

ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО ФАЗОВЫМ ДИАГРАММАМ

1. Постановка задачи. В настоящее время наблюдается повышенный интерес к способам представления и обработки графической информации с использованием ЭВМ [1—3]. С одной стороны, это вызвано потребностью создания автоматизированных систем, содержанием которых является графическая информация, с другой стороны, — возросшими возможностями современных вычислительных средств. Так, в настоящее время накоплен обширный экспериментальный материал по фазовым диаграммам многокомпонентных соединений, значение которых велико во многих областях естественных наук. Это привело к необходимости создания информационной системы на основе базы данных фазовых диаграмм (БДФД). Система предназначена для хранения и обработки: 1) графической информации по фазовым диаграммам в пространстве изменения состава, давления и температуры; 2) проекций фазовой диаграммы в пространстве изменения температуры и давления ($P-T$ -проекций); 3) необходимой справочной информации по указанной фазовой диаграмме (ФД).

Фазовая диаграмма рассматривается для n -компонентных соединений, где $n \geq 2$. Давление может быть фиксированным или переменным.

Исходная графическая информация задается в виде таблиц экспериментальных точек на ФД или $P-T$ -проекции, но хранить ее в памяти ЭВМ целесообразно в виде аппроксимаций [4]. Поэтому поверхности, заданные экспериментальными точками, перед записью в БДФД аппроксимируются и хранятся в виде коэффициентов аппроксимации. При этом все точки поверхности разбиваются на два множества Z_1 и Z_2 . Аппроксимационная функция строго проходит через точки из Z_1 и может несколько отклоняться от точек множества Z_2 , т. е. «сглаживать» их. К Z_1 обычно относят невариантные точки и граничные точки поверхностей, так как они измерены с большей точностью, и если их аппроксимировать с некоторой погрешностью, то на графике могут появиться ложные области (здесь и далее под областями будем понимать геометрические образы фаз и гетерогенных состояний смеси). С другой стороны, если все точки поверхности интерполировать, $Z_2 = \emptyset$, то в большинстве случаев аппроксимационная функция будет осциллировать из-за большой погрешности измерений многих экспериментальных точек.

Разработанное программное обеспечение БДФД рассчитано на решение следующих задач: 1) получение двумерного сечения ФД в виде графика, разрисованного на графопостроителе и/или на АЦПУ (возможно использование соответствующих дисплеев); 2) то же для $P-T$ -проекции; 3) определение кода области по фиксированной точке на ФД; 4) нахождение пределов существования области по фиксированной точке и направлению на ФД; 5) получение точек поверхностей на ФД или линий на $P-T$ -проекции, определяемых по коду или по минимальному отклонению от фиксированной точки по температуре, при этом будет выдано исправленное значение температуры в заданной точке; 6) выдача таблицы невариантных точек на ФД; 7) выдача всех кодов на ФД и $P-T$ -проекции; 8) выдача справочной информации по документу.

Следует отметить, что список введенных операций является базовым для информационной системы по фазовым диаграммам. Однако список решаемых задач может быть расширен за счет прикладных программ, которые, в свою очередь, могут быть встроены в программное обеспечение, т. е. управляться средствами языка запросов.

2. Алгоритмы аппроксимации в БДФД. В состав программ первичной обработки данных входят два основных пакета по m -мерной аппроксимации.

2.1. Смешанная полиномиальная аппроксимация. Выбор полиномов в качестве аппроксимирующих конструкций обусловлен тем, что разделяющие поверхности на фазовых диаграммах, за редким исключением, в пределах погрешности могут быть представлены полиномами невысоких степеней. Кроме того, полином является самым оптимальным по времени обработки и количеству занимаемой памяти из всех известных конструкций.

Введем некоторые обозначения: X, Y, Z_1, Z_2, F — гильбертовы пространства, где $F = [Y \times Z_2]$. Операторы $T: X \rightarrow Y, A_1: X \rightarrow Z_1, A_2: X \rightarrow Z_2, L: X \rightarrow F, L = [T, A_2]$.

