

Ćwiczenie AR

Rozszerzona_rzeczywistość

Cele laboratorium:

- Zapoznanie się z zagadnieniem kalibracji kamery.
- Zapoznanie się z zagadnieniami rozszerzonej rzeczywistości AR

Przygotowanie

W przypadku realizacji ćwiczenia w laboratorium: w katalogu wskazanym przez prowadzącego utwórz podkatalog, w którym będą zapisywane wszystkie dane podczas ćwiczenia. Nazwa katalogu powinna zawierać: **<nr ćwiczenia>_<data>_<inicjały osoby realizującej ćwiczenie>**; zapisz i rozpakuj pliki ćwiczenia (pobrane z UPEL) do utworzonego katalogu; po skończonych zajęciach pamiętaj o usunięciu danych z dysku.

W przypadku realizacji ćwiczenia zdalnie:

- wymagania: MATLAB R2023a z modułem: Image Acquisition Toolbox, Image Processing Toolbox, Audio Toolbox, DSP System Toolbox, kamera internetowa, mikrofon, następujące AddOns: MATLAB Support Package for USB Webcams, Image Acquisition Toolbox Support Package for OS Generic Video Interface
- **przed ćwiczeniem należy wydrukować planszę kalibracyjną (w MATLABie: open checkerboardPattern.pdf)**

Informacje na temat sprawozdania z ćwiczenia zostały zamieszczone na końcu instrukcji

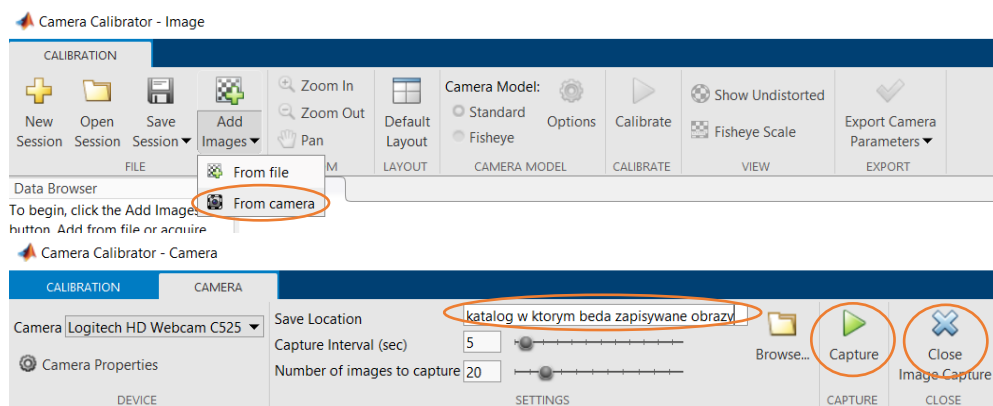
Proszę o zgłaszanie nieścisłości w instrukcji drogą emailową do prowadzącego zajęcia.

Część I – Kalibracja kamery

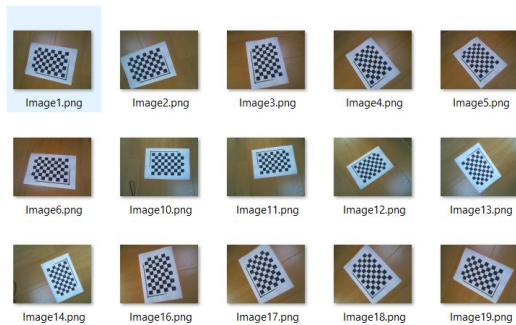
Gdzie szukać dodatkowych informacji:

- Internet – słowa kluczowe: camera calibration, augmented reality
- Dokumentacja MATLABa
- Multiple View Geometry in Computer Vision : <https://www.cambridge.org/core/books/multiple-view-geometry-in-computer-vision/0B6F289C78B2B23F596CAA76D3D43F7A>

