



Przedmiot: Sztuczna inteligencja i sensoryka.

PR063KAM3

Temat projektu: Detekcja i zliczanie osób metodą wizyjną.

Spis treści:

1.	ABSTRAKT.....	2
2.	WSTĘP.....	2
3.	KONCEPCJA PROPONOWANEGO ROZWIĄZANIA.....	5
4.	REZULTATY I WNIOSKI.....	6
5.	PODSUMOWANIE	14
6.	LITERATURA	15
7.	DODATEK A: OPIS OPRACOWANYCH NARZĘDZI I METODY POSTĘPOWANIA	15
8.	DODATEK B: REALIZACJA PROPONOWANEGO ROZWIĄZANIA.....	15
9.	DODATEK C. OPIS INFORMATYCZNY PROCEDUR.....	27
10.	DODATEK D. SPIS ZAWARTOŚCI DOŁĄCZONEGO NOŚNIKA (CD ROM).....	38

Wykonali:

Grzegorz Caban

Paweł Gutowski

III rok AiR

konsultant: dr inż. Zbigniew Mikrut

Wersja 1.1

Kraków, luty 2007.

1. Abstrakt

Celem tego projektu jest stworzenie aplikacji umożliwiającej śledzenie i zliczanie osób na filmach wideo.

Aby zrealizować to zadanie posłużyliśmy się stosunkowo prostymi algorytmami: odejmowania kolejnych klatek oraz odejmowania tła, następnie po odpowiednim etapie przetwarzania wstępnego śledzimy wykryte obiekty i na podstawie ich kształtu, wielkości oraz zachowania oceniamy czy i ile osób w rzeczywistości reprezentują.

Napisana przez nas aplikacja z dużą skutecznością wykrywa i śledzi pojedyncze osoby przechodzące przez kadr kamery. Problemатyczne bywa rozróżnianie kilku idących blisko siebie osób, np. obejmującej się pary. W celu rozróżnienia czy analizowany obiekt jest jedną osobą czy kilkoma analizowaliśmy współczynniki kształtu oraz pole obiektu.

Istotną cechą jest duża wrażliwość algorytmu na warunki, w których powstało nagranie.

2. Wstęp

Śledzenie osób na zdjęciach wideo, na przykład na zdjęciach pochodzących z nagrań kamer ochrony, może nieść ze sobą wiele cennych informacji, żeby wymienić tylko zastosowania w systemach bezpieczeństwa, logistyce czy biznesie. Oczywiście aby móc zabierać tego typu dane na odpowiednią skalę oraz z właściwą dokładnością konieczne wydaje się zautomatyzowanie tego procesu.

a. Cele i założenia projektu

Zadaniem stawianym przed opisywaną tutaj aplikacją jest wykrywanie i liczenie osób przechodzących przed kamerą na filmach nagranych na uczelnianym korytarzu oraz sfilmowanym chodniku na miasteczku studenckim, oraz zdjęciach nagranych kamerą filmującą skrzyżowanie ul. Czarnowiejskiej z Al. Mickiewicza.

Program musi być w stanie wykryć osobę pojawiającą się w kadrze, a następnie śledzić ją na kolejnych klatkach filmu, niezależnie od kierunku jej ruchu i tego, czy cały czas jest w ruchu czy też się zatrzymuje, aż do chwili gdy zniknie ona z pola widzenia kamery. W chwili zniknięcia zwiększa licznik ilości osób, które przeszły przez badany obszar. Bardzo pomocny dla naszego projektu był raport z bardzo podobnego projektu [3], zadaniem którego było policzenie osób wychodzących z budynku w trakcie ćwiczeń pożarowych. Film nagrano z kamer ochrony. Zastosowano w tym projekcie metodę odejmowania tła.

b. Zarys ogólny proponowanego rozwiązania.

Do sformułowanego powyżej problemu można podchodzić na wiele sposobów¹, my zdecydowaliśmy się na stosunkowo proste metody odejmowania tła oraz odejmowania dwóch kolejnych klatek.

Pierwszy polega na tym, że od każdej klatki filmu jest odejmowane tło i w ten sposób jesteśmy w stanie wykryć czy w kadrze pojawił się jakiś obiekt do tła nie należący, czyli przypuszczalnie szukana osoba. Posługujemy się również różnicą dwóch kolejnych klatek, aby w ten sposób łatwo wychwycić zachodzące na nich zmiany, czyli przede wszystkim ruch. Następnie oba obrazy są poddawane przetworzeniu mającemu na celu głównie usunięcie niepożądanych szumów poprzez filtracje medianową oraz połączenie informacji wydobytych z obydwu obrazów. Na tym etapie stosujemy prostą analizę obrazu w postaci obliczenia pola oraz współczynników kształtu danego obiektu aby móc go sklasyfikować jako jedną lub kilka blisko siebie idących osób, oczywiście możliwe jest również, że dany obiekt jest tylko jakimś zakłóceniem i nie powinien być brany pod uwagę.

Trzecim kryterium jakie bierzemy pod uwagę jest informacja pochodząca z śledzenia danego obiektu: na podstawie tego czy dany obiekt połączył się z jakimś innym obiektem możemy powiedzieć czy nowy obiekt reprezentuje jedną czy dwie osoby.

Na podstawie tych trzech kryteriów jesteśmy w stanie stwierdzić czy i ile osób reprezentuje dany obiekt i odpowiednio zwiększyć licznik osób.

W trakcie eksperymentów z powyższym algorytmem okazało się, że jest on stosunkowo wrażliwy na obrazy, których podłoże ma podobną jasność jak ubrania osób poruszających się na nim. Było to szczególnie dotkliwe na filmach nakręconych na korytarzu budynku C3. Gdy obszar o takim (zazwyczaj ciemnym) zabarwieniu znajduje się w obrazie powoduje mocne zniekształcenie obiektów otrzymanych metodą odejmowania tła.

Poprawnie wcześniej zidentyfikowane i śledzone osoby z chwilą wkroczenia w taki obszar potrafią zostać "rozcięte" na kilka mniejszych. Aby rozwiązać ten problem opracowaliśmy procedurę², dzięki której jesteśmy w stanie połączyć tak rozdzielone obiekty, jakkolwiek wiąże się to z pewnym zniekształceniem ich kształtu.

c. Dyskusja alternatywnych rozwiązań

Poszukując materiałów do tego projektu znaleźliśmy dużo bardzo ciekawych algorytmów

¹ Najciekawsze z nich zostaną opisane w punkcie c)

² Dokładny opis procedury znajduje się w rozdziale 8

zajmujących się problematyką detekcji ludzi na filmach.

Jedną z metod, z którą łączyliśmy duże nadzieje i długo ją testowaliśmy były operacje morfologiczne. Mieliśmy nadzieje, że przy ich użyciu uda się rozwiązać nasz główny problem czyli rozdzielać osoby znajdujące się blisko siebie, które zazwyczaj zlewają się w jeden obiekt po przetworzeniu zdjęcia. Niestety nie udało się to z zadowalającym skutkiem, zazwyczaj prowadziło do znacznej deformacji kształtu obiektu lub co gorsza do podzielenia jednej osoby na kilka obiektów, jednocześnie rzadko faktycznie rozdzielając osoby.

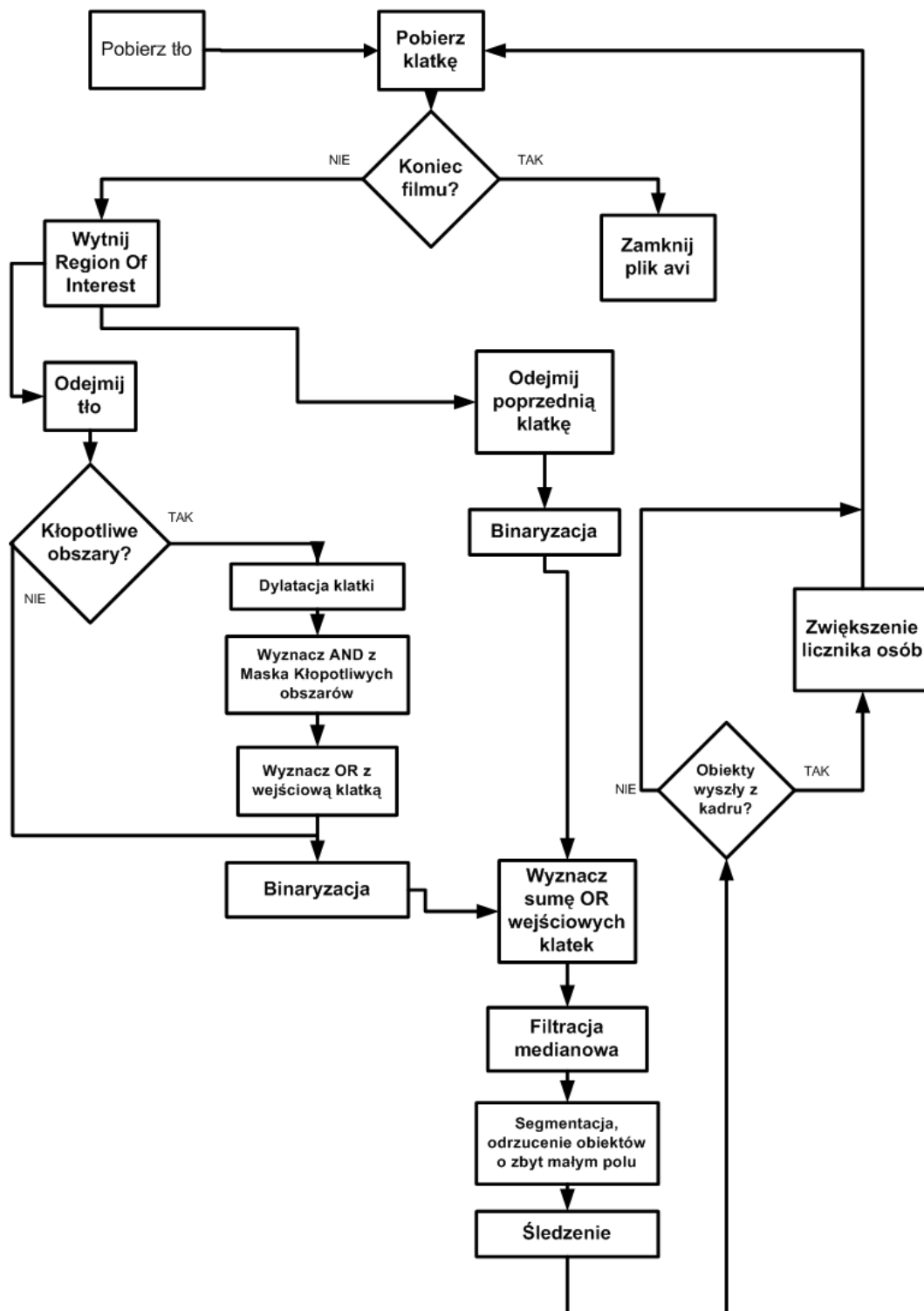
Osobnym problemem wykraczającym poza zakres tej pracy są algorytmy generacji tła. Na użytek tego projektu założyliśmy, że za tło uznawana będzie klatka wskazana przez operatora, w założeniu taka, na której nie znajdują się żadne osoby. Fakt ten jest sporym ograniczeniem i w przyszłości program powinno się wzbogacić o lepszy algorytm generacji tła.

