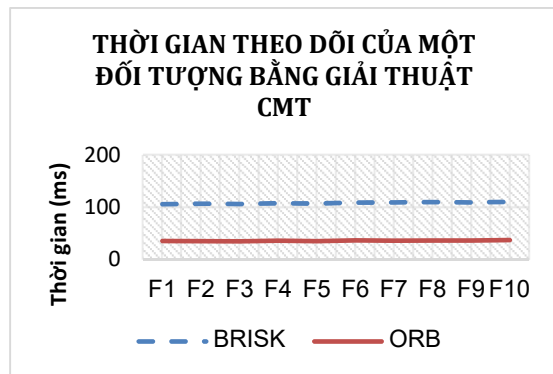
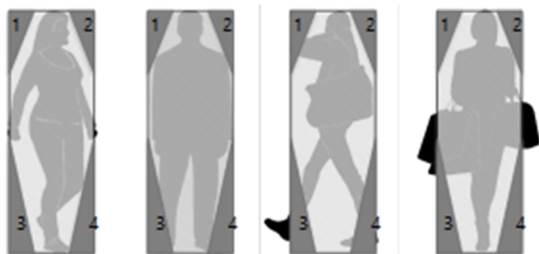


**Biểu đồ 1:** So sánh về thời gian khởi tạo của giải thuật CMT giữa hai bộ mô tả BRISK và ORB dựa trên số lượng các CMT được tạo ra

Ngoài ra, số lượng keypoint thuộc đối tượng giảm dần cho giải thuật theo dõi chạy nhanh hơn ở một số bước như theo dõi (tracking) ở bước 2, so khớp cục bộ (local matching) ở bước 7 (Hình 6).



**Biểu đồ 2:** So sánh thời gian theo dõi một đối tượng của giải thuật CMT từ khung hình thứ nhất đến khung hình thứ 10



**Hình 7:** Mặt nạ đa giác biểu diễn dáng người

### 3.4.3 Xử lý bằng kỹ thuật đa luồng

Các bước xử lý của giải thuật CMT thích nghi như mô tả trong Hình 6 đều tuần tự và đa số phụ thuộc nhau (ngoại trừ bước 2, 3 và 6, 7 có thể thực hiện song song). Chúng tôi đề xuất một giải thuật mới cải tiến tốc độ xử lý việc theo dõi bằng cách sử dụng

dùng kỹ thuật đa luồng (multithread) trên các hệ thống đa nhân (multicores).

Chúng tôi chia các bước xử lý trong Hình 6 thành 6 giai đoạn (stages): (1), (2), (3), (4, 5), (6) và (7, 8). Mỗi giai đoạn được thực thi trong một luồng (thread) khác nhau. Giải thuật hoạt động như sau: đầu tiên, thread 1 tính toán các đặc trưng cục bộ và lưu kết quả vào biến toàn cục (**A**). Thread 2 thực hiện truy vết các đặc trưng được tính ở Thread1 bằng phương pháp luồng quang học và lưu kết quả vào trong biến (**B**). Thread 3 chờ cho đến khi thread 1 hoàn thành là nó có thể bắt đầu thực hiện việc so khớp toàn cục. Kết quả của việc so khớp toàn cục sẽ được đặt vào biến lưu trữ tương ứng (**C**). Thread 4 chờ cho đến khi thread 2 và thread 3 thực hiện xong nó sẽ hợp các điểm đặc trưng lại (được lưu trữ trong biến **B** và **C**) và tính toán ước lượng tỷ lệ, góc quay, tâm đối tượng đưa vào biến lưu trữ (**D**). Thread 5 chờ cho thread 4 thực hiện xong nó sẽ tìm cluster lớn nhất theo phương pháp consensus từ thông tin lưu trữ trong **D**. Thread 6 có thể bắt đầu cùng lúc với Thread 5 sau khi thread 4 hoàn thành, thread này sẽ thực hiện so khớp cục bộ và hợp nhất các đặc trưng để cho ra kết quả sau cùng (Hình 8).

## 4 KẾT QUẢ THỰC NGHIỆM

Chúng tôi đã thực hiện cài đặt hệ thống bằng ngôn ngữ C++ sử dụng thư viện OpenCV và thư viện libCMT trên nền Qt. Chúng tôi đánh giá việc cải tiến tốc độ xử lý theo phương pháp đa luồng trên tập dữ liệu thu thập từ camera an ninh ghi lại diễn biến thực tế tại siêu thị.

### 4.1 Kết quả xây dựng mô hình nhận dạng người và hành vi

#### 4.1.1 Xây dựng mô hình nhận dạng người sử dụng phương pháp cascade classifier

Tạo tập dữ liệu để xây mô hình cho phương pháp cascade classifier có giai đoạn quan trọng là xây dựng tập ảnh chứa đối tượng (tập positive) và tập chứa ảnh nền (tập negative). Ở bước xây dựng tập ảnh chứa đối tượng, tôi sử dụng công cụ ImageClipper của tác giả Naotoshi Seo (, 2014). Một tập dữ liệu với 18.357 file là ảnh người và tập dữ liệu với 44.648 ảnh nền đã được xây dựng. Sau khi xây dựng tập dữ liệu tiến hành xây dựng mô hình. Kết quả đánh giá mô hình trên tập dữ liệu (rút trích từ video thực tế) được thể hiện ở Bảng 1.

Với tập ảnh 1 và tập ảnh 2 được rút ra ngẫu nhiên từ đoạn 2 đoạn video dài 10 phút ghi lại hình ảnh khách hàng tại siêu thị, và dữ liệu này không nằm trong tập dữ liệu huấn luyện.