

Stereowizja dwukamerowa

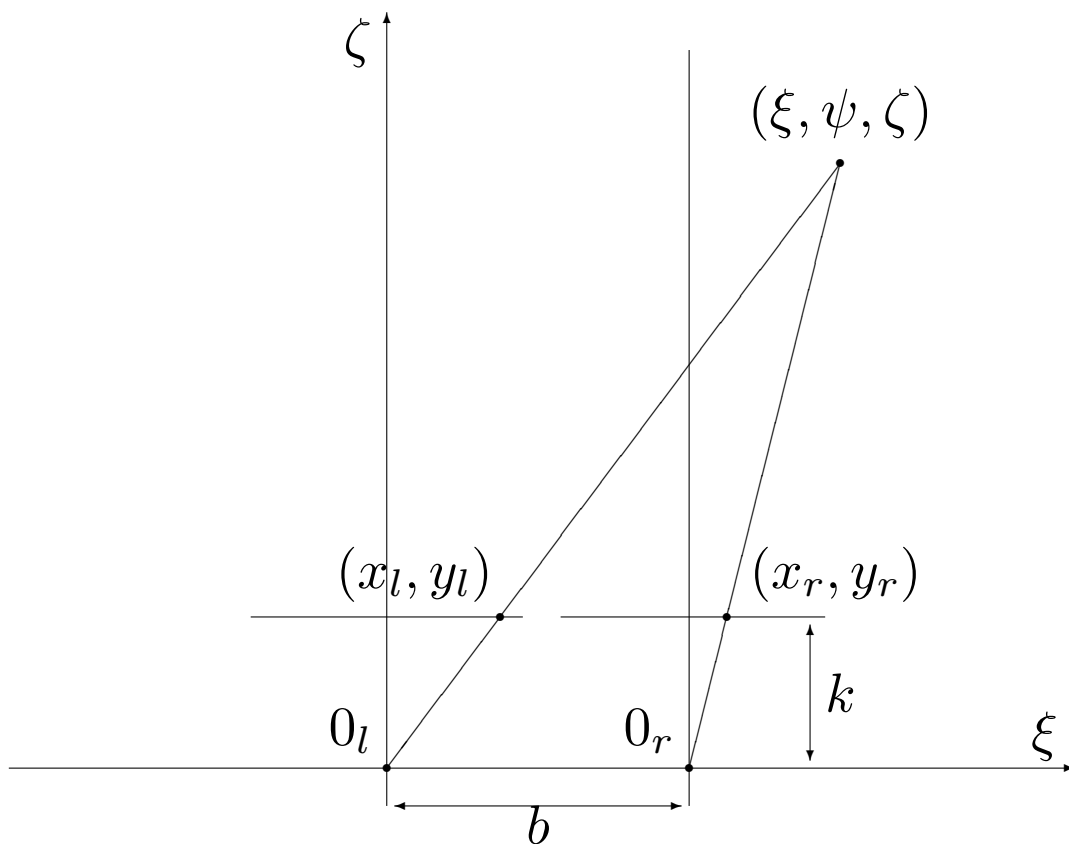
Podstawowe problemy:

- dopasowywanie obrazów
- kalibracja układu kamer
- rekonstrukcja (triangulacja)

Cechy obrazu do dopasowywania:

- jasność punktów obrazu (metody korelacyjne)
- kierunek i wartość gradientu na krawędziach
- parametry krawędzi (krzywizna, długość, orientacja)
- parametry sylwetek