Chcieliśmy również wspomnieć o kilku ciekawszych algorytmach, na które natrafiliśmy przy pracy nad naszym projektem. Poniższych metod nie zastosowaliśmy głównie z powodu na ich zbyt dużą złożoność.

Zamiast używania prostej metody obliczania różnicy kolejnych klatek filmu można posłużyć się bardziej zaawansowanymi algorytmami wykrywania ruchu *Optical Flow* [4], pozwalającymi określić czy dany piksel obrazu jest statyczny czy znajduje się w ruchu, a także pozwala wyznaczyć kierunek i szybkość tego ruchu. Algorytm ten dobrze sprawdza się w wykrywaniu szybko poruszających się obiektów, np. samochodów na autostradzie, natomiast niekoniecznie byłby lepszy w detekcji stosunkowo wolno poruszających się i czasami zatrzymujących się ludzi.

Bardzo ciekawym pomysłem jest zaproponowany w [5] algorytm wyszukujący charakterystyczne dla człowieka pozycje, określający kąty między jego kończynami, co jest zwłaszcza dobrze widoczne, gdy obserwujemy idącego człowieka z boku. Gdy algorytm ten raz wykryje człowieka jest w stanie go śledzić niezależnie od pozycji, jaką ten przyjmie dzięki zapamiętaniu schematu kolorów.

3. Koncepcja proponowanego rozwiązania



Rysunek 1. Schemat blokowy algorytmu

Zastosowany w tym projekcie algorytm opiera się na dwóch metodach uzyskania informacji z wejściowego filmu:

- odejmowanie tła
- różnica kolejnych klatek

W pierwszej z tych metod kluczowe jest odpowiednie wygenerowanie tła, w naszej aplikacji zakładamy, że tłem jest pierwsza klatka filmu. Badany materiał filmowy powinien być tak przygotowany aby na pierwszej klatce nie pojawiały się żadne osoby.

Program pobiera kolejne klatki filmu aż do ostatniej, wykonując na każdej kolejnej następujące operacje, przedstawione na rys.1.

Jeżeli klatki są w formacie kolorowym, następuje ich przekonwertowanie do skali szarości. Od aktualnej klatki jest odejmowana otrzymana wcześniej klatka tła (*Odejmij tło* na rys.1). Obraz różnicowy jest normalizowany przy użyciu wartości bezwzględnej.

Analogicznie otrzymujemy różnice dwóch kolejnych klatek (*Odejmij poprzednia klatkę* na rys.1).

Następnie każda z klatek zostaje poddana binaryzacji (*Binaryzacja* na rys.1) z ręcznie ustalonym, stałym progiem binaryzacji - innym dla różnicy dwóch klatek i dla obrazu od którego odjęto tło. Oba progi binaryzacji zostają dobrane doświadczalnie do każdego stanowiska filmowego.

Jeżeli mamy do czynienia z sytuacją, w której istnieją na filmowanym obszarze rejony, których kolor jest zbliżony do koloru³ odzienia większości osób pojawiających się na filmie, możemy wybrać specjalny sposób wstępnej obróbki tych problematycznych miejsc (*Blok wyboru Kłopotliwe obszary* na rys.1). Jeżeli użytkownik wybierze tę opcję, wtedy na podstawie klatki tła zostanie wyznaczona tzw. Maska Kłopotliwych Obszarów. Każda kolejna klatka filmu jest poddawana dylatacji (*Dylatacja klatki* na rys.1) a następnie wyznaczamy iloczyn logiczny tej klatki z Maską Kłopotliwych Obszarów (*Wyznacz AND z Maską Kłopotliwych Obszarów* na rys.1). Wynik jest następnie dodawany do początkowego obrazu (*Wyznacz OR z wejściową klatką* na rys.1), w efekcie czego otrzymujemy obraz, na którym obiekty rozdzielone przez odjęcie tła zostają ponownie skleione. W dalszej części algorytmu tak "poprawioną" klatkę traktujemy jak różnicę klatki i tła.

Aby wykorzystać informacje zawarte na obu obrazach wyznaczamy sumę logiczną obydwu obrazów różnicowych (*Wyznacz sumę OR wejściowych klatek* na rys.1), na której analizie skupi się dalsza część algorytmu.

³ Właściwie wartość luminancji, ponieważ na tym etapie film jest już przekonwertowany do skali szarości.

Na tym etapie na obrazie znajdują się interesujące nas obiekty, ale również spora ilość zakłóceń. Dokonujemy zatem filtracji medianowej filtrem (*Filtracja medianowa* na rys.1) o dużej masce, a następnie wstępnej segmentacji obrazu, po której odrzucamy wszystkie obiekty, których pole znajduje się poniżej zadanego progu (*Segmentacja i odrzucenie obiektów...* na rys.1). Zostają uznane za szумы i nie są poddawane dalszej analizie. Próg ten również trzeba wyznaczyć eksperymentalnie dla filmów nagranych na tym samym stanowisku.

Pojedyncze osoby są bardzo dobrze rozpoznawane na tym etapie przetwarzania, ale problematyczne staje się odróżnienie pojedynczej osoby od dwóch lub więcej idących blisko siebie.

Algorytm musi zatem ocenić ile osób może w rzeczywistości reprezentować każdy interesujący nas obiekt. W tym celu dla każdego z tych obiektów wyznaczamy jego pole oraz współczynnik kształtu W_7^4 .

W zależności od tego w jakim przedziale znajdują się te wartości dla danego obiektu, zostanie on uznany za jedną, dwie lub odpowiednio więcej osób. Progi wartości tych parametrów, zwłaszcza pola, muszą być doświadczalnie dobrane dla konkretnego stanowiska, na którym nagrano film, gdyż oczywiste jest, że nawet nieznaczna zmiana kąta kamery może istotnie zmienić ich wartości.

Następnym ważnym etapem jest śledzenie osób na kolejnych klatkach filmu (*Śledzenie*). Algorytm sprawdza czy obiekt z aktualnej klatki nakłada się na jakiś obiekt z poprzedniej, innymi słowy szukamy punktów wspólnych pomiędzy obiektami na dwóch kolejnych klatkach i na tej podstawie oceniamy, które obiekty są w rzeczywistości tą samą osobą⁵.

Ta wiedza daje nam jednocześnie bardzo dużo informacji na temat samych obiektów, ponieważ jeżeli dwa obiekty połączą się w jeden to z dużą pewnością możemy stwierdzić, że ten nowy obiekt reprezentuje dwie osoby.

Na podstawie łącznej informacji z tych trzech kryteriów możemy ocenić ile osób "zawierał" dany obiekt. Praktycznie decyzję o jaką liczbę zwiększyć licznik zliczonych osób podejmujemy na podstawie średniej ważonej⁶ wszystkich kryteriów w chwili, gdy obiekt opuści kadr (*Zwiększenie licznika osób*). Użytkownik może modyfikować to, jak duży wpływ będzie miało konkretne kryterium na końcowy wynik poprzez zmianę odpowiednich współczynników wagowych.

⁴ Dokładna definicja współczynnika W_7 wraz z wzorami analitycznymi znajduje się w rozdziale 8

⁵ Algorytm śledzenia zostanie dokładnie opisany w rozdziale 8

⁶ Sposób dokładnego obliczania końcowej znajduje się w rozdziale 8

4. Rezultaty i wnioski

Algorytm był testowany na sześciu filmach testowych:

- **akademik_1.avi**, **akademik_2a.avi**, **akademik_2b.avi**, **akademik_2c.avi** – są filmami nakręconymi na zewnątrz, z okna na czwartym piętrze, cyfrowym aparatem fotograficznym skierowanym pionowo w dół, na chodnik
- **C3_1.avi** – film nakręcony na korytarzu w budynku C3, kamera umieszczona powyżej podwieszanego sufitu (po zdjęciu kasetonu) – na wysokości ok. 3 metrów
- **C3_2.avi** – jw., tyle, że następowały zmiany oświetlenia polegające na włączaniu i wyłączaniu sztucznego oświetlenia

Tabela 1 przedstawia wyniki uzyskane z pracy algorytmu. Brak wyników dla filmu C3_2 zostanie wyjaśniony w dalszej części rozdziału.

Tabela 1

Nazwa filmu:	akademik_1	akademik_2a	akademik_2b	akademik_2c	C3_1
Ilość klatek:	423	1656	1970	2050	2630
Rozmiar kadru: [piksele]	224x202	224x202	224x202	224x202	273x224
Fps filmu źródłowego:	10	10	10	10	15
Czas obliczeń: [s]	98	277	282	343	482
Rzeczywista ilość osób:	20	106	83	86	74
Ilość osób wg algorytmu:	17	95	74	74	69
Za dużo zliczeń:	0	14	7	2	4
Za mało zliczeń:	3	25	16	14	9
Skuteczność globalna:	85,0%	89,6%	89,2%	86,0%	93,2%
Skuteczność rzeczywista:	85,0%	63,2%	72,3%	81,4%	82,4%

Skuteczność globalna jest to zależność ilości osób zliczonych przez algorytm w stosunku do ilości osób rzeczywiście pojawiających się na filmie. Zależność ta została wyliczona ze wzoru:

$$\eta_{glob} = \left(1 - \frac{|n_{rzecz} - n_{al}|}{n_{rzecz}} \right) \cdot 100\%$$

Równanie 1

Są przypadki w których algorytm nie zlicza pewnych osób (zliczeń za mało), lub zlicza niepotrzebnie pewne obiekty (za dużo zliczeń – np. ruszająca się gałąź). Pomimo dobrego wyniku „skuteczności globalnej” (por. tab. 1) rzeczywisty rezultat działania algorytmu zazwyczaj jest nieco

* W oparciu o te filmy algorytm był opracowywany.

** Są to filmy na których testowano algorytm.

gorszy, bowiem błędy typu „za dużo zliczeń” i „za mało zliczeń” rekompensują się nawzajem. Fakt ten został wzięty pod uwagę przy wyliczaniu skuteczności rzeczywistej, która zostaje obliczona wg poniższego wzoru:

$$\eta_{rzecz} = \left(1 - \frac{n_m + n_d}{n_{rzecz}}\right) \cdot 100\%$$

Równanie 2

W równaniach (1) oraz (2) użyto następujących oznaczeń:

η_{glob} – skuteczność globalna

η_{rzecz} – skuteczność rzeczywista

n_{rzecz} – rzeczywista ilość osób

n_{al} – ilość osób wg algorytmu

n_d – ilość osób zliczonych niepotrzebnie (za dużo zliczeni)

n_m – ilość osób nie zliczonych (za mało zliczeni)

- Po dopasowaniu parametrów i współczynników algorytm zachowywał się bardzo sensownie dla większości filmów osiągając skuteczność globalną rzędu 80%-90%. Jest to skuteczność wystarczająca do używania algorytmu w niektórych zastosowaniach (np. przy prowadzeniu statystyk zmian natężenia ruchu pieszych).
- Jedynie analiza filmu C3_2.avi przyniosła niezadowalające rezultaty. Zmiany sztucznego oświetlenia na tym filmie powodowały drastyczną zmianę tła. Tło pobrane z pierwszej klatki filmu nie pasowało do tła w klatkach po zmianie oświetlenia. Stąd też, odejmowanie raz wygenerowanego tła od kolejnych klatek w efekcie prowadziło do powstawania obrazów, których nie dało się poprawnie przeanalizować. Ten problem mógłby zostać rozwiązany poprzez zaimplementowanie algorytmu automatycznej generacji tła. Nasz algorytm takiej funkcji jednak nie posiada, więc używanie go w ww. sytuacjach jest utrudnione (wymagałoby ręcznej ingerencji użytkownika w celu wychwytywania zmian oświetlenia i odpowiedniego dzielenia filmów). Wyników dla filmu C3_2 nie zamieszczono w tabeli 1, ponieważ w tym przypadku algorytm „ogłupiał”.
- Przy kręceniu wszystkich filmów kamera była ustawiona w tryb automatycznego nastawu przysłony/jasności. Nie zauważono negatywnego wpływu na rezultaty działania algorytmu (w szczególności na słuszność doboru sztywnego progu binaryzacji). Możliwe, że powodem tego były zbyt krótkie sekwencje filmowe.

