

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЗАДАЧАХ ОБРАБОТКИ ДАННЫХ В СИСТЕМАХ РАДИОВИДЕНИЯ

Ю.А. Пирогов, А.Н. Боголюбов, В.В. Гладун, Р.А. Гуцин,  
И.Е. Могилевский, Н.Е. Шапкина

Московский государственный университет им. М.В. Ломоносова  
Центр магнитной томографии и спектроскопии

Получена 25 сентября 2008 г.

*Рассматриваются различные математические методы обработки данных в системах радиовидения с целью улучшения разрешения полученных изображений. В результате анализа ряда алгоритмов для численного эксперимента выбирается алгоритм восстановления пространства изображений (ISRA). Приводятся результаты расчетов для тестовых задач сверхразрешения в системах радиовидения.*

**Ключевые слова:** радиовидение, сверхразрешение, обработка изображений.

### Введение

В данной работе рассматриваются методы обработки данных, получаемых в системах радиовидения. Подобные системы в настоящее время находят применение во все большем количестве прикладных задач, постепенно вытесняя системы ИК-диапазона. Их основным преимуществом таких систем по сравнению с ИК-системами является независимость от метеорологических условий, обеспечивающая возможность получения сравнительно качественных изображений даже в тех случаях, когда получение изображения ИК-методами невозможно. Подобные системы применяются в управлении авиационным трафиком, экологическом мониторинге, пассивной локации, детектирования скрытого под одеждой оружия в публичных местах и т.п.

Однако в связи с использованием большей длины волны такие системы проигрывают системам ИК и видимого диапазона по таким параметрам, как угловое разрешение и скорость получения информации. Угловое разрешение для радиовидения ограничено рэлеевским пределом  $R_\varphi = \frac{\lambda}{D}$  [1], где  $\lambda$  - рабочая длина волны,  $D$  - входная апертура. Повысить угловое разрешение можно двумя путями: варьируя параметры  $\lambda$  и  $D$  или применяя методы сверхразрешения, основанные на математической обработке измеренных радиометрических данных. Цель такой обработки — компенсировать искажения изображения, вызванные конечным угловым разрешением системы. В то время как уменьшение рабочей длины волны и увеличение апертуры является либо дорогостоящим, либо вообще невозможным по техническим причинам, метод сверхразрешения может быть применен практически в любой системе радиовидения. Поскольку наибольший практический интерес представляют системы реального времени, важным требованием, предъявляемым к алгоритмам сверхразрешения, является их быстрдействие.

Скорость получения информации в системах радиовидения имеет фундаментальное ограничение. Чувствительность системы пропорциональна квадратному корню из постоянной времени интегрирования. Так, например, для двукратного повышения чувствительности необходимо в четыре раза увеличить время измерения. Характерным значением постоянной времени интегрирования является величина 0.1 с. В этом случае для построения изображения размером 128x128 точек необходимо около получаса. Для ускорения получения экспериментальных данных существует два потенциальных приема: во-первых, это улучшение характеристик используемых детекторов, усилителей и т.д. Во-вторых, это увеличение числа датчиков, т.е. использования линеек или матриц датчиков. В настоящее время используются системы, оснащенные количеством датчиков порядка одного или нескольких десятков, что заведомо недостаточно для получения изображения без использования подвижных частей установки. Различные способы

реализации влияют на геометрию получаемых данных. Таким образом, может возникать необходимость в интерполяции экспериментальных данных на регулярную прямоугольную сетку для построения полученного изображения.

### Математическая постановка задачи

Рассмотрим модель формирования изображения на выходе системы. Она описывается классическим интегральным выражением для линейных систем

$$g(\vec{r}) = \iint_D f(\vec{\rho}) h(\vec{r}, \vec{\rho}) d\vec{\rho} + n(\vec{r}), \quad (1)$$

где  $\vec{\rho}$  – двумерный вектор мировых координат;  $\vec{r}$  – двумерный вектор в лабораторной системе координат;  $f(\vec{\rho})$  – исследуемое распределение радиояркостной температуры,  $h(\vec{r}, \vec{\rho})$  – аппаратная функция системы;  $n(\vec{r})$  – аддитивный шум;  $g(\vec{r})$  – результат измерений, область  $D$  представляет собой кольцо. В дальнейшем будем пользоваться более краткой записью

$$g = h \otimes f + n \quad (2)$$

Постановка задачи сверхразрешения заключается в следующем: основываясь на экспериментальных данных  $g(\vec{r})$ ,  $h(\vec{r}, \vec{\rho})$  и, возможно, дополнительной (априорной) информации, получить решение  $\hat{f}$ , близкое к исходному распределению  $f$ . Чаще всего в качестве критерия сходимости приближенного решения к точному берется близость к нулю функционала вида

$$J(\hat{f}) = \|g - h \otimes \hat{f}\|^2,$$

где  $\|\cdot\|$  — есть норма в  $L_2$ , поскольку предполагается, что  $n$  — гауссов белый шум [3].

