия функции-решения исходной функции;  $Q(\mathbf{L}(x, y))$  – критерий качества функции-решения;  $\alpha$  – сглаживающий параметр.

Тогда оператор, отображающий функцию  $f(x, y)$  на функцию-решение

$$\text{Pr}(f(x, y), g(x, y), J, Q, \alpha) = g(x+L_x(x, y), y+L_y(x, y)), \quad (28)$$

называется *оператором морфологической проекции на функцию  $g(x, y)$* , если

$$\text{Pr}(\text{Pr}(f(x, y), g(x, y)), g(x, y)) = \text{Pr}(f(x, y), g(x, y)). \quad (29)$$

Соответствующая проективная морфология называется *проективной морфологией на базе функционала соответствия J, структурирующего функционала Q и сглаживающего параметра  $\alpha$* . ♦

**Определение 2.** ДП-разрешимой называется такая задача сравнения/отождествления функции  $f(x,y)$  с функцией  $g(x,y)$  вида (131), для которой функционал  $\Phi(f(x,y),g(x,y),L(x,y))$  может быть представлен в следующей *рекурсивно-аддитивной* форме:

$$\Phi(f(x),g(x),L(x))=\Phi(x_{\max}): \Phi(x_i) = \Phi(x_{i-1}) + \Delta\Phi(x_i,L(x_i)); \Phi(0)=\Delta\Phi(0,L(0)). \quad (30)$$

Соответствующий оператор проекции будем называть *ДП-вычислимым*.

Примечательно, что в данном случае, помимо структурирующих функционалов, введенных в статье [14], параметром морфологической проекции вновь, как ранее у Серра и Пытьева, оказывается еще и *структурирующая функция* (функция, на которую осуществляется проектирование). Только диапазон возможных трансформаций структурирующей функции оказывается здесь существенно шире.

С учетом этого, целесообразно определить и соответствующий *морфологический коэффициент корреляции двух функций*:

$$K_M(f(x,y),g(x,y))=\min(\|\Pr(f(x,y),g(x,y))\|,\|f(x,y)\|) / \max(\|\Pr(f(x,y),g(x,y))\|,\|f(x,y)\|) \quad (31)$$

с очевидными стандартными свойствами:

- 1)  $0 \leq K_M(f(x,y),g(x,y)) \leq 1$ ;
- 2)  $K_M(f(x,y),f(x,y))=1$ .

Данный коэффициент морфологической корреляции позволяет оценивать степень сходства исследуемых функций (образов) с точностью до заданного класса эквивалентности, определяемого свойствами выбранных параметров используемой процедуры морфологического проектирования – функционала соответствия J, структурирующего функционала Q и сглаживающего параметра  $\alpha$ .

### Проективные морфологии на базе структурирующих функций и функционалов

Полученный в предыдущем параграфе результат позволяет наметить пути объединения в общем математическом формализме всех ранее введенных проективных морфологий – как на базе структурирующих функций, так и на базе структурирующих функционалов.

Для одномерного случая можно дать следующие определения.

**Определение 3.** Пусть определена задача *оптимальной аппроксимации функции  $f(x)$*  следующего вида:

$$L(x): \Phi(f(x),g_1(x),\dots,g_n(x),L(x)) \rightarrow \min(L(x)), \quad (32)$$

где  $f(x)$  – аппроксимируемая функция;  $\{g_1(x),\dots,g_n(x)\}$  – базис аппроксимации порядка  $n$ ;  $L(x)$  – функция-решение;  $\Phi(f(x),g_1(x),\dots,g_n(x),L(x))$  – комплексный критерий оптимизации;  $G(f)$  – область допустимых решений для данной аппроксимируемой функции.

Тогда оператор, отображающий исходную функцию на решение задачи (32)

$$\Pr(f(x),g_1(x),\dots,g_n(x))=L(x), \quad (33)$$

называется *оператором морфологической проекции*, если



$$\Pr(\Pr(f(x),g_1(x),\dots,g_n(x)),g_1(x),\dots,g_n(x))=\Pr(f(x),g_1(x),\dots,g_n(x))). \quad (34)$$

Соответствующая проективная морфология называется *проективной морфологией на базе структурирующего функционала  $\Phi$  и набора структурирующих функций  $\{g_1(x),\dots,g_n(x)\}$* . ♦

Также в самом общем виде можно сформулировать следующие *достаточные условия* построения оператора проекции на базе функционала и набора функций.