Przyczyny powstawania błędów:

- Na ujęciach na zewnątrz widać ruch gałęzi spowodowany silnym wiatrem. W niektórych sytuacjach był on tak duży, że gałąź została uznana przez algorytm za człowieka. Przykład: ujęcie na filmie **gałąź-zliczona.avi**. Są oczywiście przypadki, kiedy algorytm słusznie uznał ruch gałęzi za szum: **gałąź-nie zliczona.avi**.
- Na ujęciu **wymiana na krawędzi.avi** algorytm nie zliczył osoby schodzącej z kadru. Problem ten pojawia się dość często, a jego przyczyną jest to, że w tym samym momencie, w bezpośredniej bliskości osoby schodzącej z kadru, pojawia się osoba wchodząca na kadr filmu. Algorytm zamiast zakończyć śledzenia poprzedniej i rozpocząć śledzenia nowej zinterpretował tę sytuację tak, jakby śledzona do tej pory postać zawróciła.
- Na ujęciu **niefortunny kolor swetra.avi** możemy zaobserwować przechodnia który został zinterpretowany jako 3 oddzielne osoby. Miał niefortunny kolor swetra – zbyt podobny do tła w miejscu, w którym przechodził. Operacja odjęcia tła, binaryzacja i filtracja doprowadziły do „pocięcia” tego obiektu na kilka mniejszych, przez co w kilku klatkach miały zbyt małą powierzchnię i zostały uznane za szumy. W efekcie pojawianie się i znikanie tego obiektu uznano za pojawianie się i znikanie kolejnych osób. W przyszłości ten problem mógłby zostać rozwiązany poprzez zmodernizowanie algorytmu śledzącego w taki sposób, że śledziłby obiekty również „znikające” na pewną ilość klatek.
- W chwili, kiedy obiekt schodzi z kadru widać tylko jego część. Przez to jego kształt ulega znacznej zmianie, zaś pole zmniejszeniu. Te deformacje nieraz mają znaczący wpływ na ostateczną ocenę ilości osób składających się na obiekt. Na filmie **kształt i pole na krawędzi.avi** widać, że z powodu błędnej oceny dwóch powyższych kryteriów algorytm zliczył o jedną osobę za mało. Prawdopodobnie problem ten można by rozwiązać poprzez nieuwzględnianie ocen kryteriów kształtu i pola obiektów schodzących z kadru. Wtedy pole i współczynniki kształtu byłyby wyznaczane tylko dla obiektów nie stykających się z krawędzią kadru.

Metody szacowania ilości ludzi składających się na obiekt:

Problemem okazało się wykorzystanie informacji o ilości osób składających się na obiekt, pochodzących z kryterium kształtu i pola. Branie pod uwagę informacji o kształcie i polu tylko z ostatniej klatki, na której obiekt został zarejestrowany nie ma sensu, gdyż widać w kadrze tylko jego część. Wynik jest więc mocno przekłamany. Nie ma sensu też branie pod uwagę średniej ze wszystkich klatek, na których został zarejestrowany obiekt, gdyż w wielu przypadkach łączył się i rozłączał, co uniemożliwia poprawne przypisanie wyniku danego kryterium do konkretnego

obiektu. Zostajemy zatem ograniczeni do uwzględnienia informacji zebranych tylko od chwili ostatniego połączenia/rozłączenia obiektu do momentu zniknięcia z kadru.

- **Łączenie i rozłączanie obiektów**

Metoda jest bardzo skuteczna, jednak nie radzi sobie w sytuacjach, kiedy osoby idą bardzo blisko siebie przez cały kadr filmu. Główną zaletą tej metody okazał się fakt, że spełniła oczekiwania zarówno w filmach kręconych z akademika jak i w budynku C3.

- **Współczynniki kształtu i momenty geometryczne**

Większość współczynników kształtu, które w założeniu miały posłużyć do rozróżniania czy na wykryty obiekt składa się jedna czy więcej osób nie sprawdziły się. Powodem mógł być fakt, że obiekty znacznie zmieniają swój kształt podczas ruchu.

Obiecująco zapowiadał się jedynie współczynnik W7, którego działanie przedstawiono na wykresie 1. Widać wartości W7 dla kolejnych obiektów pojawiających się w filmie akademik_1. Oś „X” reprezentuje kolejne klatki filmu. Na osi „Y” przedstawiono wartości współczynnika W7. Krzyżykami zaznaczono obiekty składające się z dwóch osób, natomiast kółka odpowiadają pojedynczym osobom. Można zauważyć, że W7 przybiera wartości większe dla obiektów jednoosobowych, oraz mniejsze dla wieloosobowych. Skuteczność akurat w tym przypadku tłumaczymy faktem, że film był kręcony ze znacznej wysokości, przez co kształty obiektów zasadniczo nie zmieniają się wraz ze zmianą położenia (nie zmienia się kąt widzenia - głowa zawsze zasłania tułów). Ograniczeniem okazała się możliwość rozdzielenia obiektów pomiędzy tylko dwie klasy (obiekty jedno lub wieloosobowe).

W przypadku filmu C3_1 zależność między wartościami współczynnika W7, a ilością osób składających się na obiekt ma charakter bardzo losowy, co widać oglądając film C3_1_output. Nawet obserwując obiekty na filmie C3_1 - wstępnie przetworzonym i po segmentacji, trudno jest gołym okiem wyodrębnić kształty reprezentujące jakkolwiek z klas (jedno lub dwuosobowe). Tym bardziej, że w każdej klatce filmu kształty te ulegają znacznym deformacjom. W tych warunkach wartości współczynników kształtu nie niosą cennych informacji.

Pole obiektów:

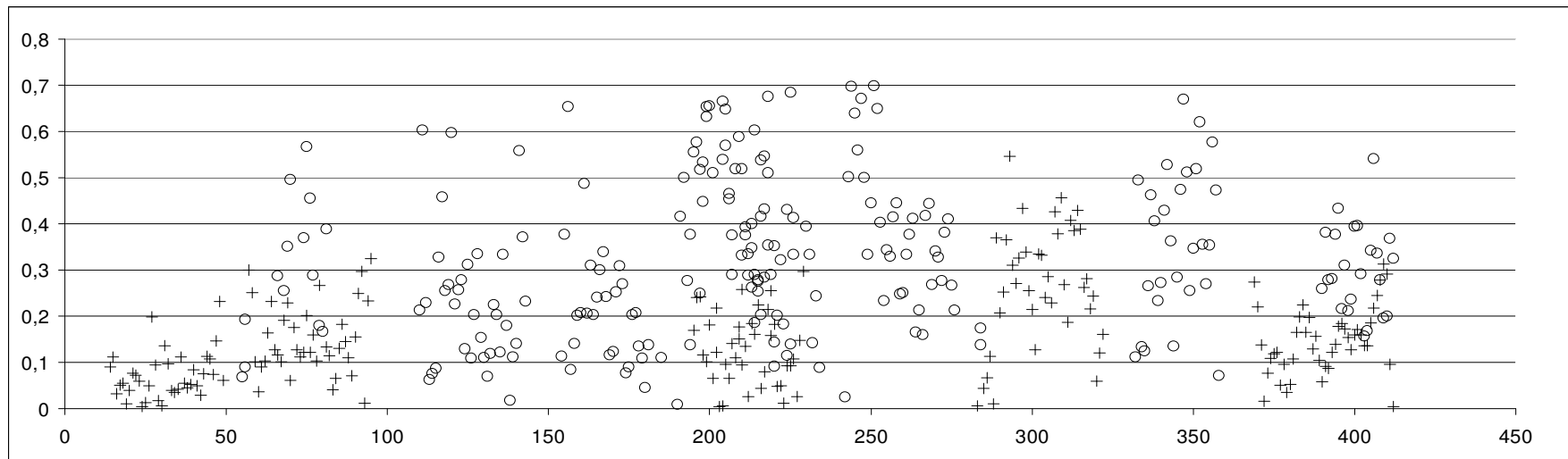
Dobrym sposobem do rozstrzygnięcia o ilości osób w obiekcie okazała się analiza pola powierzchni obiektów (wykres 2). Oś „Y” reprezentuje wartości pola (w pikselach), zaś oś „X”, tak jak na wykresie 1, przedstawia kolejne klatki filmu akademik_1. Można zaobserwować silną tendencję do grupowania się poszczególnych pól powierzchni w

okolicach pewnych wartości. Na wykresie poszczególne kształty odpowiadają ilościom osób rzeczywiście składających się na obiekt:

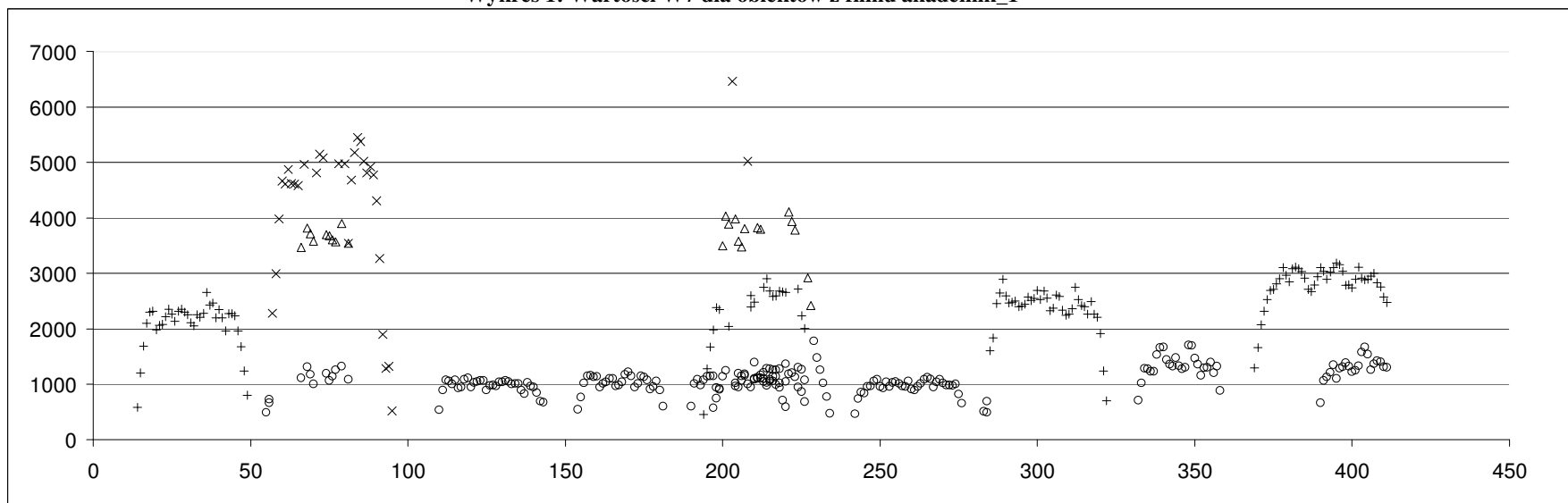
- „kółka” – 1 osoba
- „plusy” – 2 osoby
- „trójkąty” – 3 osoby
- „krzyżyki” (X) – 4 osoby

Momenty geometryczne:

Rezultaty klasyfikacji obiektów na podstawie kombinacji momentów geometrycznych różnych rzędów okazały się nieobiecujące. Nie została zaobserwowana wystarczająco silna tendencja wartości momentów obiektów jednej klasy do grupowania się.