Постановка задачи смешанной аппроксимации представлена в виде

$$\Phi_\alpha(\sigma) = \min_{x \in A_1^{-1}Z_1} \Phi_\alpha(x) = \min_{x \in A_1^{-1}Z_1} \alpha \|Tx\|_Y^2 + \|A_2x - Z_2\|_{Z_2}^2, \quad (1)$$

где $\sigma \in A_1^{-1}Z_1 = \{x \in X: A_1x = Z_1\}$; $\alpha > 0$ — коэффициент сглаживания; $\|B\|_C$ — норма B в пространстве C .

Заметим, что при $Z_2 = \emptyset$ σ — интерполяционный сплайн, а при $Z_1 = \emptyset$ σ — сглаживающий сплайн.

Посредством ограничений на свойства операторов A_1, A_2, L, T могут быть сформулированы теоремы существования и единственности сплайна σ . Их доказательство во многом аналогично доказательству теорем существования и единственности интерполяционного сплайна [5, 6].

Предположим, что X — конечно-мерное пространство полиномиальных или кусочно-полиномиальных функций с базисными элементами $\omega_i, i = 1, n$. В этом случае решение уравнения (1) приводит к блочной системе линейных алгебраических уравнений

$$\begin{pmatrix} \alpha \bar{T} + \bar{A}_2 \bar{A}_1^* \\ \bar{A}_1 & 0 \end{pmatrix} \begin{pmatrix} \bar{\sigma} \\ \bar{\lambda} \end{pmatrix} = \begin{pmatrix} \bar{f}_2 \\ \bar{f}_1 \end{pmatrix}, \quad (2)$$

где $\bar{A}_1 = (A_1\omega_i, A_1\omega_j)_{Z_1}$; $\bar{T} = (T\omega_i, T\omega_j)_Y$; $\bar{A}_2 = (A_2\omega_i, A_2\omega_j)_{Z_2}$;

(\cdot, \cdot) — скалярное произведение; \bar{A}_1^* — сопряженный оператор к $\bar{A}_1, \bar{A}_1^* = \bar{A}_1$; $\bar{f}_k = ((z_k, A_k\omega_1)_{Z_k}, (z_k, A_k\omega_2)_{Z_k}, \dots, (z_k, A_k\omega_n)_{Z_k}), k = 1, 2$; $\bar{\sigma} = (\sigma^1, \sigma^2, \dots, \sigma^n)$; σ^i — коэффициенты разложения σ по базису в пространстве X ; $\bar{\lambda}$ — вектор множителей Лагранжа.

В качестве аппроксимирующей конструкции был выбран полином следующего вида:

$$P(x) = \sum_{i_1=1}^{n+1} \dots \sum_{i_m=1}^{n+1} a_{i_1 \dots i_m} \prod_{j=1}^m x_j^{n-i_j+1}. \quad (3)$$

Здесь $x = (x_1, x_2, \dots, x_m) \in R^m$; m — размерность аппроксимируемой поверхности; n — степень полинома по каждой координате.

Из (1) следует, что решением задачи смешанной аппроксимации является полином $P^*(x)$, удовлетворяющий следующим соотношениям:

$$P^*(x^{(i)}) = f(x^{(i)}), x^{(i)} \in Z_1, i = 1, N_1; \quad (4)$$

$$\begin{aligned} \Phi_\alpha(P^*(x)) = \min_{P(x)} \Phi_\alpha(P(x)) = \min_{P(x)} \sum_{j=1}^{N_2} \beta_j (P(x^{(j)}) - f(x^{(j)}))^2 + \\ + \alpha \int_{\Omega} [D^{(h)}P(x)] d\Omega, x^{(j)} \in Z_2, j = 1, N_2, \end{aligned} \quad (5)$$

где $\beta_j, j = 1, N_2$ — статистические веса сглаживаемых точек; α — коэффициент сглаживания; Ω — область, на которой минимизируется значение

k -й производной от $P(x)$;

$$D^k = \sum_{i_1=0}^k \sum_{i_2=0}^{k-s_1} \dots \sum_{i_{m-1}=0}^{k-s_{m-2}} \frac{k!}{i_1! i_2! \dots i_{m-1}!} \left(\frac{\partial}{\partial x_1^{i_1} \dots \partial x_m^{i_m}} \right)^2;$$

$$s_j = \sum_{i=1}^j i_i, \quad i_m = k - s_{m-1}.$$

В пакете присутствует подпрограмма, в которой организована итерационная схема выбора величин коэффициента сглаживания α и наименьшего значения n -степени полинома. При этом должны одновременно выполняться два неравенства:

$$\xi(\alpha) \leq \xi_{кр}; \quad (6) \quad \varphi(\alpha) \leq \varphi_{кр}, \quad (7)$$

