

Dokumentacja końcowa

Przetwarzanie obrazów

Temat:

Tematem projektu było rozpoznawanie loga firmy AMD na obrazie. Pierwszym elementem do rozpoznania był ciąg liter *A, M, D* ułożonych obok siebie. Drugim elementem był znak rozpoznawczy firmy leżący w bliskim otoczeniu szukanego ciągu bądź występujący samotnie.

Projekt został zrealizowany w języku Java.



Przebieg rozpoznawania:

1. Pierwszym etapem jest wstępna poprawa jakości obrazu poprzez zastosowanie maski wyostrzającej (Unsharp Mask). Dzięki temu zabiegowi uzyskujemy poprawę kontrastu krawędzi obrazu. Ze względu na chęć znalezienia na obrazie liter, zastosowanie filtrów takich jak rankingowe utrudniało identyfikację obiektów dlatego nie są używane.
2. Kolejnym etapem jest wyodrębnienie obiektów z obrazu. Zważywszy na charakter szukanego tekstu, który występuje w barwach białej oraz czarnej, zastosowano wydzielenie na podstawie prostego progowania. Dodatkowo celem znalezienia symbolu wydzielana jest także barwa w przybliżeniu zielona, na podstawie analizy składowych pikseli w paletce HSV, w założeniu że znacznie różni się od tła pod względem jasności.
3. Następnym krokiem rozpoznania jest segmentacja binarnego obrazu. Segmenty obiektów (*białe* jak i *czarne*) wydzielane są do osobnych segmentów. Sama segmentacja polega na dodawaniu poszczególnych pikseli do segmentów przechodząc piksele po kolei w poziomie. W przypadku tej samej wartości pikseli sąsiadujących na planie krzyża, łączone są one w segmenty.
4. Po wydzieleniu segmentów, obliczane są wstępne ich parametry takie jak współrzędne *bounding box'a* oraz pole i obwód. Odrzucone zostają *zbyt małe* segmenty, a dla odpowiednich wyliczane są momenty geometryczne i niezmienniki momentowe.
5. Następnie elementy prawdopodobne do ułożenia loga są wybierane na podstawie porównania analizowanych cech z zakresem wzorcowym. Elementom przypisywane są przynależności, na podstawie dopasowania, odpowiednio do szukanых obiektów takich jak litera A, litera M, litera D, dolna część symbolu, górna część symbolu, jeżeli w ogóle pasują. Cechy używane celem identyfikacji to M7, M3, W7, W8, W9, M1.
6. Ostatnim etapem rozpoznania jest znalezienie odpowiednich segmentów w określonym ułożeniu. Poszczególne elementy są porównywane ze sobą na podstawie rozmiaru, odległości, linii na której leżą (kierunku) oraz kolejności. Akceptowane są ciągi liter AMD, odosobnione symbole oraz ciąg liter w sąsiedztwie symbolu.

Działanie programu:

Ze względu na zastosowanie niezmienników w celu znalezienia obiektów na obrazie, powinny zostać znalezione obiekty obrócone, rozciągnięte czy przekrzywione. Ciąg liter występować może zarówno w pionie, poziomie jak i na skos, a symbol dookoła niego.

Zdarza się, że widoczne na obrazie wyraźne różnice w barwie symbolu czy tekstu nie są poprawnie interpretowane np. w przypadku zacienionego obszaru czy odbłasku na obiekcie. Niekiedy wewnątrz segmentów widoczne są 'dziury' pixel'owe na co nieznacznie pomaga filtracja medianowa, jednakże rozmywa ona też ostre krawędzie liter dlatego nie jest stosowana.

Do poprawnego działania programu nie jest przewidziana konkretna rozdzielczość obrazu, aczkolwiek proces segmentacji dla obrazów powyżej 2 Mpx zajmuje dużo czasu.

Specyfika działania:

W celu sprawdzenia działania programu należy go uruchomić, wybrać plik obrazu poprzez naciśnięcie przycisku '*Choose image to recognition*'. Następnie można sprawdzić wgrany obraz poprzez jego wyświetlenie, przycisk '*Paint first loaded image*'. Proces rozpoznania inicjuje przycisk '*Single recognition*', po naciśnięciu którego zostaną wyświetlone kolejne okienka z obrazami:

- *Marked segments* - na którym zaznaczone są zidentyfikowane segmenty:
 - Czerwone - Litera A
 - Zielone - Litera M
 - Niebieskie - Litera D
 - Żółte - Dolna część symbolu
 - Błękitne - Górna część symbolu
- *Segmented image* - Obraz binarny po operacji progowania
- *Found AMD logos* - Obraz początkowy ze znalezionymi obiektami, zaznaczonymi przez wpisanie w czerwony prostokąt.

Znaczenie poszczególnych opcji programu:

1. *Coose image to recognition* - wybór obrazu
2. *Paint first loaded image* - narysowanie wybranego obrazu
3. *Single recognition* - rozpoznanie na wybranym obrazie
4. *Single recognition with inversion* - '*podwójne*' rozpoznanie używając wydzielonych segmentów z obrazu odwróconego
5. *Clear segmented images* - używane do iteracyjnej segmentacji do wyczyszczenia listy znalezionych segmentów
6. *Iterate segmentation* - używane do obliczania minimalnych, maksymalnych i średnich wartości momentów dla prostych obrazków. Na koniec wyświetla zebrane dane o segmentach w celu kontroli poprawności.
7. *Calculate range of moments* - obliczenie ww. wartości i wypisanie w konsoli w postaci definicji zmiennych.

Przykłady:

