

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-Механический факультет

Кафедра системного программирования

Применение алгоритмов SuperResolution к лицам

Курсовая работа студента 345 группы
Мокаева Руслана

Научный руководитель

Александр Пименов

Санкт-Петербург
2011

Оглавление

Введение	3
Постановка задачи	4
Метод SuperResolution	5
Генерация Обучающей последовательности	7
Преоброессинг Обучающей последовательности	9
Заклучение	11
Список используемой литературы	12

Введение

Чёткость – очень важный параметр в обработке сигналов. Более того, интерпретация изображения зависит от данных, которые можно из него извлечь. Чем более чёткое изображение мы имеем, тем больше информации можно получить. Но есть одна проблема: если мы хотим увеличить изображение в рамках имеющегося разрешения, то полученное изображение окажется нечётким. Более того, невозможно создать информацию, соответствующую деталям, которые нельзя увидеть в начальном изображении, и которые мы бы хотели видеть в увеличенном изображении.

Тем не менее, эту информации можно “угадать”. Это позволило бы повысить разрешение начального изображения.

Цель моей курсовой – исследование методов, позволяющих реализовать “угадывание” информации, и реализация одного из них.

Постановка задачи

Многим приложениям по обработке изображений необходимо иметь дело с чёткими изображениями. Здесь мы сталкиваемся с первой проблемой – а что означает “чёткое изображение”? Можно сказать, что это изображение, которое выглядит натурально, как в реальной жизни. Но человеческие ощущения сложно вычислить, поэтому мы скажем, что четкое изображение – это изображение в котором присутствуют частоты всех диапазонов. Напротив, нечёткое изображение – то, в котором отсутствуют высокие частоты. Изображение, полученное в результате сканирования, фотографирования во время движения, оказывается размытым. Даже если просто увеличить размер изображения, то необходимая интерполяция вызовет смазывание.

Первый решение – представлять изображение в виде многогранника. Изображение, полученное из такого представления, всегда будет иметь чёткие грани. Более того, оно не зависит от разрешения: такое изображение можно увеличивать, но грани всё равно останутся чёткими (ограничение состоит только в размерах граней). Но реальные изображения часто слишком сложны для такого представления.

Другое решение – понять, каким образом увеличивать резкость. А именно – как её угадывать. Именно этот метод и называется SuperResolution.

Метод состоит из создания четкой версии начального изображения на основе заранее подготовленной базы данных, состоящей из высокочастотных изображений. Мой метод основан на работе Фримана[1], в которой он описывает метод угадывания высоких частот высокого диапазона. В ней описано два алгоритма. Первый использует модель Сети Маркова. Но я реализую второй, Однопроходный алгоритм, который оказывается более быстрым и эффективным.