где $\xi(\alpha) = (\bar{T}, \bar{\sigma}, \bar{\sigma})$; $\xi_{кр}$ — максимально допустимое значение для $\xi(\alpha)$; $\varphi(\alpha)$ — уровень невязки; $\varphi_{кр}$ — максимальная невязка, которая задается пользователем во входных данных.

Величина $\xi(\alpha)$ нормируется в программе, что позволяет ввести универсальное значение для $\xi_{кр}$. Пакет смешанной полиномиальной аппроксимации находит решение $P^*(x)$, которое проходит через точки Z_1 и «сглаживает» остальные Z_2 (точки из Z_2).

2.2. Аппроксимация с использованием функции Грина. Если точность экспериментальных данных высокая и невозможно аппроксимировать их полиномом (3) в пределах заданной точности, т. е. не удовлетворяются неравенства (6), (7), то производится кусочная аппроксимация на основе функций Грина [5]. Для этого используется комплект программ GREEN [5], работающий в интерполяционном режиме.

В программном обеспечении имеется вспомогательный пакет для аппроксимации гиперплоскостями. Он обладает более высокой точностью и меньшим временем работы по сравнению с двумя предыдущими.

В заключение заметим, что около 90% всех поверхностей на ФД аппроксимируются первым пакетом и вспомогательным.

3. Представление n -мерной графической информации в ЭВМ. В работах [2, 3] предлагается использовать параметрическое представление функций для обработки графической информации в ЭВМ:

$$y = y(t); \quad (8) \quad x = x(t), \quad n = 2. \quad (9)$$

Такой способ имеет неоспоримое преимущество в том случае, когда при расчете каких-либо характеристик удается избежать необходимости однозначного вычисления функции: $y = y(x)$.

При параметрическом представлении (8), (9) это, как правило, не удается сделать. А если и удастся, то придется решать нелинейное уравнение для определения параметра t , что существенно сказывается на времени счета. При рассмотрении приведенных ниже алгоритмов станет ясно, что вычисление функции $y = y(x)$ является основой первых двух алгоритмов. Поэтому для представления графической информации в ЭВМ для БДФД используются однозначные функции [7].

Рассмотрим евклидово n -мерное пространство E^n с ортогональным базисом $\{e_1, e_2, \dots, e_n\}$. В E^n заданы области Ω_i , $i = \overline{1, N}$, где N конечно. Каждая область Ω_i имеет размерность n , и в совокупности они подчинены следующим условиям:

$$1) \bigcup_{i=1}^N \Omega_i = E^n;$$

2) $\Omega_i \cap \Omega_j = \Gamma_{i,j}$, $i, j = \overline{1, N}$, $i \neq j$ ($\Gamma_{i,j}$ — общая граница областей Ω_i и Ω_j размерности не больше $n-1$); $\Gamma_{i,j} = \emptyset$ для некоторых i и j ;

3) существуют $\Gamma_{i,j}^{q_i}$ такие, что $\bigcup_{q_i=1}^{k(i,j)} \Gamma_{i,j}^{q_i} = \Gamma_{i,j}$, где $\Gamma_{i,j}$ имеют размерность $n-1$ и $k(i, j)$ конечно. Кроме того, существует l_i такое, что

$$\Gamma_{i,j}^{q_i} = \{x \in E : x_{i_1} = A_{i,j}^{q_i} x', x' \in E_{i_1}^{n-1}\}, \quad (10)$$

где $A_{ij}^{q_1}$ — однозначная функция; i — номер области Ω_i , соответствующей большим значениям x_{l_1} ; j — номер области Ω_j , соответствующей меньшим значениям x_{l_1} .

