

Анализ корреляционного метода в современных архитектурах в задаче выделения объектов на аэрофотоснимке

Иванов Александр

СПбГУ, Математико-механический факультет
Кафедра системного программирования

Научный руководитель: д.ф.-м.н., проф. Граничин О.Н.

Рецензент: к.ф.-м.н., Сысоев С.С.

Санкт-Петербург, 2014

БПЛА (Беспилотный летательный аппарат)

- Технические характеристики
- Мониторинг местности
- Способы обработки данных

Задача поиска объекта

- Распознавание образов
- Вычислительные архитектуры
 - NVidia CUDA (Compute Unified Device Architecture)
 - FPGA (Field-Programmable Gate Array)

Постановка задачи

Необходимо разработать программные средства, которые позволят эффективно осуществлять распознавание плоских объектов на аэрофотоснимках.

Задачи

- Изучить существующие подходы в области анализа 2D изображений
- Предложить решение для поиска объектов малых размеров
- Выполнить реализацию на CUDA, FPGA
- Апробировать реализованный подход на конкретных снимках, сделанных в ходе съемки БПЛА и провести сравнительный анализ по полученным данным

Распознавание образов

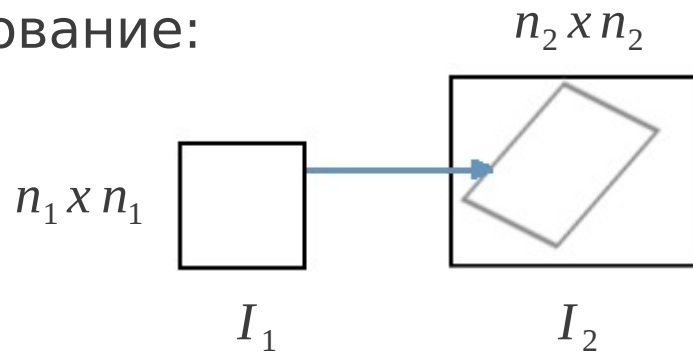
- CAD (Computer-aided diagnosis)
 - ROI (Region of Interest)
- Подобие форм
- Характерные особенности
- Корреляционный метод
 - SAD (Sum of absolute differences)

$$\text{SAD}(x, y) = \sum_{i=1}^{T_{rows}} \sum_{j=0}^{T_{cols}} \text{Diff}(x+i, y+j, i, j)$$

- Способы улучшения

Описание задачи

- Преобразование:



- Расстояние (SAD):
$$\Delta_T(I_1, I_2) = \frac{1}{n_1} \sum_{p \in I_1} |I_1(p) - I_2(T(p))|$$
- Минимальное в семействе Ψ :
$$\Delta(I_1, I_2) = \min_{T \in \Psi} \Delta_T(I_1, I_2)$$
- Цель для данного $\delta > 0$ найти преобразование T^* в Ψ :

$$|\Delta(I_1, I_2) - \Delta_{T^*}(I_1, I_2)| < \delta$$

Реализованный метод

- Гладкость: $v = \sum_{p \in I} \max_{q \in N(p)} |I(p) - I(q)|$
- Метрика: $l_{\infty}(T, T') = \max_{p \in I_1} \|T(p) - T'(p)\|_2$
- α -покрытие множества трансформаций, если $\forall T \exists T'_j$:
$$l_{\infty}(T, T'_j) = O(\alpha)$$

FAsT-Match: Fast Affine Template Matching

- Построить δn_1 -покрытие N_{δ} размера $\Theta\left(\frac{1}{\delta^6} \cdot \left(\frac{n_2}{n_1}\right)^2\right)$ и найти T^*
- Считаем SAD по разряженному шаблону из $\Theta(1/\epsilon^2)$ точек
- Уменьшаем δ и проряжаем новое покрытие методом ветвей и границ

