

Projekt i realizacja wirtualnej perkusji

autor: Michał Wielgosz gr. 59,
rok akad. 2010/2011

Promotor: prof. nzw. dr hab. inż. Barbara Putz
Konsultant: mgr inż. Maciej Przybylski



Cel pracy:

Zaprojektowanie i stworzenie systemu, który przy użyciu komputera, kamer internetowych i pałeczek perkusyjnych, będzie w możliwie najlepszy sposób symulował perkusję.



Założenia:

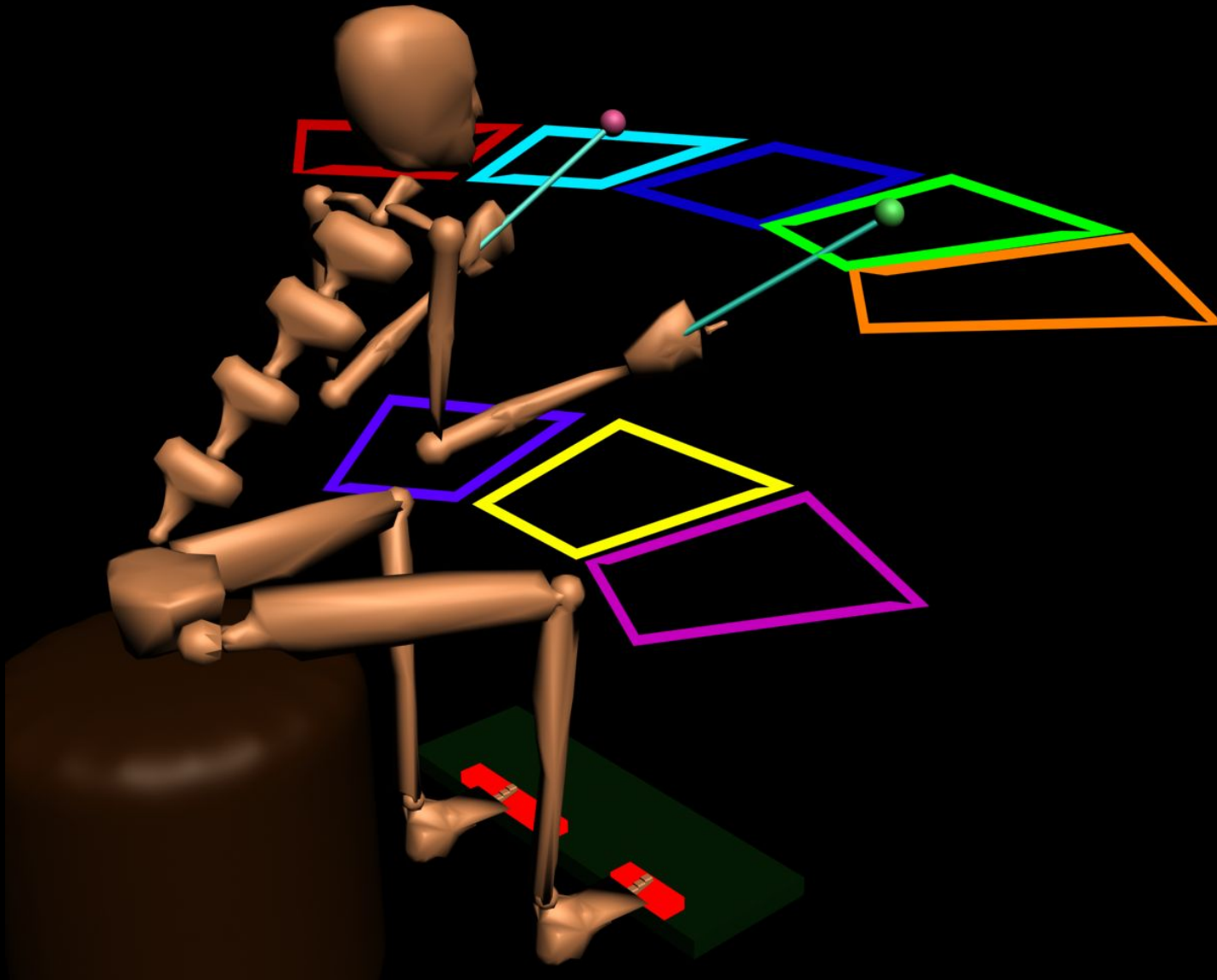
- korzystanie jedynie ze sprzętu nieprofesjonalnego, z rynku konsumenckiego
- do pobierania obrazu wykorzystanie tylko kamer internetowych
- parametry dynamiczne perkusji, określone parametrami dostępnych kamer internetowych
- możliwość modyfikacji pałeczek perkusyjnych do wspomagania przetwarzania obrazu



Temat:

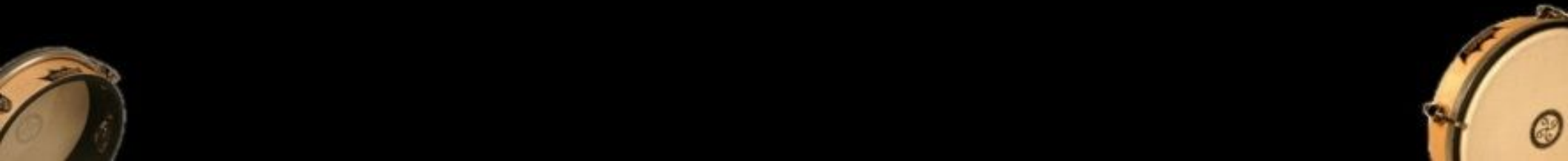


Temat:



Harmonogram pracy:

- zapoznanie się z systemami symulującymi grę na perkusji oraz z istniejącymi rozwiązaniami umożliwiającymi odczytanie w przestrzeni 3D położenia perkusyjnych pałeczek, w czasie rzeczywistym, za pomocą przetwarzania obrazu
- stworzenie założeń projektu (maks. tempo uderzeń, wymagane parametry kamer, właściwości sceny)
- wybór i zakup odpowiedniego sprzętu
- wykorzystanie istniejącego lub stworzenie oprogramowania do przechwytywania obrazów z kamer
- napisanie modułu aplikacji umożliwiającej znalezienie położenia końcówek pałeczek w przestrzeni 3D z danych otrzymanych za pomocą kamer
- utworzenie modułu aplikacji dającej użytkownikowi możliwość ustawiania poszczególnych elementów perkusji w odpowiednich miejscach w przestrzeni, edytowania ich właściwości
- ocena wyników pracy



Przegląd rozwiązań



Przegląd rozwiązań miJam Air Drummer Pro:

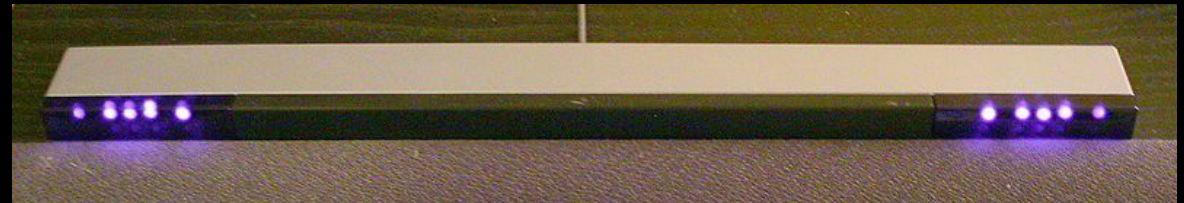
*miJam^{pro}
ready to rock
Air Drummer*



www.b2stuf.com

(źródło: http://www.b2stuf.com/main/image/gallery_29.jpg)

Przegląd rozwiązań konsola Wii:



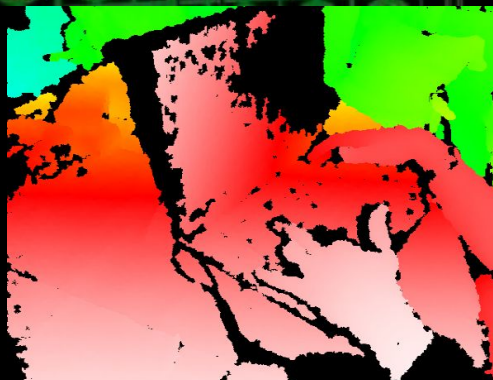
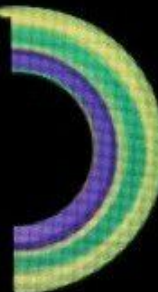
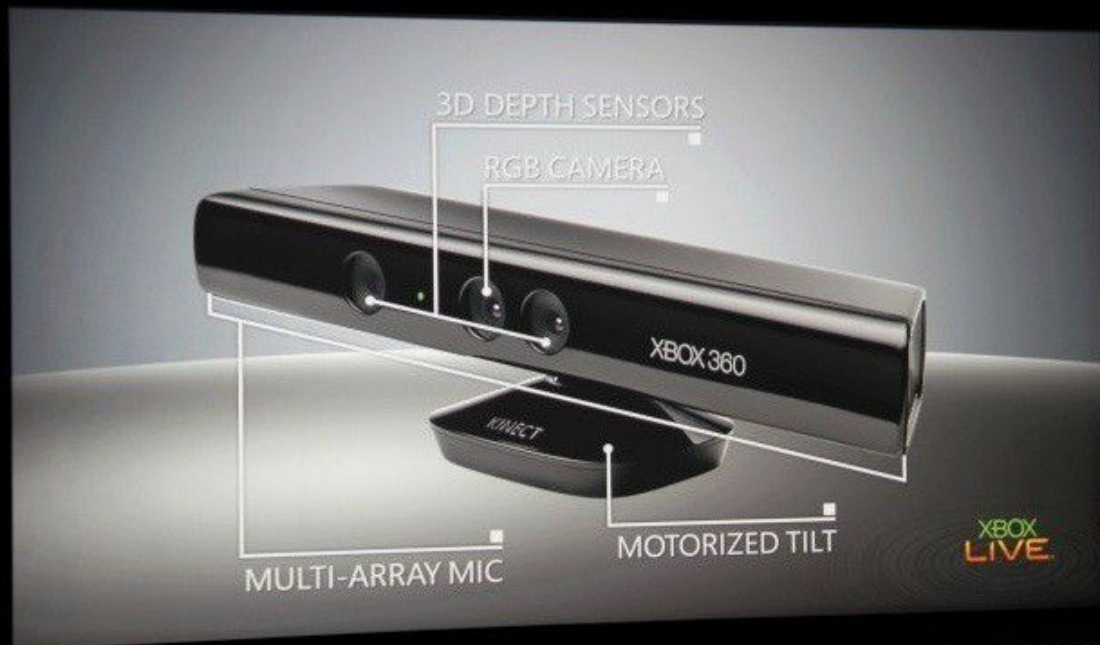
(źródło http://en.wikipedia.org/wiki/File:Wii_console.png
http://en.wikipedia.org/wiki/File:Nintendo_Wii_Sensor_Bar.jpg)

Przegląd rozwiązań Playstation Move:



(źródło: http://upload.wikimedia.org/wikipedia/commons/9/90/Move_motion_controller_over_the_floor.jpg
http://upload.wikimedia.org/wikipedia/en/5/52/PlayStation_Eye.png)

Przegląd rozwiązań Kinect:



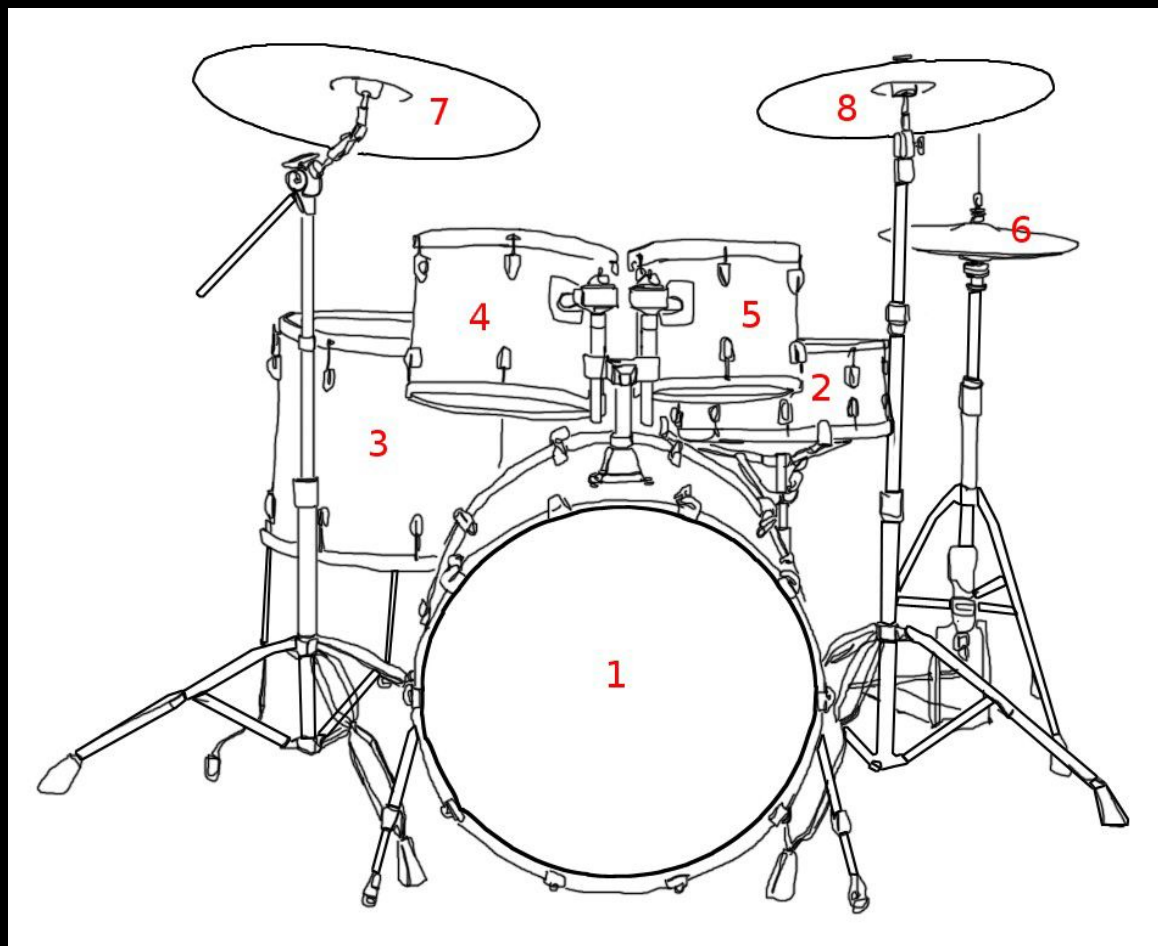
(źródło: <http://upload.wikimedia.org/wikipedia/en/a/a0/KinectTechnologiesE3.jpg>
<http://upload.wikimedia.org/wikipedia/commons/7/76/Kinect2-ir-image.png>
<http://en.wikipedia.org/wiki/File:Kinect2-deepmap.png>)

Przegląd rozwiązań - zestawienie:

	miJam Air Drummer	Nintendo Wii	PlayStation Move	Microsoft Kinect
Zalety	<ul style="list-style-type: none"> <u>- wyspecjalizowanie w perkusji</u> <u>- kontrolery są pałeczkami</u> - informacja o sile uderzenia - możliwość gry stopą 	<ul style="list-style-type: none"> - informacja o położeniu końcówek pałeczek (rozdzielność dźwięków względem przestrzeni) - informacja o sile uderzenia 	<ul style="list-style-type: none"> <u>- informacja o położeniu końcówek pałeczek (rozdzielność dźwięków względem przestrzeni)</u> <u>- informacja o sile uderzenia</u> <u>- precyzja i szybkość działania</u> 	<ul style="list-style-type: none"> - informacja o położeniu końcówek pałeczek (rozdzielność dźwięków względem przestrzeni) - brak potrzeby zakupu kontrolerów
Wady	<ul style="list-style-type: none"> - wymóg zakupu specjalnego osprzętu - kontroler wymaga zasilania bateryjnego <u>- brak informacji o położeniu końcówek pałeczek</u> 	<ul style="list-style-type: none"> - wymóg zakupu specjalnego osprzętu - kontroler wymaga zasilania bateryjnego - brak wyspecjalizowania na symulację perkusji - kontrolery nie są pałeczkami - brak możliwości gry stopą - wrażliwość na źródła światła fal podczerwonych (np. płomień świecy) 	<ul style="list-style-type: none"> <u>- wymóg zakupu specjalnego osprzętu</u> <u>- kontroler wymaga zasilania bateryjnego</u> <u>- brak wyspecjalizowania na symulację perkusji</u> <u>- kontrolery nie są pałeczkami</u> <u>- brak możliwości gry stopą</u> <u>- wrażliwość na kolorowe źródła światła</u> 	<ul style="list-style-type: none"> - wymóg zakupu specjalnego osprzętu - brak wyspecjalizowania na symulację perkusji - brak dobrej informacji o sile uderzenia - kontrolery nie są pałeczkami - brak możliwości gry stopą ? - możliwa wrażliwość na źródła światła fal podczerwonych (np. płomień świecy) <u>- odczuwalne opóźnienia w reagowaniu ruchu ciała</u> - mała precyzja znajdowania położenia - niska częstotliwość odświeżania kamery