**Определение 4.** Областью допустимых решений задачи (32)  $G(f)$  для некоторой заданной функции  $f(x)$  является множество реализаций  $L(x)$ , таких, что функционал  $\Phi(f(x),g_1(x),\dots,g_n(x),L(x))$  принимает конечные значения:

$$G(f): \{|\Phi(f(x),g_1(x),\dots,g_n(x),L(x))|<+\infty\} \quad (35)$$

**Теорема (о достаточных условиях построения проектора).** Проективная морфология может быть построена на базе решения задачи (135), если ее элементы удовлетворяют следующим условиям:

$$(a) \forall f(x): f(x) \in G(f) \quad (36)$$

$$(b) \forall f(x), L(x): L(x) \in G(f) \Rightarrow G(L) \subseteq G(f); \quad (37)$$

$$(c) \forall h(x) \in G(f): \Phi(f(x),g_1(x),\dots,g_n(x),f(x)) \leq \Phi(f(x),g_1(x),\dots,g_n(x),h(x)). \quad (38).$$

Доказательство. Пусть

$$L(x) = \Pr(f(x),g_1(x),\dots,g_n(x)),$$

$$L'(x) = \Pr(L(x),g_1(x),\dots,g_n(x)).$$

Поскольку согласно (139,140)

$$L'(x), L(x) \in G(L) \subseteq G(f),$$

значит с учетом (141)

$$\Phi(L(x),g_1(x),\dots,g_n(x),L(x)) \leq \Phi(L(x),g_1(x),\dots,g_n(x),L'(x)).$$

и при этом

$$L'(x): \Phi(f,g_1(x),\dots,g_n(x),h) \rightarrow \min(h \in G(L)).$$

Отсюда следует, что

$$L'(x) = L(x).$$

Что и требовалось доказать. ♦

Условие (38) можно назвать *условием выпуклости функционала  $\Phi$  относительно функции  $f$* .

Однако далеко не все возможные морфологии на базе теоретически мыслимых операторов проекции (33) представляют практический интерес, так как во многих случаях не удастся оценить возможность вычислительно эффективного решения соответствующей задачи (32). Чтобы говорить о построении некоторой специфической морфологии данного типа, как минимум, необходимо доказать существование такого решения, а также существование процедуры его отыскания за конечное время.

Рассмотрим возможность решения задачи (32) в рамках метода динамического программирования (ДП).

**Определение 5.** *ДП-разрешимой* называется такая задача морфологической аппроксимации функции  $f(x)$  вида (32), для которой функционал  $\Phi$  может быть представлен в следующей *рекурсивно-аддитивной* форме:

$$\Phi(f(x), g_1(x), \dots, g_n(x), L(x)) = \Phi(x_{\max}): \Phi(x_i) = \Phi(x_{i-1}) + \Delta\Phi_i(f(x), g_1(x), \dots, g_n(x), L(x)). \quad (39)$$

Соответствующий оператор проекции будем называть *ДП-вычислимым*.

Связь введенных в данном разделе общих определений с введенными ранее частными определениями становится более явной, если функционал  $\Phi$  может быть представлен в следующей форме:

$$\Phi(f(x), g_1(x), \dots, g_n(x), L(x)) = \beta J(f(x), L(x)) + \gamma W(g_1(x), \dots, g_n(x), L(x)) + \alpha Q(L(x)), \quad (40)$$

где

$J(f(x), L(x))$  – функционал соответствия функции-решения исходной функции;

$W(g_1(x), \dots, g_n(x), L(x))$  – функционал связи функции-решения с набором структурирующих функций;

$Q(L(x))$  – функционал качества, определяющий желаемый вид функции-решения;

$\alpha, \beta, \gamma$  – весовые коэффициенты соответствующих функционалов.

Обнуляя те или иные весовые коэффициенты, мы получим все частные случаи проективных морфологий, рассмотренных ранее:

- Случаю ( $\alpha > 0, \beta = 1, \gamma = 0$ ), соответствует *проективная морфология на базе функционала  $\Phi(J, Q, \alpha)$ .*
- Случаю ( $\alpha = 0, \beta > 0, \gamma > 0$ ) соответствуют:
  - во-первых, все *проективные морфологии на основе однородных морфологических разложений* по базису  $\{g_1(x), \dots, g_n(x)\}$  вида
 
$$L(x) = \lambda_1 g_1(x) \vee \dots \vee \lambda_n g_n(x)$$
  - во-вторых, *проективные морфологии на базе логических предикатов* вида
 
$$M(f(x), g_1(x), \dots, g_n(x), L(x)) = M_1(f(x), L(x)) \wedge M_2(g_1(x), \dots, g_n(x), L(x)) \wedge M_3 Q(L(x))$$
- Случаю ( $\alpha > 0, \beta > 0, \gamma > 0$ ) соответствует (при  $n=1$ ), *морфология на базе сравнения/отождествления фрагментов*, описанная в предыдущем параграфе.