CIRATEFI

1 из 2

- Для изображения B определим:

$$Cis_B(x, y, r) = \int_0^{2\pi} B(x+r \cos \theta, y+r \sin \theta) d\theta$$

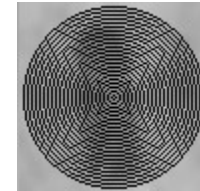
- Строим матрицы:

$$C_Q[i, k] = Cis_{Q_i}(x_0, y_0, r_k), \quad 0 \leq i < n, \quad 0 \leq k < l$$

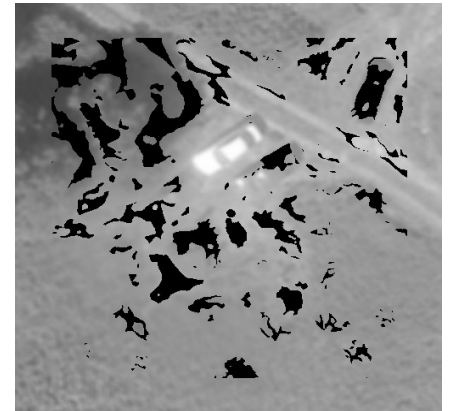
$$C_A[x, y, k] = Cis_A(x, y, r_k), \quad 0 \leq k < l$$

- Определяем начальный масштаб:

$$CisCorr_{A,Q}(x, y) = \max_{i \in [0, n-1]} |Corr(C_Q[i], C_A[x, y])|$$



Построение C_Q



Точки 1го уровня

CIRATEFI

2 из 2

- Для изображения B определим:

$$\text{Ras}_B^\lambda(x, y, \alpha) = \int_0^\lambda B(x+t \cos \alpha, y+t \sin \alpha) dt$$

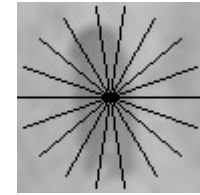
- Строим матрицы:

$$R_Q[j] = \text{Ras}_Q^{r_{i-1}}(x_0, y_0, \alpha_j), \quad 0 \leq j < m$$

$$R_A[x, y, j] = \text{Ras}_A^{s_i r_{i-1}}(x, y, \alpha_j),$$

- Определяем угол вращения:

$$\text{RasCorr}_{A,Q}(x, y) = \max_{j \in [0, m-1]} |\text{Corr}(R_A[x, y], \text{cshift}_j(R_Q))|$$



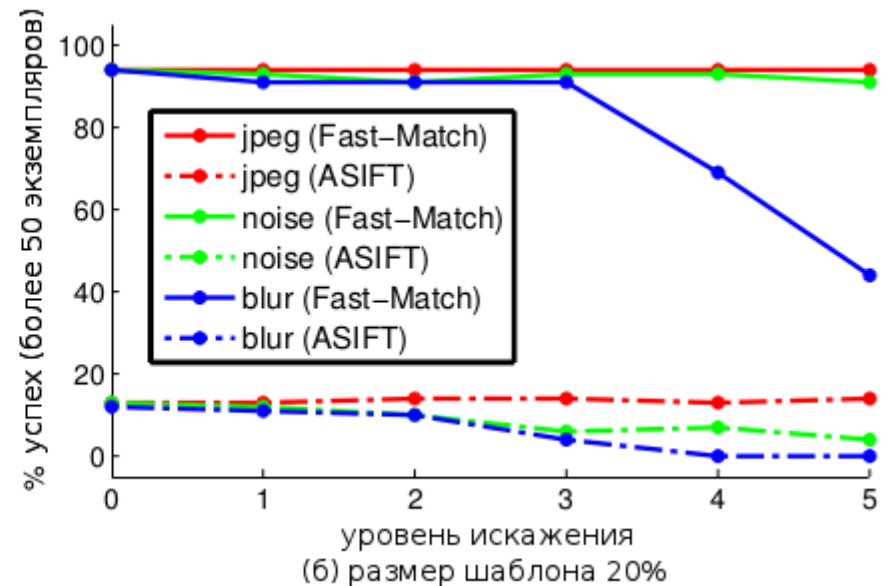
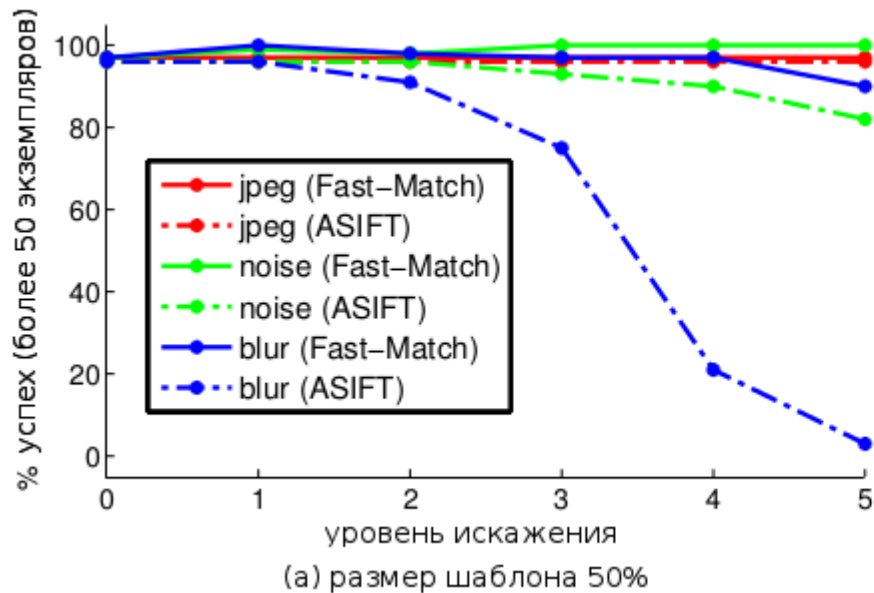
Построение R_Q