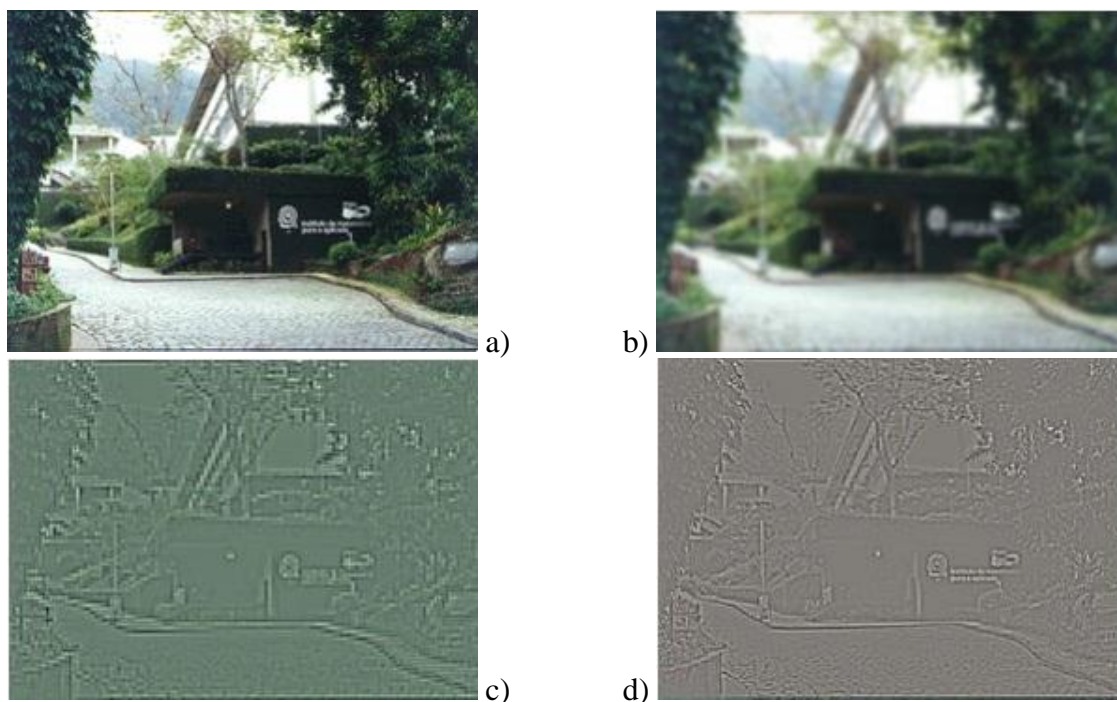
Метод SuperResolution

Главная идея состоит в использовании нескольких изображений в качестве вспомогательного множества с целью научиться повышать чёткость. Это вспомогательное множество называется Обучающая последовательность (англ. Training Set). Качество полученного изображения напрямую зависит от Обучающей последовательности, от её размера и качества.

Она состоит из чётких изображений, содержащих частоты всех диапазонов. Для начала, нам надо увеличить масштаб изображения (например, в 2 раза). При этом происходит интерполяция отсутствующих пикселей. В результате получается изображение без частот высокого диапазона.

В этот момент может показаться, что для достижения приемлемого результата нам понадобится большая Обучающая последовательность. А его обработка и вычисление может оказаться очень долгим. Таким образом, мы допустим некоторые свойства т.н. ”приемлемых изображений”, чтобы повысить эффективность Обучающей последовательности, состоящей из разумного числа элементов.

Для начала, предполагая, что мы работаем с тремя диапазонами частот (это достигается использованием пирамидальной декомпозиции) скажем, что диапазон высоких частот условно отделён от диапазона низких частот. Это позволяет не рассматривать многообразие низких частот. Затем мы предполагаем, что взаимоотношения между высокими и средними частотами независимы от локального контраста изображения.



a) Начально изображение, b) Низкие частоты,
c) Средние частоты, d) Высокие частоты
c) и d) не содержат значений от 0 до 255, поэтому для того,
чтобы их увидеть, нужно нормализовать изображение

Таким образом, нормализуя контраст в каждом изображении, мы понижаем их изменчивость, тем самым повышая эффективность Обучающей последовательности.

SuperResolution можно рассмотреть как два независимых шага: первый состоит в обработке Обучающей последовательности, чтобы на втором шаге конструировать высокие частоты, которые отсутствуют в увеличенном изображении.

Генерация Обучающей последовательности

Этот шаг является решающим для качества получаемого изображения, потому что оно создает взаимоотношения между высокими частотами и средними частотами.

Мы рассмотрим локальные области, или патчи, в каждом из диапазонов. Каждый низкочастотный патч связан с высокочастотным патчем. У них совпадают центры, но их размеры не обязательно одинаковы. Сам по себе патч не содержит достаточно информации. Для него мы можем выбрать большое число похожих низко-частотных патчей из Обучающей последовательности. Фриман показал, что соответствующие высокочастотные патчи сильно отличаются. И затем, выбор ближайшего соседнего низкочастотного патча для построения высоких частот увеличенного изображения приведёт к плохой оценке реальных высоких частот.

Однопроходный алгоритм

Основную концепцию метода, который я реализовал, легко понять: однопроходный алгоритм последовательно генерирует недостающие высокие частоты в увеличенном изображении. Для начала, мы сгладим начальное изображение (реализованным в OpenCV методом `cvSmooth()`), а потом разобьем изображение на патчи размера 5x5 пикселей.

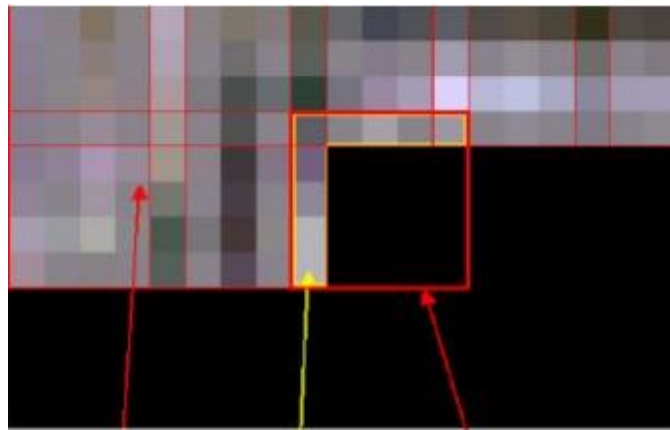
Затем мы по порядку обходим все патчи, на каждом шаге добавляя в изображение наиболее подходящий высокочастотный патч. В итоге получается чёткое изображение, содержащее частоты всех диапазонов.