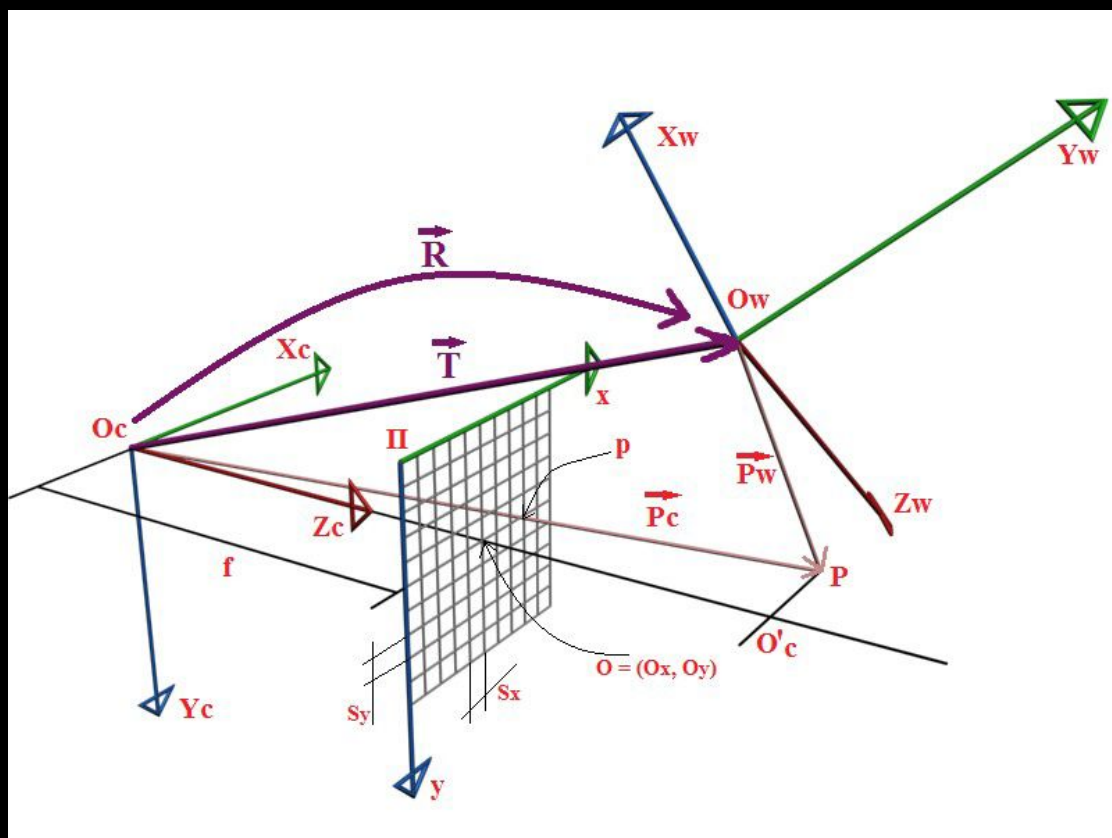


Podstawowe elementy perkusji:



- bęben wielki (1)
- werbel (2)
- duży tom-tom (3)
- średni tom-tom (4)
- mały tom-tom (5)
- hi-hat (6)
- ride (7)
- crash (8)

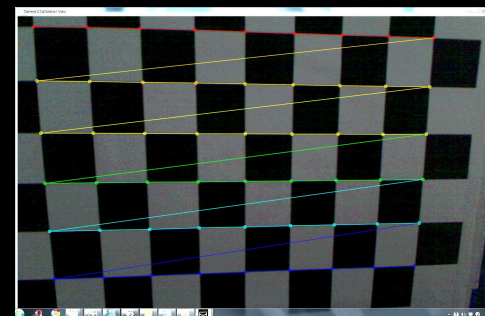
Model perpektywiczny kamery:



$$x_{corrected} = x + [2 p_1 y + p_2 (r^2 + 2 x^2)]$$

$$y_{corrected} = y + [p_1 (r^2 + 2 y^2) + 2 p_2 x]$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$



- parametry wewnętrzne (zniekształcenia radialne, przesunięcie punktu głównego o)
- parametry zewnętrzne (wektor rotacji i translacji)

Rodzaje układów stereoskopowych:

Ze względu na sposób ustawienia urządzeń optycznych:

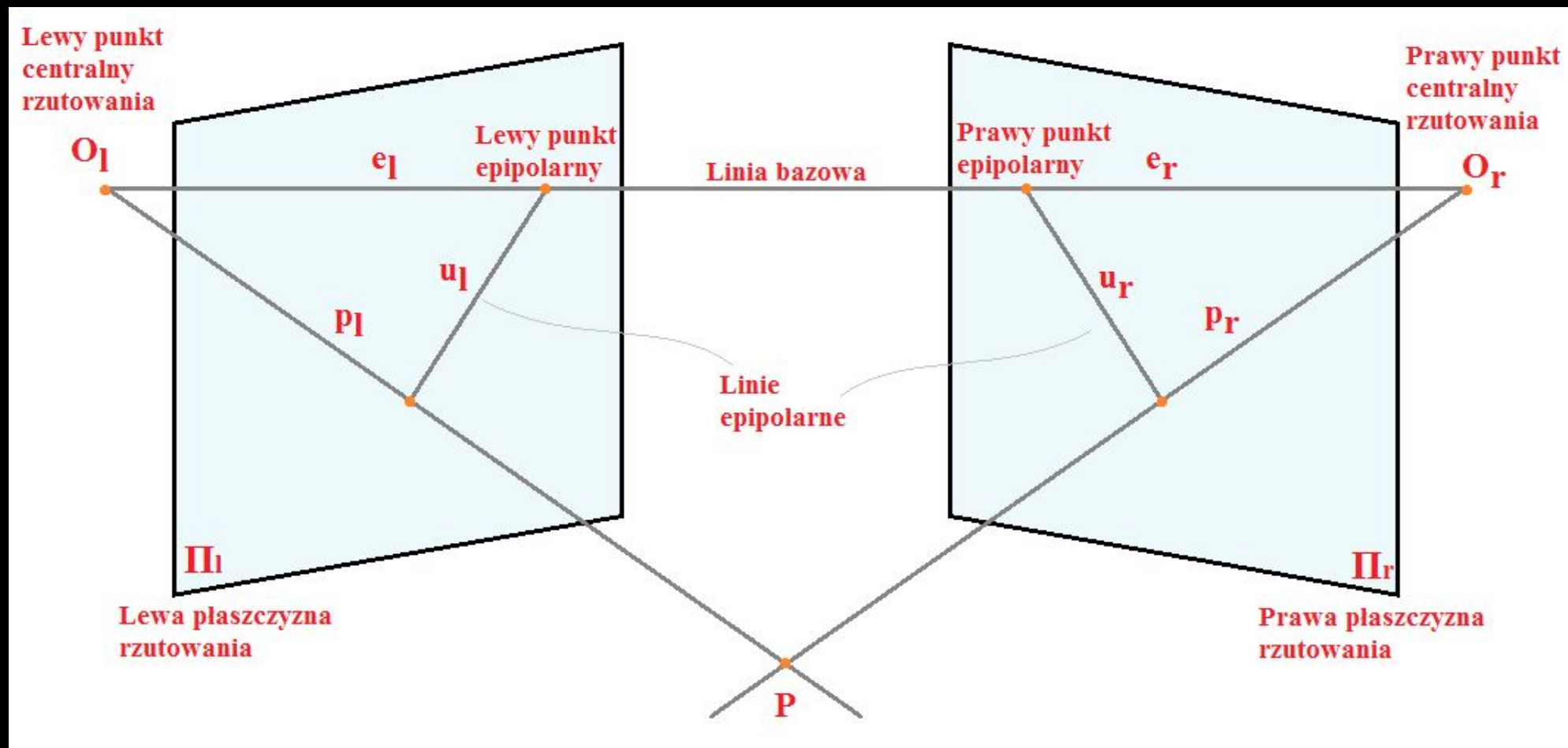
- układ dowolny kamer
- układ kanoniczny kamer
- układ katadioptryczny

