POLITECHNIKA RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

KATEDRA INFORMATYKI I AUTOMATYKI POLITECHNIKI RZESZOWSKIEJ

Geometria, kalibracja i rejestracja

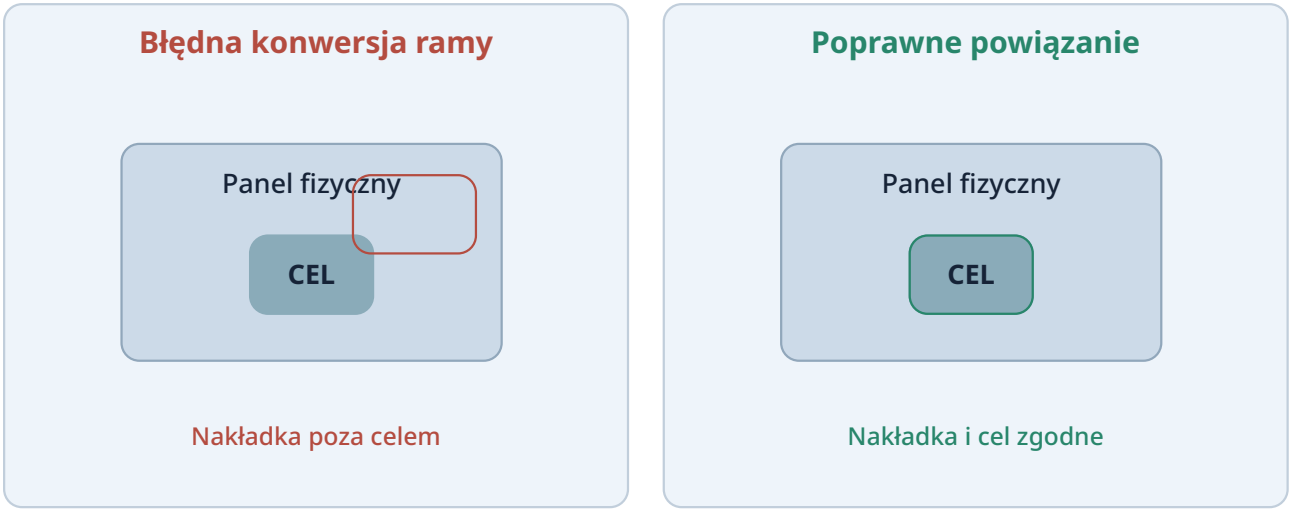
WiRR · Wykład 03 · Automatyka i Robotyka · 3 ECTS · 2026/2027 · v1.1.5

dr inż. Mateusz Pomianek

Katedra Informatyki i Automatyki, Politechnika Rzeszowska

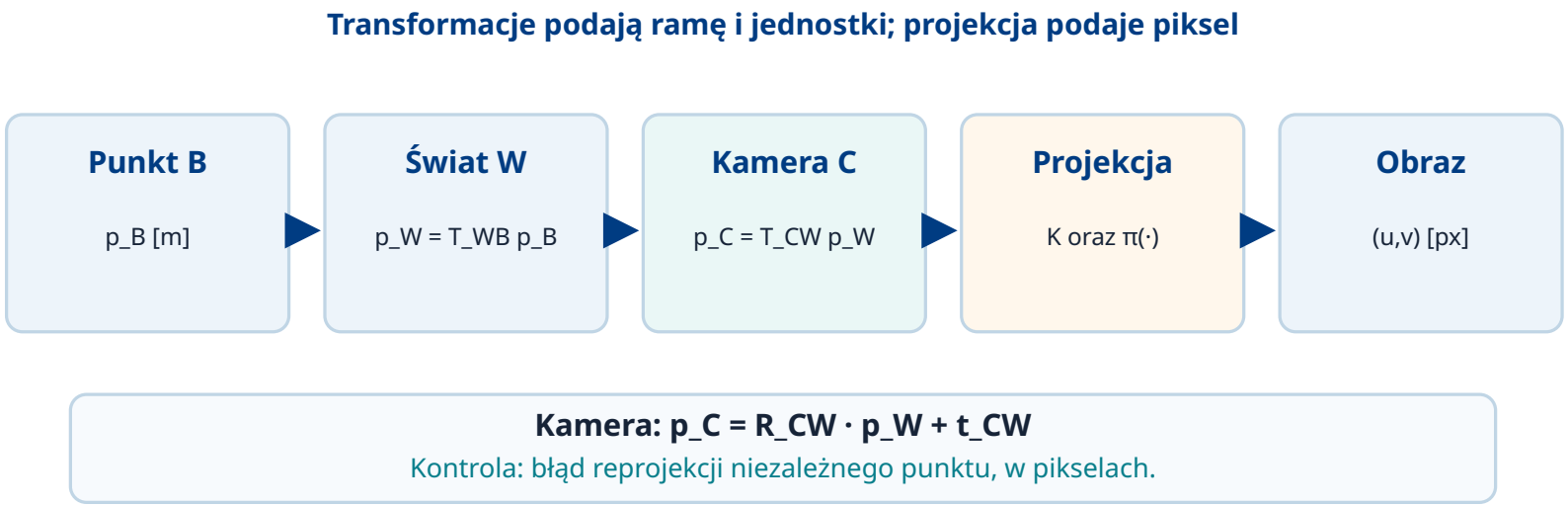
19.10.2026

Jeden obiekt, dwa wyniki rejestracji AR



Nakładka jest poprawna, gdy pozostaje związana z **tym samym fizycznym celem**.

