

Wykrywanie kostek

Kamil Piechowiak
nr albumu 136780

Daniel Nowak
nr albumu 136775

1 Wstęp

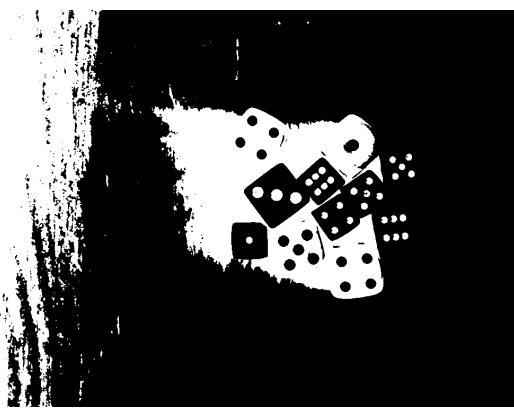
Nasz program do wykrywania kostek znajduje ściany kostek i zaznacza je na zdjęciu wynikowym. Analiza przebiega w kilku krokach. W pierwszym etapie dokonujemy wstępnego przetwarzania zdjęcia, tak aby ułatwić działanie dalszym fazom. Następnie wykrywamy regiony, które potencjalnie mogą być oczkami kostek. Proces ten wykonujemy zarówno dla oczek czarnych, jak i białych. Dalsze postępowanie wykonywane jest osobno dla dwóch kolorów oczek. Do grup potencjalnych oczek dopasowujemy wzorce będące ściankami kostek i wybieramy te o najlepszym dopasowaniu. Następnie wykryte ścianki zaznaczane są na zdjęciu przy użyciu różnych kolorów. Dla poszczególnych liczba oczek są to: 1 - czerwony, 2 - żółty, 3 - zielony, 4 - cyjanowy, 5 - niebieski, 6 - magenta.

2 Wykrywanie kostek

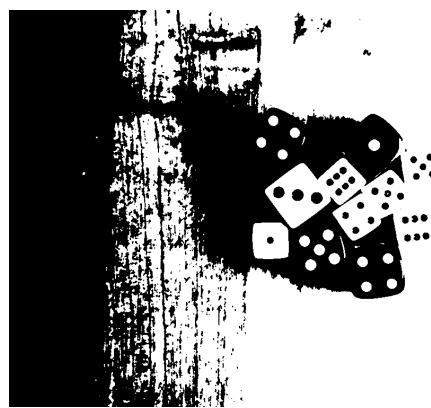
Poniżej zostały przedstawione kolejne fazy działania programu.

2.1 Wstępne przetwarzanie

Na wczytane zdjęcie aplikowany jest filtr medianowy 3×3 w celu redukcji szumu. Następnie zdjęcie konwertowane jest z przestrzeni rgb na skalę szarości i zwiększa kontrast, aby lepiej generalizować się do przypadków z różnym natężeniem światła. Na drodze eksperymentalnej ustaliliśmy jako wartość minimalną 3 percentyl rozkładu na skali szarości, a jako wartość maksymalną 97 percentyl. Następnie wykonywane jest progowanie, tak aby odróżnić od reszty obrazu potencjalne oczka. Wykonywane jest ono osobno dla białych oczek (wartość > 0.4) i czarnych oczek (wartość < 0.3) - rysunek (1). Każdy z uzyskanych obrazów jest przetwarzany osobno w dalszym etapie.



(a)

