

Bài tập lớn 1: Image Filtering and Hybrid Images

Cấu trúc

- Mẫu báo cáo: trong thư mục: writeup/. Bạn cần file hoàn tất phần báo cáo sử dụng mẫu LaTeX cho sẵn và nộp lại bạn PDF tương ứng.
- Dữ liệu: trong thư mục data/. Thư mục này bao gồm 5 cặp ảnh để các bạn thử nghiệm giải thuật sinh ảnh lai.
- Phần code cho sẵn: trong thư mục code/. Thư mục này bao gồm file các file: `student.py` (chứa phần hiện thực mà các bạn cần phải thực hiện và nộp lại), `helpers.py`, `proj1_part1.py`, `proj1_part2.py` (chứa các file giúp bạn debug quá trình hiện thực file `student.py`).
- Các file cần phải nộp: `student.py` và `writeup.pdf`. Các bạn submit 2 file này lên hệ thống BKeL. **Lưu ý không thay đổi tên file, không được nén file.**
- Hạn chót: Thứ Hai 04/05/2020, 23:59.

1. Giới thiệu

Ở bài tập lớn này, các bạn sẽ hiện thực hàm tính tích chập (convolution) và sử dụng nó để sinh ra ảnh lai (hybrid). Kỹ thuật sinh ảnh lai này được phát minh bởi nhóm nhà khoa học Oliva, Torralba, và Schyns năm 2006, và được đăng trên hội nghị SIGGRAPH (một trong những hội nghị rất uy tín trong lĩnh vực xử lý ảnh. Bằng cách pha trộn giữa tần số cao và thấp của 2 ảnh khác nhau, chúng ta có thể tạo ra ảnh lai giữa 2 ảnh ban đầu.

2. Lọc ảnh

Đây là công cụ cơ bản trong xử lý ảnh. Các bạn có thể tham khảo Chương 3.2 trong giáo trình của Szeliski và bài giảng về lọc ảnh, đặc biệt là bộ lọc tuyến tính. Hầu hết các thư viện xử lý ảnh và thị giác máy tính đều cung cấp hàm để lọc ảnh. Tuy nhiên ở bài tập lớn này, các bạn được yêu cầu phải viết lại từ đầu, kể cả hàm tính tích chập (convolution)

3. Ảnh lai

Một ảnh lai (hybrid) là tổng của ảnh thứ nhất lọc bởi bộ lọc tần số thấp (low-pass filter) và ảnh thứ hai lọc bởi bộ lọc tần số cao (high-pass filter). Các bạn phải điều chỉnh tham số này cho mỗi cặp ảnh để kiểm soát lượng tần số cao bị loại bỏ ở ảnh thứ nhất và lượng tần số thấp bị loại bỏ ở ảnh thứ 2. Tham số này được gọi là “cut-off frequency”.

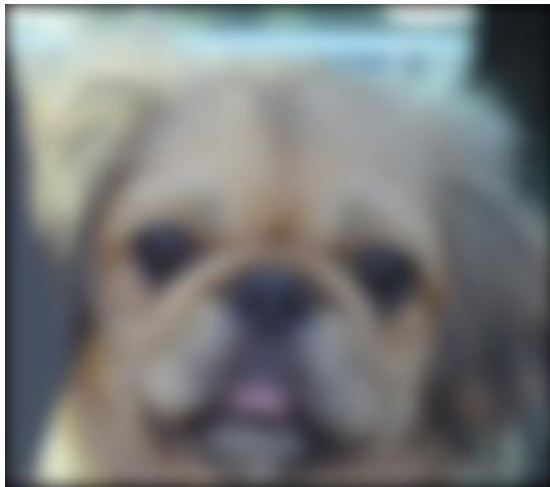
Bài tập lớn này cung cấp 5 cặp ảnh đã được canh chỉnh để có thể lai ghép với nhau một cách hợp lý. Việc canh chỉnh này là cực kỳ quan trọng bởi vì nó sẽ tác động đến (tham khảo thêm trong bài báo để nắm rõ chi tiết). Ngoài ra, các em có thể (không bắt buộc)

tạo thêm các ví dụ khác như: thay đổi biểu cảm trên khuôn mặt, hoặc khuôn mặt thay đổi theo thời gian.