- 1) Przed ćwiczeniem należy wydrukować planszę kalibracyjną (w MATLABie: `open checkerboardPattern.pdf`)
Najlepiej nakleić ją na twarde karton aby kartka się nie wyginała.
- 2) Zapoznaj się z procesem kalibracji opisanym w dokumentacji MATLABa:
What Is Camera Calibration?
<https://www.mathworks.com/help/vision/ug/camera-calibration.html>
Zwróć uwagę na:
 - model kamery „pinhole”
 - parametry zewnętrzne kamery (extrinsic parameters)
 - parametry wewnętrzne kamery (intrinsic parameters)
- 3) Upewnij się, że masz podłączoną i widoczną z poziomu MATLABa kamerę, a następnie otwórz aplikację Camera Calibrator
`>> cameraCalibrator`
- 4) Pobierz przynajmniej 20 obrazów, w taki sposób, aby plansza kalibracyjna była widziana w całości z różnych punktów widzenia.
 - Utwórz katalog na obrazy i ustaw go odpowiednio w narzędziu.
 - Ustaw i zanotuj rozdzielczość pobierania obrazów (camera properties) – w dalszej części ćwiczenia obrazy powinny mieć taką samą rozdzielczość jak podczas kalibracji.
 - W trakcie pobierania obrazów przy pomocy tego narzędzia mogą się zdarzyć źle pobrane obrazy. Narzędzie automatycznie potrafi odrzucić te które są nieodpowiednie, ale możesz ręcznie usunąć obrazy z utworzonego katalogu. Należy wtedy utworzyć nową sesję w narzędziu (*New Session*).
 - Nie zapomnij zakończyć akwizycji (*Close Image Capture*). W przeciwnym przypadku przycisk *Calibrate* będzie nieaktywny.
 - Obrazy możesz również pobrać bez użycia tego narzędzia (np. przy pomocy *imaqtool/webcam*) – istotne jest, aby umieścić je w jednym katalogu – można je wskazać w narzędziu (*From File*).

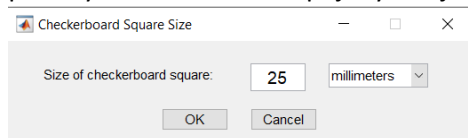


Przykładowe obrazy:

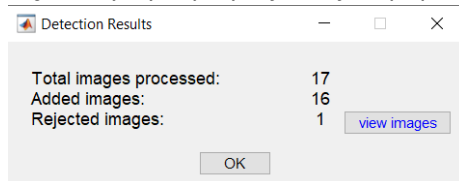


5) Kalibracja:

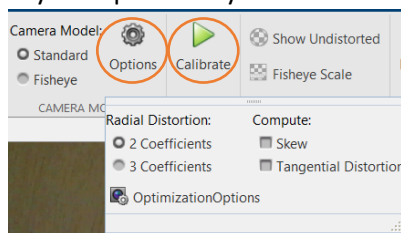
- Po wskazaniu obrazów pojawi się pytanie o rzeczywiste rozmiary wzorca na wydrukowanej planszy. Zmierz wielkość pojedynczej kratki liniijką i podaj w narzędziu (w mm).



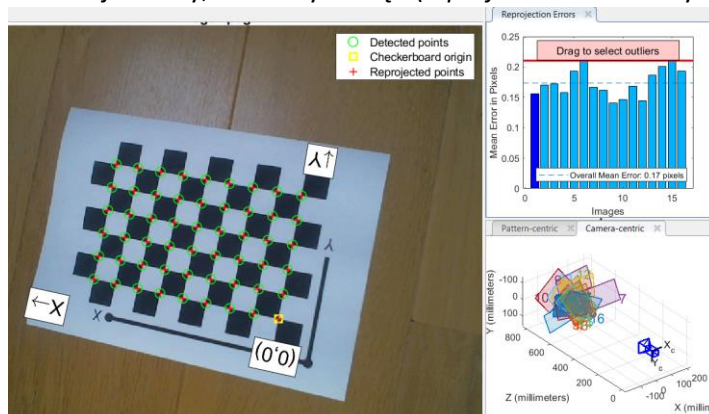
- Następnie aplikacja spróbuje automatycznie wykryć wzorec – w przypadku, gdy na zdjęciach nie wykryto zostanie wzorec, te obrazy automatycznie zostaną odrzucone z dalszej analizy. Upewnij się, że wykryto przynajmniej 15 poprawnych obrazów.



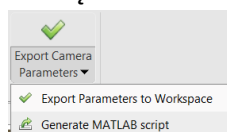
- Wybierz parametry oraz uruchom kalibrację



- Sprawdź rezultaty kalibracji – czerwone punkty *reprojected points* powinny pokrywać się z zielonymi punktami *detected points*. Jeśli zauważysz zbyt duże różnice, spróbuj usunąć z kalibracji obrazy, dla których błąd (*reprojection error* – wykres w prawym górnym rogu) jest duży.



