**Dokumentacja użytkowa programu OpenCVPingPoint (C++)**

Wchodzący w skład projektu zespołu nr 10: „PingPoint”

Zadaniem programu OpenCVPingPoint jest przetwarzanie obrazu odtwarzanego w czasie rzeczywistym, oraz umożliwienie automatycznego zaliczania punktów w zależności od sytuacji na stole. Program jest bezpośrednio powiązany z aplikacją C#, porozumiewając się w celu wskazania która strona otrzymała punkt. Program może analizować albo obraz podpiętej kamery internetowej do komputera, albo podany plik wideo, z uprzednio nagraną rozgrywką.

Program zakłada że kolor piłki pingpongowej jest **pomarańczowy,** lecz jeżeli jest to konieczne, może on również pracować z piłką **białą,** lecz nie gwarantujemy poprawności działania.

Program został napisany w środowisku Visual Studio Community 2017, korzystając z biblioteki OpenCV w wersji 3.4.1. Program został skompilowany w trybie **Release**, oraz jako program **64-bitowy**. Użytym językiem programowania jest **C++.**

Składa się tylko z jednego pliku: „OpenCVPingPoint.cpp” zawierający całość działania programu. Każda linia kodu w tym pliku została szczegółowo skomentowana w celu łatwiejszego napisana szczegółowej dokumentacji.

Aby uruchomić aplikację OpenCVPingPoint, należy spełnić następujące wymagania wstępne**:**

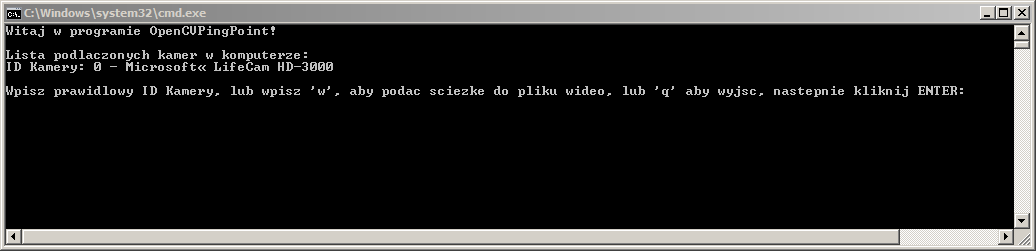
**1 – plik „opencv\_world341.dll”** – główna biblioteka OpenCV, jest dostępna z oficjalnej strony OpenCV. Ściągamy „Win pack” w wersji 3.4.1 tutaj <https://opencv.org/releases.html>, następnie go rozpakowujemy.

Plik znajduje się w katalogu **opencv\build\x64\vc15\bin.** Kopiujemy go do lokalizacji aplikacji.

**2 – Visual Studio 2017 x64 Redistributable –** biblioteki Visual Studio (m.in. plik msvcp140.dll),

ściągamy je tutaj i instalujemy: <https://aka.ms/vs/15/release/vc_redist.x64.exe>

**3 – uruchomiona Aplikacja C# PingPoing w tle** - program nie uruchomi się, jeżeli nie utworzy potoku z aplikacją C# w celu przesyłania sygnałów o zdobytych punktach

Po uruchomieniu programu pokaże się okno wiersza poleceń, które informuje o wszystkich dostępnych opcjach programu, oraz o wykrytych kamerach podpiętych do komputera i przypisanych do nich numerów ID :

UWAGA: Program obsługuje maksymalnie 10 podpiętych kamer (ID od 0 do 9), próba uruchomienia kamery o ID większym niż 9, skończy się niepowodzeniem i uruchomieniem innej kamery.

Po podaniu prawidłowego ID Kamery (w naszym przypadku 0) pojawi się pytanie o szerokość i rozdzielczość kamery. Zalecamy aby rozdzielczość wynosiła co najmniej 800x600, gdyż niższa rozdzielczość uniemożliwia pokazanie interfejsu kalibracji w całości.

Jeżeli natomiast chcemy pracować na wcześniej nagranym materiale, należy wpisać literkę „**w”,** wtedy pojawi się okno dialogowe z możliwością wyboru pliku wideo. Obsługa plików wideo zależy od biblioteki OpenCV, przetestowane zostały filmy z rozszerzeniami \*.avi, oraz \*.mp4.