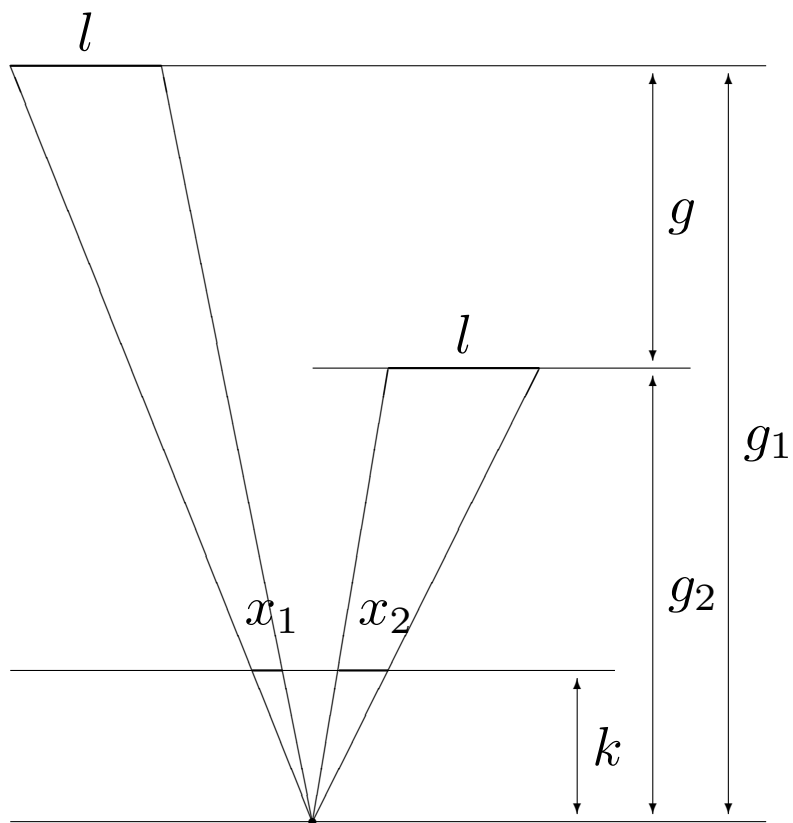
Rekonstrukcja trójwymiarowa



$$\frac{\xi}{x_l} = \frac{\xi - b}{x_l - d} = \frac{\psi}{ay_l} = \frac{\zeta}{k}; \quad d = x_l - x_r$$

$$\xi = \frac{x_l}{d}b; \quad \psi = \frac{y_l}{d}ab; \quad \zeta = \frac{1}{d}kb$$

Kalibracja odległości obrazowej



$$g = g_1 - g_2 = kl \left(\frac{1}{x_1} - \frac{1}{x_2} \right)$$

$$k = \frac{gx_1x_2}{l(x_2 - x_1)}$$

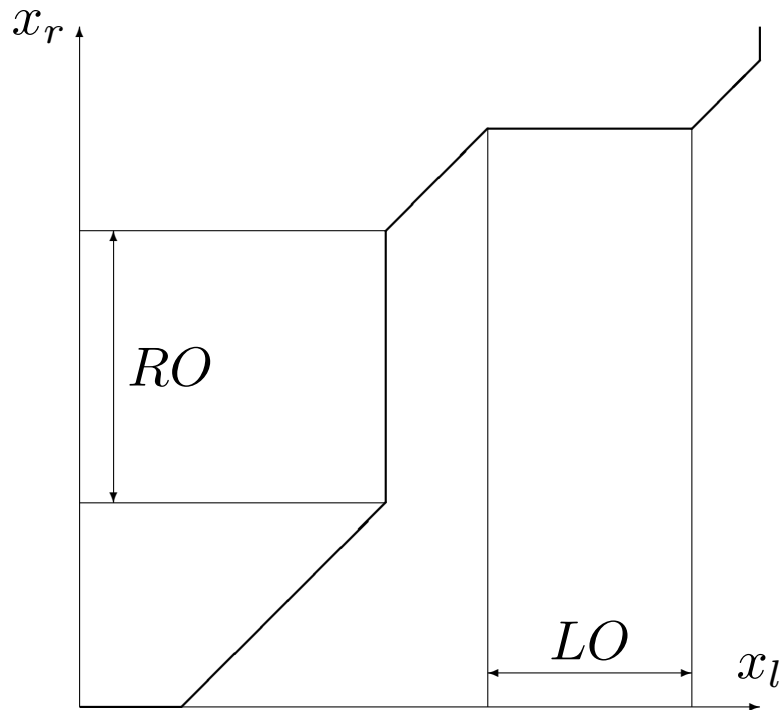
Algorytm dopasowywania linii epipolarnych

$$Q_y = \sum_{x=0}^{N-1} (f_l(x, y) - f_r(x + \delta(x, y), y))^2$$

Założenia i ograniczenia:

1. nośnikiem informacji o scenie jest jasność punktów obrazu $f(x, y)$;
2. dopasowaniu podlegają jednowymiarowe przebiegi funkcji jasności będące przekrojami obrazów wzdłuż odpowiadających sobie linii epipolarnych.
3. jednoznaczność - punkt lewego obrazu można dopasować co najwyżej do jednego punktu prawego obrazu (i odwrotnie);
4. monotoniczność - jeśli punkt $(x_{l,1}, y)$ jest dopasowany do $(x_{r,1}, y)$, i $(x_{l,2}, y)$ jest dopasowany do $(x_{r,2}, y)$, to jeśli $x_{l,1} < x_{l,2}$ to również $x_{r,1} < x_{r,2}$.