С точки зрения мощности описания процедур морфологического анализа, данный формализм на базе структурирующих функционалов, по-видимому, является наиболее общим из всех ранее рассмотренных проективных морфологических формализмов, поскольку любое условие однородной разложимости по базису функций может быть рассмотрено как предикат:

$$M(f(x), g_1(x), \dots, g_n(x), L(x)) = \{\exists \lambda_1 \dots \lambda_n: L(x) = \lambda_1 g_1(x) \vee \dots \vee \lambda_n g_n(x)\},$$

а любой предикат, в свою очередь может быть описан штрафным функционалом:

$$\Phi(f(x), g_1(x), \dots, g_n(x), L(x)) = \{0: M(f(x), g_1(x), \dots, g_n(x), L(x)) = \text{true} \\ +\infty: M(f(x), g_1(x), \dots, g_n(x), L(x)) = \text{false}\}.$$

Между тем, обратное не справедливо, так как не всякий функционал может быть заменен предикатом, и не всякий предикат – алгебраической комбинацией функций.

### Проективные морфологии ансамблей функций на базе функционалов

Наконец, для того, чтобы сделать наш обзор методов проективного морфологического анализа данных при помощи метода динамического программирования наиболее полным, рассмотрим также решение задачи фильтрации/сегментации *ансамбля* (набора) регистраций зашумленной функции. Практическим примером, который мы при этом все время будем иметь в виду, является фильтрация/сегментация набора параллельных прорезок (одномерных профилей яркости) двумерного изображения одномерного штрихового кода, подвергнутого воздействию различного рода яркостных (но не геометрических) искажений типа затираний, загрязнений, бликов и т.п (рис.1).

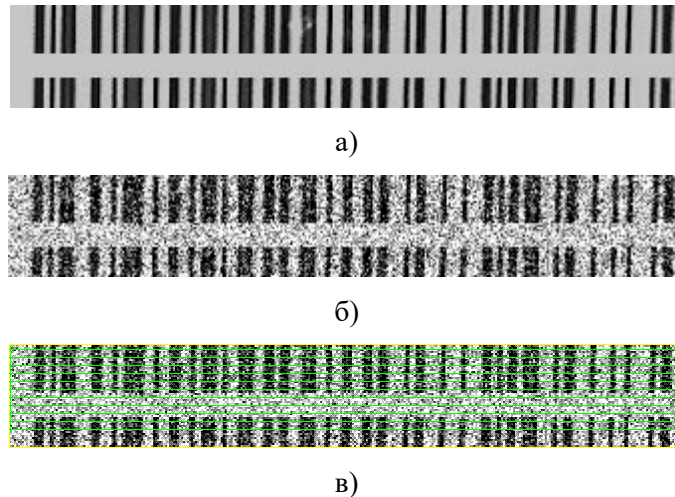


Рис. 1. а) Исходное изображение «разрезанного» штрихового кода; б) зашумленное изображение; в) набор горизонтальных прорезок искаженного и зашумленного изображения, порождающих ансамбль одномерных функций, подлежащих фильтрации/сегментации.

Пусть наблюдаемая дискретная функция  $f(x)$  определена на отрезке  $x \in [0..x_{\max}]$ , принимает значения на множестве  $l=0..N-1$ , представляет собой комбинацию неизвестной идеальной исходной функции  $v(x)$ , принадлежащей некоторому классу функций  $V$ , и случайного шума  $\xi$ . Пусть теперь регистрируется не единичная функция, а некоторый набор (ансамбль) из  $M$  различных реализаций данной функции:

$$\{f_m(x)\}: f_m(x) = v(x) + \xi_m, m=1..M. \quad (41)$$

Требуется найти такую *функцию-решение*  $L(x) \in V$ , которая наилучшим образом описывает наблюдаемые данные в смысле некоторого *критерия соответствия*  $J$ :

