

Template matching

Cho ba bức ảnh sau: `barton2.jpg`, `barton2-small.jpg` và `barton2-window.jpg`. Hãy thực hiện các yêu cầu sau đây:

Bài tập 1. Viết hàm `correlation(img1: np.ndarray, img2: np.ndarray)` thực hiện các thao tác sau:

- Chuyển ảnh `img1` và `img2` về dạng vectơ
- Tính hệ số tương quan chéo chuẩn hóa của `img1` và `img2` theo công thức sau:

$$\frac{1}{n} \sum_{x,y} \frac{1}{\sigma_1 \sigma_2} (f(x,y) - \mu_f)(g(x,y) - \mu_g) \quad (1)$$

với n là số pixel của hai ảnh `img1` và `img2`. $f(x,y)$, μ_f và σ_f lần lượt là giá trị điểm ảnh tại (x,y) , giá trị trung bình và độ lệch chuẩn của ảnh `img1`. Tương tự, $g(x,y)$, μ_g và σ_g lần lượt là giá trị điểm ảnh tại (x,y) , giá trị trung bình và độ lệch chuẩn của ảnh `img2`

Bài tập 2. Viết hàm `get_correlation_map(img: np.ndarray, template: np.ndarray)` tính toán giá trị tương quan chéo của từng miếng vá trong ảnh `img` có kích thước bằng ảnh `template`. Lưu ý: không cần padding ảnh `img`.

Bài tập 3. Viết hàm `find_matched(image: np.ndarray, corr_map: np.ndarray, template_size: tuple[int, int], threshold: float)` thực hiện tìm các vị trí ảnh có tương quan lớn hơn ngưỡng `threshold` và đánh dấu lại trên ảnh `image`, bằng cách vẽ một hình chữ nhật tại vị trí đó (kích thước của hình chữ nhật là `template_size`)

Bài tập 4. Viết hàm `main()` thực hiện đọc ảnh `barton2-small.jpg` và `barton2-window.jpg`, chuyển ảnh về thang xám và sử dụng hai ảnh xám trên để đánh dấu các vùng trên ảnh `barton2-small.jpg` có hình dạng giống với ảnh `barton2-window.jpg` (`barton2-window.jpg` là ảnh template). Làm tương tự như vậy với ảnh `barton2.jpg`