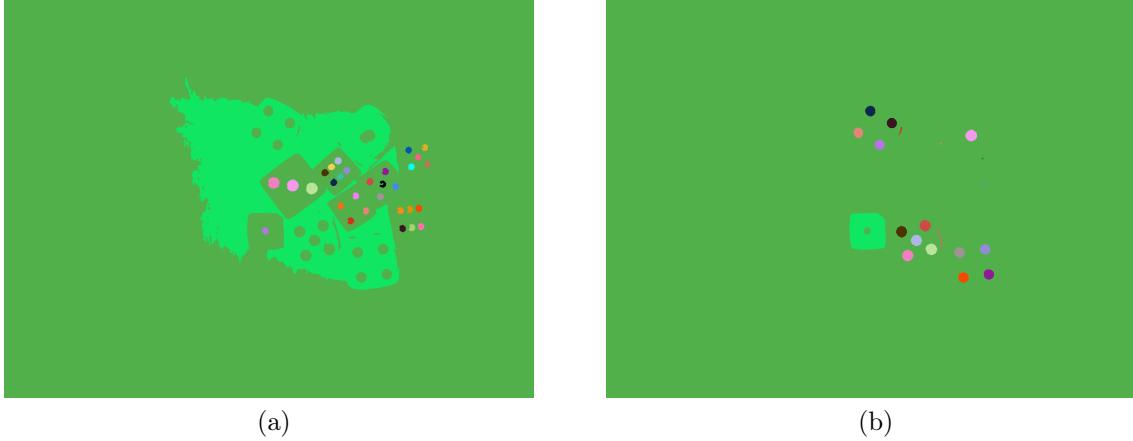


(b)

Rysunek 1: Progowanie w celu wykrycia ciemnych (a) i jasnych obszarów (b)

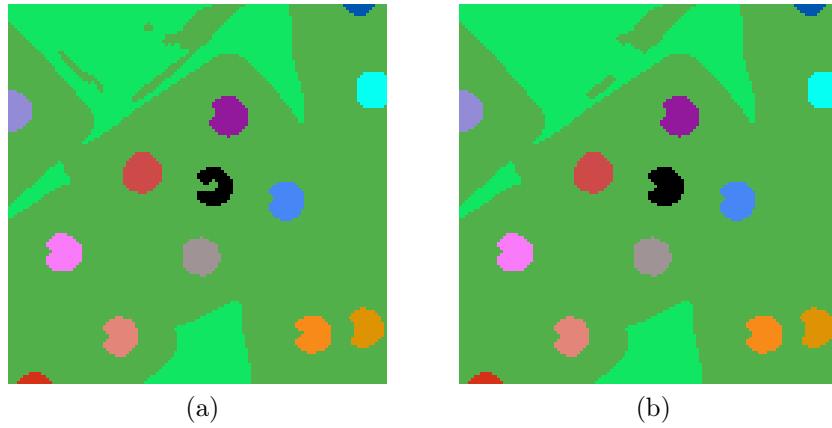
Z obrazu po progowaniu tworzony jest graf G . Każda spójna składowa obrazu po progowaniu staje się jednym wierzchołkiem wynikowego grafu, a krawędziami połączone są sąsiadujące składowe. Wagą

krawędzi zostają obliczone na podstawie wartości krawędzi między regionami na obrazie po zastosowaniu filtra krawędziowego. Wierzchołki, między którymi krawędź ma wartość poniżej progu łączone są w jeden wierzchołek. Dzięki temu zmniejsza się liczba regionów, co przyspiesza późniejszą analizę. Usuwane są także regiony, które mogłyby zostać zaklasyfikowane jako oczka mimo, że nimi nie są - regiony o okrągłym kształcie, ale o słabo zarysowanych krawędziach. Efekt widoczny jest na rysunku 5



Rysunek 2: Regiony po zastosowaniu operacji łączenia dla ciemnych (a) i jasnych (b) obszarów

Na nowo uzyskane regiony aplikowana jest dylatacja, aby wygładzić obiekty i ułatwić następne usunięcie dziur wewnętrznych regionów zainteresowania. Ostatecznie stosowana jest erozja, aby powrócić z obiektami do ich wcześniejszych rozmiarów - jest to potrzebne w kolejnej fazie dopasowywania wzorca. Rezultat można zobaczyć na rysunku 3