Введем в рассмотрение области $\Omega_{ij}^{q_1}$, на которых заданы функции $A_{ij}^{q_1}$. По сути дела, $\Omega_{ij}^{q_1}$ — ортогональная проекция $\Gamma_{ij}^{q_1}$ на подпространство $E_{l_1}^{n-1}$, состоящее из векторов $(x_1, x_2, \dots, x_{l_1-1}, 0, x_{l_1+1}, \dots, x_n)$.

Для области $\Omega_{ij}^{q_1}$ введем в рассмотрение минимальный включающий прямоугольник $P_{ij}^{q_1} \subset E_{l_1}^{n-1}$, определяемый отрезками на осях $(\alpha_{l_1 s}^1, \alpha_{l_1 s}^2)$, где $s = \overline{2, n}$, $\alpha_{l_1 s}^1$ — максимально возможное число, а $\alpha_{l_1 s}^2$ — минимально возможное число и $\alpha_{l_1 s}^1 < \alpha_{l_1 s}^2$ при условии $\Omega_{ij}^{q_1} \subset P_{ij}^{q_1}$;

4) все $\alpha_{l_1 s}^1$ и $\alpha_{l_1 s}^2$ для $P_{ij}^{q_1}$, $s = \overline{2, n}$, должны быть конечны.

Нетрудно показать, что при выполнении введенных ограничений существует только одна область $\Omega_{ij}^{q_1}$, проекция которой на $E_{l_1}^{n-1}$ не принадлежит объединению всех $\Omega_{ij}^{q_1}$. Будем называть ее пустой.

Выделим одну из переменных, например x_{l_2} , $l_2 \neq l_1$, и построим аппроксимацию границ области $\Omega_{ij}^{q_1}$ относительно переменной x_{l_2} :

$$x_{l_2} = A_{ijp}^{q_1 q_2} (x_1, x_2, \dots, x_{l_2-1}, x_{l_2+1}, \dots, x_{l_1-1}, x_{l_1+1}, \dots, x_n), \quad (11)$$

где $A_{ijp}^{q_1 q_2}$ — однозначные функции, $q_2 = \overline{1, k_2(i, j, q_1)}$; $p = 0$, если точки области $\Omega_{ij}^{q_1}$ соответствуют меньшим значениям x_{l_2} ; $p = 1$, если точки области $\Omega_{ij}^{q_1}$ соответствуют большим значениям x_{l_2} .

Получая проекции границ вдоль координат и аппроксимируя их, находим

$$x_{l_n} = A_{ijp}^{q_1 q_2 \dots q_n},$$

где $q_s = \overline{1, k_s(i, j, q_1, q_2, \dots, q_{s-1})}$, $s = \overline{1, n}$; $k_s(i, j, q_1, q_2, \dots, q_{s-1})$ — конечные натуральные числа.

При представлении структур s -го шага получения проекций ($s = \overline{2, n}$) нет необходимости учитывать образующие поверхностей, параллельных оси проецирования x_{l_s} [7]. Заметим, что выполняется взаимно однозначное соответствие между исходными поверхностями и тройками (i, j, q_1) .

Стало уже традиционным, что графическая информация может быть представлена в виде древовидной структуры, например, [2], однако в вершины графа всякий раз вкладывается различное содержание. Нетрудно заметить, что описанное выше представление графической информации также может быть структурировано в виде дерева. Каждая поверхность

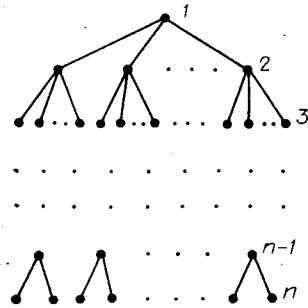


Рис. 1.

$A_{ij}^{q_1}$ будет представлена графом (рис. 1). Вершины графа s -го уровня, $s = \overline{1, n}$, соответствуют границам структур $s - 1$ -го шага получения проекций. Дуги графа характеризуют принадлежность структур s -го уровня, $s = \overline{2, n}$, причем дуга имеет код 0, если $p = 0$ для структуры $A_{ijp}^{q_1 q_2 \dots q_s}$, и код 1, если соответствующее $p = 1$. Поскольку области всех уровней проекций могут иметь общие границы, то все графы, представляющие общую структуру графической информации, могут иметь общие вершины на уровнях $s = \overline{2, n}$, при этом дуги, прихо-

дящие в вершину от различных графов, содержат различные коды в зависимости от значения p . Это позволяет избавиться от избыточности представления графической информации.