Ze względu na sposób rekonstrukcji głębi sceny:

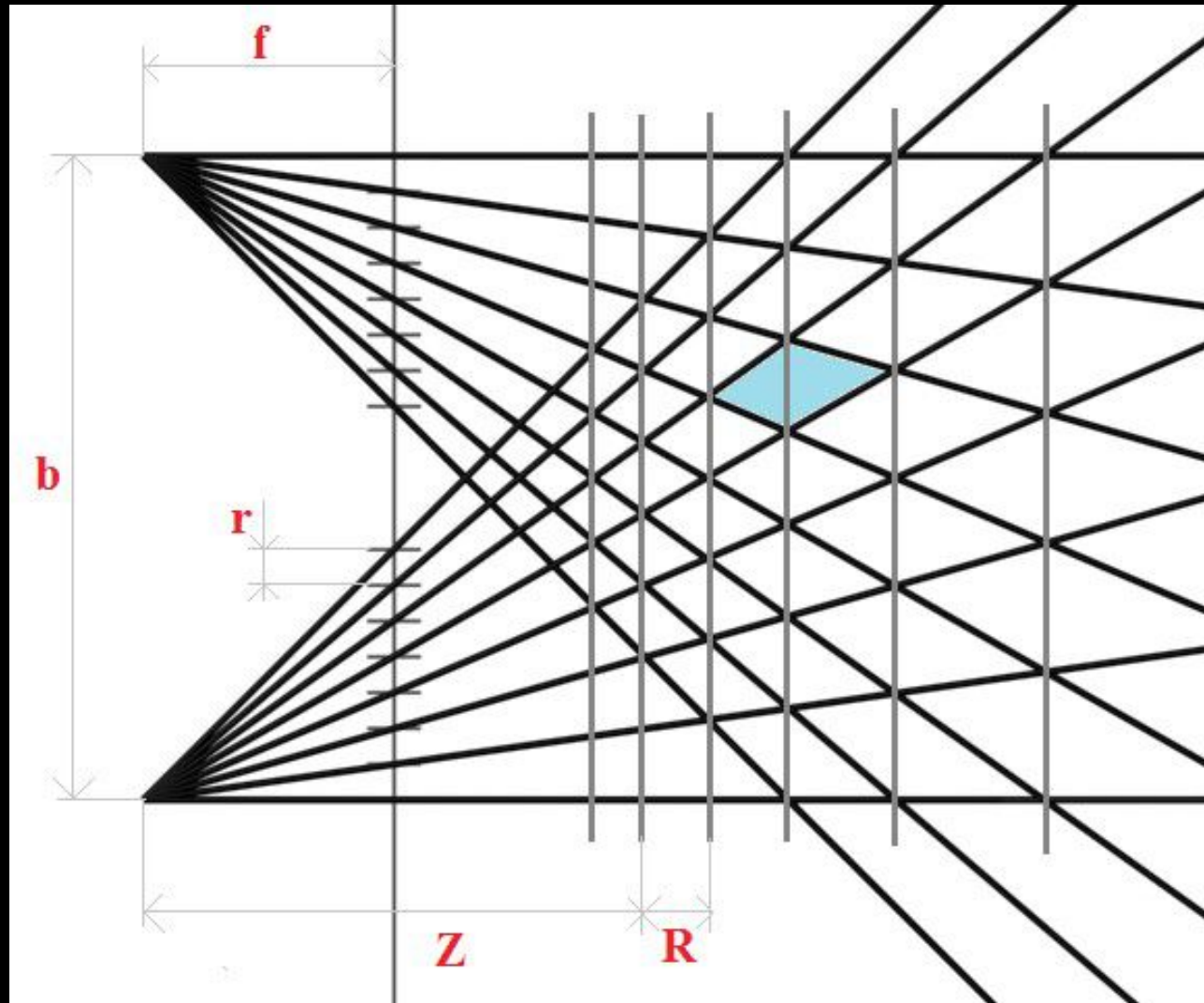
- kamery ustawione w układzie dowolnym z możliwością użycia geometrii epipolarnej
- kamery ustawione w układzie kanonicznym z możliwością użycia geometrii epipolarnej
- triangulacja przy pomocy przecięcia się promieni z obu kamer ustawionych w układzie dowolnym



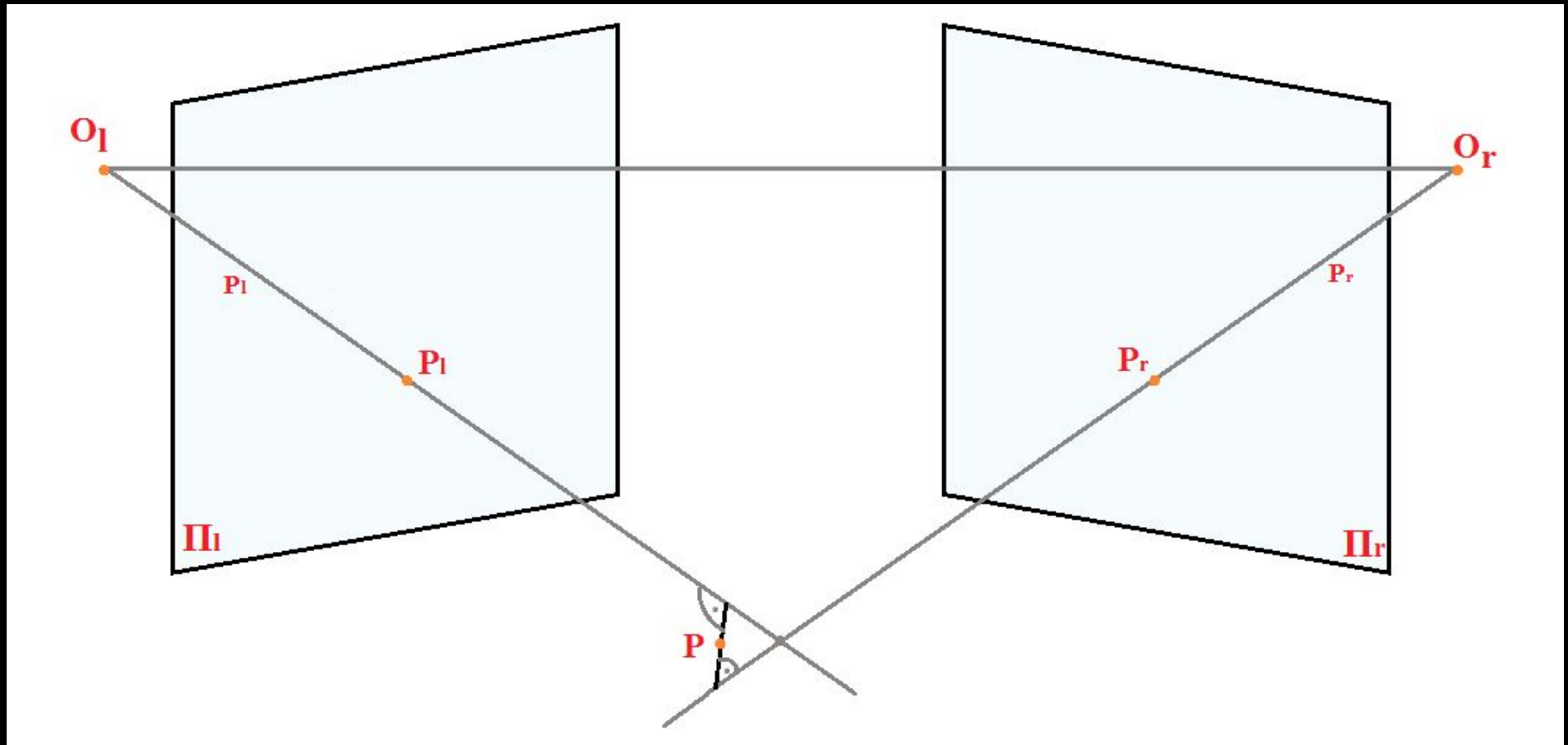
Geometria epipolarna:



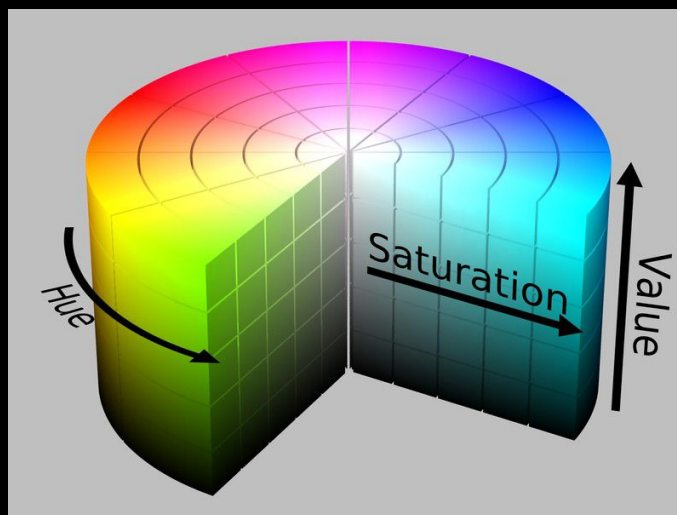
Układ kanoniczny:



Układ dowolny + triangulacja przecięcia się promieni:



Użyta przestrzeń barwna - HSV:



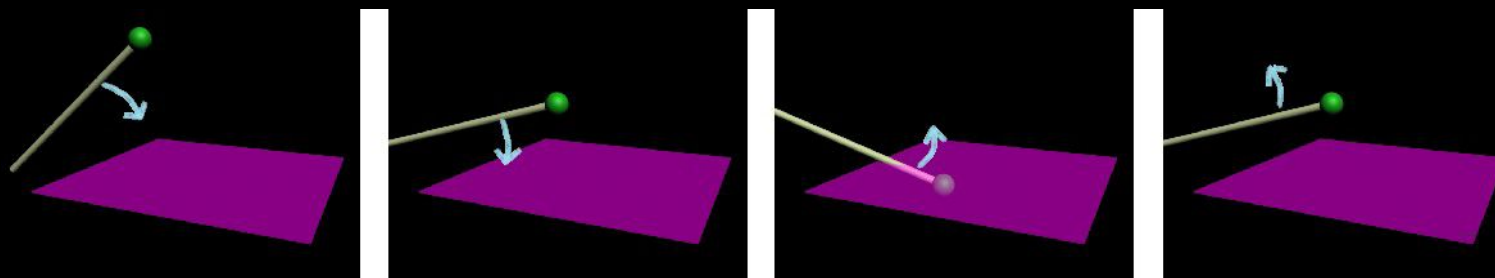
Charakter wirtualnej perkusji:

- system dostępny dla zwykłego użytkownika komputera:
możliwie tani i łatwy w konfiguracji
- konsola, a nie symulator
- pokazanie idei użytkowania zwykłej perkusji:
rozdzielność ruchów dłoni, stóp, przyzwyczajenia
trzymania pałeczek oraz zobrazowanie zalet i
możliwych trudności w grze

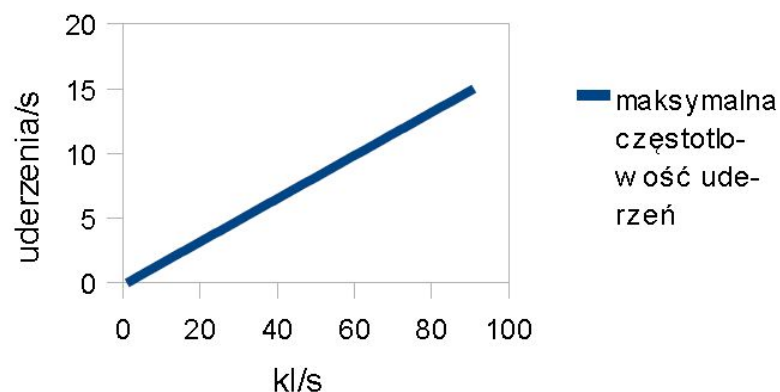


Możliwość realizacji wymagań:

- maksymalna możliwa do uzyskania prędkość uderzeń perkusisty (500ud/min \rightarrow 8,3ud/s jedną dłonią)
- dynamika uderzeń i ich głośność



Maksymalna liczba uderzeń/s
od kl/s używanej kamery

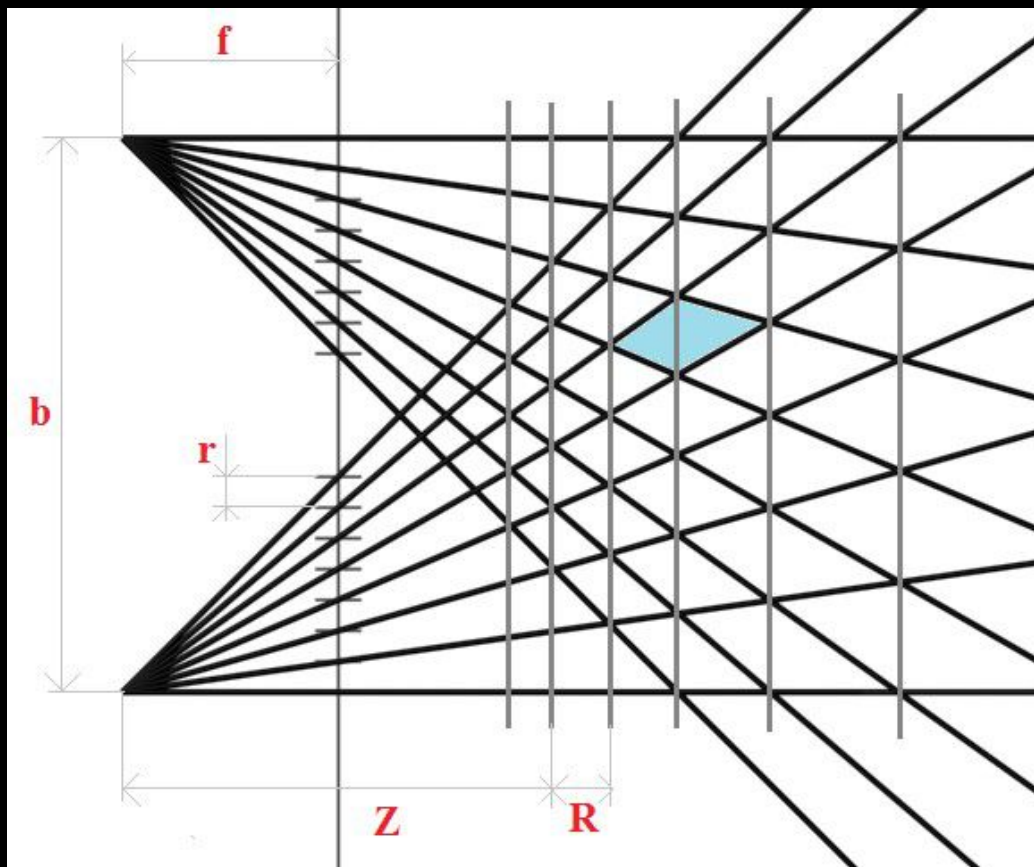


Możliwość realizacji wymagań:

- fizyczne wrażenia zwrotne z gry na elementach perkusyjnych: używanie zwykłych pałeczek perkusyjnych, odbijanie pałeczki po uderzeniu w bęben, pochłanianie uderzenia przy zderzeniu pałeczki z talerzem



Możliwość realizacji wymagań:



$$R = \frac{Z^2}{\frac{R_h b}{2 \operatorname{tg} \frac{\alpha}{2}} - Z}$$

- dokładność położenia pałeczek i rozmiary elementów perkusyjnych

Dokładność wirtualnej perkusji:

- liczby użytych kamer i ich jakości (rozdzielczość, poziom szumów, liczby kl/s),
- położenia i orientacji kamer względem siebie – parametry zewnętrzne (wektor translacji i rotacji kamer względem siebie)
- znajomości parametrów wewnętrznych tych kamer – parametry wewnętrzne (aberracje układu optycznego, kąt widzenia, położenie punktu głównego)
- jakości oświetlenia sceny i zawartości tej sceny (różnorodność obiektów, ilość ruchu w kadrach kamer)
- jakości, rodzaju i wielkości użytych markerów



Środowisko programistyczne:

- **Visual C++ 2008 Express Edition**



- **OpenCV 2.1** (Open Computer Vision) – przetwarzanie obrazu z kamer



- **OpenGL** (Open Graphics Library), GLUT (OpenGL Utility Toolkit) – generowanie grafiki w interfejsie 3D – wizualizacja rozmieszczenia perkusji oraz jej działania



- **irrKlang** – operacje audio – odtwarzanie dźwięków w programie i funkcje z tym związane