Wykres 1: Wartości W7 dla obiektów z filmu akademik_1



Wykres 2: Wartości pola dla obiektów z filmu akademik_1

5. Podsumowanie

Udało się zaprojektować i zaimplementować algorytm, który z powodzeniem mógłby być wykorzystany do śledzenia dla celów statystycznych np. ilości przechodniów. Dowiedziono, że odpowiednio przygotowane stanowisko (zapewnienie odpowiedniego oświetlenia, odpowiedniego umieszczenia kamery, dobranie odpowiednich współczynników dla algorytmu) umożliwia uzyskanie satysfakcjonująco dokładnych wyników dla niektórych rozwiązań.

Algorytm może sprawiać jednak problemy w obsłudze - chociażby z powodu konieczności doboru dużej ilości bardzo istotnych współczynników. Dla każdego stanowiska należy wprowadzić (w wersji obecnej) metodą prób i błędów odpowiednie wartości.

Tak jak przewidywaliśmy i było to opisane w literaturze, problemem okazały się postacie zasłaniające się nawzajem, lub idące bardzo blisko siebie. W odróżnieniu z ilu osób składają się tak wykryte obiekty miały pomóc:

1. Współczynniki kształtu
2. Momenty
3. Pole powierzchni obiektów
4. Wychwytywanie i zapamiętywanie sytuacji, w których obiekty łączą się lub rozdzielają.

Jak się okazało czwarty ze sposobów okazał się najbardziej skuteczny. Dwa pierwsze stwarzają problemy przy ostatecznej interpretacji

Czas działania algorytmu zależał od ilości pikseli w analizowanym obrazie oraz ilości obiektów „przewijających się” pod kamerą. Na komputerze klasy PC, z procesorem AMD Athlon 3200+ oraz 1GB RAMu, algorytm jest w stanie przetworzyć ok. 6 klatek na sekundę. Jest to szybkość bardzo zbliżona do szybkości filmów odtwarzanych na żywo. Co prawda, nasz algorytm działał w trybie offline – na wcześniej nagranych filmach. Jednak prawdopodobnie optymalizacja algorytmu i implementacja w środowiskach szybszych niż Matlab (np. język C, C++) ułatwiłaby takiemu systemowi pracę online.

W kolejnych fazach rozwijania projektu należałoby również zaimplementować algorytm automatycznie generujący tło, bowiem nasza metoda (gdzie za tło uznawaliśmy pierwszą klatkę filmu) mogła zostać wykorzystana jedynie w warunkach laboratoryjnych, na stosunkowo krótkich filmach.

Pomocne dla użytkowników zapewne okazałoby się również dodatkowe wprowadzenie algorytmów automatycznie dobierających współczynniki (minimalna powierzchnia obiektów, minimalny czas, na który muszą się pojawić, próg binaryzacji, rozmiar maski filtrującej itp.)

6. Literatura

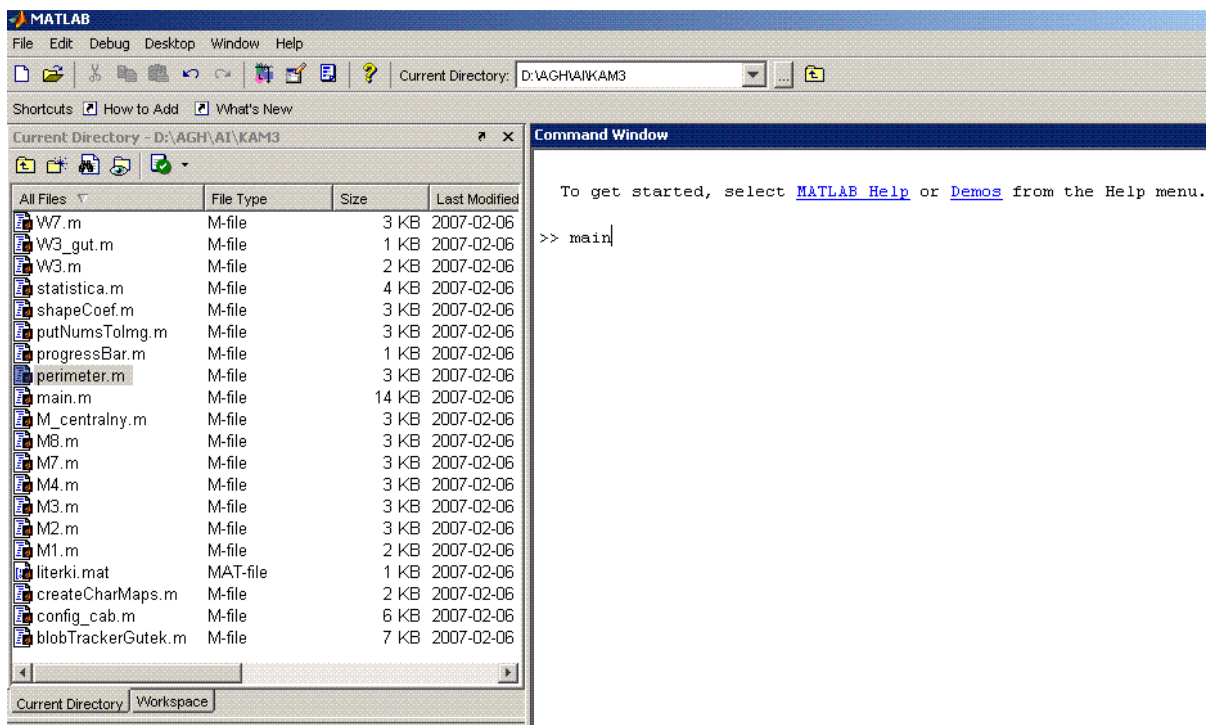
Literatura

- [1] Tadeusiewicz R., Korohoda P., „Komputerowa analiza i przetwarzanie obrazów.” Wyd. Fundacji Postępu Telekomunikacji, Kraków, 1997.
- [2] Matlab Image Processing Toolbox help
- [3] " People Counting and Tracking for Surveillance ", Anup Doshi, CSE252C Project Report Draft- Nov. 29,2005
- [4] Optical Flow:
<http://www.cs.otago.ac.nz/research/vision/Research/OpticalFlow/opticalflow.html>
http://www.cs.cf.ac.uk/Dave/Vision_lecture/node45.html
- [5] "Strike a Pose: Tracking People by Finding Stylized Poses" Deva Ramanan and D.A. Forsyth and Andrew Zisserman, University of California, Berkeley–Berkeley, CA94720
University of Oxford –Oxford, OX14AJ, UK
- [6] Współczynniki kształtu i momenty geometryczne:
www.agh.edu.pl/uczelnia/tad/Przetwarzanie_obrazow_medycznych/7-analiza_kształtu.ppt

7. DODATEK A: Opis opracowanych narzędzi i metody postępowania

Aby rozpocząć pracę z aplikacją konieczne jest uruchomienie programu Matlab oraz ustawienie katalogu roboczego na ten zawierający pliki filmowe kod programu. Filmy muszą być zapisane w formacie obsługiwany przez Matlab. Najlepszym rozwiązaniem jest zapisanie filmów wejściowych do formatu nie skompresowanego (dużo szybsza praca algorytmu). Filmy zamieszczone na płycie są skompresowane kodekiem xVid (również znajdującym się na płycie). Do dekompresji najlepiej użyć Virtual Duba. Głównym plikiem aplikacji jest plik main.m. Aby uruchomić naszą aplikację należy w Matlab Command Window wpisać :

main



Rysunek 2. Widok przed uruchomieniem obrazu

Jak widać na Rysunek 2 aby aplikacja działała poprawnie konieczne jest aby w folderze roboczym, z którego uruchamiamy program znajdowały się następujące pliki:

createConfig.m

blobTracker.m

shapeCoef.m

progressBar.m

perimeter.m

W7.m

putNumsToImg.m

literki.mat

Powyższe pliki są niezbędne do pracy programu, natomiast jeżeli chcemy również obliczyć wartości innych współczynników oraz momentów geometrycznych musimy także dodać pliki :

statistica.m

M_centralny.m

m.m

M1.m

M2.m

M3.m

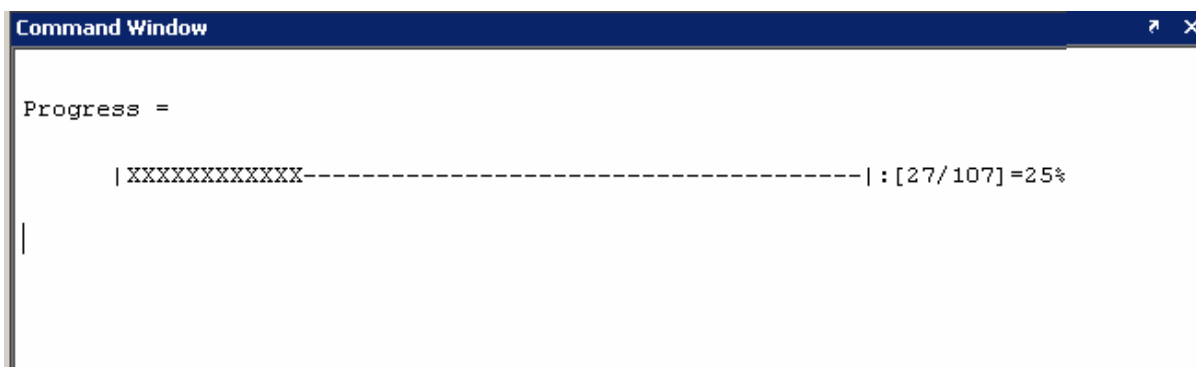
M4.m

M7.m

M8.m

W3.m

Po uruchomieniu skryptu w Matlab Command Window powinniśmy zobaczyć widoczny na Rysunek 3 „pasek postępu” pokazujący ilość już przetworzonych klatek względem długości całego przetwarzanego filmu.