Выбор подходящего шага на каждом шаге является очень важной частью. Но у нас имеется два ограничения:

- Частотное ограничение: высокочастотный патч должен соответствовать низкочастотному патчу из Обучающей последовательности, который должен быть близок (по Евклидовой норме) рассматриваемому низкочастотному патчу.
- Пространственное: мы хотим обеспечить непрерывность при построении высоких частот, что означает, что новый высокочастотный патч должен быть связан с предыдущим добавленным в изображение.

Первое условие легко выполнить: низкочастотному патчу из увеличенного изображения мы ищем наиболее близкое совпадение в Обучающей последовательности, состоящее из пары (низкочастотный патч, высокочастотный патч).

Для второго ограничения потребуем, чтобы на каждом шаге добавляемый высокочастотный патч пересекался с пикселями, уже содержащим высокие частоты.

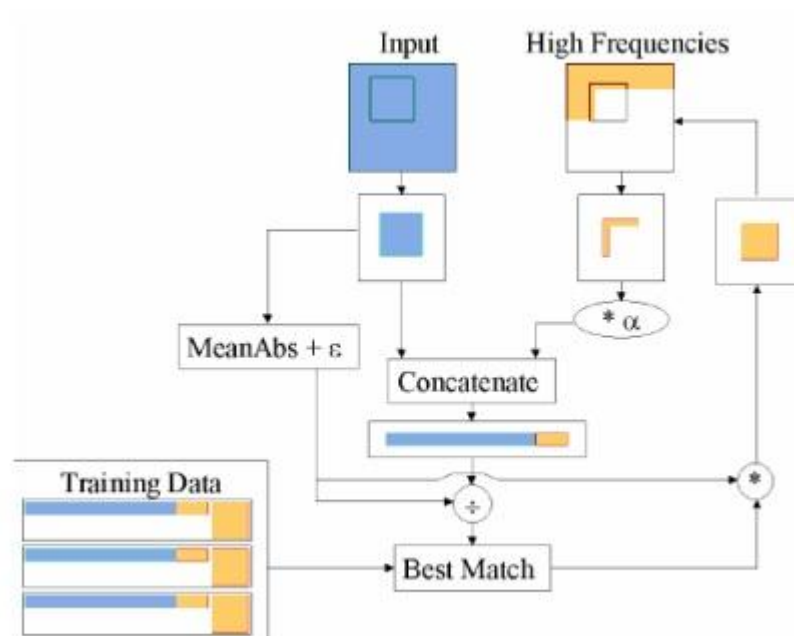


a) b) c)

- a) Уже вычисленные высокие частоты
- b) Пересечение патча с уже вычисленной областью
- c) Следующий патч, который необходимо вычислить

Мы добавляем эту информацию и используем её для выбора хорошей пары во время поиска в Обучающей последовательности. Для того чтобы понять важность этих ограничений, введём параметр α . В моих вычислениях он будет равен 0,5.

Обучающая последовательность состоит из поискового вектора и высокочастотного патча, поисковой вектор состоит из низкочастотного патча и “уголка” высокочастотного патча:



Обработка патча на каждом шаге.
Синий цвет – низкие частоты, оранжевый - высокие

Преобработка обучающей последовательности

Перед тем, как учитывать описанные выше ограничения, необходимо обработать изображения, составляющие Обучающую последовательность.

1. Для начала уберём все высокие частоты (т.е. создадим низкочастотную версию изображения). Это сделаем следующим образом:
 - Сгладим изображение (методом `cvSmooth()` из библиотеки `OpenCV`);
 - Уменьшим вдвое масштаб (методом `cvResize()`);
 - Увеличим вдвое масштаб;
2. Различие будет состоять как раз из высоких частот.
3. После этого убираем низкие частоты из полученного изображения (я реализовал это с помощью метода `cvSobel()`), тем самым останутся только средние частоты.
4. Полученные изображения разбиваем на патчи. Мы получим высокочастотные патчи (размера 5×5) и низкочастотные патчи (7×7).

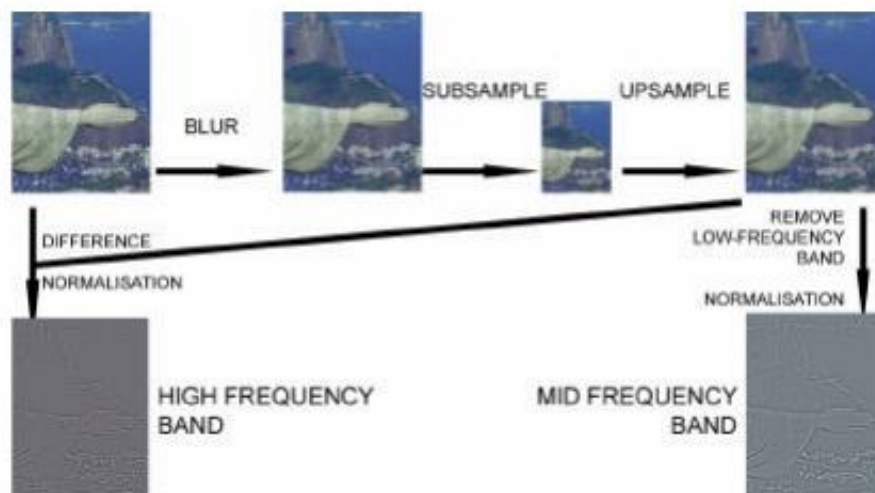
Таким образом, Обучающая последовательность состоит из пар (высокочастотный патч 5×5 , низкочастотный патч 7×7).