Рассматриваемая задача, представляющая собой интегральное уравнение Фредгольма 1-го рода, является некорректно поставленной. Даже небольшой шум во входных данных приводит к сильному искажению получаемых результатов.

Для одномерного случая типичен выбор функции  $h(x, y) = \frac{\sin(x-y)}{x-y}$  в качестве модельной аппаратной функции. Если отсутствует шум, то есть  $n(x) \equiv 0$ , задача восстановления изображения может быть решена точно с помощью разложения по собственным функциям интегрального оператора с ядром  $h(x, y)$  [3]. Однако построенное таким образом решение является неустойчивым, наличие даже небольшого шума приводит к сильным отклонениям от точного решения.

### Методы решения

Существует классический, наиболее обоснованный с теоретической точки зрения, метод решения уравнения (1) с использованием процедуры регуляризации Тихонова.

Однако для практического применения он подходит плохо, так как требует введения поправок, получаемых либо эмпирическим путем, либо на основе достаточно полных априорных данных, что делает практически невозможным создание полностью автоматизированных систем. В то же время разработано большое количество практических алгоритмов, преимущественно итерационных, не требующих таких данных.

В работах [2,4-6] проведено сравнение различных алгоритмов и выявлено преимущество нелинейных итерационных алгоритмов, изначально нацеленных

на задачи обработки изображения — это алгоритмы восстановления пространства изображений (ISRA) и Люси-Ричардсона.

### Алгоритм восстановления пространства изображений (ISRA)

При построении данного алгоритма предполагается, что шум  $n$  представляет собой случайную величину, подчиняющуюся распределению Гаусса с нулевым средним значением и дисперсией  $\sigma_i$  [4]. Результаты измерений  $g_i$  (интенсивность в рассматриваемой  $i$ -ой точке) являются независимыми случайными величинами. Для каждого пикселя « $i$ » результат измерения  $g_i$  будет суммой детерминированной величины  $(Hf)_i$  и случайной величины  $n_i$ , где  $Hf = h \otimes f$ . Вероятность того, что величина  $g_i$  принимает значение  $(Hf)_i$ , определяется как

$$L(f) \propto \prod_i \exp \left[ -\frac{(g_i - (Hf)_i)^2}{2\sigma_i^2} \right].$$

Соответствующая логарифмическая функция правдоподобия есть

$$\log[L(f)] \propto -\frac{1}{2} \sum_i \frac{(g_i - (Hf)_i)^2}{\sigma_i^2}.$$

Решение  $f$  ищется путем минимизации евклидовой нормы с соответствующим весом для разности между результатами измерения и восстановленным средним значением интенсивности:

$$J(f) = -\log[L(f)] = \frac{1}{2} \sum_i \frac{(g_i - (Hf)_i)^2}{\sigma_i^2} = \frac{1}{2} \|g - Hf\|_{\sigma}^2,$$

Обозначим диагональную матрицу с элементами  $\frac{1}{\sigma_i^2}$  как  $R$ . Используя выражение для градиента  $\nabla_f J(f) = H^T R H f - H^T R H g$ , получаем следующий итерационный алгоритм [3], основанный на методе градиентного спуска:

$$f^{k+1} = f^k + \alpha \left[ -\nabla_f J(f^k) \right] = f^k + \alpha H^T R (g - Hf^k).$$

Достаточным условием сходимости такого алгоритма будет  $0 < \alpha < \frac{2}{\|H^T R H\|}$  [5].

Решение  $f^*$ , построенное с помощью данного итерационного процесса при  $k \rightarrow \infty$ , формально может быть записано в виде  $\nabla_f J(f) = (H^T R H)^{-1} H^T R g$ , что соответствует следующему условию минимума функционала первого порядка:  $\nabla J(f^*) = 0$ . Как уже отмечалось, восстановление функции, входящей в свертку, является некорректно поставленной задачей, поэтому полученное по данному алгоритму численное решение может быть неустойчивым. Более того, это решение может быть отрицательным, что не имеет физического смысла. Очевидно, при построении алгоритма следует наложить условие неотрицательности решения. Такому условию удовлетворяет итерационный алгоритм восстановления пространства изображения, являющийся модификацией метода градиентного спуска (Image Space Reconstruction Algorithm (ISRA) - алгоритм восстановления пространства изображений). Данный алгоритм может быть записан в виде:

$$\hat{f}_i^{k+1} = \hat{f}_i^k \frac{(H^T R g)_i}{(H^T R H \hat{f}^k)_i} \quad \forall i, \quad (3)$$

где  $k$  — номер итерации, в качестве начального приближения берется  $\hat{f}^0 = g$ .