Rysunek 3: Regiony przed (a) i po (b) operacji zamknięcia

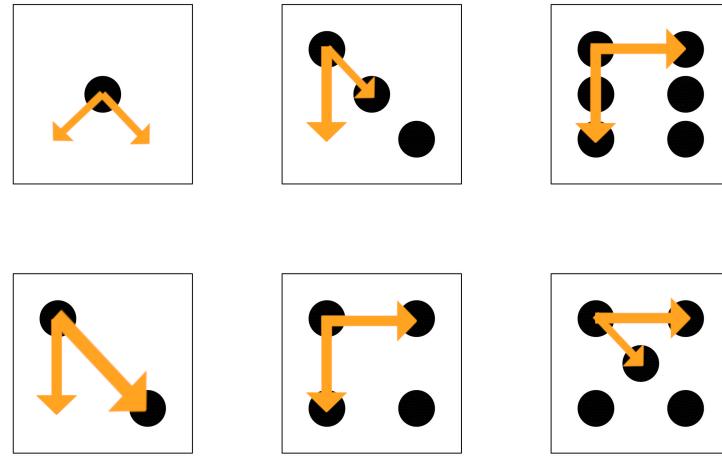
Do dopasowywania wzorców potrzebne jest też sąsiedztwo dla każdego regionu - takie regiony, które razem mogą stworzyć ściankę kostki. Jako sąsiedztwo $s(v)$ danego wierzchołka v uznawane są wszystkie wierzchołki u w grafie G , takie, że najkrótsza ścieżka (co do liczby wierzchołków) z v do u przechodzi przez dokładnie jeden inny wierzchołek. Dodatkowo środki tych regionów mogą być oddalone co najwyżej o $8r$, gdzie r jest szacowanym promieniem koła jako regionu odpowiadającego wierzchołkowi v . Zmniejsza to liczbę par regionów do rozważenia, a jednocześnie umożliwia wykrywanie ścianki z dwoma oczkami (minimalna odległość między oczkami jest na tej ściance największa).

2.2 Przygotowanie do dopasowania wzorców

Aby wykryć ścianki z 4, 5 lub 6 oczkami, stosowane jest dopasowanie wzorców dla grup trzech wierzchołków v, a, b , takich że $a, b \in s(v)$. Mapowane są one na trzy oczka kostek, tak aby uzyskane wektory na wzorcu były liniowo niezależne. Zastosowane wektory można zobaczyć na rysunku 4. Umożliwia to

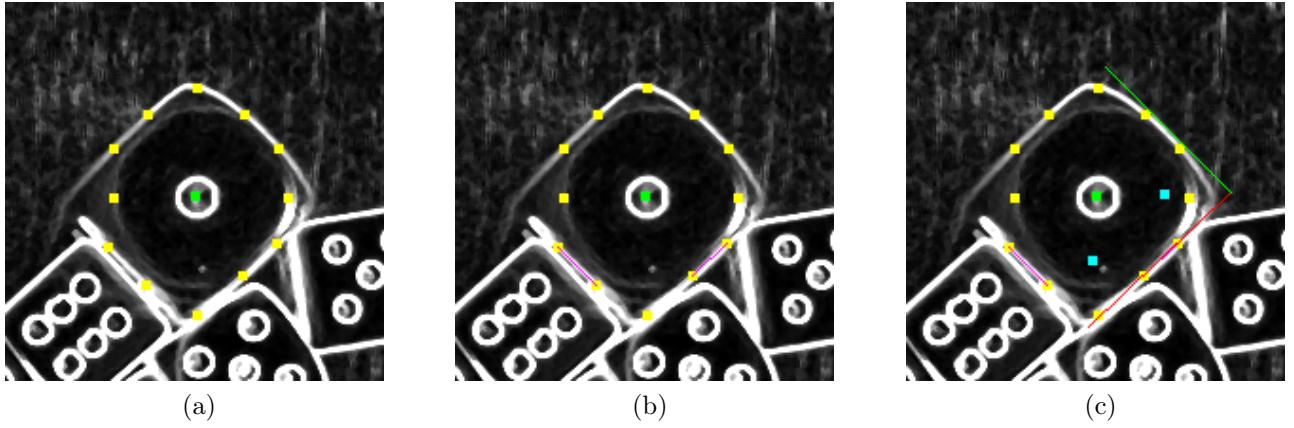
jednoznaczną transformację wzorca do przestrzeni obrazka.

Dla pozostałych oczek niemożliwe jest uzyskanie wektorów liniowo niezależnych. W celu wykrycia ścianek z 2 lub 3 oczkami, rozpatrywane są pary wierzchołków v, a , takie że $a \in s(v)$. Aby uzyskać jednoznaczną transformację, tworzony jest sztuczny wierzchołek o ustalonym położeniu na wzorcu. Po przez zmiany jego odległości od prostej zawierającej środki wierzchołków a i v sprawdzane są różne możliwości dopasowania wzorca.