Затем необходимо нормализовать контраст. Это может быть реализовано с помощью разделения диапазонов по энергии изображения, которая описывает среднее абсолютное значение изображения в каждой его точке. Вычисления производятся по формуле

$$energy = 0,01 + \sqrt{\sum |m_i|^2}$$

где m_i – значение i -го пикселя низкочастотного патча. $0,01$ добавилось для предотвращения деления на ноль.

Но в итоге я решил провести это процесс локально: вместо нормализации всего изображения, я нормализовал каждый патч, разделяя высокочастотные и низкочастотные патчи по локальной энергии. После процесса подбора проводится обратный к нормализации процесс, в результате чего получается высокочастотный патч.



Шаги препроцессинга

Подбор патчей

Процесс подбора патчей использует Евклидову норму, модифицированную параметром α . Если запишем поисковой вектор из Обучающей последовательности как $V = (V_l, V_h)$, где V_l – вектор, содержащий низкочастотные данные, V_h – высокочастотные (данные не обо всем патче, а только об уголке). Тогда

$$d(V, W)^2 = |V_l - W_l|^2 + \alpha |V_h - W_h|^2$$

Процесс поиска

Для начала я реализовал прямое вычисление. По данному вектору, я обходил всю Обучающую последовательность и сравнивал данный вектор с каждым. Количество патчей и размер вектора делают этот процесс очень долгим, так как Обучающая последовательность обычно состоит из 100000 – 500000 патчей, а размер поискового вектора составляет 174 (3 на каждый пиксель (RGB), всего пикселей 58 (49 из низкочастотного патча, 9 – из высоко частотного уголка)). Поэтому я попытался как-нибудь ускорить этот процесс, но об это в Заключение.

Влияние параметра α

Поскольку параметр α участвует в норме, по которой мы сравниваем патчи, то становится понятно, что от него много зависит при поиске наиболее близкого патча. При $\alpha = 0$ “уголок” не играет никакой роли, поэтому увеличенное изображение получалось с искажениями (хоть и не очень значительными). Понятно, что с ростом α возрастает роль этого самого “уголка”. Из моих результатов я понял, что оптимальное значение $\alpha = 0,5$.

Заключение

Область, в которой я проводил исследование, изучается уже довольно долгое время, поэтому сказать что-то новое – довольно сложно, все фундаментальные работы уже написаны.

Целью курсовой являлось ознакомление с методами повышения чёткости изображения, да и в целом – с принципами и методами решения задач компьютерного зрения.

В процессе реализации алгоритма мной была частично изучена библиотека OpenCV. Я ознакомился со многими способами представления и обработки изображения.

В моей работе ещё много надо дополнить, например, выбор Обучающей последовательности, количество, качество, размер изображений. Но в первую очередь нужно оптимизировать алгоритм поиска наиболее близкого патча, потому что по большому счёту нет необходимости всё время обрабатывать Обучающую последовательность, есть смысл один раз её обработать, а затем хранить в виде какой-либо структуры с удобным и быстрым обходом (бинарное дерево, например). В статье, на которой я основывал изучение вопроса, автор предложил метод (TrainingSetVectorQuantification), который, после обработки Обучающей последовательности, позволял бы искать похожий патч не во всей ОП, а только в небольшой её части (*codebook*), в которой содержатся векторы из ОП, наиболее близкие по норме друг к другу. К сожалению, у меня не хватило времени реализовать этот метод.

Список использованной литературы

- [1] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):24-47, 2000.
- [2] Régis DESTOBBELEIRE . Super-Resolution , 2002
- [3] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. MERL, TR-2001-30.
- [4] Jianbo Shi and Carlo Tomasi, "Good features to track", *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 593-600, 1994.
- [5] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, volume 1, pages 275-282, 2004.
- [6] S. Y. Dai, M. Han, W. Xu, Y. Wu, and Y. H. Gong. Soft edge smoothness prior for alpha channel super resolution. In *CVPR*, 2007.
- [7] Niranjan Damera-Venkata, Nelson L. Chang . Realizing Super-Resolution with Superimposed Projection
- [8] Frederick W. Wheeler, Xiaoming Liu and Peter H. Tu. Multi-Frame Super-Resolution for Face Recognition