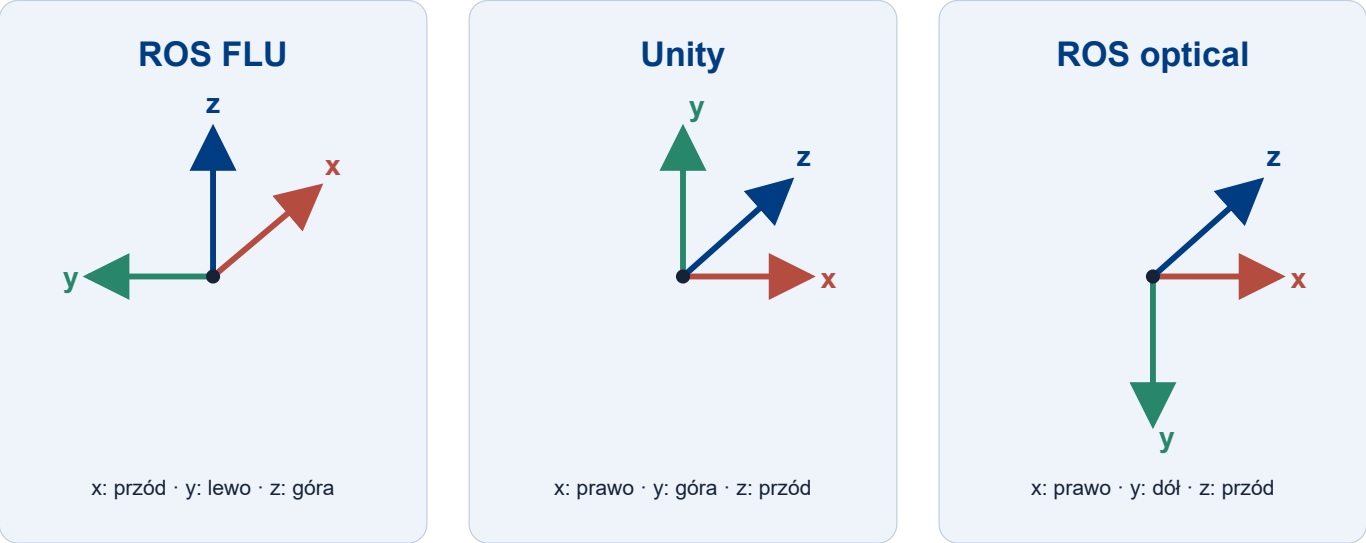
Łańcuch danych: od robota do piksela



Poza robota → transformacja świata → kamera → projekcja → obraz → kontrola błędu.

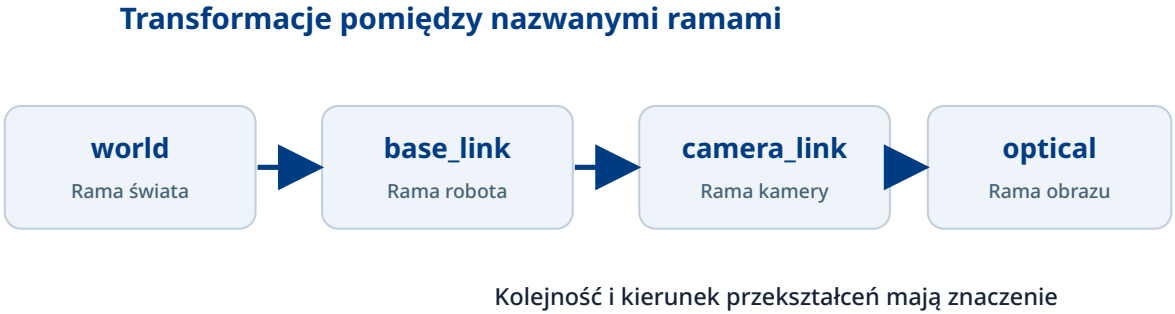
Każdy etap wymaga zgodnych **jednostek, ramek i czasu**.

ROS FLU, Unity i rama optyczna kamery



- ROS FLU: x do przodu, y w lewo, z w górę.
- Unity: x w prawo, y w górę, z do przodu.
- ROS optical: x w prawo, y w dół, z do przodu.

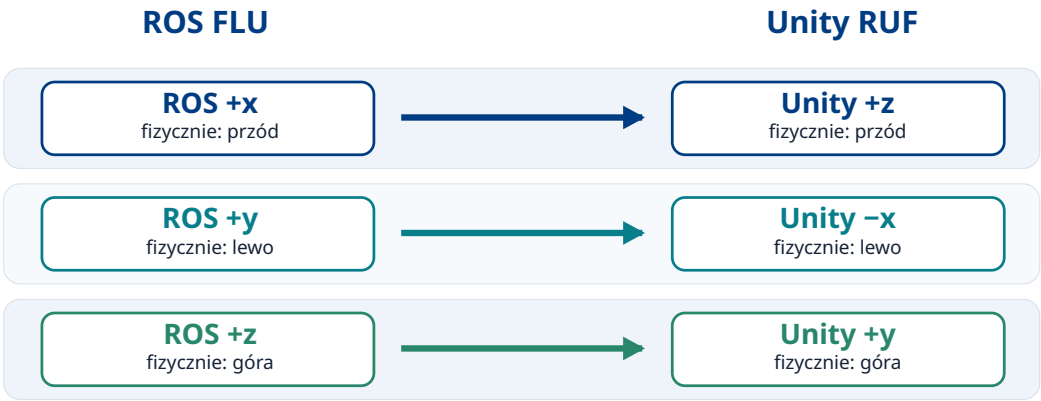
Ramy odniesienia określają znaczenie wektora