Rysunek 4: Wektory pomiędzy punktami użytymi do dopasowania poszczególnych wzorców

Natomiast dla ścianek z jednym oczkiem propozycje ścianek generowane są na podstawie analizy krawędzi obrazu. Posiadamy tu informację tylko o jednym punkcie, więc nieznany jest ani kąt obrotu, ani kąt między sąsiednimi bokami ścianki. W celu zaproponowania ustawienia, ze środka regionu prowadzone są promienie w różnych kierunkach o długości $8r$, gdzie r jest oszacowanym promieniem regionu. Wykrywają one punkty będące lokalnymi maksimami na promieniu o wartości powyżej wartości progowej. Następnie analizowane są średnie wartości krawędzi pomiędzy sąsiadującymi punktami i na podstawie najlepszych proponowane jest ustawienie boków ścianki.



Rysunek 5: Proponowanie dopasowania wzorca dla ścianki z jednym oczkiem: wykrycie punków na krawędziach (kolor żółty) (a), wybór najbardziej nasyconych krawędzi (kolor magenta) (b), propozycja dopasowania wraz z pozycją sztucznych wierzchołków (kolor cyjanowy) (c)

2.3 Dopasowanie wzorców

Dla każdej propozycji dopasowania wzorca do oryginalnego zdjęcia obliczamy wartość funkcji celu, którą chcemy minimalizować. Przyjmijmy oznaczenia, że biały obszar to obszar dominujący w miejscu dopasowania wzorca (potencjalne tło ścianki), a pozostałe obszary - czarne to te, które mają szansę zostać

oczkami kostki.

Pewne rozwiązania od razu odrzucamy, na przykład takie, które nie mają czarnych regionów pod środkami oczek ze wzorca. Dla pozostałych obliczamy wartość funkcji celu jako stosunek niezgadzającej się liczby pikseli pomiędzy wzorcem a obrazem (czarny piksel obrazu i biały wzorca lub odwrotnie) do liczby czarnych pikseli na wzorcu. Im mniejszy ten iloraz tym lepsze dopasowanie.

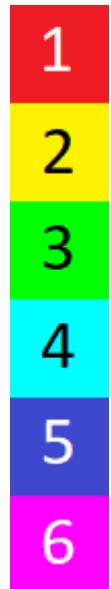
2.4 Wybór rozwiązań

Propozycje dopasowań są sortowane według rosnącej wartości funkcji celu. Wybierane są najlepsze, których wartość funkcji celu jest niższa niż wartość progowa. Po zaakceptowaniu danej propozycji regiony, z których została stworzona zostają oznaczone jako odwiedzone i nie mogą być bazą dla kolejnych propozycji. Wykryte ścianki zaznaczane są na wejściowym zdjęciu korzystając z algorytmu wykrywania konturów marching squares.

3 Wyniki

Wyniki działania programu są widoczne na poniższych zdjęciach

Na zdjęciach łatwych (Rysunek 7) zostały zawarte tylko ścianki jednego typu, na kontrastującym tle, robione od góry. Program radzi sobie na nich bez problemów. Warto zauważyć, że na zdjęciu (b) wykrył też czwórkę w prawym górnym rogu.



Rysunek 6:
Skala



(a)



(b)



(c)



(d)