Przystanianie (*occlusion*)



Jednokrokowa funkcja kosztu dopasowania:

$$C(i, j) = \min \left\{ \begin{array}{l} C(i-1, j-1) + c(i, j), \\ C(i-1, j) + O, \\ C(i, j-1) + O \end{array} \right\}$$

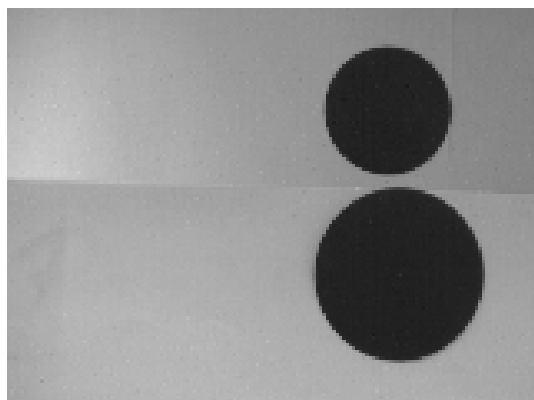
$c(i, j)$ – lokalny koszt błędu dopasowania punktu i na lewym obrazie do punktu j na prawym obrazie

Algorytm ML (*Maximum Likelihood*)

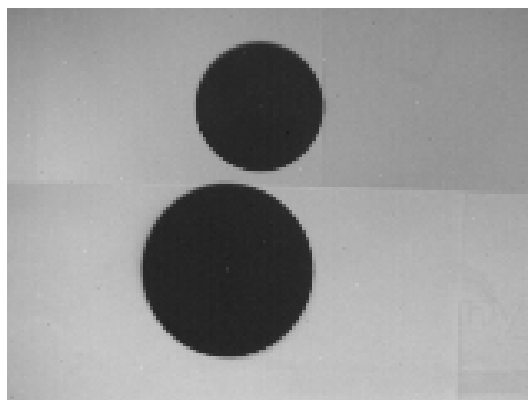
```
for (i=1; i<=N; i++) C(i,0) = i * Occlusion;
for (j=1; j<=N; j++) C(0,j) = j * Occlusion;
for (i=1; i<=N; i++) {
    for (j=1; j<=M; j++) {
        min1 = C(i-1,j-1) + c(i,j);
        min2 = C(i,j-1) + Occlusion;
        min3 = C(i-1,j) + Occlusion;
        C(i,j) = cmin = min(min1,min2,min3);
        if(min1==cmin) M(i,j) = 1;
        if(min2==cmin) M(i,j) = 2;
        if(min3==cmin) M(i,j) = 3;
    }
}
p=N; q=M;
while ( p!=0 && q!=0 ) {
    P(p,q) = 1;          /* dodanie (p,q) do sciezki */
    switch ( M(p,q) ) {
        case 1:  p--; q--; break; /* p i q tworza pare */
        case 2:  p--; break;      /* p jest bez pary */
        case 3:  q--; break;      /* q jest bez pary */
    }
}
```