$$\tilde{\mathbf{p}}_A = {}^{A}T_B \, \tilde{\mathbf{p}}_B \qquad \text{(we współrzędnych jednorodnych)}.$$

Symbol ${}^{A}T_B$ oznacza przekształcenie współrzędnych punktu z ramy B do ramy A.

ROS FLU do Unity: jawna zmiana bazy



$$M = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{p}_U = M\mathbf{p}_R, \qquad \det M = -1.$$

M nie jest fizycznym obrotem obiektu: zmienia reprezentację i skrętność układu.

Jeden metr do przodu: dwie reprezentacje



$$\mathbf{p}_R = (1,0,0)^T\,\mathrm{m} \quad \Rightarrow \quad \mathbf{p}_U = (0,0,1)^T\,\mathrm{m}.$$

Wynik dotyczy **ramy ciała FLU**, a nie ramy optycznej kamery.

Orientację przelicza się inaczej niż pozycję



Zmiana bazy dla R wymaga M z lewej i M^T z prawej

$$R_U = M R_R M^T, \qquad R_R^T R_R = I.$$

$\det R_U = 1$ mimo $\det M = -1$. W konwersji orientacji trzeba uwzględnić **dwie zmiany bazy**; nie mnożymy kwaternionu jak wektora pozycji.

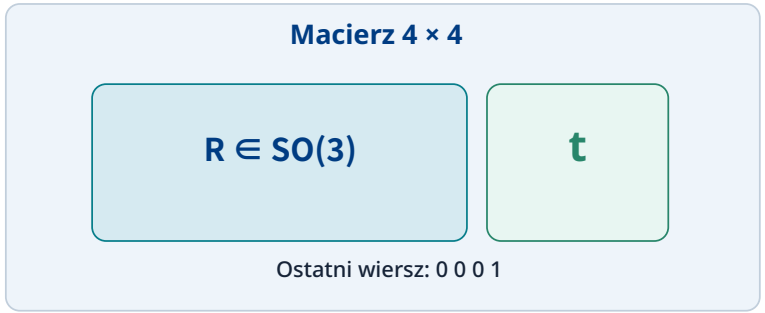
Test trzech osi i obrotu

```
assert ros_body_to_unity((1, 0, 0)) == (0, 0, 1)
assert ros_body_to_unity((0, 1, 0)) == (-1, 0, 0)
assert ros_body_to_unity((0, 0, 1)) == (0, 1, 0)
assert matmul(M, transpose(M)) == IDENTITY
assert det3(M) == -1
assert matvec(ros_rotation_to_unity(RZ90), (0, 0, 1)) == (-1, 0, 0)
```

Pełny plik i test orientacji.

Wynik numeryczny nie zastępuje kontroli poprawności w Unity 6000.6.5f1.

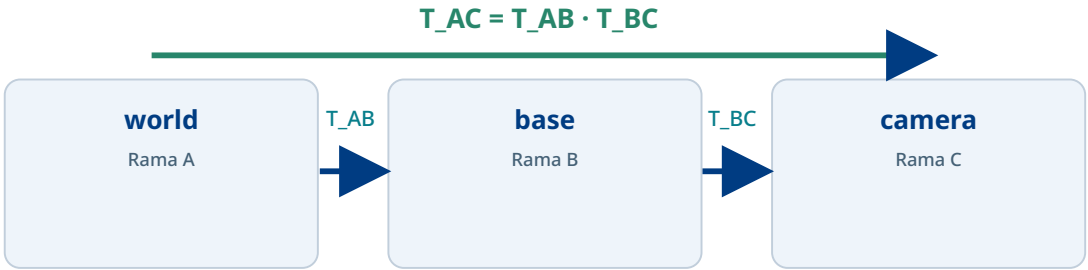
Macierz jednorodna: obrót i translacja



$$ {}^{A}T_{B} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^{T} & 1 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{p}_{A} \\ 1 \end{bmatrix} = {}^{A}T_{B} \begin{bmatrix} \mathbf{p}_{B} \\ 1 \end{bmatrix}. $$

Translacja $\mathbf{t}$ jest wyrażona w ramie A; R zmienia współrzędne kierunków.