В аддитивной форме формула (3) может быть записана как

$$\hat{f}_i^{k+1} = \hat{f}_i^k + \frac{\hat{f}_i^k}{(H^T R H \hat{f}^k)_i} \left[ H^T R (g - H \hat{f}^k) \right]_i. \quad (4)$$

Из (4) видно, что это модифицированный алгоритм типа наискорейшего спуска, в котором направление спуска на каждой итерации задается формулой

$$d^k = \text{diag} \left[ \frac{f_i^k}{\left( H^T R H f^k \right)_i} \right] H^T R (g - H f^k),$$

где  $\text{diag}[\cdot]$  представляет собой вектор, составленный из диагональных элементов матрицы. Легко видеть, что алгоритм гарантирует положительность решения, если начальное приближение положительно, так как аппаратная функция также положительна.

Вопрос о сходимости данного алгоритма исследован в работе [5], где показано, что последовательность (4) сходится к величине  $f$ , минимизирующей функционал  $J(f) = \frac{1}{2} \|y - Hf\|_{\sigma}^2$  при дополнительном условии  $f \geq 0$ .

В работе [2] проведено сравнение большого числа различных алгоритмов сверхразрешения. Алгоритм ISRA был признан по многим параметрам наиболее удачным для применения в задачах повышения разрешения систем радиовидения. Поэтому в настоящей работе предпочтение отдано именно этому алгоритму.

### **Интерполяция данных на прямоугольную сетку Метод обратных расстояний с весом (The inverse-distance weighted procedure)**

Способ интерполяции, основанный на использовании величин, обратных расстояниям между узлами интерполяции, возведенным в некоторую степень, универсален, легко реализуем вычислительной машиной и понятен, а также обладает высокой степенью точности.

В рамках этого метода, искомое значение функции в каждой точке, для которой ищется решение, дается следующим образом:

$$P_i = \frac{\sum_{j=1}^G \frac{P_j}{D_{ij}^n}}{\sum_{j=1}^G \frac{1}{D_{ji}^n}},$$

где  $P_i$  – найденное значение в точке  $i$ ;

$P_j$  – значение в узле интерполяции  $j$ ;

$D_{ij}$  – расстояние между  $i$ -той и  $j$ -той точками;

$G$  – количество узлов интерполяции;

$n$  – степень, в которую возводятся расстояния  $D_{ij}$ .

Значение  $n$  фактически определяет, насколько влияет расположение  $j$ -тых точек на прогнозируемое значение  $P_i$  в точке  $i$ . По мере того, как  $n$  увеличивается, область влияния уменьшается до тех пор, пока в пределе не станет окрестностью точки  $i$ , из любой точки которой до  $i$  ближе, чем до любой из точек  $j$ . Когда  $n$  принимается равным нулю, метод становится идентичен обыкновенному нахождению среднего арифметического рассматриваемых величин.

Ватсон (*Watson*) и Филип (*Philip*) в 1985 году составили список некоторых ограничений использования этого метода, например, его нельзя использовать для случая радиальной симметрии, когда в результате такой интерполяции могут появиться элементы неоправданной линейности.

Экспериментально было показано, что для  $n \leq 1$  полная производная функции  $z=f(x,y)$  в узлах интерполяции претерпевает разрыв. Обычно значение  $n$  подбирается эмпирически для конкретной задачи. Значения 1,65 и 2 выбирались для  $n$  Келвеем (*Kelway, 1974*) и NOAA (1972) соответственно, для прогнозирования количества выпадающих осадков, в то время как модель ARMOS (1990) для обнаружения месторождений нефти рекомендует использовать значения  $n$  от 4 до 8.

Ватсон и Филип показали, как нужно модифицировать этот способ интерполяции, чтобы устранились ограничения, упоминаемые выше. В их



методе степень  $n$  – функция как расстояний  $D_{ij}$ , так и погрешностей входных величин.

### Интерполяция с минимизирующей погрешность функцией (Punctual Kriging (PKG))

В рамках этого метода значение в каждом узле интерполяции берется с весом, а погрешность минимизируется путем решения системы следующих уравнений:

$$P_i = \sum_{j=1}^G W_j P_j ;$$

$$\sum_{j=1}^G W_j = 1 ;$$

$$\sum_{j=1}^G W_j \gamma(D_{kj}) + \lambda = \gamma(D_{ki}), k=1, 2, \dots, G,$$

где  $W_j$  – поправочные коэффициенты значений  $P_j$ , учитывающие близость каждого из узлов  $j$  к точке  $i$ ;

$\lambda$  – переменная, вводимая для минимизации погрешности;

$\gamma(D_{kj})$  – функция, восстанавливающая значение в точке, лежащей между точками  $k$  и  $j$ .

Даже в том случае, когда функция  $\gamma$  неизвестна, система уравнений все же может быть решена, если предположить, что искажающие эффекты отсутствуют. В этом случае она определяется следующим образом:

$$\gamma(D_{kj}) = A \cdot f(D_{kj}),$$

где  $f$  – функция от  $D_{kj}$ , вид которой определяется начальными условиями;

$A$  – константа.