Набор графов (см. рис. 1) является внутренним представлением n -мерной фазовой диаграммы. Графы для $P-T$ -проекций и $T-X$ -диаграмм имеют всего два уровня.

4. Алгоритмы обработки n -мерной графической информации. На основе представления, описанного в 3, в пакете прикладных программ реализованы следующие алгоритмы.

4.1. *Определение номера области по фиксированной точке.* Для работы алгоритма необходима фиксированная точка $x^0 = (x_1^0, x_2^0, x_3^0, \dots, x_n^0) \in E^n$.

В шагах с 1-го по $n-1$ проверяем выполнение неравенств для троек (i, j, q_i) :

$$A_{ijp'}^{q_1 q_2 \dots q_s} (x^0) \leq x_{i_s}^0 \leq A_{ijp''}^{q_1 q_2 \dots q_s} (x^0), \quad (12)$$

где $s = n, n-1, \dots, 2$; $p' = 1, p'' = 0$; $A_{ijp'}^{q_1 q_2 \dots q_s} (x^0) = \max_{q_s} A_{ijp'}^{q_1 q_2 \dots q_s} (x^0)$;

$$A_{ijp''}^{q_1 q_2 \dots q_s} (x^0) = \min_{q_s} A_{ijp''}^{q_1 q_2 \dots q_s} (x^0).$$

Если на каком-либо шаге для тройки (i, j, q_i) ни разу не удовлетворились неравенства (12), то на последующем шаге эта тройка не рассматривается. Нетрудно проверить, что после выполнения шага $n-1$ останутся тройки, соответствующие поверхности для которых имеют непустое пересечение с прямой:

$$x_1 = x_1^0, \dots, x_{i_1-1} = x_{i_1-1}^0, x_{i_1+1} = x_{i_1+1}^0, \dots, x_n = x_n^0.$$

На шаге n для оставшихся троек строится последовательность

$$a_{imjm}^{q_1 m} = A_{imjm}^{q_1 m} (x^0) - x_{i_1}^0, \quad m = \overline{1, r}. \quad (13)$$

Если в последовательности (13) нет нулевых элементов, то из нее выбирается наибольший отрицательный элемент $a_{i_s j_s}^{q_1 s}$ и наименьший положительный $a_{i_p j_p}^{q_1 p}$. При этом могут возникнуть следующие ситуации:

- 1) оба элемента найдены, тогда должно выполняться $i_s = j_p$ и это будет решением алгоритма;
- 2) все элементы отрицательны, тогда i_s — решение;
- 3) все элементы положительны, тогда i_p — решение;
- 4) $r = 0$, тогда x^0 принадлежит пустой области, номер которой определяется однозначно.

Если в последовательности (13) имеются нулевые элементы, то все индексы i и j для них дадут решение алгоритма, т. е. номера областей, для которых точка x^0 является граничной.

Следует отметить, что описанный алгоритм — основа для последующих.

4.2. *Алгоритм определения предельной точки области по фиксированной точке x^0 и направлению β .* Осью L зададим вектором относительных приращений $\beta = (\beta_1, \beta_2, \dots, \beta_n)$; тогда $L = \{x: x_i = x_i^0 + \alpha \beta_i, i = \overline{1, n}, \alpha \in [0, \infty)\}$. Будем считать направление β вырожденным, если $\beta_{i_s} = 0, s = \overline{2, n}$. Идея этого алгоритма состоит в последовательном просмотре всех троек (i, j, q_i) с проверкой на пересечение соответствующих поверхностей с L . Предварительно по алгоритму 4.1 определим i^0 такое, что $x^0 \in \Omega_{i^0}$ и $\alpha_{кр} = +\infty$.