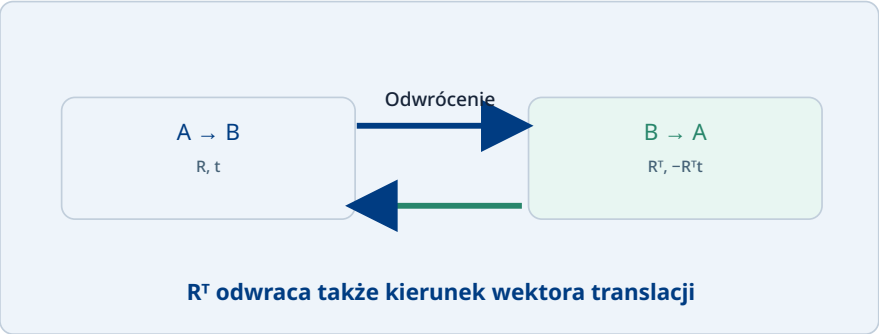
Składanie transformacji: zgodne ramy



$$ {}^{A}T_{C} = {}^{A}T_{B}\,{}^{B}T_{C}. $$

Transformacje składa się tylko przez **tę samą ramę pośrednią**. Kolejność mnożenia nie jest dowolna.

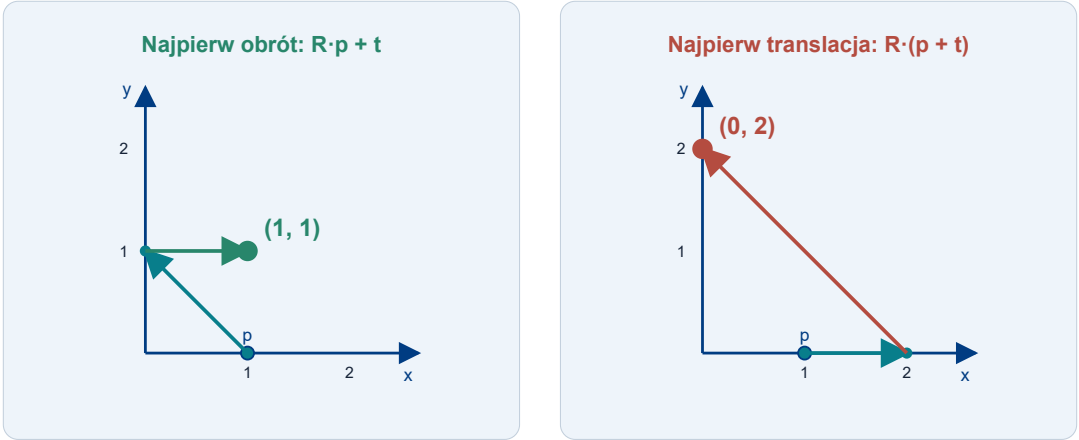
Odwrotność macierzy jednorodnej



$$({}^{A}T_{B})^{-1} = {}^{B}T_{A} = \begin{bmatrix} R^{T} & -R^{T}\mathbf{t} \\ \mathbf{0}^{T} & 1 \end{bmatrix}.$$

Przy niezerowym obrocie nie wystarcza zmiana samego znaku translacji.

Dlaczego kolejność działań zmienia wynik?

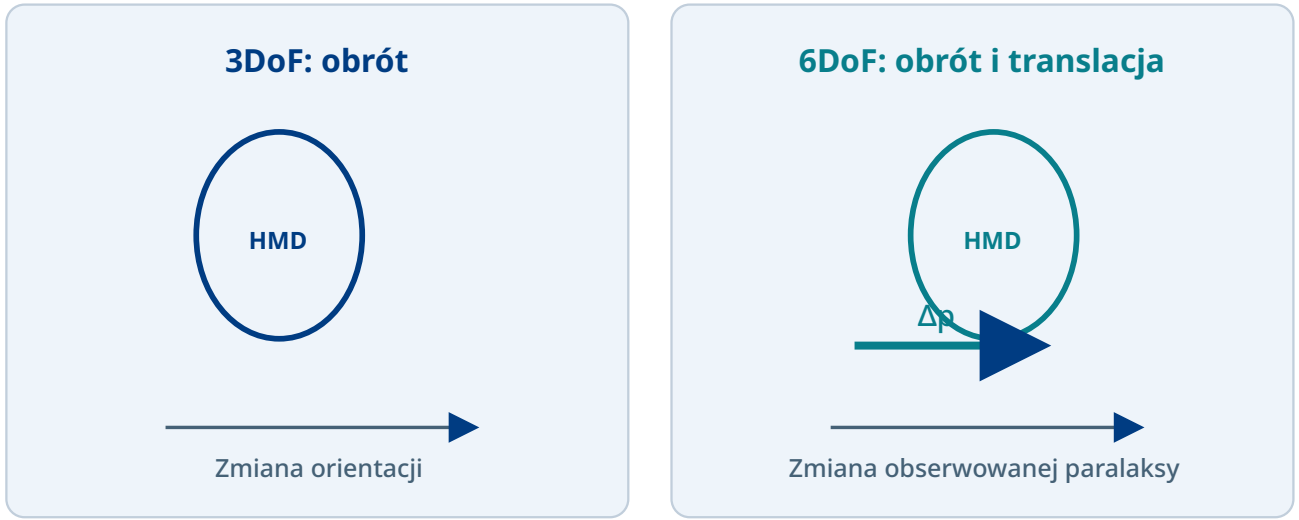


Dla $\mathbf{p} = (1, 0)$, $\mathbf{t} = (1, 0)$ i $R_z(90°)$:

$$R\mathbf{p} + \mathbf{t} = (1, 1), \qquad R(\mathbf{p} + \mathbf{t}) = (0, 2).$$

To dwa różne wyniki, mimo użycia tych samych operacji.