Количество таких функций  $f$ , которые могут быть использованы для

вычисления функции  $\gamma$ , сильно ограничено. МакБретни (McBratney) и Вебстер (Webster) в 1986 составили список таких функций и дали объяснение, почему остальные функции не подходят. В самом простом случае, когда  $f(D_{kj}) = D_{kj}$ , уравнение

$$\sum_{j=1}^G W_j \gamma(D_{kj}) + \lambda = \gamma(D_{ki}), \quad k=1, 2, \dots, G$$

может быть заменено следующим:

$$\sum_{j=1}^G W_j D_{kj} + \beta = D_{ki}, \quad k=1, 2, \dots, G,$$

где  $\beta$  – константа (нет необходимости находить  $\beta$  или  $\lambda$  для того, чтобы вычислить значения  $P_i$ ).

В этом упрощении функция  $\gamma$  линейна. Оно применимо к тем случаям, когда количество узлов сетки интерполяции недостаточно для точного определения этой функции.

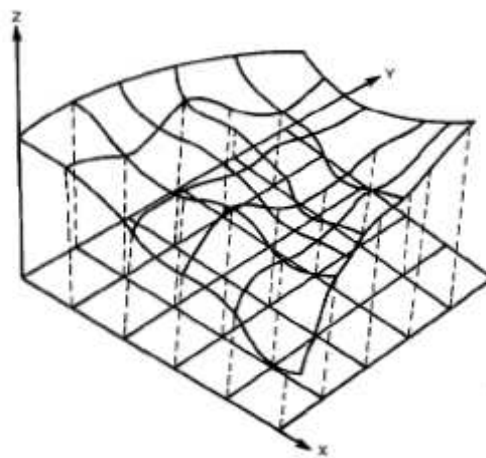
Исследования показали, что несмотря на то, что вид функции  $\gamma$  влияет на погрешности, на сами интерполируемые величины она оказывает небольшое влияние. Кук (Cooke) и Мостахими (Mostaghimi) в 1992 использовали линейную функцию  $\gamma$  для того, чтобы оценить значение осадков и получили значения, которые незначительно отличались от тех, что прогнозировались путем использования стандартных методов.

### **Интерполяция с помощью триангуляции на плоскости (Bivariate Interpolation)**

Для стандартного компьютерного дизайна (CAD) или тестирования (CAT), огромное количество входных данных внутри многомерной матрицы должно быть итеративно проанализировано за короткое время и, желательно, с низкими материальными затратами. Непосредственные вычисления для получения дополнительных входных данных, которые не располагаются в узлах матрицы, дорогостоящи. Это является достаточной причиной для поиска хорошо

приближающих методов интерполяции. Интерполяционные полиномы, приводимые в литературе по численным методам, обычно применимы только к регулярной сетке интерполяции. Однако для решения различных инженерных задач, например, таких, как структурная разработка параболических рефлекторов для спутниковых антенн, исследователь должен пользоваться нерегулярной сеткой интерполяции для создания соответствующей модели рефлектора. Для того, чтобы провести интерполяцию для величин, находящихся внутри сетки нерегулярно расположенных узлов, необходимо иметь формулу интерполяции, подходящую для распределенных таким образом узлов.

В общем случае, такая интерполяция используется для прямоугольной равномерной сетки (см. рис. ниже). Значение функции  $z=f(x,y)$  определено в узлах интерполяции  $(x_i, y_i)$  и интерполяция используется для определения значений функции  $z$  в других точках, не являющихся узлами.



Простой алгоритм нахождения значений, например, такой, как интерполяция многочленом Лагранжа, может быть использован для нахождения значений гладких непрерывных функций. Применительно к случаю, когда значения функции  $z_{ij}=f(x_i, y_i)$  определены в узлах интерполяции  $(x_i, y_i)$  регулярной сетки ( $i=0, \dots, I; j=0, \dots, J$ ), и в этих точках существуют первые производные, т. е. определены  $z_x(x_i, y_i)$  и  $z_y(x_i, y_i)$ , в ходе интерполяции происходит попытка максимально приблизить  $z=f(x, y)$  к гладкой функции, принимающей заданные значения в узлах интерполяционной сетки. В этом

случае функция находится следующим образом:

$$f(x, y) = \sum_{m=0}^M \sum_{n=0}^N a_{mn}^{ij} (x - x_{i-1})^m (y - y_{i-1})^n,$$

где  $x_{i-1} \leq x \leq x_i$ ;

$M, N$  выбираются так, чтобы коэффициенты  $a_{mn}^{ij}$  могли быть найдены посредством решения полученных уравнений

$$z_{ij} = 0;$$

$$z_x(x_i, y_j) = 0;$$

$$z_y(x_i, y_i) = 0;$$

$$z_{xy}(x_i, y_j) = 0.$$

В том случае, когда  $M=N=3$ , функция  $z$  называется *бикубической*.

Область применения этого метода интерполяции ограничивается данными, расположенными регулярно, образующими прямоугольную сетку интерполяции.