На первом шаге алгоритма проверяем вырожденность направления β . Если β вырождено, то из последовательности (13) сразу находим пре-

дельную точку и выходим из алгоритма. На втором шаге для невырожденного направления β определяем интервал $I_{ij}^{q_1}$ для очередной тройки (i, j, q_1) , который удовлетворяет соотношению

$$L_{l_1} \cap P_{ij}^{q_1} = L_{l_1}(i, j, q_1) = \{x: x_{i_s} = x_{i_s}^0 + \alpha \beta_{i_s}, s = \overline{2, n}, \alpha \in I_{ij}^{q_1}, x_{l_1} = 0\},$$

где L_{l_1} — проекция L на $E_{l_1}^{n-1}$. Если $I_{ij}^{q_1} = \emptyset$, то переходим в начало шага 2. Если $I_{ij}^{q_1} \neq \emptyset$, то вычисляем k равноотстоящих точек x^t , $t = \overline{1, k}$, с оси L таких, что $x_{i_s}^t = x_{i_s}^0 + \alpha^t \beta_{i_s}$, $s = \overline{1, n}$, $\alpha^t \in I_{ij}^{q_1}$. Величина k определяется погрешностью представления поверхностей и их топологическими свойствами. Если все $x_{i_s}^t$ принадлежат Ω_{i^0} , то переходим в начало шага 2. В противном случае для $x^t \notin \Omega_{i^0}$ проверяем: если $\alpha^t \leq \alpha_{кр}$, то $\alpha_{кр} = \alpha^t$, если $\alpha^t > \alpha_{кр}$, то $\alpha_{кр}$ остается без изменений. Если тройки (i, j, q_1) не исчерпаны, то переходим в начало шага 2. В противном случае для $\alpha_{кр}$ вычисляем $x_{кр} = x^0 + \alpha_{кр} \beta$. Если $\alpha_{кр} = +\infty$, то L принадлежит пустой области с предельной точкой в бесконечности. В противном случае нетрудно показать, что отрезок $\{x: x = x^0 + \alpha \beta, \alpha \in [0, \alpha_{кр}]\}$ в пределах точности принадлежит только двум областям Ω_{i^0} и Ω_{j^0} , $i^0 \neq j^0$. Поэтому каким-либо итерационным методом легко уточнить предельную точку области Ω_{i^0} в заданном направлении.

Существует еще одна версия описанного алгоритма. Основная идея в ней заключается в последовательном «шагании» вдоль оси L с последующим уточнением граничной точки с точностью ϵ . При этом приходится выполнить $N_1 \leq \frac{k}{\min_{(i,j,q_1)} (\alpha_{i_s}^2 - \alpha_{i_s}^1)} + m$ итераций. Под итерацией

понимается одноразовое выполнение алгоритма 4.1, и с использованием дихотомии $1/2^m < \epsilon$. Для исходной версии алгоритма 4.2 необходимо $N_2 \leq kN_0 + m$ итераций, где N_0 — количество троек (i, j, q_1) . Применительно к фазовым диаграммам $N_1 \leq 10^9$, $N_2 \leq 10^3$, что и определило выбор исходной версии алгоритма.

4.3. Алгоритм разрисовки $P - T$ -проекций и двумерных сечений ФД. Для этого алгоритма фиксируется m точек с каждой поверхности, имеющей непустое пересечение с двумерным сечением на ФД или с каждой линией на $P - T$ -проекции, и передают их для разрисовки на периферийное устройство ($C_1 \leq m \leq C_2$; величины C_1 и C_2 фиксированы и в процессе эксплуатации БДФД могут быть заменены). Как и в алгоритме 4.2, существует еще одна версия алгоритма 4.3, связанная с последовательным «шаганием» вдоль графика. Однако эта версия была отвергнута по той же причине, что и вторая версия алгоритма 4.2. Заметим, что точность представления графической информации визуально не различима для обеих версий алгоритма 4.3.

5. Структура программного обеспечения. Информационная система для ФД имеет структуру, представленную на рис. 2, где ПО — программное обеспечение; 1 — перфокарты с первичной информацией; 2 — программы обработки первичной информации, перевода ее во внутримашинное представление, выявления синтаксических ошибок

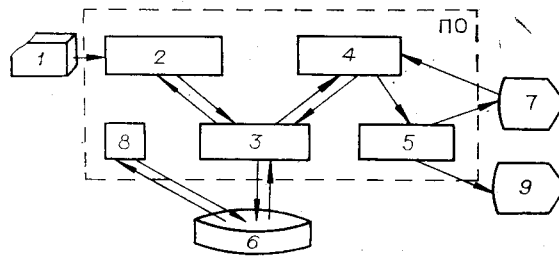


Рис. 2.