3DoF i 6DoF: orientacja oraz przemieszczenie

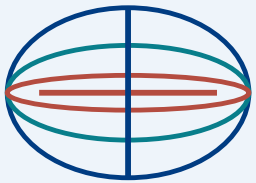


3DoF: trzy stopnie swobody orientacji. **6DoF:** dodatkowo trzy składowe pozycji.

Podczas translacji zmienia się paralaksa obiektów znajdujących się w różnych odległościach.

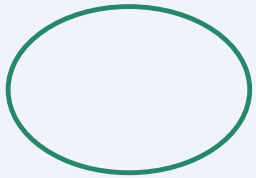
Euler, gimbal lock i kwaterniony

Kąty Eulera



Przy szczególnej pozy dwie osie pokrywają się

Kwaternion



Brak osobliwości parametrów Eulera

Kąty Eulera wymagają określenia **kolejności osi** i mogą mieć osobliwość reprezentacji.

Kwaternion jednostkowy spełnia $\|q\| = 1$ i nie ma osobliwości kątów Eulera. Kwaterniony q i $-q$ oznaczają tę samą orientację.

Kontrola pełnej pozy w środowisku 3D



Dla pozycji ROS $(1; 0; 0)$ m i obrotu $R_z(90^\circ)$ otrzymujemy w Unity $(0; 0; 1)$ m oraz obrót $R_y(-90^\circ)$. Test orientacji tożsamościowej nie wystarcza.

Sprawdzamy pozycję i orientację, zanim zaczniemy dopasowywać obraz.

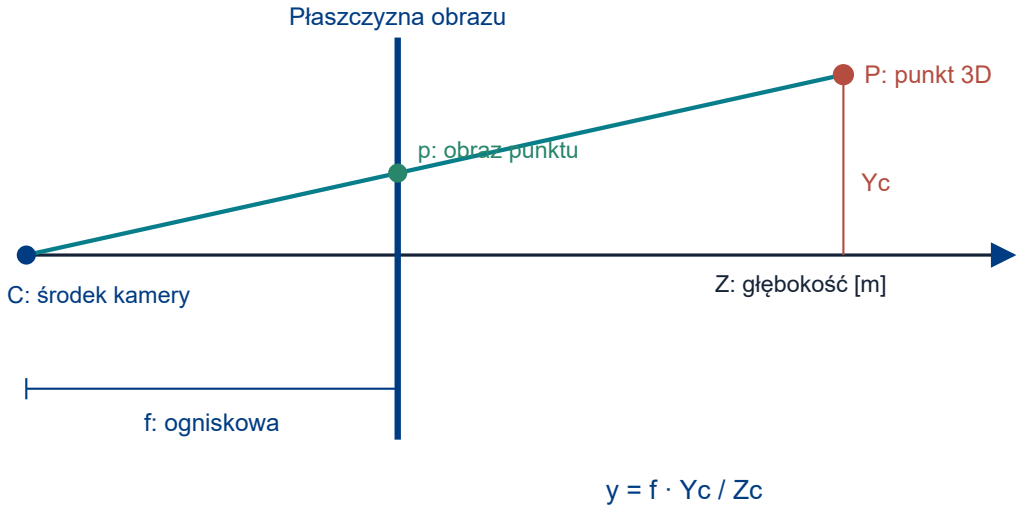
Przerwa 5 minut

5 minut przerwy

Pytania do samodzielnego powtórzenia

1. Przelicz $(0,4; 0,1; 0,8)$ m z ROS FLU na Unity.
2. Dlaczego odwrotność macierzy jednorodnej zawiera translację $-R^T\mathbf{t}$?
3. Wyznacz piksel dla $(0,3; -0,15; 2,0)$ m, $f_x = f_y = 800$ px i $(c_x; c_y) = (640; 360)$ px.
4. Dlaczego niski FRE nie oznacza automatycznie niskiego TRE w punkcie docelowym?

Kamera otworkowa: geometria punktu i piksela

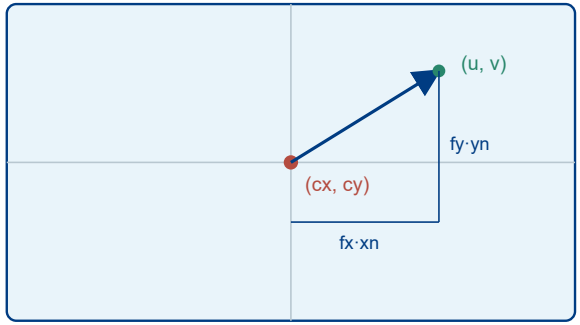


Dla punktu (X_C, Y_C, Z_C) przed kamerą ($Z_C > 0$):

$$x_n = X_C/Z_C, \qquad y_n = Y_C/Z_C.$$

Współrzędne znormalizowane powstają przez **rzut perspektywiczny**.