Для того чтобы получить функцию, пригодную для интерполяции нерегулярно расположенных исходных данных, вначале необходимо провести разбиение плоскости  $(x, y)$  на непересекающиеся треугольники с вершинами в опорных точках (*триангуляцию*), причем вершинами каждого из треугольников являются проекции на его плоскость узлов интерполяции; каждому треугольнику ставится в соответствие полином пятой степени от переменных  $x$  и  $y$ . Вычисляются значения частных производных в каждой исходной точке и затем используются для определения коэффициентов для полиномов, каждый из которых задается для своего треугольника. Значения переменной  $z$  вычисляются посредством этих полиномов, зависящих от  $x$  и  $y$ :

$$z(x, y) = \sum_{j=0}^5 \sum_{k=0}^{5-j} a_{jk} x^j y^k. \quad (5)$$

Посредством подстановки в формулу (5) известных значений  $z$  в вершинах треугольника и частных производных  $z_x, z_y, z_{xx}, z_{xy}, z_{yy}$  (это дает 18 линейно независимых уравнений) определяется 21 коэффициент  $\{a_{jk}\}$ . Еще три

уравнения можно получить, подставив значения частных производных  $z$ , взятых по направлениям, перпендикулярным каждой из трех сторон треугольника соответственно.

Весь процесс можно кратко описать следующим образом:

1) провести триангуляцию точек на плоскости  $x, y$ , считая их вершинами треугольников триангуляции;

2) найти частные производные  $z$  в каждой точке  $(x_i, y_i)$ ;

3) для точки  $(x, y)$ , в которой ищется значение  $z(x, y)$  найти проекцию на плоскость  $x, y$  и треугольник, внутри которого эта проекция располагается;

4) исходя из заданных и найденных условий, используя формулу (5), вычислить искомое значение  $z(x, y)$ .

Если узлы интерполяции расположены регулярно, можно воспользоваться интерполяционным многочленом Лагранжа (нет необходимости проводить триангуляцию в плоскости  $(x, y)$ ).

### **Алгоритм триангуляции**

*Триангуляция Делоне* - это разбиение нерегулярного множества опорных точек на такую сеть треугольников, когда система взаимосвязанных неперекрывающихся треугольников имеет наименьший диаметр, если ни одна из вершин не попадает внутрь ни одной из окружностей, описанных вокруг образованных треугольников [7].

Это означает, что образовавшиеся треугольники при такой триангуляции максимально приближаются к равносторонним, а каждая из сторон образовавшихся треугольников из противоположащей вершины видна под максимальным углом из всех возможных для точек соответствующей полуплоскости.

При создании алгоритма, автоматизирующего построение триангуляции Делоне, в качестве базового было выбрано свойство триангуляции Делоне давать локально наиболее правильные треугольники. Это позволило, во-первых, не бояться ошибок, связанных с машинными погрешностями, во-

вторых, получить *инкрементный алгоритм*, то есть алгоритм, постепенно включающий точки в уже построенную триангуляцию и по определению позволяющий легко реагировать на ситуацию при появлении новых данных и, в-третьих, данный алгоритм работает за время, пропорциональное числу точек.

Ещё одним преимуществом этого алгоритма является простота удаления ранее внесённых и обработанных точек.

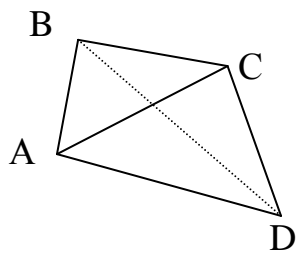
Благодаря возможности легко вносить и удалять точки из ранее построенной триангуляционной сетки, получается алгоритм, способный работать в режиме реального времени.

Одной из основных трудностей при использовании инкрементного алгоритма является проблема отыскания того треугольника, в который попала новая точка. Эта проблема решена следующим образом: вводится вспомогательная очень крупная прямоугольная сетка, в каждую из ячеек которой попадает некоторое число треугольников. Определив, к какой ячейке принадлежит новая точка, локализируются "подозрительные" треугольники. Проверив принадлежность точки к этим треугольникам, либо получаем искомый номер, либо выясняется, что точка лежит вне уже построенной конструкции.

Обработка последнего случая вызывает некоторые трудности, требуется проверить каждую из внешних точек имеющейся триангуляции Делоне на возможность соединения с новой точкой, и последующего образования треугольника. Эта проблема решается путём исследования отрезка, соединяющего новую точку с рассматриваемой, на предмет пересечения его с каким-либо внешним из ребер имеющейся триангуляции Делоне. Пересечение (или его отсутствие) устанавливается изучением системы линейных уравнений, образованной уравнениями прямых, содержащих вышеупомянутые отрезки.