Po poprawnym uruchomieniu kamery bądź pliku wideo, pojawią się **dwa** dodatkowe okna:

**1 – Obraz na żywo –** jest to obraz podglądowy mający za zadanie sprawdzić poprawność działania programu podczas spotkania, zawiera obraz na żywo kamery/pliku wideo, obraz kalibracji stołu, obraz toru lotu piłki, oraz poglądowe kółka sytuacji na stol2

**2** – **Kalibracja** – zawiera obraz kalibracji piłki (jako tło), narzędzia do kalibracji stołu i piłki, oraz włącznik/wyłącznik automatycznego liczenia punktów

Oba te okna, można dowolnie rozszerzać i zmniejszać. Należy zwrócić uwagę, że interfejs okna kalibracji skaluje się wraz z rozmiarem okna, dlatego też należy je rozszerzyć aby przeczytać tam zawarty tekst.

Gdy mamy otwarty plik **wideo** można naciskać klawisz „**p”** aby zatrzymać lub odtworzyć z powrotem plik wideo, w celu dokonania kalibracji. Nie działa to zbyt precyzyjnie, trzeba naciskać klawisz tak długo, aż plik wideo się zatrzyma/odtworzy. Wynika to z ograniczeń biblioteki OpenCV.

UWAGA: Jeżeli otworzymy plik wideo, w oknie wierszu poleceń, mogą się pojawiać komunikaty typu:

„Failed to parse avi: index was not found