Obrazy do rekonstrukcji stereo

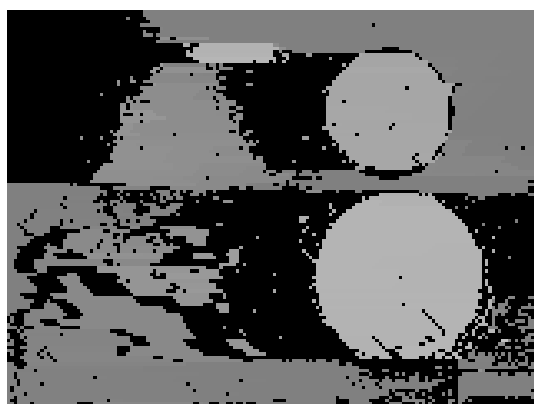
lewa kamera



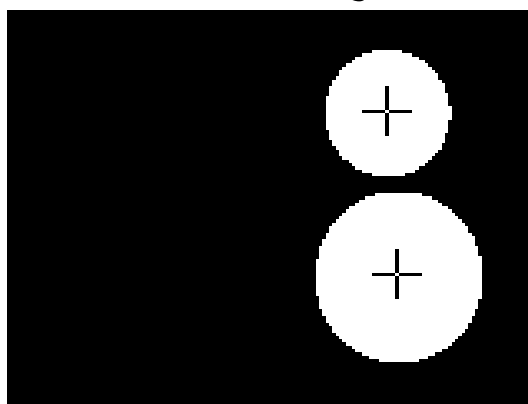
prawa kamera



rozbieżność



lokalizacja



Niejednoznaczność dopasowania linii

```

L  1 1 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
R  1 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0

```

```

L  1 1 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
   | | | | | | | | | | | | \ \ \ \ \ \ \ \ \
R  1 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0

```


Algorytm MLMH+V (*Maximum Likelihood, Minimum Horizontal plus Vertical discontinuities*)

Macierze DVv, DVh, DVd – ilość nieciągłości przy dopasowaniu pierwszych i punktów lewego obrazu do pierwszych j punktów prawego przy osiągnięciu (i, j) krokiem w odpowiednim kierunku.

```
for (i=1; i<=N; i++) C(i,0) = i * Occlusion;
for (j=1; j<=N; j++) C(0,j) = j * Occlusion;
for (i=1; i<=N; i++) {
  for (j=1; j<=M; j++) {
    min1 = C(i-1,j-1) + c(i,j);
    min2 = C(i,j-1) + Occlusion;
    min3 = C(i-1,j) + Occlusion;
    C(i,j) = cmin = min(min1,min2,min3);

    if(min1==cmin) {
      vc=IsMatched(i); /* dopasowanie w poprzedniej linii */
      DVd(i,j)=imin(DVd(i-1,j-1),DVh(i-1,j-1)+1,
                  DVv(i-1,j-1)+1)+vc;
    }
    else DVd=HUGE;
  }
}
```

```

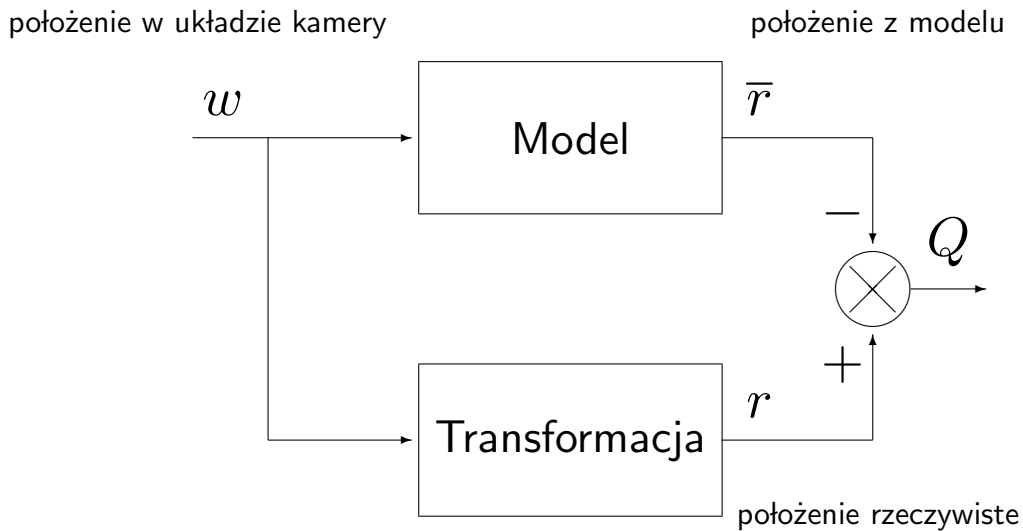
if(min2==cmin) {
    vc=IsLeftOcclusion(i); /* lewe przysłonicie */
    DVh(i,j)=imin(DVd(i-1,j)+1,DVh(i-1,j),DVv(i-1,j))+vc;
else DVh=HUGE;

if(min3==cmin) {
    vc=IsRightOcclusion(i); /* prawe przysłonicie */
    DVv(i,j)=imin(DVd(i,j-1)+1,DVh(i,j-1),DVv(i,j-1))+vc;
else DVv=HUGE;
}
}

p=N; q=M;
d1=d2=d3=e=f=0;
while ( p>0 && q>0 ) {
    P(p,q) = 1; /* dodanie (p,q) do sciezki */
    p-=e; q-=f; break;
    switch (imin(DVd(p,q)+d1,DVv(p,q)+d2,DVh(p,q)+d3)) {
        case 1: /* p i q tworza pare */
            d1=0; d2=1; d3=1; e=1; f=1; break;
        case 2: /* p jest bez pary */
            d1=1; d2=0; d3=1; e=1; f=0; break;
        case 3: /* q jest bez pary */
            d1=1; d2=1; d3=0; e=0; f=1; break;
    }
}
}