$$L(x): J(\{f_m(x)\}, L(x)) \rightarrow \text{extr}. \quad (42)$$

Естественно использовать, например, критерий минимума суммы квадратов отклонений, обеспечивающий решение, оптимальное в случае, когда  $\xi$  представляет собой Гауссов белый шум:

$$J(\{f_m(x)\}, L(x)) = \sum_x \sum_m (f_m(x) - L(x))^2, \quad (43)$$

$$L(x): J(\{f_m(x)\}, L(x)) \rightarrow \min.$$

При этом класс функций, в котором ищется решение, также может быть задан через некоторый дополнительный *функционал качества решения*:

$$Q(L(x)) \rightarrow \text{extr.} \quad (44)$$

Например, можно использовать функционалы качества, штрафующие негладкость функций-решений:

$$Q(L(x)) = \sum_x (L'(x))^2 \text{ или } Q(L(x)) = \sum_x (L''(x))^2, \quad (44)$$

$$Q(L(x)) \rightarrow \text{min.}$$

Таким образом, в простейшем случае задача фильтрации/сегментации ансамбля регистраций зашумленной функции (1) представляет собой задачу поиска экстремума комбинированного функционала

$$\Phi(\{f_m(x)\}, L(x)) = J(\{f_m(x)\}, L(x)) + \alpha Q(L(x)), \quad (46)$$

$$L(x): \Phi(\{f_m(x)\}, L(x)) \rightarrow \text{extr.},$$

где  $\alpha$  - весовой коэффициент, определяющий целевую значимость качества решения по отношению к степени соответствия получаемого решения наблюдаемым данным.

Для того, чтобы задача (6) могла быть решена методом динамического программирования, как функционал соответствия, так и функционал качества решения должны быть представимы в следующей *рекурсивно-кумулятивной* форме:

$$J(\{f_m(x)\}, L(x)) = J(x_{\max}): J(x_i) = J(x_{i-1}) + \Delta J(x_i, L(x_i)); J(0) = \Delta J(0, L(0)). \quad (47)$$

$$Q(L(x)) = Q(x_{\max}): Q(x_i) = Q(x_{i-1}) + \Delta Q(L(x_{i-1}), L(x_i)); Q(0) = 0, \quad (48)$$

где  $\{x_i\}$ ,  $i=1..x_{\max}$  - множество дискретных значений аргумента рассматриваемой функции в ее области определения.

Для приведенных выше примеров это будет соответственно

$$\Delta J(x_i, L(x_i)) = \sum_m (f_m(x_i) - L(x_i))^2; \quad (49)$$

$$\Delta Q(L(x_{i-1}), L(x_i)) = (L(x_{i-1}) - L(x_i))^2. \quad (50)$$

Простейшая реализация решения данной задачи методом динамического программирования (ДП) основана на использовании двумерного *аккумулятора*  $A(x, l)$  размера  $(1+x_{\max}) \times N$ , где  $N$  - количество элементов дискретизации области значений рассматриваемой функции

Алгоритм динамического программирования для этой задачи будет полностью аналогичен описанному ранее для случая фильтрации одной функции с учетом изменений в формуле (49).

В случае реконструкции штрихового кода необходимо решить задачу *сегментации* наблюдаемой функции  $f(x)$  на *области постоянного значения*. Это означает, что функция-решение  $L(x)$  должна принадлежать классу  $V$  кусочно-постоянных функций, причем количество и протяженность отрезков (областей) постоянного значения могут быть произвольными. Алгоритм динамического программирования позволяет решить задачу *ДП-сегментации*, если изменить критерий качества решения (50) на следующий:

$$\Delta Q(L(x_{i-1}), L(x_i)) = \begin{cases} 0, & \text{если } L(x_{i-1}) = L(x_i), \\ 1, & \text{если } L(x_{i-1}) \neq L(x_i). \end{cases} \quad (51)$$

Иными словами, штрафной функционал качества решения  $Q(L(x))$  в этом случае будет иметь значение, равное числу переходов от одной области постоянного значения к другой. Если при этом выбрать весовой коэффициент  $\alpha$  достаточно большим, то любое изменение значения  $L(x)$  будет оплачиваться столь дорогим «штрафом», что оно окажется оправданным лишь при достаточно сильном соответствующем уменьшении критерия соответствия  $J(\{f_m(x)\}, L(x))$ .