Macierz K i kąt pola widzenia



u = fx·xn + cx

v = fy·yn + cy

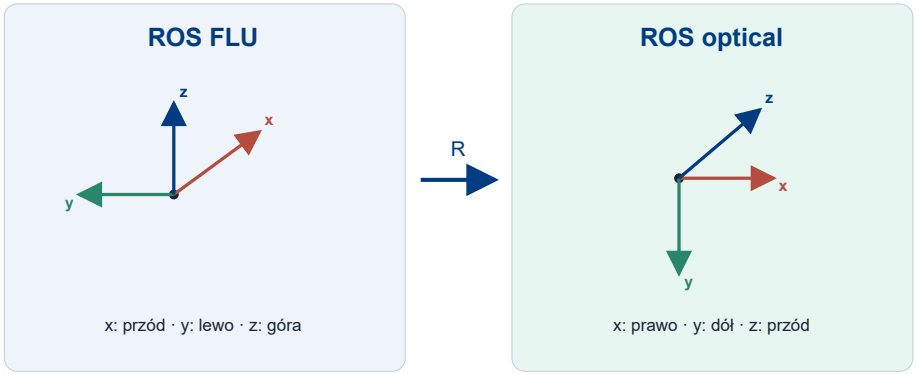
fx, fy: piksele

cx, cy: punkt główny

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{FOV}_x = 2\arctan\left(\frac{W}{2f_x}\right).$$

Dla $W = 1280$ px i $f_x = 800$ px otrzymujemy $\mathrm{FOV}_x \approx 77,3^\circ$. Unity standardowo stosuje pionowy FOV: $\mathrm{FOV}_y = 2\arctan(H/(2f_y))$.

Układ optyczny kamery i zniekształcenia



Optyczna rama ROS ma x w prawo, y w dół oraz z do przodu. FLU $\rightarrow$ optical: $x_o = -y_F$, $y_o = -z_F$, $z_o = x_F$; jest to obrót o wyznaczniku $+1$, a zmiana bazy ROS $\rightarrow$ Unity ma wyznacznik -1.

Wzór projekcji przez K bez modelu dystorsji stosujemy dla **obrazu skorygowanego**. Obraz surowy wymaga zgodnych współczynników zniekształceń.

Punkt świata → piksel: przykład obliczeniowy

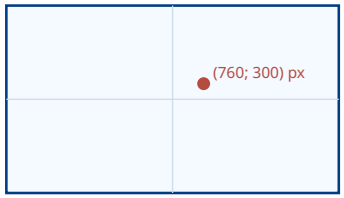
Punkt kamery: (0,3; −0,15; 2,0) m

fx = fy = 800 px

cx = 640 px; cy = 360 px

u = 800 · 0,3 / 2 + 640 = 760

v = 800 · (−0,15) / 2 + 360 = 300



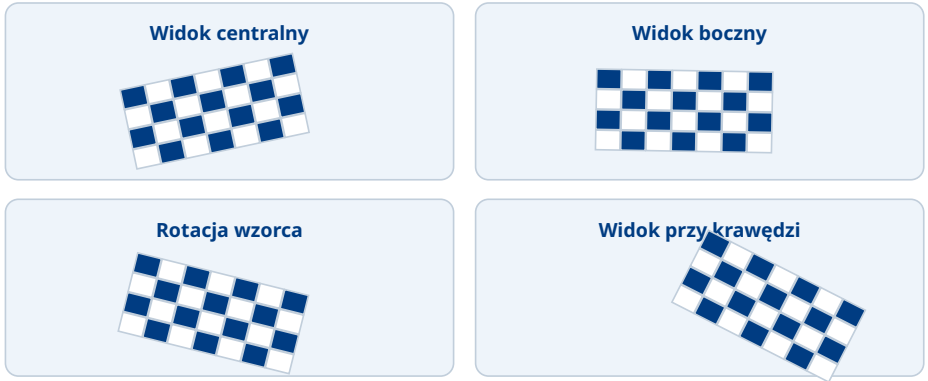
Obraz 1280 × 720 px

Punkt leży w granicach obrazu

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \mid \mathbf{t}]\begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}.$$

Dla obrazu 1280 × 720 px, $(X_C; Y_C; Z_C) = (0,3; -0,15; 2,0)$ m, $f_x = f_y = 800$ px i $(c_x; c_y) = (640; 360)$ px otrzymujemy $\mathbf{(u; v) = (760; 300)}$ **px**. Tutaj $R = I$, $\mathbf{t} = 0$ i $s = Z_C$.

Kalibracja kamery: różne widoki tego samego wzorca



Zróżnicowanie położenia, kąta i obszaru obrazu wzorca pomaga ograniczyć niejednoznaczności kalibracji.

Metoda Zhanga: różnorodne widoki wzorca pozwalają oszacować K, dystorsję i błąd RMS reprojekcji. Odrzucamy ujęcia rozmyte; zachowujemy docelową rozdzielczość, ostrość i ogniskową.