Unexpected list type. Expected: hdrl. Got: movi.”

Można je zignorować, nie są to błędy, lecz mało ważne ostrzeżenie nie mające wpływu na działanie programu.

Aby program działał prawidłowo, należy najpierw przygotować odpowiednie ujęcie do kalibracji, które będzie widoczne w obrazie na żywo, oto przykład którym należy się kierować:



**Ważne elementy ujęcia:**

- stół widoczny bokiem

- wyraźny kontrast pomiędzy piłką a resztą obrazu

- jak najmniejsza ilość szumów

- brak elementów o podobnym kolorze do piłki

- wyraźnie widoczna piłka na stole

- wyraźnie widoczne brzegi stołu

- brak źródeł światła (lampy, żarówki)

Należy zwrócić uwagę, że w zależności od mocy komputera, rozdzielczości filmu, oraz jego liczby klatek na sekundę, płynność obrazu na żywo nie może być zagwarantowana. Wynika to z ciągłej obróbki odczytywanych klatek kamery/pliku wideo w czasie rzeczywistym i jest ograniczeniem biblioteki OpenCV.

Gdy już mamy odpowiednie ujęcie, możemy przejść do **kalibracji**:

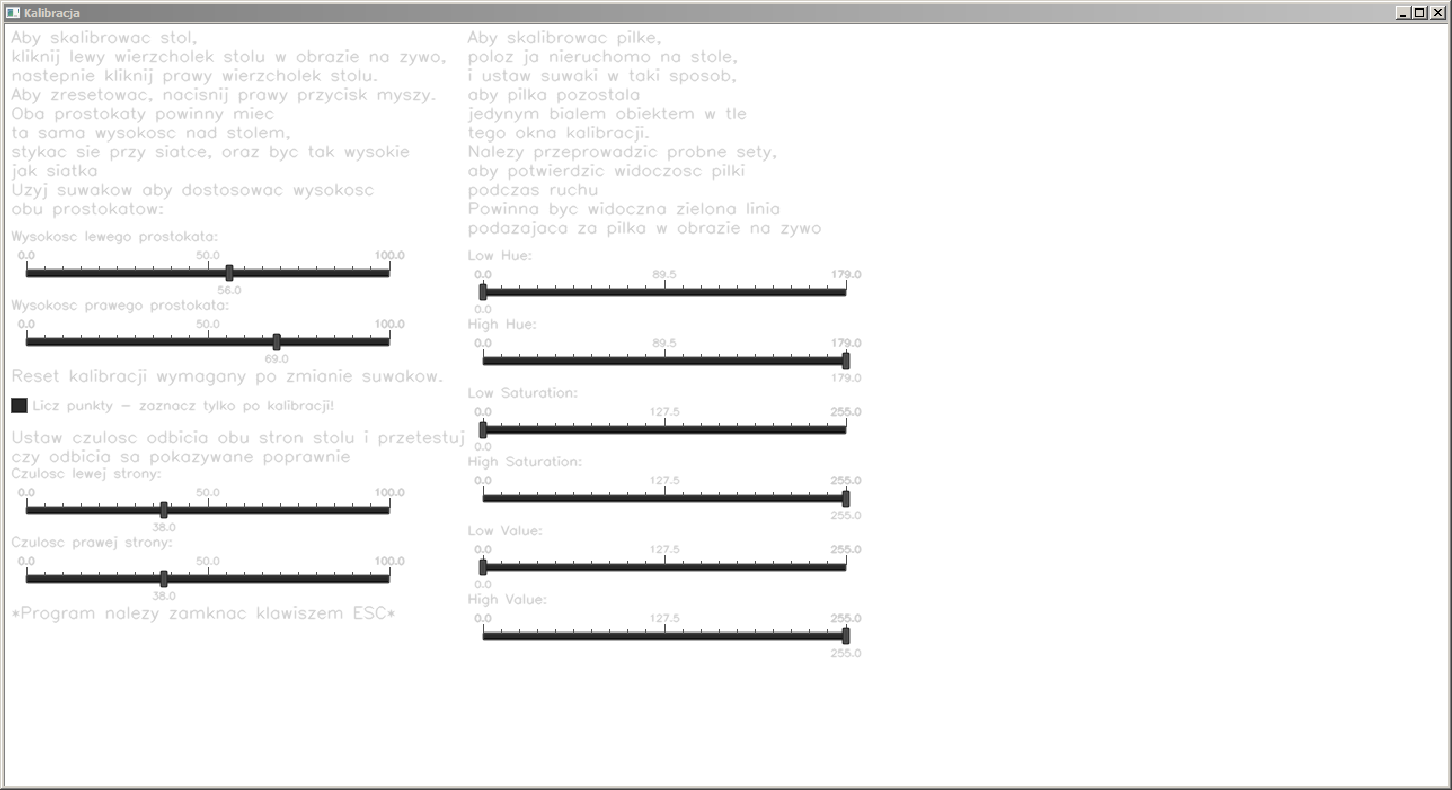
1. Należy zacząć od **kalibracji stołu**. Instrukcje zawarte są w oknie kalibracji po jego lewej stronie.