Результат Tefi

Сравнение результатов

FAsT-Match vs ASIFT



- Pascal VOC 2010 data-set

Размер шаблона	90%	70%	50%	30%	10%
сред. ош. Fast-Match SAD	5.5	4.8	4.4	4.3	4.8
сред. ош. оптим. SAD	4.1	4.1	4.0	4.4	6.1
сред. ош. перекрытие обл.	3.2%	3.3%	4.2%	5.3%	13.8%

Сравнение результатов

FAsT-Match vs Ciratefi

Эксперимент. 1

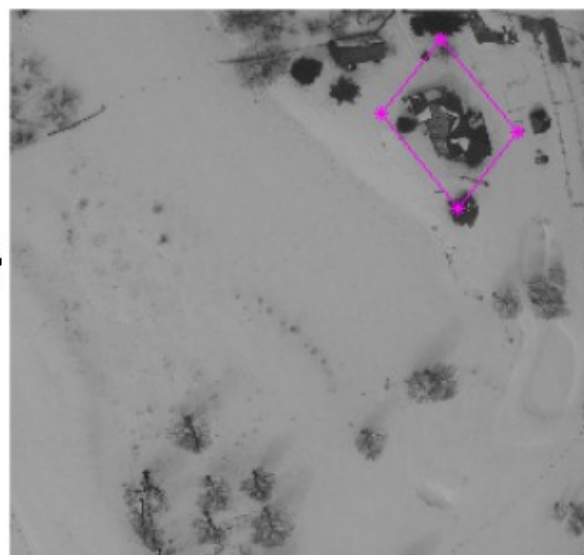
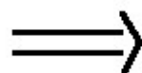
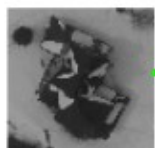
	10 приг. снимков	50%	20%	10%
Overlap. error FAsT-M	4.5%	7.1%	9.3%	14.2%
Overlap. error Ciratefi	5.2%	12.6%	12.1%	10.3%
Время FAsT-M CPU	54с.	38с.	47с.	52с.
Время Ciratefi CPU	27с.	37с.	31с.	26с.
Время FAsT-M GPU	6.3с.	2.1с.	5.4с.	6.5с.
Время Ciratefi GPU	2.5с.	3.8с.	3.2с.	2.7с.