Rysunek 7: Zdjęcia łatwe

3.1 Średnie

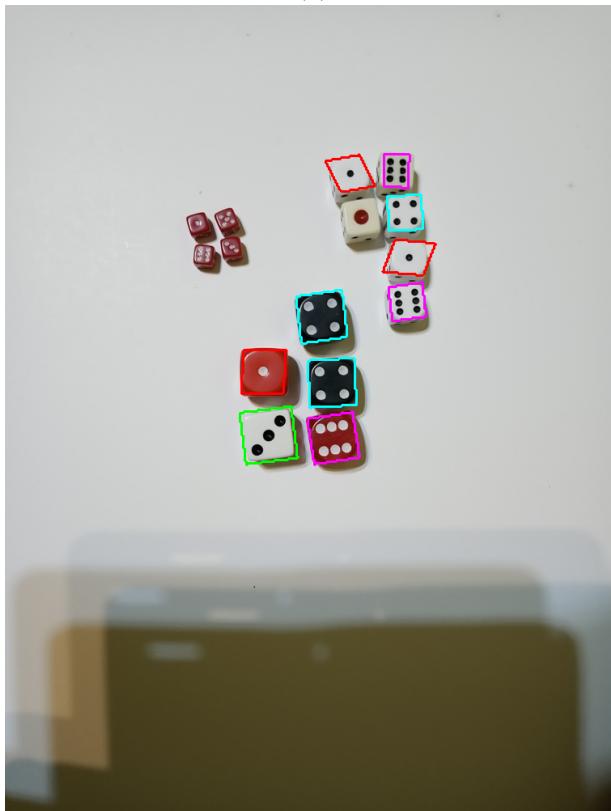
Wybrane zdjęcia średnie (rysunek 8) charakteryzują się bardziej niejednolitym tłem lub większą ilością cieni. Niektóre kostki słabiej kontrastują z otoczeniem. Jednak zdjęcia dalej wykonywane są od góry. Tutaj program także poradził sobie dobrze, ale pojawiły się drobne błędy. Na zdjęciu (b) błędnie zaklasyfikował boczną ściankę jako 4, jest to spowodowane bardzo małym kątem padania. Na zdjęciu (c) nie udało się zaklasyfikować niektórych kostek, gdyż są innego typu niż te, do których przygotowany był program.



(a)



(b)

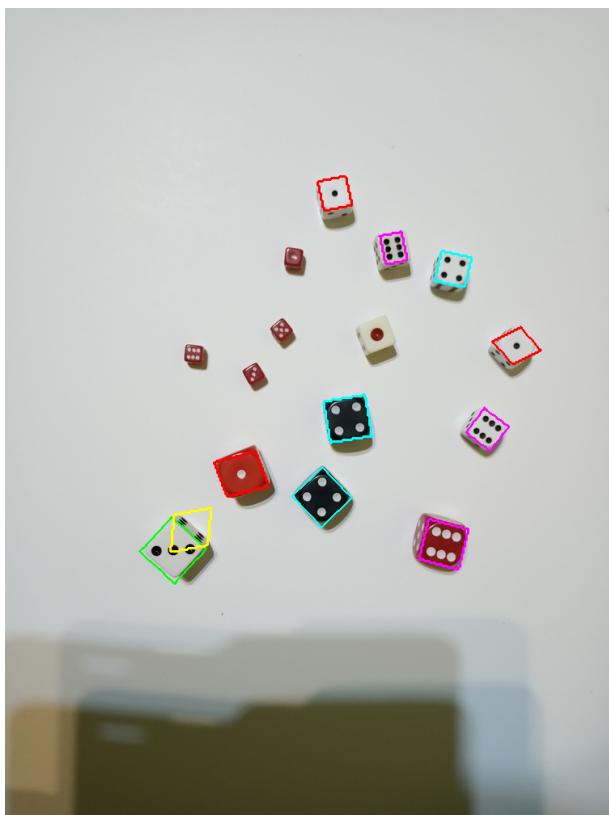


(c)

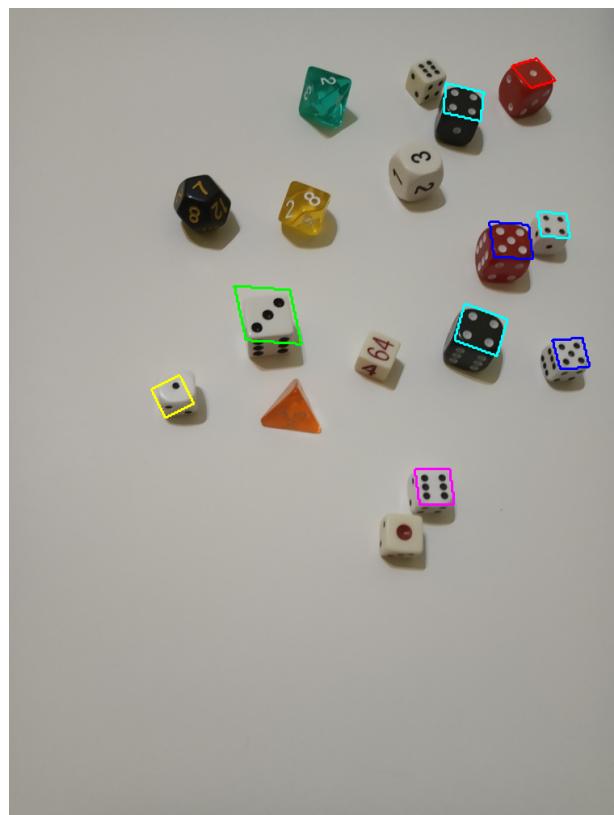


(d)

