Графові моделі. Комп'ютерне завдання

Святослав Хмелевський, ФІ-82

7 грудня 2021 р.

Завдання: Маємо множину $P = \{1, \ldots, h\} \times \{1, \ldots, w\}$ пікселів, відношення сусідства $\tau = \{\langle\langle i, j\rangle, \langle i, j+1\rangle\rangle : i \in \{1, \ldots, h\}, j \in \{1, \ldots, w-1\}\}$, множину $C \subseteq \mathbb{R}^3$ кольорів, скінченну множину $K = \{1, \ldots, m\}$ міток, скінченний набір $x = \langle x^1, \ldots, x^m\rangle$ зображень $x^k : P \to C,\ k \in K$ так скінченний набір $b = \langle b^1, \ldots, b^m\rangle$ масок $b^k : P \to \{0, 1\},\ k \in K$. Задача полягає у відшуканні такої розмітки $k^* : P \to K$, що для $q_p(k) = \alpha\dot{(}1 - b^k_p),\ \alpha \in \mathbb{R}_+ \cup \{0, +\infty\},\ g_{pp'}(k, k') = \beta\dot{(}|x^k_p - x^{k'}_p| + |x^k_{p'} - x^{k'}_{p'}|),\ \beta \in \mathbb{R}_+ \cup \{0, +\infty\}$ виконується:

$$\sum_{p \in P} q_p(k^*_p) + \sum_{pp' \in \tau} g_{pp'}(k^*_p, k^*_{p'}) = \min_{k : P \to K} \left(\sum_{p \in P} q_p(k_p) + \sum_{pp' \in \tau} g_{pp'}(k_p, k_{p'}) \right)$$

Розв'язок: $\tau \subseteq T^2$ складається з ланцюжків. Задача сформульована на напівкільці $\langle \mathbb{R}_+ \cup \{0, +\infty\}, \min, +\rangle$. Те, що дана алгебраїчна структура є напівкільцем було вже доведено в попередній частині цієї домашньої роботи. Алгоритм для роботи з довільним напівкільцем, коли $\tau \subseteq T^2$ складається з ланцюжків було представлено в основній частині даної домашньої роботи.

Весь алгоритм буде виглядати наступним чином.

1. Ініціалізація:
   - 1.1 На вхід подаються $q_i(k_i)$,
     $g_{i,i+1}(k_i, k_{i+1}),\ k_i, k_{i+1} \in K,\ i = \overline{1, n-1}$.
   - 1.2 $f_{n-1}(k_{n-1}) = \min\limits_{k_n \in K} (q_n(k_n) + g_{n-1,n}(k_{n-1}, k_n)),\ k_{n-1} \in K$.
2. Цикл $i = \overline{2, n-1}$:
   - 2.1 Обчислюємо $f_{n-i}(k_{n-i}) = \min\limits_{k_{n-i+1} \in K} (q_{n-i+1}(k_{n-i+1}) +$
     $+ g_{n-i,n-i+1}(k_{n-i}, k_{n-i+1}) + f_{n-i+1}(k_{n-i+1})),\ k_{n-i} \in K$.
3. Наступний етап:

3.1 Обчислюємо $G = \min\limits_{k_1 \in K}(q_1(k_1) + f_1(k_1))$.

3.2 Якщо немає сенсу в пошуку такого $k^* \in K^T$,
що $G = q_n(k^*_n) + \sum\limits_{i=1}^{n-1}(q_i(k^*_i) + g_{i,i+1}(k^*_i, k^*_{i+1}))$, то:
Робота алгоритму завершена.

3.3 Знаходимо таке $k^*_1 \in K$, що $G = q_1(k^*_1) + f_1(k^*_1)$.

4. Цикл $i = \overline{2, n-1}$:

4.1 Знаходимо таке $k^*_i \in K$, що $f_{i-1}(k^*_{i-1}) = q_i(k^*_i) +$
$+ g_{i-1,i}(k^*_{i-1}, k^*_i) + f_i(k^*_i)$.

5. Останній етап:

5.1 Знаходимо таке $k^*_n \in K$, що $f_{n-1}(k^*_{n-1}) =$
$= q_n(k^*_n) + g_{n-1,n}(k^*_{n-1}, k^*_n)$.

Під час виконання роботи використовувалися значення $\alpha = 90,\ \beta = 5$. З ними були отримані найкращі результати на тестових зображеннях.