Результаты ДП-сегментации искаженного и зашумленного изображения штрихового кода (рис.1), рассматриваемого как ансамбль одномерных прорезок, представлены на рис.2.

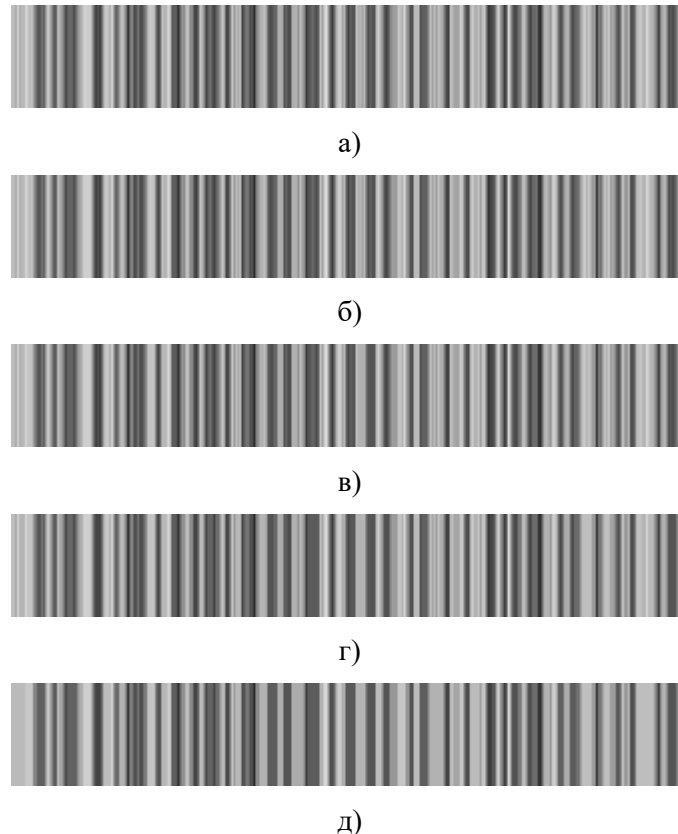


Рис. 2. Результаты ДП-сегментации ансамбля горизонтальных прорезок искаженного и зашумленного изображения штрихового кода: а)  $\alpha=100$ ; б)  $\alpha=500$ ; в)  $\alpha=1000$ ; г)  $\alpha=2000$ ; д)  $\alpha=5000$ .

Данный подход также позволяет решать задачу *суперразрешения*. Для того чтобы найти решение по ансамблю функций с субпиксельной точностью (с точностью до  $1/k$ ) необходимо:

- 1) Оцифровать (интерполировать) входной ансамбль функций с разрешением в  $k$  раз больше (перейти к массивам размера  $x_{\max} \times k$ ).
- 2) Осуществить взаимную привязку массивов увеличенного разрешения по сдвигу с пиксельной точностью (что соответствует точности привязки до  $1/k$  в исходном разрешении).
- 3) Осуществить фильтрацию/сегментацию ансамбля реализаций повышенного разрешения как описано выше.

Таким образом, задача фильтрации/сегментации ансамбля функций является естественным и практически применимым обобщением рассмотренной ранее задачи фильтрации/сегментации

одной функции. Соответственно введенные выше обобщенные определения проективной морфологии могут быть модифицированы с учетом возможности работы по ансамблю входных регистраций.

**Определение 3'.** Пусть определена задача *оптимальной аппроксимации ансамбля функций*  $\mathbf{f}(x)$  следующего вида:

$$\mathbf{L}(x): \Phi(\mathbf{f}(x), \mathbf{g}(x), \mathbf{L}(x)) \rightarrow \min(\mathbf{L}(x)), \quad (52)$$

где  $\mathbf{f}(x) = \{f_1(x), \dots, f_m(x)\}$  – аппроксимируемый ансамбль функций порядка  $m > 0$ ;

$\mathbf{g}(x) = \{g_1(x), \dots, g_n(x)\}$  – базис аппроксимации порядка  $n > 0$ ;

$\mathbf{L}(x) = \{L_1(x), \dots, L_m(x): L_1(x) = \dots = L_m(x) = L(x)\}$  – ансамбль-решение;  $L(x)$  – функция-решение;

$\Phi(\mathbf{f}(x), \mathbf{g}(x), \mathbf{L}(x))$  – комплексный критерий оптимизации;

$\mathbf{G}(\mathbf{f})$  – область допустимых решений для данного ансамбля аппроксимируемых функций.