PnP i RANSAC: geometria znana, poza nieznana

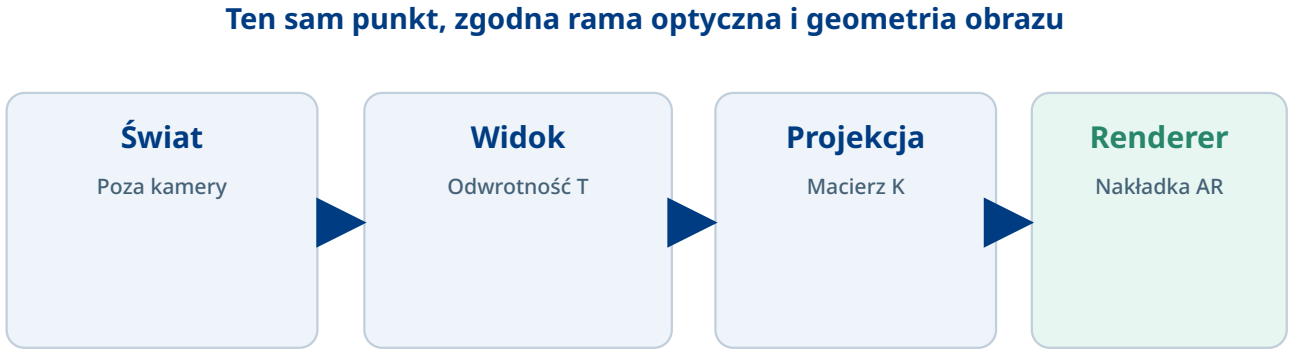


Odstające korespondencje nie powinny dominować wyniku

Wejście: punkty 3D, odpowiedniki 2D, macierz K i model dystorsji.

Wynik: `rvec` (oś–kąt, Rodrigues) i `tvec` (jednostki punktów 3D). `solvePnPRansac` odrzuca odstające korespondencje; dla płaskich markerów należy kontrolować możliwą dwuznaczność pozy.

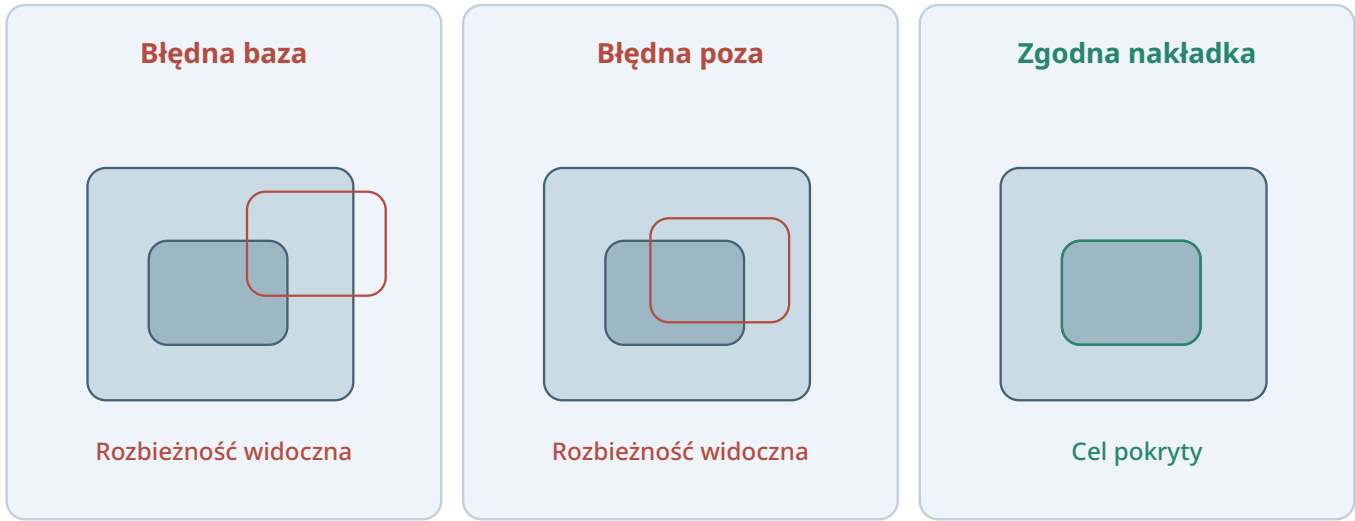
Macierz widoku łączy kamerę fizyczną i wirtualną



$$ {}^{C}T_{W} = ({}^{W}T_{C})^{-1}. $$

Macierz widoku odwraca pozę kamery w tej samej ramie. Projekcję renderera należy uzgodnić z K, pionowym FOV, punktem głównym i kierunkiem osi obrazu.

Rejestracja: przed i po korekcji osi i pozy



Zmiana ramy, ustawienie kamery i kalibracja mogą powodować **różne wzory odchylenia nakładki**.

Mapa reszt reprojekcji



$$e_i = \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2} \quad [\mathrm{px}].$$

Badamy kierunki i wielkości reszt w całym polu obrazu, nie tylko jedną średnią.