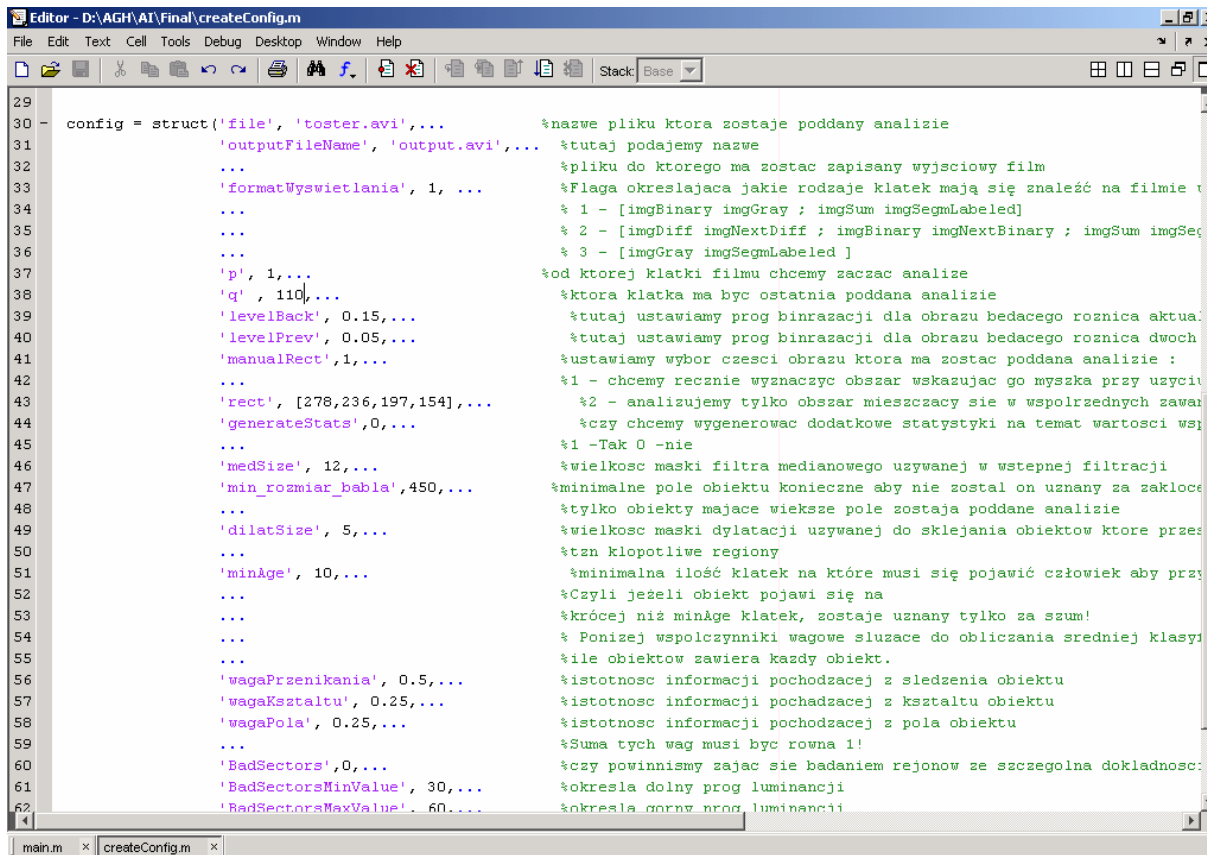
Rysunek 3. Widok w trakcie działania programu

Za paskiem postępu widzimy trzy liczby, są to odpowiednio: ilość przetworzonych klatek, całkowita ilość klatek do przetworzenia, oraz procentowo wyrażony stosunek dwóch poprzednich liczb. Innymi słowy wyraża procentowo jaka część obliczeń została już wykonana.

Gdy zobaczymy, że pasek postępu doszedł do końca (tzn. do 100%) i Matlab wyświetla znak zachęty, oznacza to, że obliczenia się zakończyły. Możemy obejrzeć film wynikowy, został on utworzony pod nazwą wybraną przez użytkownika w *config.outputFileName*.

Jeżeli użytkownik wybrał opcje generacji dodatkowych statystyk, wpisując 1 w *config.generateStats*, zakończenie obliczeń potrwa kilkanaście sekund dłużej niż w innym przypadku. Statystyki są dostępne jako wektory w Matlab Workspace, gdzie są łatwo dostępne do wizualizacji lub dalszych obliczeń w Matlabie lub poprzez Exel Link w MS Exel.

Jeżeli chcemy dokonać jakiś zmian w parametrach obliczeń aplikacji konieczna jest ręczna edycja skryptu config.m. Aby to zrobić otwieramy plik config.m w edytorze skryptów Matlab'a:



Rysunek 4. Skrypt konfiguracyjny

Jak widzimy na rysunku 4 plik createConfig.m zawiera strukturę przechowującą wszystkie parametry aplikacji, możemy je modyfikować poprzez zmianę odpowiednich pól tej struktury. Poniżej znajdują się lista pól struktury config oraz parametrów aplikacji, które można w nich modyfikować, wartości domyślne znajdują się w nawiasach kwadratowych zaraz za nazwą pola

nazwaPola [wartości domyślne]

file ['input.avi']- nazwę pliku filmowego, która zostaje poddany analizie

Dla optymalnej prędkości działania aplikacji wskazane jest aby plik wejściowy nie był skompresowany, gdyż w innym wypadku czas trwania obliczeń wzrasta nawet kilkudziesięciokrotnie przy dużych plikach! Jeżeli zatem pliki filmowe, które chcemy poddać analizie są w skompresowane najlepiej będzie je zdekompresować przed przystąpieniem do przetwarzania ich przy użyciu tej aplikacji. Można do tego użyć na

przykład programu Virtual Dub, z którego dość często korzystaliśmy w trakcie pracy nad tym programem.

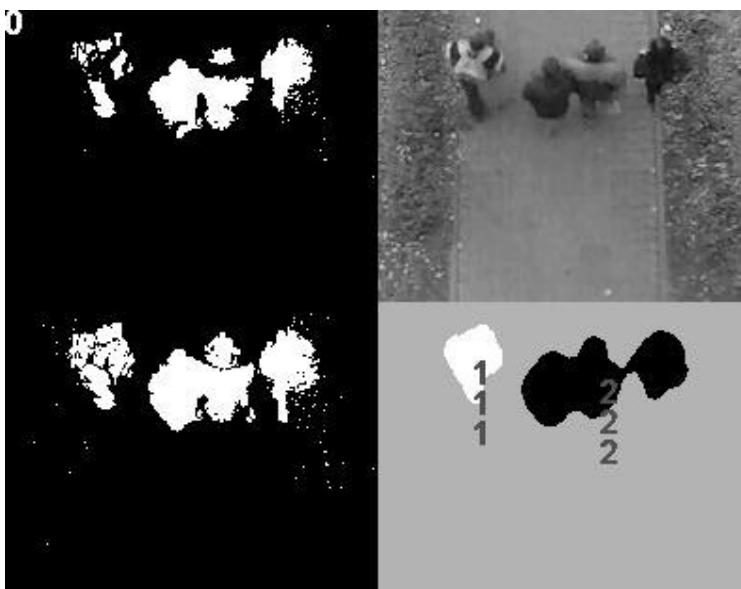
Jeżeli jednak zdecydujemy się przetwarzać plik skompresowany musi to być którymś z formatów obsługiwanych przez środowisko Matlab [2]. W naszych testach używaliśmy kompresji Indeo video 5.1.

outputFileName ['output.avi']- tutaj podajemy nazwę pliku do którego ma zostać zapisany wyjściowy film

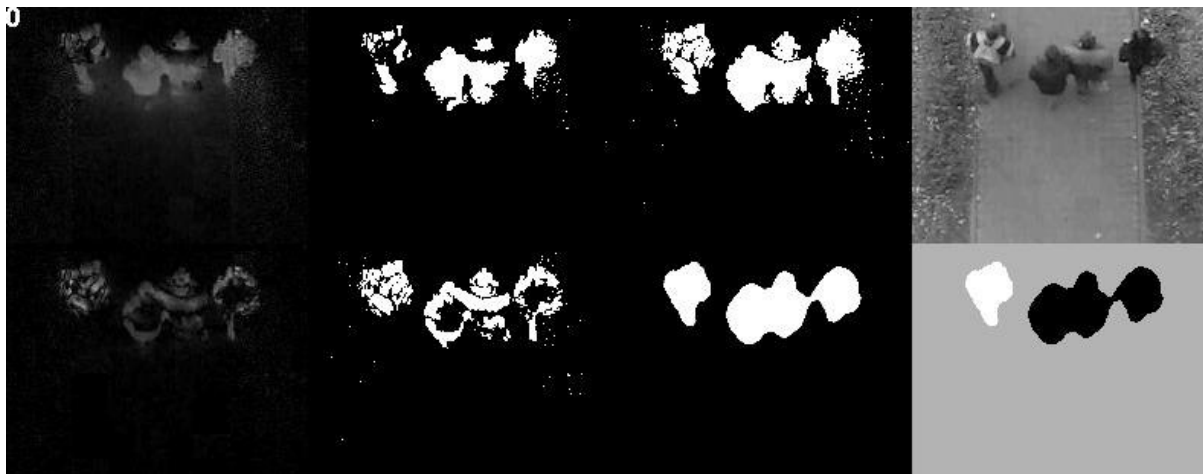
formatWyswietlania [1]- Flaga określająca jak będzie wyglądał film wyjściowy, tzn z jakich rodzajów klatek ma się on składać:

1 - [imgBinary imgGray ; imgSum imgSegmLabeled]

Po wybraniu tej opcji film końcowy będzie wyglądał jak klatka poniżej:



2 - [imgDiff imgBinary imgSum imgGray;imgNextDiff imgNextBinary imgMesh imgSegm]



3 - [imgGray imgSegmLabeled]



p [1]- od której klatki filmu chcemy zacząć analizę

q [*n*] - która klatka ma być ostatnia poddana analizie, gdzie 'n' ilość klatek

levelBack [0.2]- tutaj ustawiamy próg binaryzacji dla obrazu będącego różnicą aktualnej klatki i tła

levelPrev [0.1]- tutaj ustawiamy próg binaryzacji dla obrazu będącego różnicą dwóch kolejnych klatek

manualRect [0]- ustawiamy wybór części obrazu, która ma zostać poddana analizie

1 - chcemy ręcznie wyznaczyć obszar wskazując go myszka przy użyciu funkcji `imcrop()`

W przypadku wybrania '1' po uruchomieniu głównego skryptu pojawi się okno z pierwszą klatką, pozwalające na ręczne wybranie obszaru poddawanego późniejszej analizie.

Użytkownik zobaczy poniższe okno:



0 - analizujemy tylko obszar mieszczący się w współrzędnych zawartych w prostokącie 'rect'

rect [[278, 236, 197, 154]]- zawiera współrzędne obszaru wybranego do analizy

generateStats [0]- czy chcemy wygenerować dodatkowe statystyki na temat wartości współczynników kształtu 1 -Tak 0 -nie

medSize [9]- wielkość maski filtra medianowego używanej w wstępnej filtracji

min_rozmiar_babla [450]- minimalne pole obiektu konieczne aby nie został on uznany za zakłócenie 07835393235

dilatSize [5]- wielkość maski dylatacji używanej do sklejania obiektów, które przeszły przez tzw. kłopotliwe regiony

minAge [10]- minimalna ilość klatek na które musi się pojawić człowiek aby przy zniknięciu został zliczony. Czyli jeżeli obiekt pojawi się na krócej niż *minAge* klatek, zostaje uznany tylko za szum!