В случае попадания новой точки в треугольник, ранее построенной триангуляции, происходит ликвидация этого треугольника и появление трёх новых (на базе четырёх точек – новой и трёх вершин ликвидированного

треугольника). Появившиеся треугольники исследуются на возможность построения флипов (*флип* - операция переброски диагонали выпуклого четырехугольника, т.е. если у четырехугольника  $ABCD$ , разбитого диагональю  $BD$  на два треугольника, удалить эту диагональ и заменить ее на  $AC$ , то получится другая триангуляция четырехугольника). Преобразованные после флипов треугольники добавляются в разряд исследуемых.



Выбор структуры для представления триангуляции оказывает существенное влияние на теоретическую трудоёмкость алгоритмов, а также на скорость конкретной реализации. Кроме того, выбор структуры может зависеть от цели дальнейшего использования триангуляции.

В триангуляции можно выделить три основных вида объектов: *узлы* (точки, вершины), *рёбра* (отрезки) и *треугольники*.

В работе используемого алгоритма построения триангуляции Делоне возникают следующие операции с объектами триангуляции:

1. *Треугольник*  $\rightarrow$  *узлы*: получение для данного треугольника координат образующих его узлов.

2. *Треугольник*  $\rightarrow$  *рёбра*: получение для данного треугольника списка образующих его рёбер.

3. *Треугольник*  $\rightarrow$  *треугольники*: получение для данного треугольника списка соседних с ним треугольников.

4. *Ребро*  $\rightarrow$  *узлы*: получение для данного ребра координат образующих его узлов.

5. *Ребро*  $\rightarrow$  *треугольники*: получение для данного ребра списка соседних с ним треугольников.

6. *Узел*  $\rightarrow$  *рёбра*: получение для данного узла списка смежных рёбер.

7. Узел  $\rightarrow$  треугольники: получение для данного узла списка смежных треугольников.

В используемой структуре для каждого узла триангуляции хранятся его координаты на плоскости и список указателей на соседние узлы (список номеров узлов), с которыми есть общие рёбра. По сути, список соседей определяет в неявном виде рёбра триангуляции. Треугольники же при этом не представляются вообще, что является обычно существенным препятствием для дальнейшего применения триангуляции. Кроме того, недостатком является переменный размер структуры узла, зачастую приводящий к неэкономному расходу оперативной памяти при построении триангуляции.

### **Экспериментальные данные**

Используемая экспериментальная установка устроена следующим образом: измерительные датчики линейно расположены на неподвижной планке, в то время как сферическая поверхность, служащая для фокусировки сигнала, совершает вращательно-поступательное движение вокруг оптической оси установки. Таким образом, геометрия получаемых экспериментальных данных имеет весьма нетривиальный характер. Все точки, в которых проводятся измерения, расположены на нескольких (порядка 10) сдвинутых относительно друг друга окружностях. Причем расстояние между соседними точками на одной окружности постоянно, как и расстояние между центрами соседних окружностей. Количество точек на одной окружности — порядка 500.

Плотность покрытия точками исследуемой области сильно зависит от соотношения диаметра окружностей и сдвига между ними. Также характерной особенностью является очень высокая плотность точек на окружностях и большие незаполненные области между ними.

### **Восстановление изображений**



Поскольку все экспериментальные точки лежат на не слишком сложной кривой (на окружности), удобно перейти к одномерной задаче на этой кривой. Таким образом удастся максимально уточнить (восстановить) данные на окружностях.

Занумеруем все экспериментальные точки по порядку, сначала на первой окружности, потом на второй, и так далее. Точки, находящиеся на разных окружностях, будем обрабатывать отдельно. Численный эксперимент показал, что влияние сравнительно удаленных точек, находящихся на других окружностях, не очень велико, и им можно пренебречь. После восстановления значений на каждой окружности осуществляется интерполяция данных на прямоугольную сетку. Основные преимущества подобного подхода состоят в следующем:

во-первых, исходная двумерная задача сводится к решению набора одномерных задач, что гораздо проще с вычислительной точки зрения;

во-вторых, в случае неисправности одного из датчиков это не оказывает влияние на обработку данных с других датчиков;

в-третьих, решается проблема с нормировкой аппаратной функции, так как у каждой точки при таком подходе существует конечное число соседних точек на заранее известных расстояниях;

в-четвертых, возможно разделение задачи на несколько независимых задач по числу окружностей, что позволяет решать задачу с помощью параллельных вычислений на нескольких ЭВМ и получить многократный выигрыш в скорости.

Аппаратная функция каждой одномерной задачи фактически представляет собой квадратную симметричную матрицу  $h_{ij}$  размера  $N \times N$ . Заменяем свертки по площади исходной области на одномерные суммы вида

$$(h \otimes f)_k = \sum_t h_{tk} f_t \quad (6).$$

Таким образом, формулы используемого алгоритма ISRA принимают вид:

$$\hat{f}_k^{i+1} = \hat{f}_k^i \frac{\sum_j h_{kj} g_j}{\sum_m h_{km} \sum_l h_{ml} \hat{f}_l^i}.$$

Сходимость данного алгоритма может быть доказана методом проектирования на выпуклые множества [2].