в 1, корректировки старых документов (к этой части относятся и пакеты по аппроксимации); 3 — программы чтения, записи, поиска и сортировки информации на диске; 4 — программы обработки языка запросов к БДФД (ЯЗБДФД); 5 — на-

кет прикладных программ (ППП); 6 — магнитный диск (МД); 7 — терминал пользователя; 8 — сервисные процедуры анализа, состояния файлов на МД и восстановления их структуры после машинных сбоев; 9 — графопостроитель и/или графический дисплей.

В разработанном программном обеспечении реализованы основные требования к базам данных и информационным системам [8]: независимость хранимой информации от программного обеспечения, несложный и неизбыточный язык запросов, отсутствие необходимости прикладному программисту знать физическую реализацию хранимых данных и т. п.

ЛИТЕРАТУРА

1. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики.— М.: Мир, 1976.
2. Гиллой В. Интерактивная машинная графика.— М.: Мир, 1981.
3. Фокс А., Пратт М. Вычислительная геометрия.— М.: Мир, 1982.
4. Дегтярев С. А., Воронин Г. Ф. Применение сплайнов в термодинамике растворов.— В кн.: Математические проблемы фазовых равновесий. Новосибирск: Наука, 1983.
5. Василенко В. А. Сплайн-функции: теория, алгоритмы, программы.— Новосибирск: Наука, 1983.
6. Лоран П. Ж. Аппроксимация и оптимизация.— М.: Мир, 1975.
7. Дробышев Ю. П., Зыкин С. В. Идентификация областей, ограниченных n -арными поверхностями.— В кн.: Системы и методы анализа данных. Новосибирск: ВЦ СО АН СССР, 1984.
8. Мартин Дж. Организация баз данных в вычислительных системах.— М.: Мир, 1980.

Поступила в редакцию 2 июля 1985 г.

УДК 681.3

К. И. КУЧЕРЕНКО, Е. Ф. ОЧИН

(Ленинград)

МУЛЬТИМИКРОПРОЦЕССОРНАЯ СИСТЕМА МЕДИАННОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ С ПОРАЗРЯДНЫМ ПОИСКОМ МЕДИАНЫ

Эффективным методом подавления импульсных помех является медианная фильтрация [1, 2]. Двумерная медианная фильтрация изображения $0 \leq D_{ij} < 1$, дискретизованного в виде матрицы размером $I \times J$ элементов и квантованного на 2^q уровней, определяется следующим образом:

$$D'_{ij} = \text{med}(D_{i-m, j-n}), \quad -M \leq m \leq M, \quad -N \leq n \leq N, \quad (1)$$

где оператор med обозначает некоторую процедуру поиска элемента последовательности $D_{i-m, j-n}$ длиной $L = (2M + 1)(2N + 1)$, для которого существует $(L - 1)/2$ элементов, меньших или равных ему по величине, и $(L - 1)/2$ элементов, больших или равных ему по величине.

Известны конвейерные алгоритмы медианной фильтрации [3, 4]. Время фильтрации изображения посредством данных алгоритмов не зависит от размеров окна фильтра, а затраты оборудования, необходимые для реализации алгоритмов, пропорциональны числу элементов локальной окрестности. Широко применяются гистограммные алгоритмы медианной фильтрации [2]. Поиск медианы локальной окрестности изображения сводится к определению локальной гистограммы

$$H_{ij}(q) = \sum_{m=-M}^M \sum_{n=-N}^N [D_{i-m, j-n} = q2^{-q}], \quad q \in \{0, 1, \dots, (2^q - 1)\}, \quad (2)$$

где
$$D_{ij} = \sum_{r=1}^Q d_{i,j,r} 2^{-r}, \quad d_{i,j,r} \in \{0, 1\},$$

$$[D_{ij} = q2^{-q}] = \begin{cases} 1, & \text{если } D_{ij} = q2^{-q}; \\ 0, & \text{если } D_{ij} \neq q2^{-q}; \end{cases}$$