- Zapisz sesję kalibracji (*Save session*) pod dowolną nazwą, a następnie wygeneruj kod MATLABa (Export Camera Parameters-> Generate MATLAB Script). Zapisz wygenerowany m-plik pod nazwą: `calibrationParams.m`



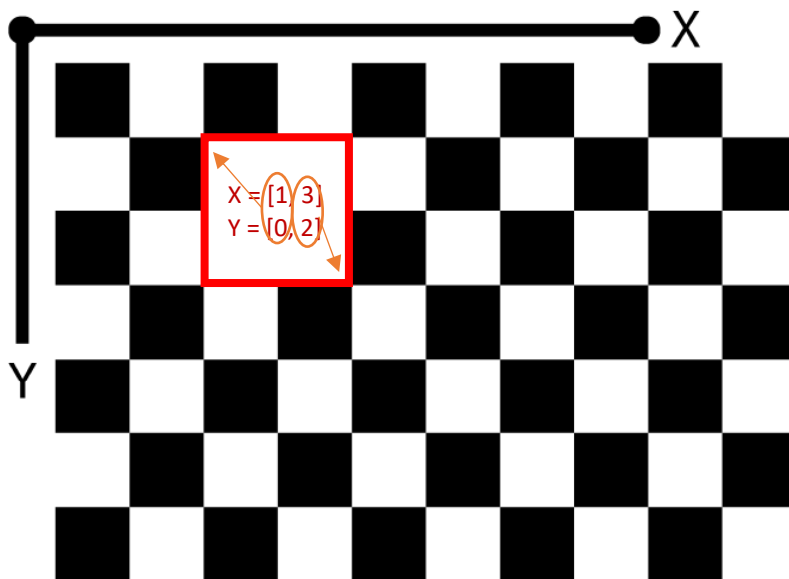
Część II – rzutowanie obiektów 3D na obraz

- 1) Otwórz skrypt `AR_przyklad_1.m` i wykonuj kolejne sekcje kodu, notując w sprawozdaniu rezultaty.
- 2) W sekcji 1 wywoływany jest wygenerowany automatycznie skrypt kalibracji oraz zapisywane parametry kalibracji do pliku MAT, który będzie później wykorzystywany w ćwiczeniu.
Zanotuj w sprawozdaniu parametry kamery: *focal length*, *principal point*, *radial distortion*.
Umieść w sprawozdaniu wykres „*extrinsic parameters visualization*” oraz „*Mean reprojection error per image*”
- 3) W sekcji 2 realizowane jest wyznaczanie parametrów rotacji i translacji kamery na podstawie wykrytych punktów planszy kalibracyjnej.
W tym celu najpierw zrób nowe zdjęcie planszy kalibracyjnej przy pomocy narzędzia. Zdjęcie powinno być zrobione w takim ujęciu, aby później łatwiej było zaobserwować w perspektywie nakładany obiekt AR. Rozdzielczość obrazu powinna być taka sama jak przy kalibracji.



- Kolejność operacji jest następująca:
 - opcjonalna korekta zniekształceń obiektywu kamery `undistortImage` (w przypadku większości kamer internetowych zniekształcenia są niewielkie – można to sprawdzić wyświetlając obydwa obrazy przy pomocy funkcji `imshowpair` lub sprawdzając czy zmienna `newOrigin` zwracana przez funkcję jest różna od [0 0])
 - wykrycie punktów wzorca na obrazie `imagePoints` – podobnie jak przy kalibracji. Jeśli nie zostaną wykryte wszystkie punkty wyświetlony będzie błąd – konieczne jest zrobienie innego zdjęcia planszy.
 - Na podstawie parametrów kamery `cameraParams`, wykrytych punktów na obrazie `imagePoints` oraz znajomości położenia punktów w przestrzeni 3D `worldPoints` można wyznaczyć translację i rotację układu współrzędnych kamery względem planszy kalibracyjnej. Zwróć uwagę, że punkty `worldPoints` zawierają dwie współrzędne (X, Y) – zakłada się, że cała płaszczyzna planszy jest położona w Z=0.

- 4) W sekcji 3 przedstawiony jest sposób zamiany współrzędnych obrazu na współrzędne świata, przy użyciu wcześniej wyznaczonych parametrów wewnętrznych kamery oraz translacji i rotacji.
 - Zapoznaj się z dokumentacją funkcji `pointsToWorld`
- 5) W sekcji 4 przedstawiony jest sposób rzutowania punktów 3D do współrzędnych obrazu. Zwróć uwagę, że współrzędna Z powinna być ujemna (kwestia przyjętego kierunku osi układu współrzędnych).
 - Zapoznaj się z dokumentacją funkcji `worldToImage`
- 6) Sekcja 5 – wybierz w którym miejscu ma pojawić się obiekt 3D na planszy, określając położenie 4 punktów w układzie współrzędnych planszy, np.



Uzupełnij fragment kodu (rzutowanie punktów 3D do współrzędnych obrazu) i zamieść w sprawozdaniu uzupełniony kod. Wybrane punkty są zapisywane do pliku MAT wykorzystywanego w dalszej części ćwiczenia.