Kalibracja będzie tym łatwiejsza, im stół jest bardziej prosty. Lewa strona stołu powinna być obejmowana przez zielononiebieski prostokąt, prawa strona przez różowy prostokąt

Po wykonaniu podanych instrukcji, kalibracja stołu wyglądać w następujący sposób:

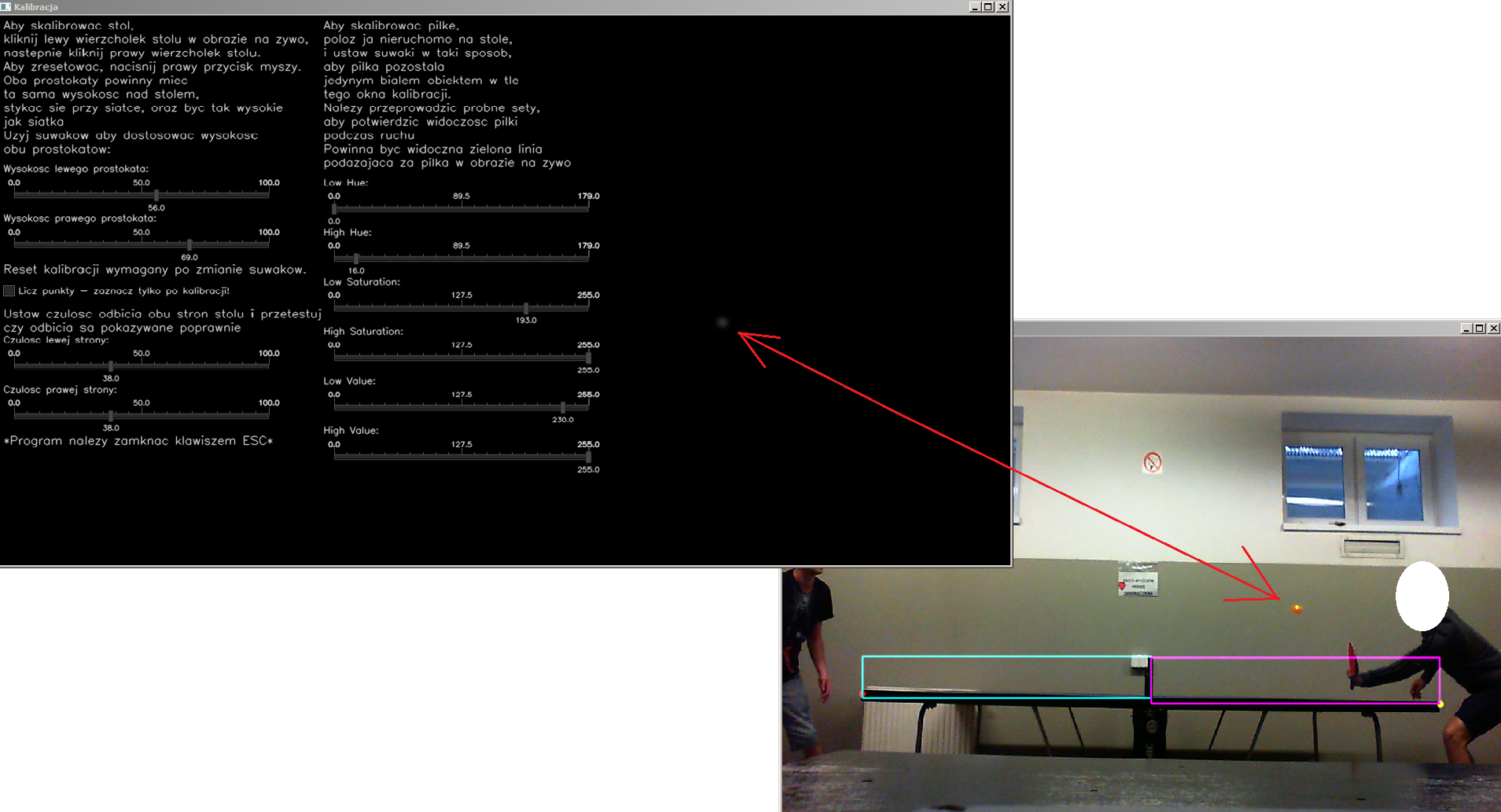


1. Następnie zaczynamy od **kalibracja piłki.** Połóż piłkę na stole nieruchomo, upewnij się że kadr jest nieruchomy. Po prawej stronie znajduje się sześć suwaków. Początkowo zalecamy ustawić suwaki w ten sposób, aby tło było całe białe, czyli wartości suwaków powinny być na przemian najmniejsze oraz największe, tak jak na poniższym przykładzie (0, 179, 0, 255, 0, 255), będą to wartości początkowe:



Gdy już mamy taką sytuację, należy przesuwać suwaki w taki sposób, aby widoczny w tle obraz zawierał tylko jeden biały obiekt - piłkę. Należy zacząć od suwaka **Low Hue**, a następnie po kolei ustawiać pozostałe, aż piłka pozostanie jedynym obiektem. Ustawianie suwaków polega na tym, aby znaleźć jak najbardziej ich skrajną pozycję względem pozycji początkowej, w której piłka nie znika.

Finalnie powinno to wyglądać następująco, można wtedy zauważyć że w oknie poglądowym, widzimy zieloną kropkę w środku piłki, jest to linia toru piłki którą będziemy obserwowali i jest ściśle powiązana z funkcją kalibracji stołu, oraz funkcją zaliczania punktów:



1. Jeżeli wszystko zostało wykonane prawidłowo, należy zaznaczyć opcję **Licz punkty.**

Należy przeprowadzić próbne sety, w celu sprawdzenia poprawności kalibracji. W tym celu w obrazie na żywo będzie pojawiać się zielona linia toru piłki, oraz kolorowe koła kontrolne symbolizujące następujące sytuacje:

- **różowe** koło nad stołem – lewa strona rozpoczęła serw

**- niebieskie** koło nad stołem – prawa strona rozpoczęła serw

- **czerwone** koło nad lewą stroną stołu – nastąpiło odbicie po lewej stronie stołu

- **żółte** koło nad prawą stroną stołu – nastąpiło odbicie po prawej stronie stołu