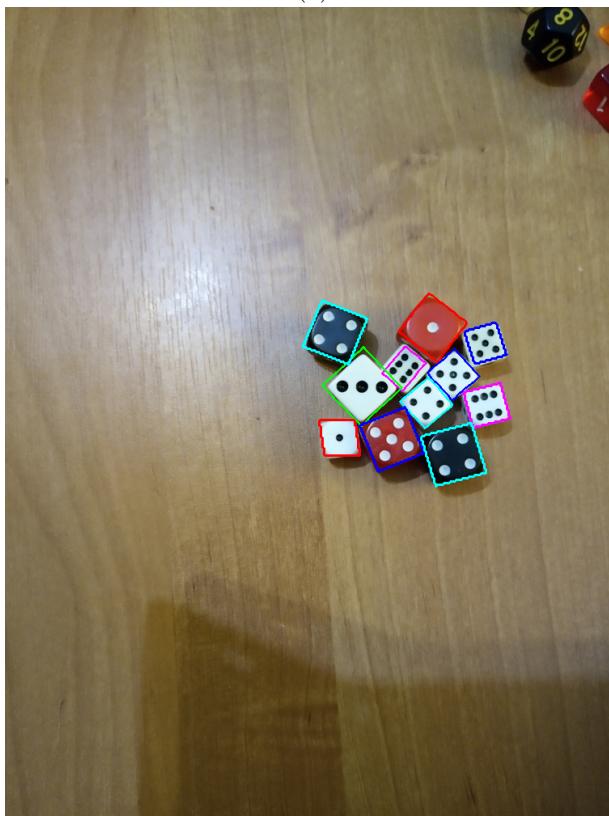
Rysunek 8: Zdjęcia średnie



(a)



(b)



(c)



(d)

Rysunek 9: Zdjęcia trudne



(a)



(b)

Rysunek 10: Zdjęcia trudne 2

3.2 Trudne

Na zdjęciach trudnych (rysunek 9) pojawiły się obiekty innego typu, większe refleksy i kąt robienia zdjęcia odbiegający od pionu. Na zdjęciu (a) została wykryta nadmiarowo dwójka na bocznej ścianie (kolor żółty). Można by tego uniknąć, jeśli sprawdziłoby się jeszcze czy wykryte ścianki nie znajdują się w siebie w zbyt dużym stopniu. Na zdjęciu (b) nie została zaklasyfikowana jedna szóstka - jej oczka zbytnio się ze sobą zlewają, aby program je odróżnił. Błędnie zostały połączone także dwie ścianki jednej z białych kostek i zaklasyfikowane jako dwójka. Natomiast ze zdjęciami (c) i (d) program poradził sobie bardzo dobrze. Zdjęcia trudne zostały pokazane także na rysunku 10. W przypadku (a) została nadmiarowo jedynie wykryta jedynka na ścianie bocznej. Natomiast w przypadku (b) ze względu na nierównomierne oświetlenie nie została wykryta część kostek - program miał problem z odpowiednim segmentowaniem regionów ze względu na cienie, które pojawiły się wewnątrz oczek.

4 Wnioski

Stworzony przez nas program najlepiej rozpoznaje przypadki, w których zdjęcia wykonywane są od góry i nie ma dużych zmian w natężeniu światła. W przypadku zdjęć robionych pod innym kątem czasem błędnie znajdują się zaklasyfikowane ściany boczne. Dzięki dopasowaniu wzorców program nie ma problemów z odróżnianiem kostek od innych obiektów. Podsumowując, program dobrze radzi sobie z pewnymi grupami przypadków, ale stosując techniki klasyczne ciężko zgeneralizować rozwiązanie do większej liczby sytuacji.