- **OpenMP** (Open Multi-Processing) – tworzenie aplikacji wieloprocessorowych – rozłożenie przetwarzania obrazu każdej z kamer na jeden rdzeń procesora



Skrót algorytmu programu:

- I – deklaracja i inicjalizacja zmiennych globalnych i pomocniczych
- II – proces kalibracji każdej kamery z osobna – parametry wewnętrzne i dystorsji (XML)
- III – proces kalibracji obu kamer – parametry zewnętrzne (wzajemne położenie kamer)
- IV – nauka tła widzianego przez kamery
- V – oznaczenie kolorów markerów przez użytkownika
- VI – pętla główna programu, wyliczanie pozycji markerów w kolejnych pobieranych ramkach – w pierwszej fazie, za pomocą markerów, pozwala na rozstawienie elementów perkusyjnych w przestrzeni wokół użytkownika



Skrót algorytmu programu:

Równoległe działające wątki:

- VII – wyliczanie pozycji markerów w kolejnych pobieranych ramkach – sprawdzanie czy jakiś marker uderzył, w któryś z ustawionych elementów perkusyjnych
- VIII – dwie pętle odtwarzania dźwięków, po jednej pętli dla każdego markera – sprawdzanie jaki element perkusyjny został trafiony i odtwarzają odpowiedni dźwięk
- IX – pętla okna interfejsu 3D wirtualnej perkusji, wizualizująca względne położenia: kamer, markerów i elementów perkusyjnych + daje dodatkowe informacje
- X – pętla głównego interfejsu umożliwia modyfikację parametrów jej działania, rodzajów wyświetlanych okien, podglądu 3D sceny z wirtualną perkusją.



Nauka tła – metoda codebook:

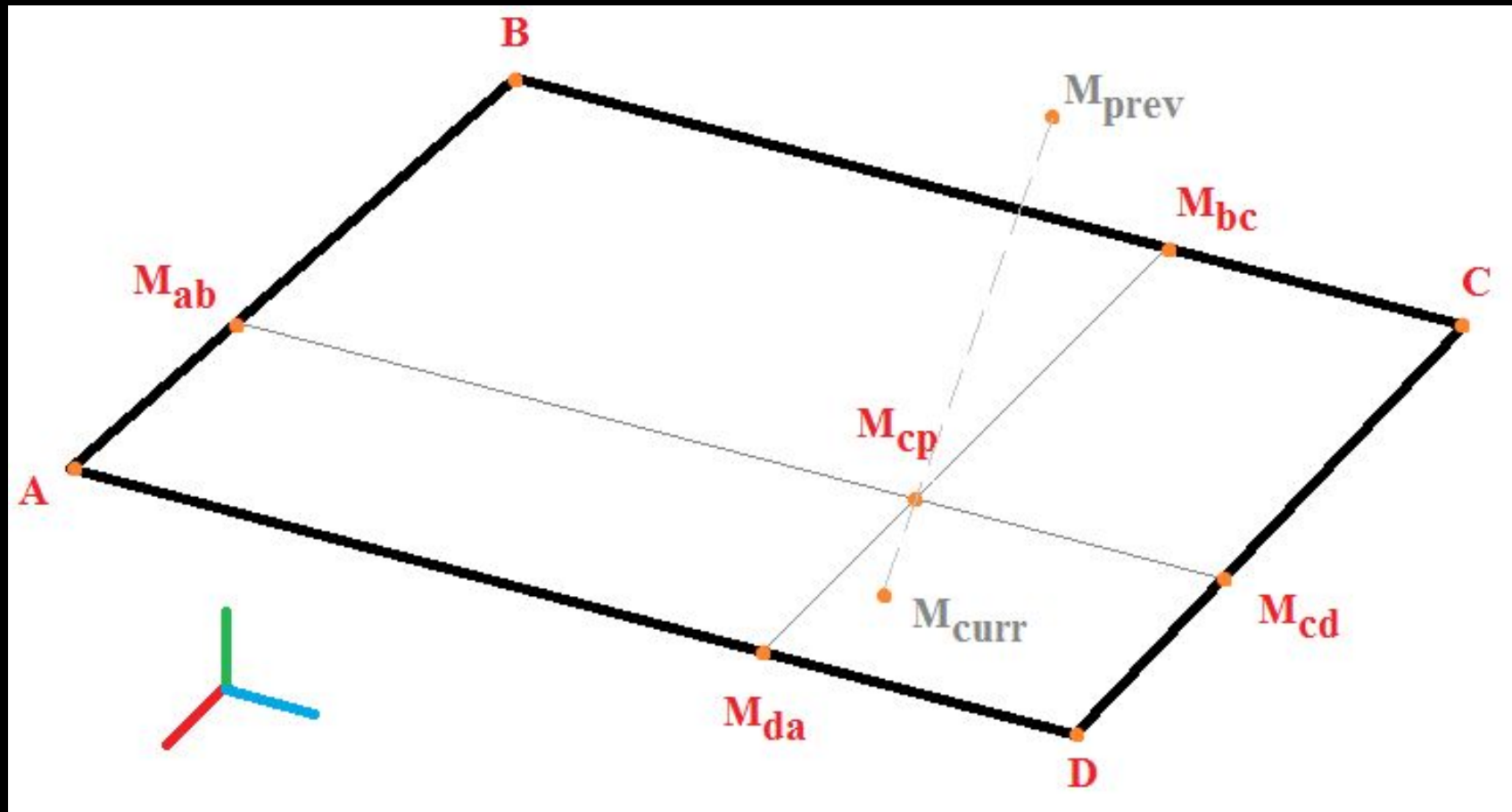
- RGB do YCrCb – uzależnienie od jasności
- rozszerzanie zakresu nauki jeżeli zmiana jasności nie jest duża
- dodawanie nowego wpisu po przekroczeniu ustalonego progu
- usuwanie starych wpisów



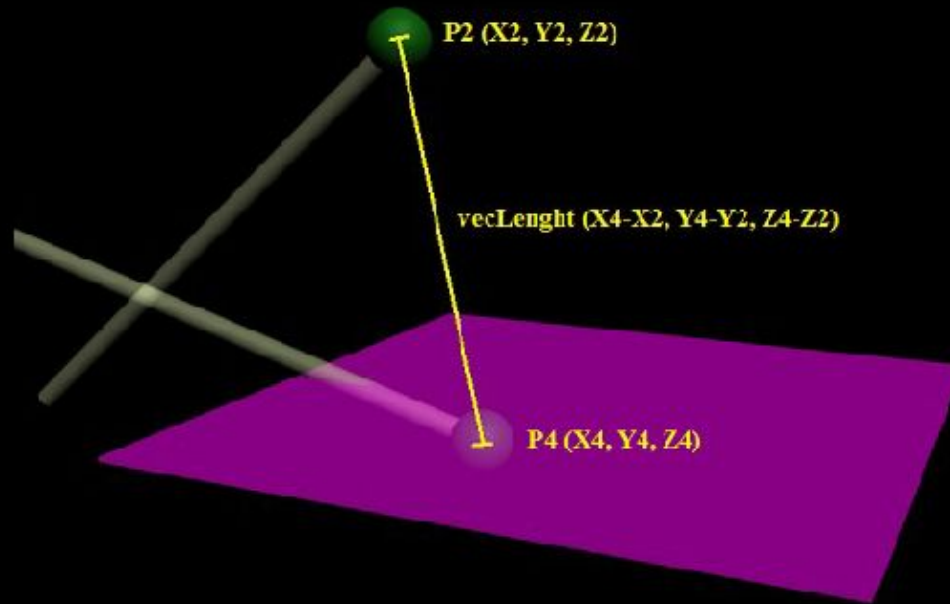
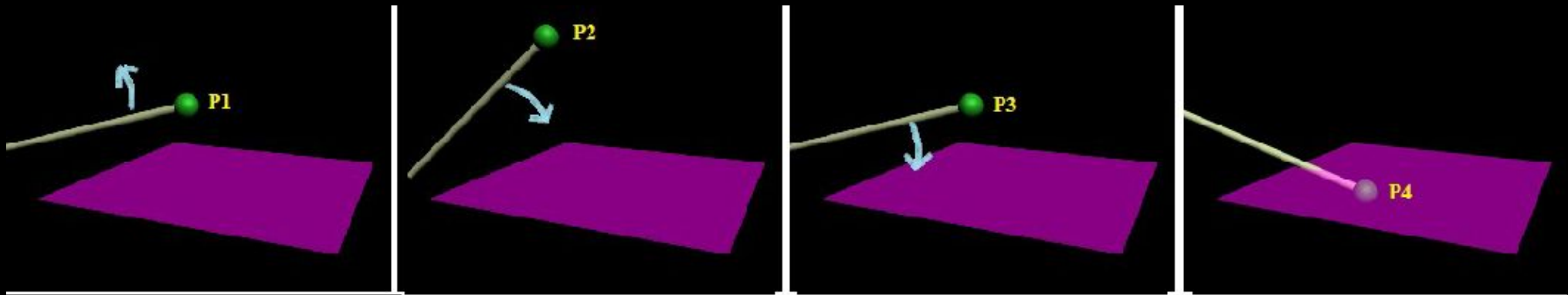
Główne etapy przetwarzania obrazu:



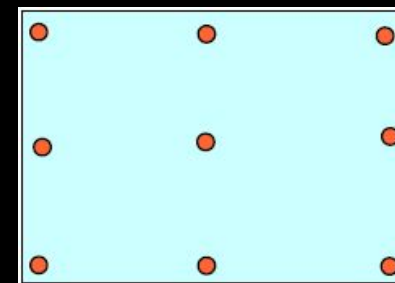
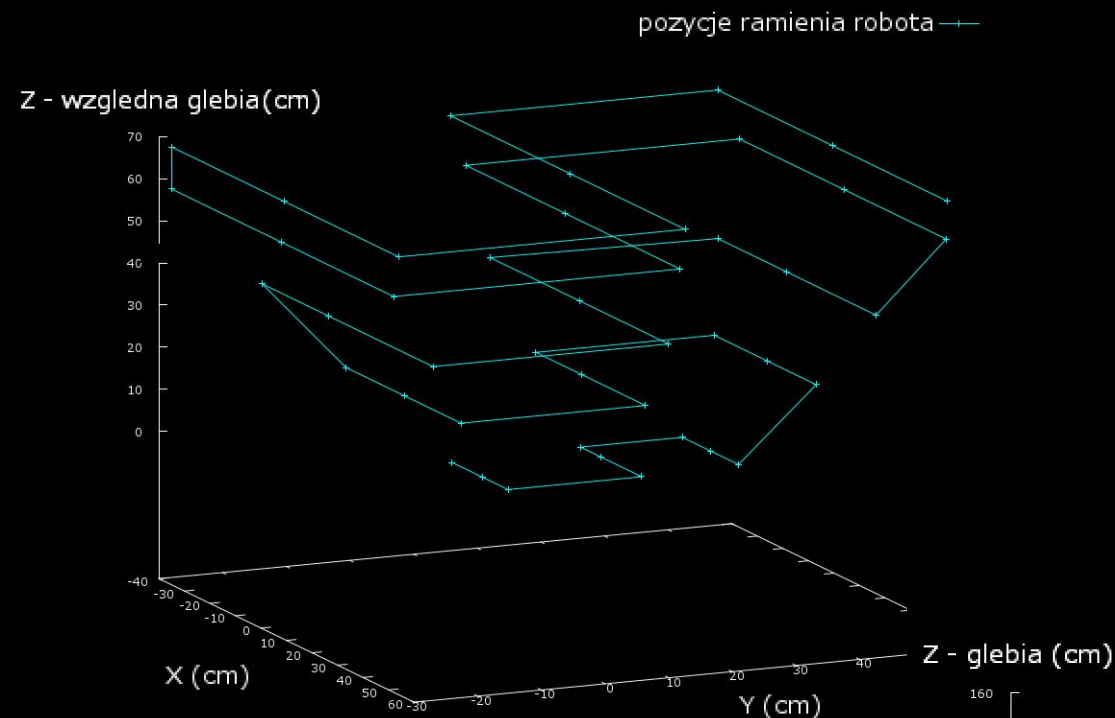
Wykrywanie uderzenia:



Dynamika uderzenia:



Dokładność odczytu pozycji:

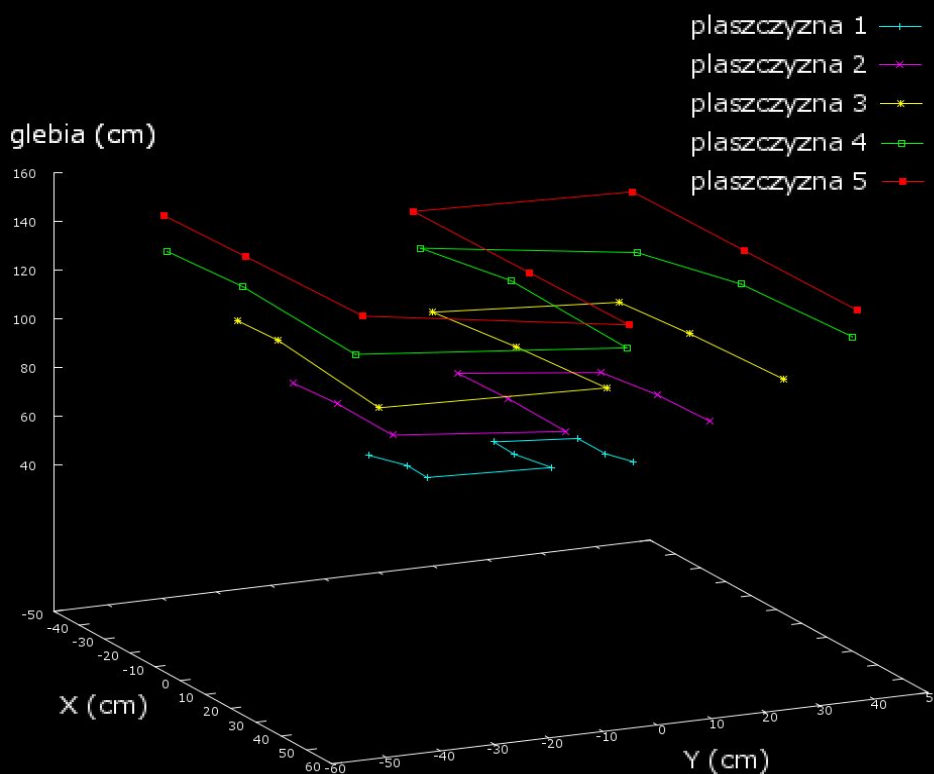


Rozrzut wartości (dokładność):

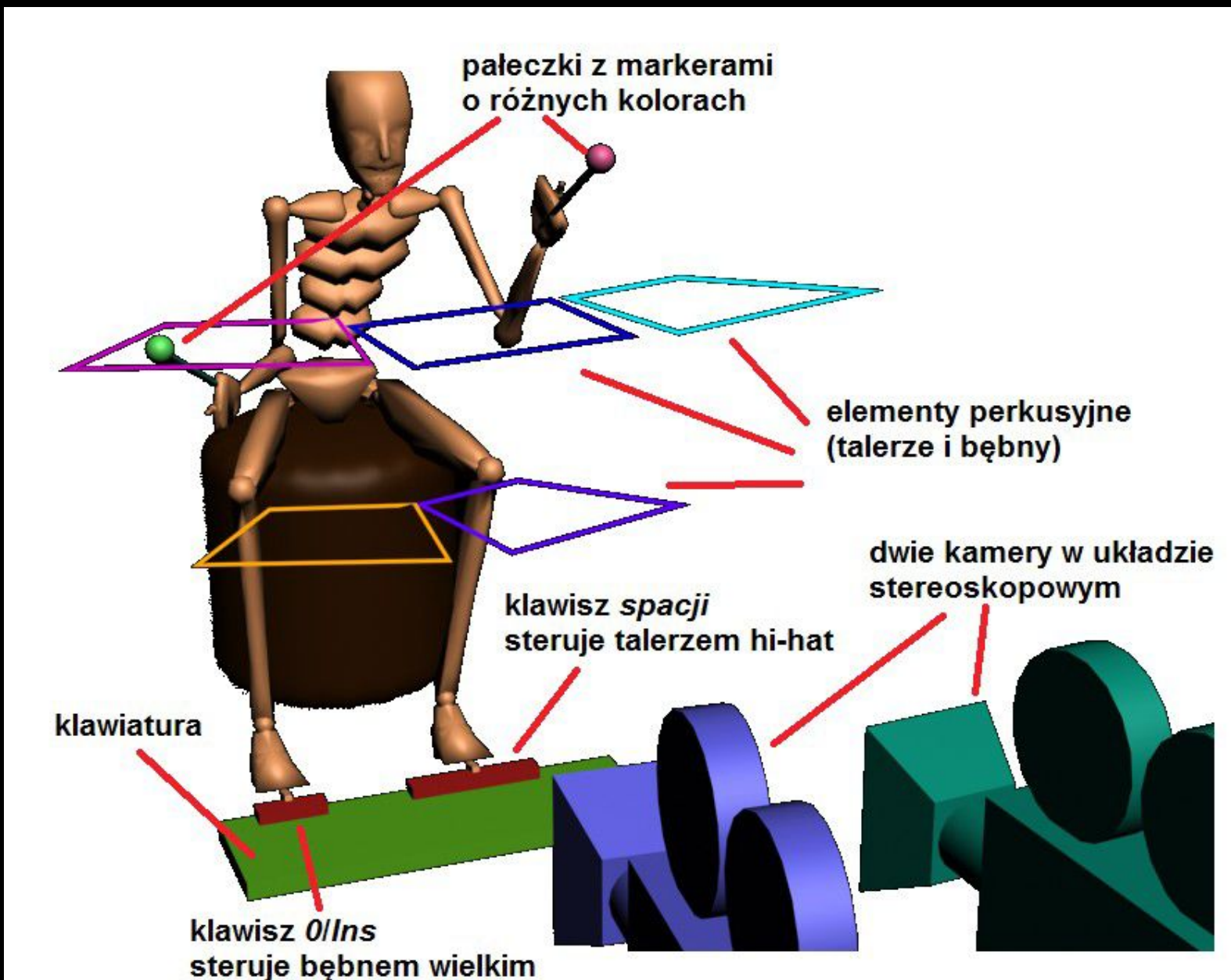
$\Delta z_{\max} = 30\text{cm}$ - przy największej odległości od kamer równej 130cm