Poniżej współczynniki wagowe służące do obliczania średniej ważonej na podstawie której klasyfikujemy ile osób zawiera każdy obiekt. Każda z tych trzech wag określa jak istotne dla ostatecznej decyzji będzie konkretne kryterium:

wagaPrzenikania [0.5]- istotność informacji pochodzącej z śledzenia obiektu

wagaKształtu [0.25]- istotność informacji pochodzącej z kształtu obiektu

wagaPola [0.25]- istotność informacji pochodzącej z pola obiektu

Suma tych wag musi być równa 1!

Zmieniając poniższy parametr użytkownik może zdecydować czy aplikacja powinna zająć się przetwarzaniem niektórych, kłopotliwych rejonów ze szczególną dokładnością

BadSectors [0]- ustawienie tej flagi na '1' powoduje uruchomienie dodatkowej procedury obiektów rozdzielonych przez wyjątkowo ciemne obszary tła.

Te wartości służą do wyznaczenia tzw. Maski Kłopotliwych Obszarów, znajduje się w niej wszystkie piksele, których wartość luminancji,

na pierwszej klatce filmu, znajduje się w tym przedziale pomiędzy tymi progami

BadSectorsMinValue [30]- określa dolny próg luminancji

BadSectorsMaxValue [60]- określa górny próg luminancji

Poniższe parametry służą do ustalenia wartości progowych pola obiektów od których zostaną one sklasyfikowane jako odpowiednia ilość osób:

progDwochOsob [1000]- wartość pola, od której uznajemy dany obiekt za dwie osoby

progTrzechOsob [2500]- wartość pola, od której uznajemy dany obiekt za trzy osoby

progCzterechOsob [3000]- wartość pola, od której uznajemy dany obiekt za cztery osoby

8. DODATEK B: Realizacja proponowanego rozwiązania

Procedura łączenia obiektów rozdzielonych przez wyjątkowo ciemne obszary

tła.

Procedura ta jest uruchamiana na życzenie użytkownika, poprzez ustawienie flagi *config.BadSectors* w pliku konfiguracyjnym *createConfig.m* na wartość 1.

Pierwszym krokiem tej procedury, wykonywanym jeszcze przed wejściem w pętlę główną programu, jest wyznaczenie tzw. Maski Kłopotliwych Obszarów. Wyznaczana jest ona przy użyciu funkcji *roicolor()*⁷. Funkcja umożliwia stworzenie binarnej maski, w której wartość 1 mają tylko te

⁷ help Matlab Image Processing Toolbox

piksele, których wartość na wejściowym obrazie znajduje się pomiędzy wybranym przez użytkownika przedziałem wartości.

Następnie już wewnątrz głównej pętli każda kolejna klatka już po binaryzacji przechodzi następującą procedurę:

- wyznaczany jest obraz wejściowej klatki po dylatacji z maska utworzoną przy użyciu funkcji `Strel()` o kształcie 'Disk' i rozmiarze określonym przez użytkownika w `config.dilatSize`
- wyznaczana jest obraz będący iloczynem logicznym Maski Kłopotliwych Obszarów i obrazu po dylatacji
- obliczana jest suma logiczna obrazu otrzymanego w poprzednim punkcie i wejściowej klatki

Tak uzyskana klatka jest poddawana dalszej analizie.

Obliczanie współczynników kształtu i momentów geometrycznych⁸

Współczynnik W3 (współczynnik Malinowskiej)

$$R_M = \frac{L}{2 \cdot \sqrt{\pi \cdot S}} - 1$$

gdzie

L - obwód obiektu

S - pole powierzchni obiektu

Współczynnik W7

$$W7 = \frac{r_{\min}}{R_{\max}}$$

gdzie

r_{\min} – minimalna odległość konturu od

środką ciężkości obiektu

R_{\max} – maksymalna odległość konturu

od środka ciężkości obiektu

Momenty geometryczne :

⁸ w oparciu o prezentację [6]

Dla cyfrowego obrazu podwójne całki występujące w wzorach na momenty mogą być aproksymowane sumami.

Wzór na moment zwykły :

$$m_{pq} = \sum_{i=1}^m \sum_{j=1}^n i^p \cdot j^q \cdot x_{ij}$$

Gdzie:

x_{ij} - wartość piksela (i, j)

p, q – określa rząd momentu geometrycznego

Współrzędne środka ciężkości:

$$\tilde{i} = \frac{m_{10}}{m_{00}} \qquad \tilde{j} = \frac{m_{01}}{m_{00}}$$

Moment centralny :

$$M_{pq} = \sum_{i=1}^m \sum_{j=1}^n (i - \tilde{i})^p \cdot (j - \tilde{j})^q \cdot x_{ij}$$

Wprowadza się często momenty znormalizowane :

$$N_{pq} = \frac{M_{pq}}{m^{\zeta}_{00}}$$

gdzie:

$$\zeta = \frac{p+q}{2} + 1$$

p+q=2,3,4,...

Oto wzory analityczne na momenty geometryczne użyte przez nas w naszym projekcie:

$$\begin{aligned} \underline{M1} &= N_{20} + N_{02} \\ \underline{M2} &= (N_{20} - N_{02})^2 + 4 \cdot N_{11}^2 \\ \underline{M3} &= (N_{30} - 3 \cdot N_{12})^2 + (3 \cdot N_{21} - N_{03})^2 \\ \underline{M4} &= (N_{30} - N_{12})^2 + (N_{21} - N_{03})^2 \\ \underline{M5} &= (N_{30} - 3 \cdot N_{12}) \cdot (N_{30} + N_{12}) \cdot \\ &\quad \cdot [(N_{30} + N_{12})^2 - 3 \cdot (N_{21} + N_{03})^2] + \\ &\quad + (3 \cdot N_{21} - N_{03}) \cdot (N_{21} + N_{03}) \cdot \\ &\quad \cdot [3 \cdot (N_{30} + N_{12})^2 - (N_{21} + N_{03})^2] \end{aligned}$$

$$\begin{aligned}
\underline{M6} &= (N_{20} - N_{02}) \cdot [(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2] + \\
&+ 4 \cdot N_{11} \cdot (N_{30} + N_{12}) \cdot (N_{21} + N_{03}) \\
\underline{M7} &= N_{20} \cdot N_{02} - N_{11}^2 \\
\underline{M8} &= N_{30} \cdot N_{12} + N_{21} \cdot N_{03} - N_{12}^2 - N_{21}^2 \\
\underline{M9} &= N_{20} \cdot (N_{21} \cdot N_{03} - N_{12}^2) + N_{02} \cdot (N_{30} \cdot N_{12} - N_{21}^2) - \\
&- N_{11} \cdot (N_{30} \cdot N_{03} - N_{21} \cdot N_{12}) \\
\underline{M10} &= (N_{30} \cdot N_{03} - N_{12} \cdot N_{21})^2 - \\
&- 4 \cdot (N_{30} \cdot N_{12} - N_{21}^2) \cdot (N_{03} \cdot N_{21} - N_{12})
\end{aligned}$$

Algorytm śledzący oraz metoda określająca ilość osób składających się na jeden obiekt.

Po szeregu operacji wykonywanych osobno dla każdej z klatek filmu (odjęcie tła, binaryzacja, filtracja, segmentacja) otrzymaliśmy wyodrębnione, pokolorowane i poindeksowane obiekty. Zadaniem algorytmu śledzącego jest szukanie par obiektów, z których dwóch kolejnych klatek, których położenie zmieniło się nieznacznie. Algorytm ma zliczać obiekty w chwili ich zniknięcia (czyli w momencie, kiedy dla obiektu z klatki poprzedniej nie potrafimy znaleźć jego następnika w klatce następnej).

Zadanie okazuje się jednak znacznie trudniejsze, gdyż ludzie przechodzący przed kamerą często zasłaniają się nawzajem, lub idą bardzo blisko siebie. W takich sytuacjach zostają oni wyodrębnieni jako jeden obiekt. Algorytm śledzący rozpatruje sytuacje, kiedy ludzie oddalają się od siebie (następuje rozłączenie się obiektów), lub przechodnie znacznie zbliżają się do siebie (następuje połączenie się obiektów). To utrudnienie, polegające na przenikaniu się śledzonych obiektów okazuje się być bardzo użyteczne przy określaniu ilości osób składających się na jeden wyodrębniony obiekt. Jeżeli np. dwa śledzone przez kilka klatek obiekty łączą się, to po kilku następnych klatkach, kiedy ten obiekt znika, wiemy, że składały się na niego dwie osoby. Z tych zależności korzysta opracowana przez nas metoda określająca ilość osób składających się na jeden obiekt.

Algorytm śledzący wykonuje każdą swoją iterację dla dwóch kolejnych klatek (w programie jest to wywołanie funkcji blobTracker dla dwóch kolejnych klatek).

Analizowaliśmy filmy o niskim współczynniku ilości klatek na sekundę: nawet 10 fps. Obserwacje wykazały, że nawet wtedy nałożone na siebie obiekty z dwóch kolejnych klatek w znacznym stopniu się pokrywają (Rysunek 5). Nawet jeżeli tymi obiektami są szybko biegnący ludzie – ich przesunięcie pomiędzy dwoma klatkami jest nieznaczne.



Rysunek 5: Dwie kolejne klatki oraz obraz powstały z nałożenia jednej na drugą

Zależność ta została wykorzystana w algorytmie śledzącym. W tym celu zostaje stworzona „macierz wspólności”. Każdy z jej wierszy odpowiada jednemu obiektowi z klatki poprzedniej, każda z kolumn odpowiada jednemu obiektowi z klatki teraźniejszej. Liczba znajdująca się na przecięciu i-tego wiersza oraz j-tej kolumny reprezentuje część wspólną (wyrażoną w ilości pokrywających się pikseli) pomiędzy i-tym obiektem z klatki poprzedniej oraz j-tym obiektem z klatki teraźniejszej.

Dalsza analiza oraz określanie ilości osób składających się na jeden obiekt jest wykonywana w oparciu właśnie o macierz wspólności, wg poniższych kryteriów:

- Każdy obiekt niesie ze sobą informację o ilości składających się na niego osób (zapisaną jako liczba w zmiennej numOfPeople)
- Jeżeli obiekty łączą się (w j-tej kolumnie macierzy wspólności znajduje się więcej niż jedna liczba niezerowa) to numOfPeople dla nowopowstającego obiektu jest sumą numOfPeople obiektów łączących się
- Jeżeli obiekt się dzieli (w i-tym wierszu macierzy wspólności znajduje się więcej niż jedna liczba niezerowa) to wartość numOfPeople każdego z nowopowstających obiektów jest wprost proporcjonalna do numOfPeople obiektu dzielącego się oraz do wspólnej powierzchni. Suma numOfPeople nowopowstałych obiektów jest zatem równa numOfPeople obiektu dzielącego się. Zależność ta została opisana w Równanie 3.

$$n_j = \frac{m_{w_{i,j}} \cdot numOfPeople_i}{\sum_i m_{w_{i,j}}}$$

Równanie 3

W Równanie 3 użyto zmiennych:

n_j – wartość numOfPeople nowopowstającego obiektu

$m_{w_{i,j}}$ – wartość komórki w i-tym wierszu i j-tej kolumnie „macierzy wspólności”

$numOfPeople_i$ – wartość numOfPeople dzielącego się, j-tego obiektu

- Nowopowstający obiekt (człowiek wchodzący w kadr ekranu) zawsze ma przypisywaną wartość numOfPeople = 1.