Благодаря сохранению геометрии исходных данных возможно кэширование значительного количества вычисляемых величин: значений аппаратной функции для всех пар точек, расстояний между ними, расстояний между исходными и интерполяционными точками. Несмотря на значительный объем кэшируемых данных, обращение к ним может происходить последовательно, что в сочетании с большими размерами доступной памяти на современных компьютерах делает возможным ускорить процесс обработки изображения в десятки раз.

В целом процесс обработки экспериментальных данных выглядит следующим образом:

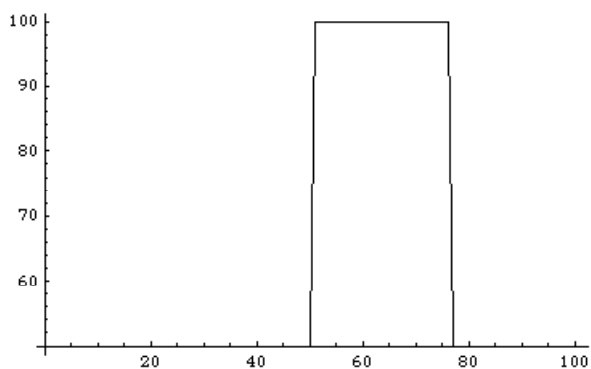
1. при необходимости осуществляется прореживание экспериментальных данных;
2. отдельно для каждой окружности реализуется алгоритм восстановления изображения ( $n$ -итераций);
3. оценивается качество полученного изображения, принимается решение о продолжении или завершении процесса;
4. восстановленные данные интерполируются на прямоугольную сетку;
5. полученное изображение сохраняется и демонстрируется на экране.

## Результаты

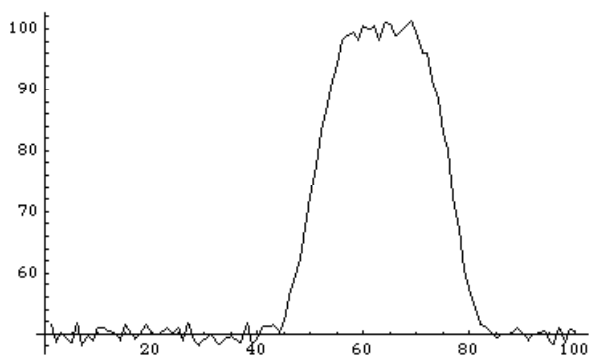
Изучение сходимости алгоритма ISRA было проведено на примере одномерной задачи. Рассматривалась единичная окружность с

экспериментальными точками. На модельных данных установлена сходимость алгоритма в норме  $L_2$ .

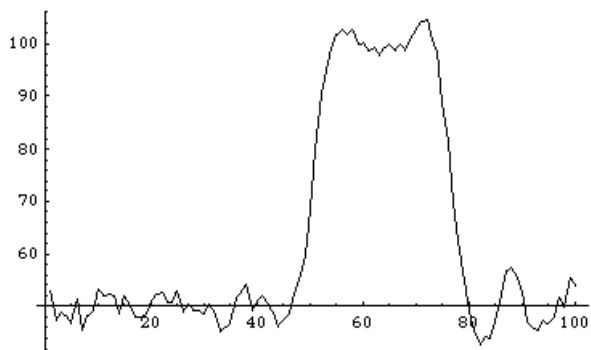
Ниже представлены графики исходного сигнала (график 1), сигнала после воздействия аппаратной функции и наложения шума (график 2), график восстановленного сигнала (график 3) и график разницы между исходным и восстанавливаемым сигналом на различных итерациях (график 4), то есть график зависимости нормы разности исходного и восстанавливаемого сигнала от номера итерации.



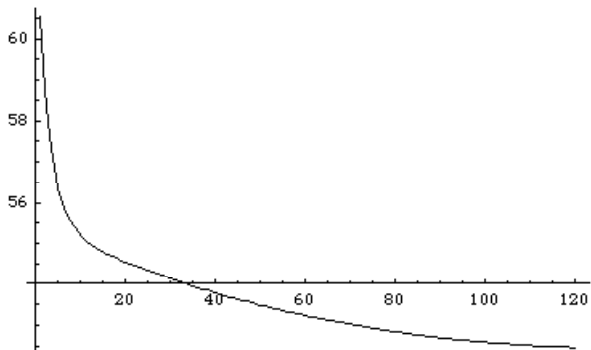
*график 1*



*график 2*



*график 3*

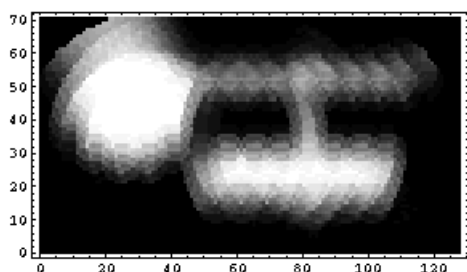


*график 4*

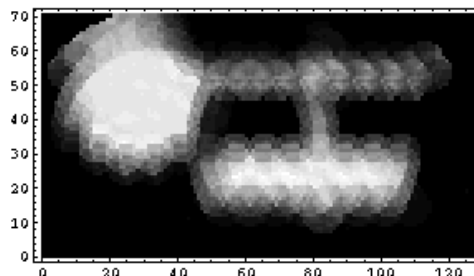
Восстановлению основного характера изображения (в данном случае: ширины и формы пика) сопутствует также усиление фонового шума. К сожалению, этот недостаток характерен для всех алгоритмов восстановления изображения, и для его устранения требуются дополнительные сведения о начальных данных.