Ví dụ: cho hai ảnh đầu vào như bên dưới:



Phiên bản low-pass và high-pass của hai ảnh này như sau:



Ảnh tần số cao thường có giá trị trung bình là 0 và chứa các giá trị âm nên để minh họa bằng hình ảnh thông thường sẽ được cộng thêm 0.5. Trên hình, các vị trí sáng là giá trị dương và vị trí tối là giá trị âm.

Ghép tần số cao và thấp lại với nhau sẽ tạo ra ảnh lai như bên dưới. Nếu bạn gặp khó khăn trong việc quan sát ảnh lai kết quả này, một hiệu ứng có khá là hữu ích là tạo ra các ảnh có độ phân giải thấp hơn của ảnh lai này:



Phần code cho sẵn đã cung cấp hàm `vis_hybrid_image()` trong `helpers.py` để lưu và hiển thị hiệu ứng này.

4. Yêu cầu và cách chấm điểm

- +6 điểm: Hiện thực toán tử tích chập (convolution) trong file `student.py` (hàm `my_imfilter()`). Giải thuật của các bạn cần:
 1. Chèn ảnh đầu vào sử dụng giá trị 0.
 2. Hỗ trợ cả ảnh xám và ảnh màu.
 3. Hỗ trợ bộ lọc kích thước lẻ tùy ý (ví dụ 3x3, 3x7, 7x9).
 4. Thông báo lỗi (raise an **Exception**) khi bộ lọc có kích thước chẵn (ví dụ 3x4, 4x6).
 5. Trả về ảnh đầu ra có cùng độ phân giải (kích thước) với ảnh đầu vào.
 - File `proj1_part1.py` được cung cấp để giúp các bạn debug giải thuật lọc ảnh.
- +3 điểm: Hiện thực giải thuật tạo ảnh lai trong file `student.py` (hàm `gen_hybrid_image()`)
 - File `proj1_part2.py` được cung cấp để giúp các bạn tạo các ảnh lai.
- +1 điểm: Viết báo cáo.
 - Mẫu có sẵn trong thư mục `writeup/`. Vui lòng mô tả giải thuật bạn sử dụng và đính kèm kết quả trong báo cáo này. Mẫu LaTeX đã được cung cấp sẵn. Các bạn dịch file LaTeX thành PDF và đặt vào trong thư mục `writeup/`, sau đó nộp cùng với code mà bạn hiện thực.
- +1 điểm: Điểm cộng (Bonus).
 - Tối đa 1 điểm: Thực hiện tích chập dựa trên biến đổi Fourier nhanh (Fast Fourier Transform – FFT). Các bạn có thể sử dụng các hàm có sẵn cho biến đổi Fourier. Phần hiện thực này nên nằm trong hàm `my_imfilter_fft()` trong file `student.py`. Lưu ý bạn vẫn phải thực hiện hàm `my_imfilter()` để lọc ảnh trong miền không gian vì đây là phần bắt buộc.

Phần hiện thực hàm `my_imfilter()` cần phải chạy hoàn tất trong vòng 01 (một) phút trên cặp ảnh chó và mèo. Nếu toàn bộ phần hiện thực không hoàn tất trong vòng 10 (mười) phút, bạn sẽ nhận điểm 0 (zero) cho bài tập lớn này.

Các hàm không được phép sử dụng: tất cả các hàm liên quan đến bộ lọc, tích chập (convolution) hay tương quan (correlation) đều KHÔNG được phép sử dụng.

Các hàm có thể có ích: các toán tử cơ bản trong Numpy như: cộng `numpy.add()` (hay đơn giản sử dụng `+`), nhân từng phần tử `numpy.multiply()` (hay đơn giản là sử dụng `*`), tính tổng `numpy.sum()`, lật ma trận `numpy.flip()`, cắt tỉa `numpy.clip()`, chèn thêm `numpy.pad()`, xoay `numpy.rot90()`, ...

5. Lời cảm ơn

Bài tập lớn này sử dụng lại bài tập lớn của GS. James Tompkin trong khóa học [CSCI 1430: Introduction to Computer Vision](#). Bài tập lớn này ban đầu được phát triển bởi GS. James Hays dựa theo phần hiện thực của TS. Derek Hoiem. Hiện thực này được chuyển sang ngôn ngữ Python bởi Seungchan Kim and Yuanning Hu.