- Jeżeli w wyniku podzielenia obiektu wartość numOfPeople obiektu nowopowstającego wynosi 0 to jest jej przypisywana wartość 1. Przykładowo: Przez kilka klatek śledzimy jeden, nowy obiekt (wartość numOfPeople = 1). Dochodzi do podzielenia tego obiektu na dwa, wtedy numOfPeople jednego z nich przyjmuje wartość 1, a drugiego 0. Jest to sytuacja w której prawdopodobnie dwoje ludzi idących dotąd blisko siebie, oddaliło się. Zatem drugi z obiektów również powinien mieć wartość numOfPeople równą 1, a nie 0.

Każda iteracja blobTracker zwraca informację o ilości osób, które w danej klatce zeszły z kadru (zmienna numOfDeadBubbles). Ilość osób niesionych przez właśnie znikający obiekt jest wyliczana przy użyciu średnich ważonych z decyzji podejmowanych przez:

- „metodę pola”
- „metodę ze współczynnikiem W7”
- „analizę przenikania obiektów”.

Przy czym najpierw liczymy średnią arytmetyczną z wyników każdej z tych trzech metod – uzyskanych dla klatek od ostatniej „sytuacji krytycznej” do zniknięcia obiektu. „Sytuacją krytyczną” nazywamy ostatnią klatkę filmu, w której obiekt uległ podzieleniu/połączeniu. Mając już 3 wartości średnich (po jednej dla każdej z trzech metod) wyliczamy ostateczną decyzję przy użyciu średniej ważonej. Wagami w tym wypadku są parametry startowe, ustawiane przez użytkownika, takie jak:

- wagaPrzenikania
- wagaKształtu
- wagaPola

O tak wyliczoną wartość (dla znikającego obiektu) jest powiększana zmienna numOfDeadBubbles (dla teraźniejszej klatki filmu).

9. DODATEK C. Opis informatyczny procedur

Napisana przez nas aplikacja powstała przy użyciu środowiska MATLAB w wersji 7.1.0.246 (R14) Service Pack 3, w systemie operacyjnym MS Windows XP.

Oto opis głównej procedury programu zawartej w pliku main.m

```
% /******* */
% /* */
% /* Główny skrypt programu MAIN */
% /* */
% /* Przeznaczenie: */
% /* Jest to główna procedura programu, z jej */
% /* wnetrza uruchamiane sa pozostale moduly */
% /* */
% /* Uzywane funkcje: */
% /* */
% /* Funkcje z Matlab Image Processing Toolbox: */
% /* roicolor() */
% /* imcrop() */
% /* aviinfo() */
% /* avifile() */
% /* aviread() */
% /* regionprops() */
% /* bwlabel() */
% /* rgb2gray() */
% /* imdilate() */
% /* strel() */
% /* immultiply() */
% /* imabsdiff() */
% /* im2bw() */
% /* im2frame() */
% /* Wlasne funkcje : */
% /* progressBar() - funkcja rysujaca progress bar */
% /* putNumsToImg() - funkcja wypisujaca cyfry na obrazie */
% /* blobTracker() - implementuje algorytm sledzenia */
% /* statistica() - oblicza i zapisuje wartosci wszystkich */
% /* wspolczynnikow ksztaltu i momentow geometrycznych dla kazdego */
% /* obiektu z kazdej klatki filmu */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* */
% /* Uzywane zmienne: */
% /* config - Skrypt main korzysta z zmiennej config, */
% /* tworzonej przez skrypt config_cab. Konieczne jest */
% /* poprawne zaincjowanie wszystkich pol struktury config. */
% /* Wszystkie pola struktury sa opisane dokladnie w pliku */
% /* config_cab.m */
% /* */
% /* LICZNIK_LUDZI - zmienna przechowujaca zliczana */
% /* ludzi ilosc */
% /* */
% /* God - tablica struktur przechowujaca dane z kazdej klatki */
% /* na temat wszystkich obiektow wykrytych w tej klatce */
% /* */
% /* outputFileMovie - przechowuje uchwyt do pliku wyjsciowego
```

```

% /*
% /* Zmienne przechowujące kolejne, przetworzone klatki obrazu:
% /* background - przechowuje obraz tła (pierwsza klatka)
% /* tmpFrame - przechowuje aktualna klatkę filmu
% /* imgGray - konwertuje klatkę filmu z RGB do skali szarości
% /* imgDiff - zawiera obraz różnicowy klatki aktualnej i tła
% /* imgBinary - zawiera zbinaryzowany obraz imgDiff
%
% /* Dodatkowy klatki potrzebne do sklejania obietek należących do
% /* Maski Kłopotliwych Obszarów
% /* maskaKłopotliwychObszarów - zawiera piksele których luminancja
% /* mieści się w pewnym określonym przez użytkownika przedziale
% /* imgDilated - przechowuje obraz po dylatacji
% /* imgWycięte - zawiera tylko te obszary z imgDilated które należą
% /* jednocześnie do maskaKłopotliwychObszarów
% /* imgReparatedDiff - obraz końcowy po dodaniu imgWycięte do
% /* początkowego obrazu imgBinary
% /* imgNextDiff - obraz będący różnicą dwóch kolejnych klatek
% /* imgNextBinary - zbinaryzowany imgNextDiff
% /* imgSum - obraz będący sumą logiczną obrazów imgNextBinary i
% /* imgBinary
% /* imgSegm - obraz img Sum po segmentacji funkcja bwlabel
% /* numObjects - ilość znalezionych na klatce obiektów
% /* graindata - tablica struktur przechowująca dane otrzymane z
% /* funkcji regionprops()
% /* imgMesh - obraz imgSegm po usunięciu obiektów o polu mniejszym niż
% /* zadany prog
% /* imgSegmLabeled - obraz po segmentacji na który wpisujemy wartości
% /* współczynników takich jak ilość osób itp.
% /* imgFinal - końcowy obraz będący złożeniem wybranych przez
% /* użytkownika klatek
% /* prevGraindata - zmienna przechowująca tablice graindata z
% /* poprzedniej iteracji głównej petli, umożliwia to porównie tych
% /* informacji z dwóch kolejnych klatkach co jest konieczne w
% /* algorytmie śledzenia
% /* prevImgSegm - przechowywany obraz po segmentacji z poprzedniej
% /* klatki, również konieczne przy segmentacji
% /* Stats - struktura przechowująca wyliczone wartości wszystkich
% /* współczynników kształtu i momentów geometrycznych dla każdego
% /* obiektu z każdej klatki filmu
% /* cyferki - wektor przechowujący liczby, które zostaną zapisane na
% /* klatce
% /*
% /*
% /*
% /* Uwagi: */
% /* Pliki konieczne do uruchomienia tego skryptu. */
% /* config_cab.m
% /* shapeCoef.m
% /* progressBar.m
% /* putNumsToImg.m
% /* M_centralny.m
% /* W7.m
% /* perimeter.m
% /* literki.mat
% /* statistica.m
% /* M1.m
% /* M2.m
% /* M3.m
% /* M4.m
% /* M7.m
% /* M8.m
% /* W3.m
% /* m.m

```

```

% /* */
% /* Autor: */
% /* Grzegorz Caban, rok III AiR */
% /* Paweł Gutowski, rok III AiR */
% /* */
% /* Ostatnia modyfikacja: */
% /* 5 stycznia 2007 */
% /* */
% /*****/

```

Opis funkcji śledzącej blobTracker:

```

% ****
%
% Funkcja: blobTracker
%
% wywołanie:
% [newGraindata, numOfDeadBubbles] = blobTracker( X, Xprev, grain,
%                                               prevGrain, config);
%
% Przeznaczenie:
% Funkcja śledzi obiekty zlokalizowane na kolejnych klatkach filmu.
% W ramach śledzenia zostają w odpowiedni sposób przeniesione do
% obiektów z klatki aktualnej informacje o ilości osób stanowiących
% obiekt w klatce poprzedniej.
% Działanie polega na przeanalizowaniu informacji z dwóch kolejnych
% klatek filmu. Celem jest skojarzenie obiektów z klatki aktualnej z
% obiektami z klatki poprzedniej. Skojarzenie oznacza, że są to te
% same, częściowo przesunięte obiekty.
% Obiekty z dwóch kolejnych klatek filmu zostają
% uznane za ten sam, przemieszczający się obiekt,
% jeżeli znaczne ich części się pokrywają.
% Funkcja musi być wywoływana po kolei, dla każdej klatki filmu,
% zaczynając
% od drugiej.
%
% Argumenty funkcji:
% X - Macierz o wymiarach [m n] - aktualnie przetwarzana
% klatka filmu, otrzymana po przetworzeniu i
% posegmentowaniu.
% Każdy obiekt jak i tło muszą mieć inny odzien szarości.
% Xprev - Poprzednia klatka filmu o wymogach j.w.
% grain - Tablica struktur zawierających informacje o każdym z
% obiektów znalezionych w aktualnej klatce.
% Rozmiar tablicy odpowiada ilości obiektów otrzymanych
% w wyniku segmentacji.
% prevGrain - Tablica struktur znalezionych w poprzedniej klatce.
% config - Struktura zawierające informacje konfiguracyjne
% algorytmu zliczania osób
%
% Funkcja zwraca:
% newGraindata - Tablica struktur zawierających informacje o każdym z
% obiektów znalezionych w aktualnej klatce.
% W odróżnieniu od zmiennej wejściowej "grain",
% "newGraindata" jest strukturą poszerzoną -
% zawiera szereg informacji dodatkowych
% numOfDeadBubbles - Ilość ludzi składających się na obiekty,
% których śledzenie zostało
% zakończone w aktualnej klatce. W praktyce jest to
% ilość osób która
% w porównaniu do klatki poprzedniej zeszła z pola
% widzenia
%

```

```

%
% Używane funkcje:
%     blobTrackerGutek korzysta z podstawowych funkcji środowiska MatLab,
%     głównie operujących na macierzach i liczbach.
%
% Używane zmienne:
%     m_wspolnosci - "Macierz wspólności", przechowująca informację o tym w
%                    jakim
%                    stopniu obiekty z klatki aktualnej odpowiadają
%                    obiektom z klatki
%                    poprzedniej (nakładają się).
%                    Każdy wiersz odpowiada kolejnemu obiektowi z
%                    klatki poprzedniej, każda kolumna obiektowi z
%                    klatki aktualnej. Indeks wiersza/kolumny pokrywa
%                    się z indeksem obiektu z klatki
%                    poprzedniej/aktualnej.
%                    Element o współrzędnych i,j jest liczbą
%                    określającą wspólną powierzchnię (wyrażoną w
%                    pikselach): i-tego obiektu z klatki poprzedniej,
%                    oraz j-tego obiektu z klatki aktualnej.
%
%     numOfPrevs - Ilość obiektów na klatce poprzedniej
%                  (ilość wierszy macierzy "m_wspolnosci")
%     numOfActs - Ilość obiektów na klatce aktualnej
%                  (ilość kolumn macierzy "m_wspolnosci")
%     newX       - Klatka (macierz) powstała z połączenia klatek X oraz
%                  prevX
%                  Wartość każdego piksela jednoznacznie określa czy
%                  jest to:
%                  a) tło w obrazie X oraz prevX
%                  b) tylko część i-tego obiektu z prevX lub
%                     tylko część j-tego obiektu z X
%                  c) jednocześnie i-ty obiekt z prevX oraz j-ty
%                     obiekt z X (nałożone obiekty)
%     koloryPrev - Tablica zawierająca liczby odpowiadające
%                  odcieniom szarości kolejnych obiektów z poprzedniej
%                  klatki
%                  (długość tablicy jest równa ilości wierszy
%                  macierzy m_wspolnosci).
%                  Zmienna potrzebna do utworzenia "m_wspolnosci".
%     kolory     - Tablica zawierająca liczby odpowiadające
%                  odcieniom szarości kolejnych obiektów z aktualnej
%                  klatki
%                  (długość tablicy jest równa ilości kolumn
%                  macierzy m_wspolnosci).
%                  Zmienna potrzebna do utworzenia "m_wspolnosci".
%
% Autor:
%     Paweł Gutowski, rok III AiR
%
% Ostatnia modyfikacja:
%     07.01.2007 r.
%
% *****