Эксперимент. 2

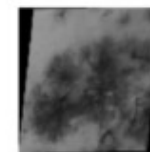
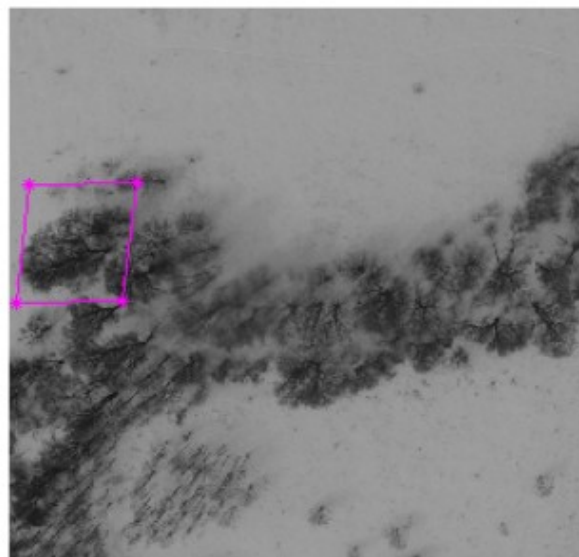
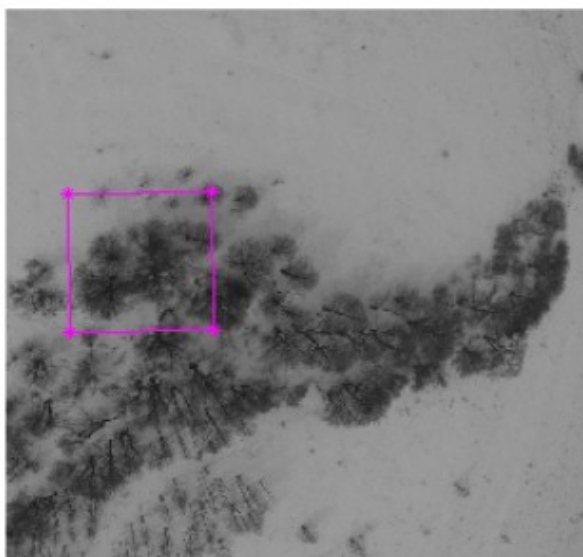
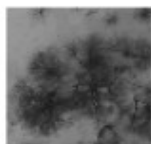
	1 колл.	2 колл.	3 колл.
Overlap. error FAsT-M	3.2%	3.6%	2.4%
Overlap. error Ciratefi	3.6%	4.4%	7.0%
Время FAsT-M CPU	48с.	44с.	49с.
Время Ciratefi CPU	26с.	31с.	29с.
Время FAsT-M GPU	5.9с.	2.5с.	5.8с.
Время Ciratefi GPU	2.3с.	3.3с.	3.6с.

Примеры работы алгоритма FAsT-Match

Шаблон
(дом)



Шаблон
(дерево)



Примеры работы алгоритма FAsT-Match

Шаблон
(машина)



Рис. 1

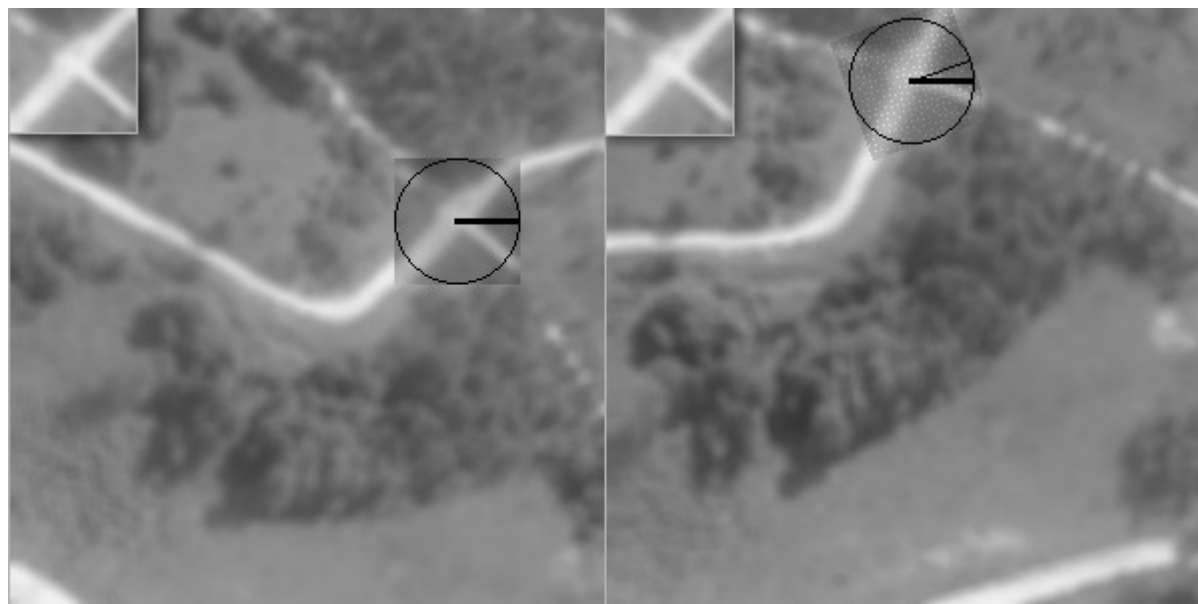


Рис. 2



Рис. 3

Примеры работы алгоритма Ciratefi



Перекресток



Машина

Заключение

- Произведен сравнительный анализ существующих подходов
- Выбраны для реализации два современных алгоритма FasT-Match и Ciratefi
- Апробация на тестовых наборах реальных аэрофотоснимков