- 7) Sekcja 6a, 6b – rzutowanie prostego obiektu (cylinder) na obraz
Uruchom sekcję eksperymentując z różnymi parametrami `myWorldPoints` z sekcji 5 oraz skalą obiektu w osi Z i promieniem cylindra.
Utwórz bardziej zaawansowany obiekt manipulując parametrem funkcji `cylinder` np.

```
t1 = 0:pi/20:0.5*pi;
r1 = 0.5 + sin(t1);
[X,Y,Z] = cylinder(r1);
surf(X,Y,Z)
```

Uzupełnił kod w skrypcie i zamieść w sprawozdaniu.

- 8) Sekcja 7 – W MATLABie można zaimportować model 3D w postaci chmury punktów i wykorzystać jako obiekt do wyświetlenia. Uruchom sekcję kodu wczytującą przykładowy model. Spróbuj dobrać skalowanie osi X, Y, Z tak aby obiekt dopasować do pola planszy. Jeśli dysponujesz innym modelem 3D w formacie STL możesz spróbować go zaimportować i użyć. Zamieść rezultaty w sprawozdaniu.

Część III – rozszerzona rzeczywistość

- 1) Otwórz skrypt `testHarness_AR.m` i zapoznaj się z nim. Ustaw parametry (numer kamery i rozdzielczość obrazu) w sekcji oznaczonej `UZUPEŁNIJ_1` :
Uwaga – rozdzielczość obrazu musi być taka sama jak podczas kalibracji kamery.
Ta część ćwiczenia wymaga do działania kamery. Jeśli kamera nie działa, możesz spróbować zrobić kalibrację offline używając nagranego pliku video. W tym celu w skrypcie należy odpowiednio zmodyfikować fragmenty kodu oznaczone `%VIDEO` (2 miejsca).

Skrypt realizuje następującą funkcjonalność:

- pobranie ramki z kamery
 - wywołanie algorytmu detekcji planszy i wyznaczania parametrów sceny 3D (`myAlgorithmAR.m`)
 - obliczenie rotacji i translacji kamery na podstawie parametrów kalibracji `parametryAlg.calibfilename`
 - skala i translacja (środek) obiektu na planszy kalibracyjnej wyznaczana na podstawie wybranych punktów `parametryAlg.pointsfilename` i `parametryAlg.skalaZ`
 - wizualizacja (`myVisualizationAR.m`)
- 2) Uruchom skrypt i zaobserwuj jego działanie. W tym celu w polu widzenia kamery powinna być widoczna cała plansza kalibracyjna. Zadbaj o to, aby kartka z planszą była prosta a nie pofałdowana np. naklejając ją na tekturkę. W jakich granicach nakładanie obiektu 3D na obraz działa poprawnie? Zamieść w sprawozdaniu przykładowe zrzuty ekranu.

Raport z ćwiczenia

Raport z ćwiczenia należy dostarczyć poprzez system UPEL, w formacie **PDF**. Raport składa się z 3 części. Pierwsza część zawiera rezultaty otrzymane w trakcie realizacji ćwiczenia i należy ją umieścić na UPEL pod koniec ćwiczeń. Kolejne części (analiza i wnioski oraz odpowiedzi na pytania), należy umieścić na UPEL w terminie określonym przez prowadzącego zajęcia (najczęściej jest to tydzień czasu – przed następnym ćwiczeniem).

W sprawozdaniu proszę zamieścić:

Data realizacji ćwiczenia i godzina: Imię i nazwisko:
<u>Rezultaty</u> 1) Cz.I: Zamieść rezultaty kalibracji kamery. 2) Cz.II: Zamieść rezultaty z tej części ćwiczenia oraz fragmenty kodu uzupełnione w trakcie ćwiczenia. 3) Cz.III: Zamieść rezultaty z tej części ćwiczenia
<u>Analiza i wnioski</u> 1) Zamieść obserwacje i wnioski z wszystkich części ćwiczenia 2) Cz.I: Wymień parametry zewnętrzne i wewnętrzne modelu kamery typu „pinhole”. Jakie jest ich znaczenie?
<u>Pytania</u> 1) Wymień i krótko uzasadnij 3 potencjalne praktyczne zastosowania algorytmów przedstawionych w ćwiczeniu. 2) Wymień i krótko uzasadnij 2 elementy z ćwiczenia, które zwróciły Twoją uwagę lub są Twoim zdaniem istotne do zrozumienia działania opisanych algorytmów. 3) Zadaj 1 pytanie związane z ww przykładami - co jest dla Ciebie niezrozumiałe lub niejasne.