```

Opis funkcji shapeCoef():

```

/*****
/*
/* shapeCoef()
/*
/*
*/
*/

```

```

/* Wywołanie:
/*     [rW7, rArea] = shapeCoef(grain, config);
/*
/*
/* Przeznaczenie:
/*     Funkcja shapeCoef określa ile osób reprezentuje dany obiekt
/*     otrzymany z segmentacji.
/*
/*
/* Argumenty funkcji:
/*     struct grain - jest to Matlabowska struktura zwrócona
/*     przez funkcje regionprops() zawierająca wybrane informacje
/*     na temat każdego znalezionego obiektu
/*
/* Funkcja zwraca:
/*     int rW7 - jest to ilość osób będąca w danym obiekcie
/*     na podstawie współczynnika kształtu W7
/*     int rArea - jest to ilość osób będąca w danym obiekcie
/*     na podstawie pola danego obiektu
/*
/* Używane funkcje:
/*     W7 - funkcja obliczająca współczynnik W7
/*
/*
/*
/* Autor:
/*     Grzegorz Caban, rok III AiR
/*
/*
/* Ostatnia modyfikacja:
/*     15 grudnia 2006
/*
/*
/*****

```

Funkcje obliczające momenty geometryczne:

```

% /*****
% /*
% /* Funkcja: m
% /* Wywołanie:
% /*     r = m(p,q,x);
% /*
% /*
% /* Przeznaczenie:
% /*     Oblicza moment geometryczny zadanego rzędu
% /*
% /*
% /* Argumenty funkcji:
% /*     p - określa rząd momentu zwykłego względem zmiennej x
% /*     q - określa rząd momentu zwykłego względem zmiennej y
% /*     X - wejściowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /*     r - wartość momentu
% /*
% /*
% /* Autor:
% /*     Paweł Gutowski, rok III AiR
% /*
% /*
% /* Ostatnia modyfikacja:
% /*     15 grudnia 2006
% /*
% /*
% /*****

% /*****

```

```

% /* */
% /* M_centralny */
% /* */
% /* Przeznaczenie: */
% /* Oblicza moment centralny zadanego rzędu */
% /* */
% /* Wywołanie: */
% /* r = M_centralny(p,q,X); */
% /* */
% /* */
% /* Argumenty funkcji: */
% /* p - określa rząd momentu zwykłego względem zmiennej x */
% /* q - określa rząd momentu zwykłego względem zmiennej y */
% /* X - wejsciuowy zbinaryzowany obraz obiektu */
% /* */
% /* Funkcja zwraca: */
% /* r - wartość momentu */
% /* */
% /* Używane funkcje: */
% /* m - funkcja obliczająca moment zwykły */
% /* */
% /* */
% /* */
% /* Autor: */
% /* Paweł Gutowski, rok III AiR */
% /* */
% /* Ostatnia modyfikacja: */
% /* 15 grudnia 2006 */
% /* */
% /* **** */

```

```

% /* **** */
% /* */
% /* M1 */
% /* */
% /* Przeznaczenie: */
% /* Oblicza moment M1 */
% /* */
% /* Wywołanie: */
% /* */
% /* r = M1(X); */
% /* */
% /* Argumenty funkcji: */
% /* X - wejsciuowy zbinaryzowany obraz obiektu */
% /* */
% /* Funkcja zwraca: */
% /* r - wartość momentu */
% /* */
% /* Używane funkcje: */
% /* m - funkcja obliczająca moment zwykły */
% /* M_centralny - funkcja obliczająca moment centralny */
% /* */
% /* */
% /* */
% /* Autor: */
% /* Paweł Gutowski, rok III AiR */
% /* */
% /* Ostatnia modyfikacja: */
% /* 15 grudnia 2006 */
% /* */
% /* **** */

```



```

% /*****
% /*
% /* M2
% /*
% /* Przeznaczenie:
% /* Oblicza moment M2
% /*
% /* Wywołanie:
% /*
% /* r = M2(X);
% /*
% /* Argumenty funkcji:
% /* X - wejsciowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartosc momentu
% /*
% /* Uzywane funkcje:
% /* m - funkcja obliczajaca moment zwykly
% /* M_centralny - funkcja obliczajaca moment centralny
% /*
% /*
% /*
% /* Autor:
% /* Pawel Gutowski, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% /*****/

```

```

% /*****
% /*
% /* M3
% /*
% /* Przeznaczenie:
% /* Oblicza moment M3
% /*
% /* Wywołanie:
% /*
% /* r = M3(X);
% /*
% /* Argumenty funkcji:
% /* X - wejsciowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartosc momentu
% /*
% /* Uzywane funkcje:
% /* m - funkcja obliczajaca moment zwykly
% /* M_centralny - funkcja obliczajaca moment centralny
% /*
% /*
% /*
% /* Autor:
% /* Pawel Gutowski, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% /*****/

```

```

% /*****
% /*
% /* M4
% /*
% /* Przeznaczenie:
% /* Oblicza moment M4
% /*
% /* Wywołanie:
% /*
% /* r = M4(X);
% /*
% /* Argumenty funkcji:
% /* X - wejściowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartość momentu
% /*
% /* Używane funkcje:
% /* m - funkcja obliczająca moment zwykły
% /* M_centralny - funkcja obliczająca moment centralny
% /*
% /*
% /*
% /* Autor:
% /* Paweł Gutowski, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% *****/

```

```

% /*****
% /*
% /* M7
% /*
% /* Przeznaczenie:
% /* Oblicza moment M7
% /*
% /* Wywołanie:
% /*
% /* r = M7(X);
% /*
% /* Argumenty funkcji:
% /* X - wejściowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartość momentu
% /*
% /* Używane funkcje:
% /* m - funkcja obliczająca moment zwykły
% /* M_centralny - funkcja obliczająca moment centralny
% /*
% /*
% /*
% /* Autor:
% /* Paweł Gutowski, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% *****/

```

```

% /*****
% /*
% /* M8
% /*
% /* Przeznaczenie:
% /* Oblicza moment M8
% /*
% /* Wywołanie:
% /*
% /* r = M8(X);
% /*
% /* Argumenty funkcji:
% /* X - wejsciowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartosc momentu
% /*
% /* Uzywane funkcje:
% /* m - funkcja obliczajaca moment zwykly
% /* M_centralny - funkcja obliczajaca moment centralny
% /*
% /*
% /*
% /* Autor:
% /* Pawel Gutowski, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% /*****/

```

```

% /*****
% /*
% /* perimeter
% /*
% /* Przeznaczenie:
% /* Oblicza obwod danego obiektu
% /*
% /* Wywołanie:
% /*
% /* r = perimeter(X);
% /*
% /* Argumenty funkcji:
% /* X - wejsciowy zbinaryzowany obraz obiektu
% /*
% /* Funkcja zwraca:
% /* r - wartosc obwodu
% /*
% /* Uzywane funkcje:
% /* bwboundaries - zwraca wspolrzedne puntow na krawedzi obrazu
% /*
% /*
% /*
% /* Autor:
% /* Grzegorz Caban, rok III AiR
% /*
% /* Ostatnia modyfikacja:
% /* 15 grudnia 2006
% /*
% /*****/

```

```

% /*****/

```

```

% /* */
% /* W7 */
% /* */
% /* Przeznaczenie: */
% /*     Oblicza współczynnik W7 */
% /* */
% /* Wywołanie: */
% /* */
% /*     r = W7(X); */
% /* */
% /* Argumenty funkcji: */
% /*     X - wejściowy zbinaryzowany obraz obiektu */
% /* */
% /* Funkcja zwraca: */
% /*     r - wartość W7 */
% /* */
% /* */
% /* */
% /* Autor: */
% /*     Paweł Gutowski, rok III AiR */
% /* */
% /* Ostatnia modyfikacja: */
% /*     15 listopada 2006 */
% /* */
% /* **** */

```

```

% /* **** */
% /* */
% /* W3 */
% /* */
% /* Przeznaczenie: */
% /*     Oblicza współczynnik W3 Malinowskiej */
% /* */
% /* Wywołanie: */
% /* */
% /*     r = W3(X); */
% /* */
% /* Argumenty funkcji: */
% /*     X - wejściowy zbinaryzowany obraz obiektu */
% /* */
% /* Funkcja zwraca: */
% /*     r - wartość W3 */
% /* */
% /* Używane funkcje: */
% /*     perimeter - oblicza długość obwodu danego obiektu */
% /* */
% /* */
% /* */
% /* Autor: */
% /*     Grzegorz Caban, rok III AiR */
% /* */
% /* Ostatnia modyfikacja: */
% /*     15 listopada 2006 */
% /* */
% /* **** */

```

```

% /* **** */
% /* */
% /* statistica */
% /* */
% /* Przeznaczenie: */

```


10. DODATEK D. Spis zawartości dołączonego nośnika (CD ROM)

Płyta zawiera:

- Katalog program – ze wszystkimi m-plikami programu.
- Katalog „Filmy wejściowe” – filmy na których pracuje algorytm. UWAGA: przed uruchomieniem programu należy przekonwertować te filmy do formatu nie skompresowanego. Można użyć np. Virtual Duba i kodeka xVid zawartego na płycie.
- Katalog „filmy wyjściowe” – filmy zawierające efekt pracy algorytmu (nazwy odpowiadają filmom wejściowym). Wewnątrz znajduje się katalog „filmy do zaprezentowania”, w którym umieszczono krótkie sekwencje z widocznymi ciekawymi zachowaniami algorytmu.
- Katalog „Prezentacja” – katalog z prezentacją w PowerPoint, oraz plikami niezbędnymi do jej prawidłowego funkcjonowania (np. filmy prezentacyjne)
- Katalog „materiały elektroniczne” – zawiera pliki z dokumentami w wersji PDF, podanymi w dziale „Literatura”.
- Katalog „kodek xvid” – katalog z kodekiem niezbędnym do uruchomienia filmów zawartych na płycie.
- Plik SprawozdanieKAM3.doc oraz pdf – Plik tekstowy z treścią sprawozdania

Wszystkie filmy zawarte na płycie DVD zostały skompresowane kodekiem Xvid.

Przed przystąpieniem do pracy z programem należy zdekompresować filmy wejściowe!