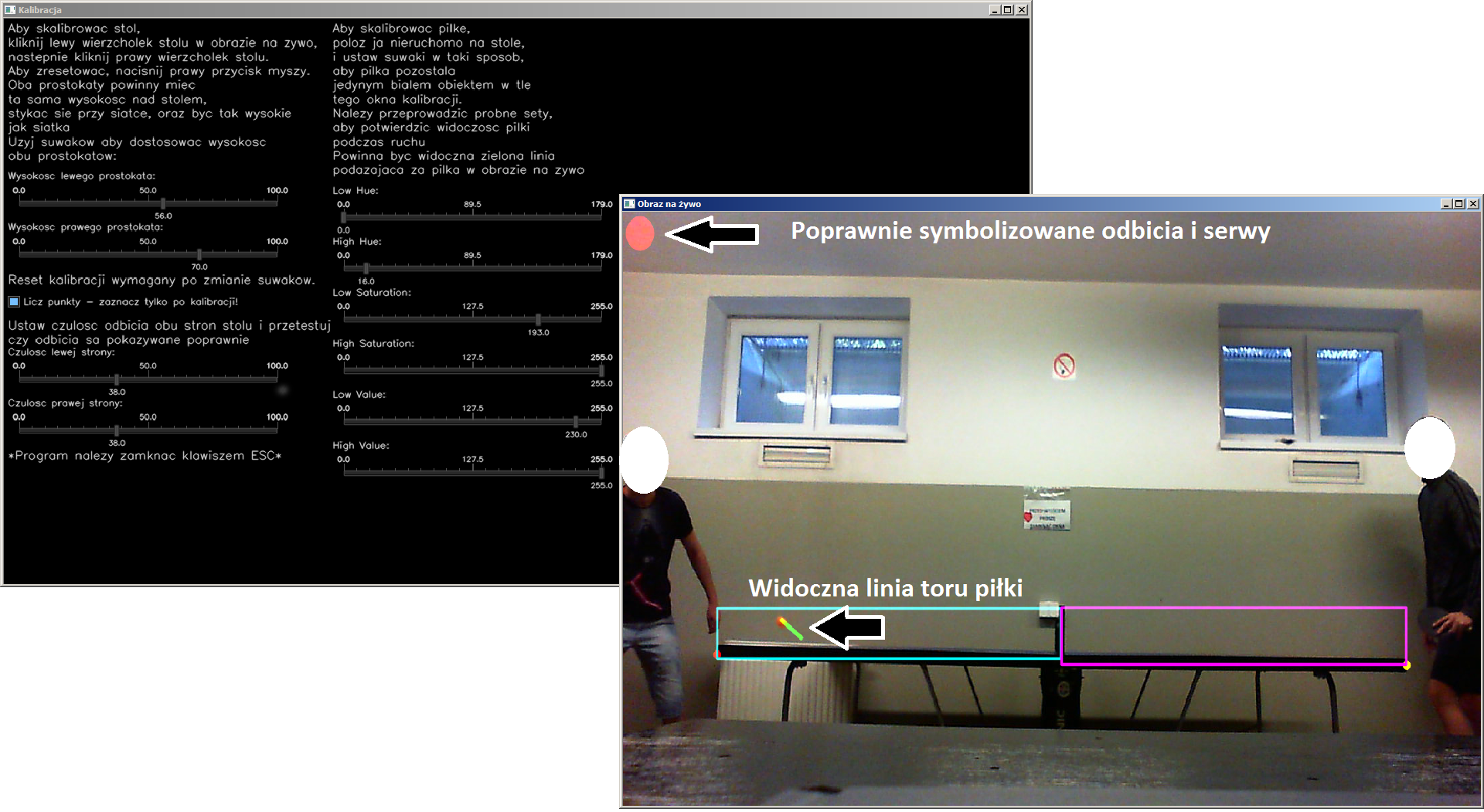
Jeżeli odbicia nie są wykrywane poprawnie na danej stronie stołu, należy zwiększyć czułość odbić dla lewej, lub prawej strony stołu. Suwaki te znajdują się pod opcją Licz punkty.

Czułość ta ustala, jak duża jest tolerancja odległości piłki od stołu podczas odbicia od niej.

1. Jeżeli testy przebiegły pomyślnie, należy zacząć mecz w aplikacji C#, oraz rozpocząć pierwszy set**.** Punkty powinny być automatycznie liczone i uaktualniane w aplikacji C#. W celu unikania fałszywie zaliczanych punktów, warto zatrzymywać liczenie w razie potrzeby, jeśli jest to konieczne.

Należy pamiętać że aplikacja czeka dokładnie **2.5 sekundy** na kolejne odbicie piłki w sytuacji gdy piłka opuszcza obraz po lewej lub prawej stronie.

Przykład dobrej kalibracji:



Program zamykamy klikając klawisz **ESC** gdy jedno z dwóch okien programu jest zaznaczone jako aktywne. Pozwala to na prawidłowe opróżnienie buforów, oraz zapisanie ustawień rozmiarów okien do późniejszego użycia. Zamknięcie w inny sposób nie zapamięta rozmiaru okien.