Odchylenie standardowe (powtarzalność):

$\sigma_{z \max} = 8\text{cm}$ - dla poziomu niepewności 0.95, przy największej odległości od kamer równej 130cm



Wirtualna perkusja



Wirtualna perkusja – porównanie **zalet:**

Zalety systemów:

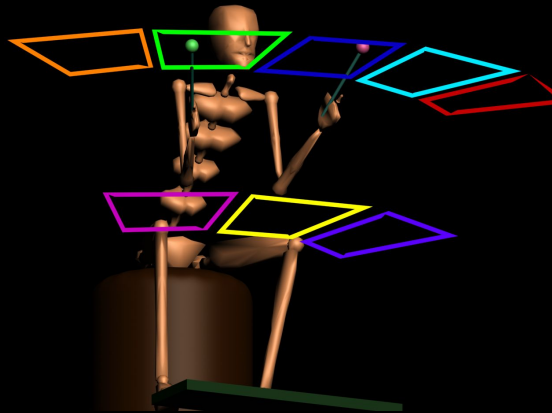
- wyspecjalizowanie pod kątem perkusji – **TAK**
- kontrolery są pałeczkami perkusyjnymi – **TAK**
- informacja o sile uderzania – **POŚREDNIO**
- możliwość gry stopą – **POŚREDNIO** – za pomocą klawiatury
- informacja o położeniu końcówek pałeczek (rozdzielność dźwięków względem miejsc w przestrzeni) – **TAK**
- precyzja i szybkość działania – **TAK**
- brak potrzeby zakupienia kontrolerów – **TAK** – wystarczy użycie zwykłych pałeczek perkusyjnych z zamontowanymi na ich końcach kolorowymi markerami (np. piłeczki do ping ponga)



Wirtualna perkusja – porównanie **wad**:

Wady systemów:

- wymóg zakupienia specjalnego osprzętu – **POŚREDNIO** – potrzeba posiadania dwóch kamer internetowych umożliwiających ręczne sterowanie ich ekspozycją, dodatkowo wymagany jest przeciętny komputer z procesorem dwurdzeniowym i 1GB pamięci RAM
- kontroler wymaga zasilania bateryjnego – **NIE**
- wrażliwość na źródła światła fal podczerwonych (np. płomień wiecey) – **NIE**
- wrażliwość na kolorowe źródła światła – **TAK**
- odczuwalne opóźnienia w reagowaniu na ruchy – **NIE**



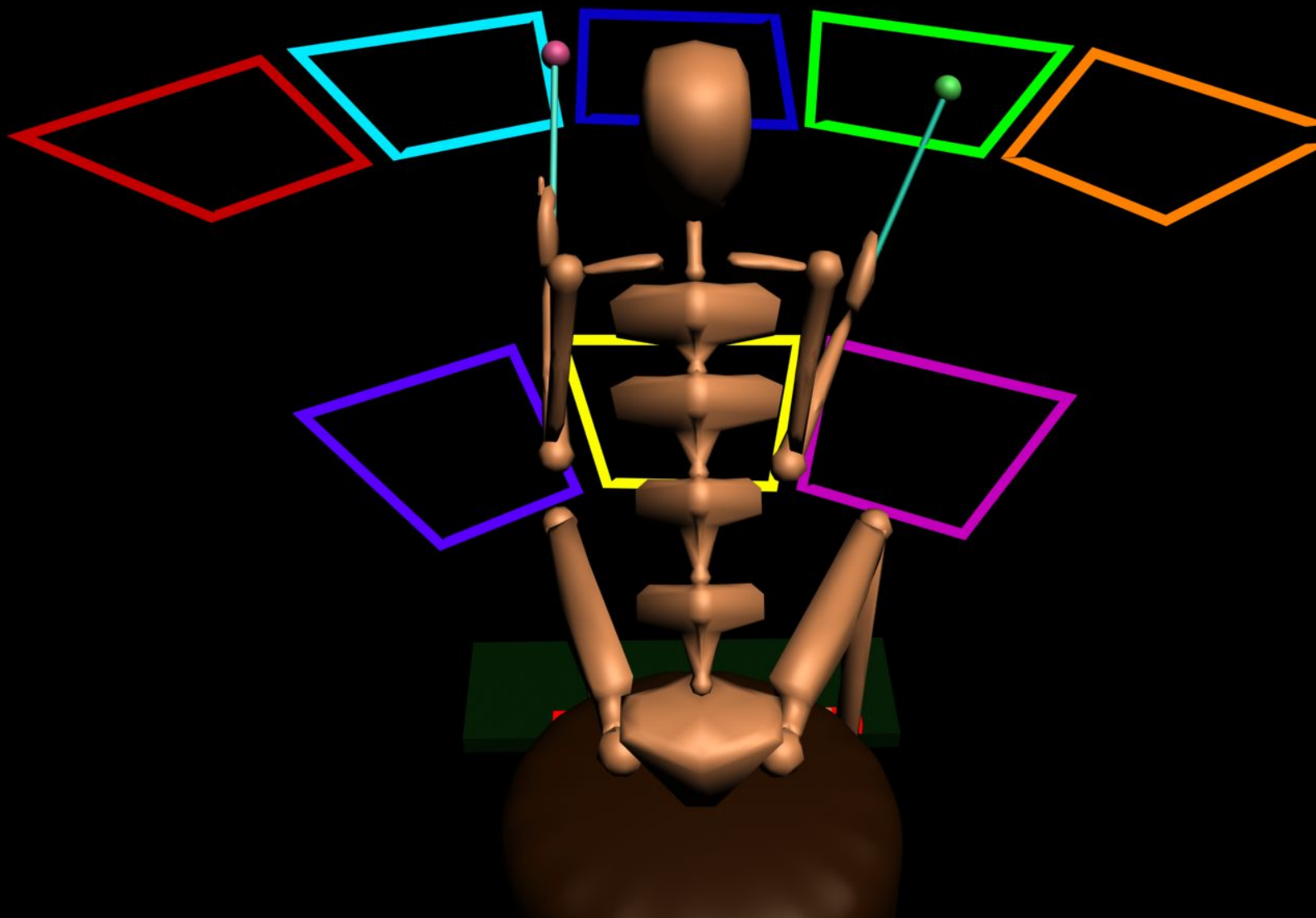
Wirtualna perkusja – kierunki rozwoju:

- pałeczki z świecącymi markerami lub specjalnym kształcie
- dodawanie kolejnych kamer (możliwość załonięcia, wzrost dokładności, potrzebna większa moc obliczeniowa)
- specjalne kontrolery do grania stopami (bęben basowy, talerz hi-hat)

Ulepszenia → wzrost skomplikowania urządzeń i wzrost kosztów



Podsumowując....



Literatura:

- O. Reilly – Learning OpenCV
- Bogusław Cyganek – Komputerowe przetwarzanie obrazów Trójwymiarowych



Dziękuję za uwagę