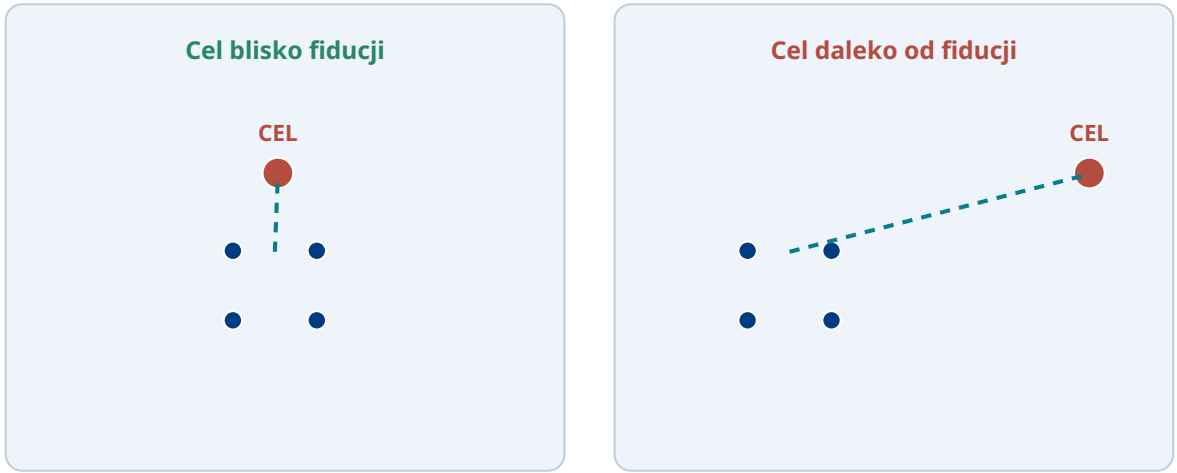
FRE i TRE: dopasowanie a punkt kontrolny



FRE to reszty na punktach dopasowania. **TRE** ocenia błąd w niezależnym punkcie zadania.

Metoda Kabscha: centroidy → kowariancja → SVD → korekta wyznacznika obrotu do $+1$. Niski FRE **nie gwarantuje** niskiego TRE.

Rozmieszczenie punktów i efekt dźwigni

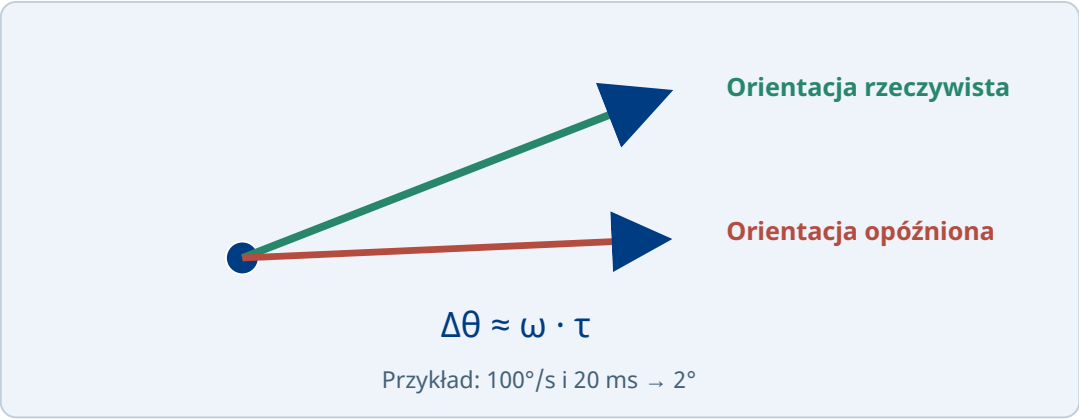


Wzrost ramienia L wzmacnia wpływ błędu kąta na przesunięcie

$$\| \Delta \mathbf{p} \| \approx L \, \Delta\theta \quad (\Delta\theta \ \text{ w radianach}).$$

L to odległość celu od środka punktów odniesienia. Dla błędu $0,5^\circ$: przy $L = 1$ m przesunięcie wynosi około **8,7 mm**, a przy $L = 3$ m około **26 mm**.

Opóźnienie powoduje dynamiczny błąd rejestracji



$$\Delta\theta \approx \omega\tau, \quad 100^\circ/\mathrm{s} \cdot 0{,}020\,\mathrm{s} = \boxed{2^\circ}\,.$$

Błąd 2° odpowiada około **3,5 cm** przy ramieniu 1 m. Poprawna kalibracja statyczna nie usuwa skutków opóźnienia.

Procedura diagnozy błędu nakładki



Sprawdzamy **osie → kierunek transformacji → jednostki → orientację → projekcję → czas i opóźnienie → niezależny punkt**.

Kryterium końcowe to zgodność nakładki z fizycznym celem w obszarze zadania.

Raport odbioru kalibracji: co trzeba udokumentować?

- **Układy odniesienia:** nazwy ram, osie, jednostki i kierunek transformacji.
- **Kalibracja:** wersja parametrów i sposób wyznaczania transformacji.
- **Walidacja:** niezależne punkty kontrolne, położenie celu i błąd w obszarze zadania.
- **Dynamika:** znaczniki czasu, opóźnienie i warunki ruchu urządzenia.

Niski FRE w próbie kalibracyjnej nie zastępuje niezależnej oceny TRE.

Bibliografia

- ROS, REP-103: konwencje jednostek i osi.
- ROS 2 Humble, tf2: przekształcenia ram.
- Unity, Quaternion and Euler rotations.
- OpenCV, Camera Calibration and 3D Reconstruction.
- B. K. P. Horn, Closed-form solution of absolute orientation, 1987.
- M. A. Fischler, R. C. Bolles, Random Sample Consensus, *Commun. ACM*, 1981.
- Z. Zhang, A Flexible New Technique for Camera Calibration, *IEEE TPAMI*, 2000.
- J. M. Fitzpatrick, J. B. West, *The Distribution of Target Registration Error in Rigid-Body Point-Based Registration*, *IEEE TMI*, 2001.