Кроме того, проведен ряд численных экспериментов с модельными данными для двумерного случая: имеет место сходимость итерационного

процесса восстановления, а также визуальное повышение качества изображения. Ниже представлены примеры работы программы. Созданное вручную прямоугольное изображение (круг и три прямоугольника разных размеров) было размыто с помощью стандартной модельной аппаратной функции и интерполировано на окружности. Результат преобразования отображен на первом рисунке. Далее было проведено 100 итераций алгоритма ISRA. Результат восстановления показан на втором рисунке. Заметно повышение резкости контуров контрастных областей (особенно заметно для верхнего горизонтального прямоугольника).



*модельное изображение*



*изображение после 100 итераций ISRA*

Отдельно рассматривался вопрос о качестве восстановленного изображения в зависимости от геометрии расположения экспериментальных точек. Естественно, что оптимальным является наиболее равномерное покрытие области. На плохо покрытых экспериментальными точками частях области, например, вблизи границы, появляются значительные искажения. В то же время большая плотность точек на окружностях при малом количестве самих окружностей не улучшает качества восстанавливаемого изображения, а при использовании интерполяционных методов, основанных на триангуляции области, может приводить и к ухудшению качества восстановления. Этот вопрос рассматривался при обработке ряда экспериментальных данных.

Создана программа, осуществляющая восстановление и интерполяцию экспериментальных данных. Программа реализована на языке C и способна работать в любой UNIX-подобной операционной системе (тестирование проводилось на linux, mac os x). Реализованы 3 режима работы:

- режим генерации таблиц;
- нормальный режим;
- режим работы с модельными данными.

Для повышения производительности максимально возможное число вычислений перенесено на этап генерации таблиц, при этом формируется большой (порядка 100 мегабайт) кэшируемый массив данных, использующихся в дальнейшей работе. Это позволило вплотную приблизиться к требованию работы в реальном времени: для модельных данных (512 точек, 1000 итераций ISRA, inverse-distance weighted procedure, размер конечной сетки 128x128) время обработки данных составило менее секунды на компьютере PowerPC G4 (1.42 GHz, 512 Mb RAM).

### **Заключение**

Разработка математических методов обработки радиометрических изображений позволяет частично компенсировать физические недостатки метода. Особенно актуальной является задача первичного анализа получаемых изображений: выделение контрастных областей, оконтуривание потенциальных предметов. Наложение изображений, получаемых радиометрическим и обычным (в видимом диапазоне) способом, особенно в реальном времени, позволяет решать задачи выявления скрытого под одеждой оружия и других предметов. Пассивный характер радиометрии позволяет сделать этот процесс внешне незаметным.

Развитие экспериментальной техники на базе Центра Магнитной Томографии МГУ, осуществляемое в настоящее время (создание многоканальных систем, повышение скорости сбора информации) создает

необходимость в развитии соответствующих математических методов, позволяющих с небольшими временными затратами обрабатывать и визуализировать радиометрические изображения.

### Литература

1. Пирогов Ю.А. Пассивное радиовидение в миллиметровом диапазоне длин волн // Известия вузов. Радиофизика. - 2003. - Том XLVI. №8-9. С.660-670.
2. Тимановский А.Л. Сверхразрешение в системах пассивного радиовидения // Диссертация на соискание ученой степени кандидата физ.-мат. наук.- М.: физич.ф-т МГУ, 2007.
3. Madisetti V.K., Williams D.B. The Digital Signal Processing Handbook.- CRC Press, 1999.
4. Lanteri H., Soummer R., and Aime C. Comparison between ISRA and RLA algorithms. Use of a Wiener Filter based stopping criterion // Astronomy & astrophysics supplement series, Vol. 14, 1999, pp. 235-246.
5. De Pierro A.R. On the Relation between the ISRA and the EM Algorithm for Positron Emission Tomography, 1993, IEEE. Trans. Med. Imaging, MI-12, 328-333.
6. Пирогов Ю.А., Гладун В.В., Тищенко Д.А., Тимановский А.Л., Шлемин И.В, Джен С.Ф., // Журнал радиоэлектроники (<http://jre.cplire.ru>), 2004, №3.
7. Скворцов А.В. Триангуляция Делоне и ее применение //ТГУ. 2002. С. 27-29.