Тогда оператор, отображающий исходный ансамбль функций на решение задачи (52)

$$\text{Pr}(\mathbf{f}(x), \mathbf{g}(x)) = \mathbf{L}(x), \quad (53)$$

называется *оператором морфологической проекции ансамблей функций*, если

$$\text{Pr}(\text{Pr}(\mathbf{f}(x), \mathbf{g}(x)), \mathbf{g}(x)) = \text{Pr}(\mathbf{f}(x), \mathbf{g}(x)). \quad (54)$$

Соответствующая проективная морфология называется *проективной морфологией ансамблей функций  $\mathbf{f}(x)$  на базе структурирующего функционала  $\Phi$  и набора структурирующих функций  $\mathbf{g}(x)$* . ♦

Аналогичным образом можно модифицировать и остальные введенные ранее определения.

Функционал  $\Phi$  также может быть представлен в аналогичной форме:

$$\Phi(\mathbf{f}(x), \mathbf{g}(x), \mathbf{L}(x)) = \beta J(\mathbf{f}(x), \mathbf{L}(x)) + \gamma W(\mathbf{g}(x), \mathbf{L}(x)) + \alpha Q(\mathbf{L}(x)), \quad (55)$$

где

$J(\mathbf{f}(x), \mathbf{L}(x))$  – функционал соответствия ансамбля-решения исходному ансамблю;

$W(\mathbf{g}(x), \mathbf{L}(x))$  – функционал связи ансамбля-решения с набором структурирующих функций;

$Q(\mathbf{L}(x))$  – функционал качества, определяющий желаемый вид ансамбля-решения;

$\alpha, \beta, \gamma$  – весовые коэффициенты соответствующих функционалов.

Обнуляя те или иные весовые коэффициенты, мы получим все необходимые частные случаи проективных морфологий уже для ансамблей функций.

### Заключение

Данная статья завершает цикл статей, посвященных разработке единого математического формализма для построения процедур структурного анализа цифровых изображений, получившего название «проективная морфология изображений». В этих статьях на основе введенного формализма проективной морфологии был предложен единый унифицированный подход к разработке методов и процедур морфологической фильтрации и морфологического сжатия изображений, морфологического сравнения изображений, определены меры структурного сходства между изображениями, описано решение задач взаимной структурной привязки

изображений и задач эффективного в вычислительном смысле обнаружения на изображении объектов заданной структуры.

В данной статье на основе всех ранее построенных в статьях [1-4] частных проективных морфологий предложен обобщающий формализм построения проективных морфологий изображений на базе моделей, описываемых структурирующими функционалами. В этой наиболее общей форме проективная морфология позволяет описывать с единых позиций большинство используемых в настоящее время регулярных подходов к структурному анализу изображений, таких как частотные методы фильтрации (на базе преобразования Фурье, БПФ, ДКП и др.), пространственно-частотные методы фильтрации (кратномасштабный вейвлет-анализ), процедуры морфологического анализа Ю.П. Пытьева, операторы математической морфологии Серра, методы выделения объектов и геометрических примитивов путем голосования, восходящие к преобразованию Хафа, методы структурно-лингвистического анализа изображений, методы стереотождествления, методы анализа движения и ряд других. Показано также, что данный подход может быть непосредственно обобщен на случай анализа ансамбля функций, в том числе – при решении задачи «суперразрешения».

#### **Литература:**

1. ВИЗИЛЬТЕР Ю.В., ЖЕЛТОВ С.Ю. Сравнение и локализация фрагментов изображений с использованием проективных морфологий. Вестник компьютерных и информационных технологий, N2, 2008, с.14-22.
2. ВИЗИЛЬТЕР Ю.В., ЖЕЛТОВ С.Ю. Обнаружение и идентификация объектов на изображениях, с использованием проективных морфологий. Вестник компьютерных и информационных технологий, N3, 2008.
3. ВИЗИЛЬТЕР Ю.В., ЖЕЛТОВ С.Ю. Морфологический анализ изображений на основе неоднородных структурных моделей, описываемых логическими предикатами. Известия РАН. Теория и системы управления, N2, 2009. *(в печати)*
4. ВИЗИЛЬТЕР Ю.В., ЖЕЛТОВ С.Ю. Проективные морфологии на базе операторов фильтрации и сегментации изображений, вычисляемых методом динамического программирования. Вестник компьютерных и информационных технологий, N2, 2009. *(в печати)*