```

Związek między układami współrzędnych systemu wizyjnego i robota



$$r = Tw \quad r = [\xi, \psi, \zeta, 1]^T \quad w = \left[\frac{x_l}{d}, \frac{y_l}{d}, \frac{1}{d}, 1 \right]^T$$

$$\begin{bmatrix} \xi \\ \psi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x_l}{d} \\ \frac{y_l}{d} \\ \frac{1}{d} \\ 1 \end{bmatrix}$$

Estymacja parametrów transformacji

Serie pomiarowe:

$$W_p = [w_1, w_2, \dots, w_p] \quad R_p = [r_1, r_2, \dots, r_p]$$

Kryterium jakości:

$$Q = \sum_{i=1}^p (r_i - Tw_i)^T (r_i - Tw_i)$$

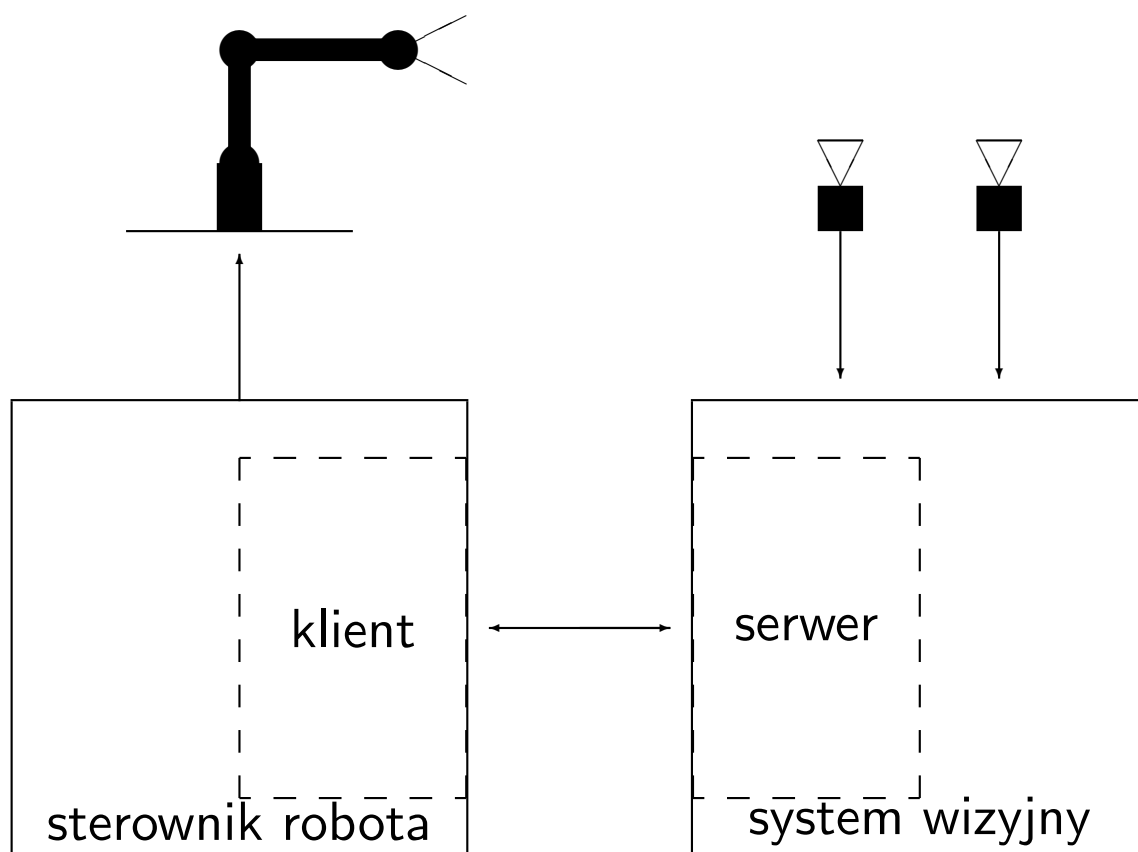
Model transformacji:

$$\bar{T}_p = R_p W_p^T (W_p W_p^T)^{-1}$$

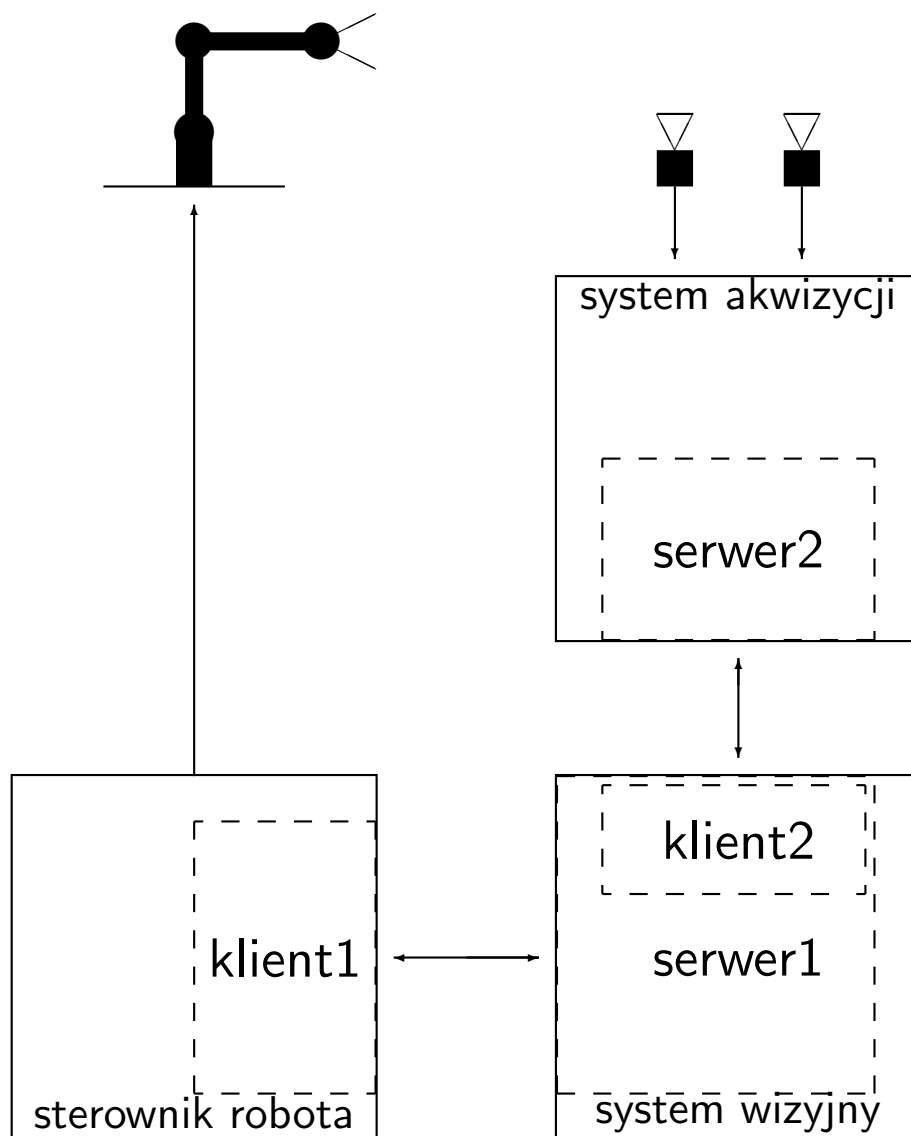
Transformacja zredukowana do przypadku dwuwymiarowego:

$$\begin{bmatrix} \xi \\ \psi \\ \zeta \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & 0 & t_{14} \\ t_{21} & t_{22} & 0 & t_{24} \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

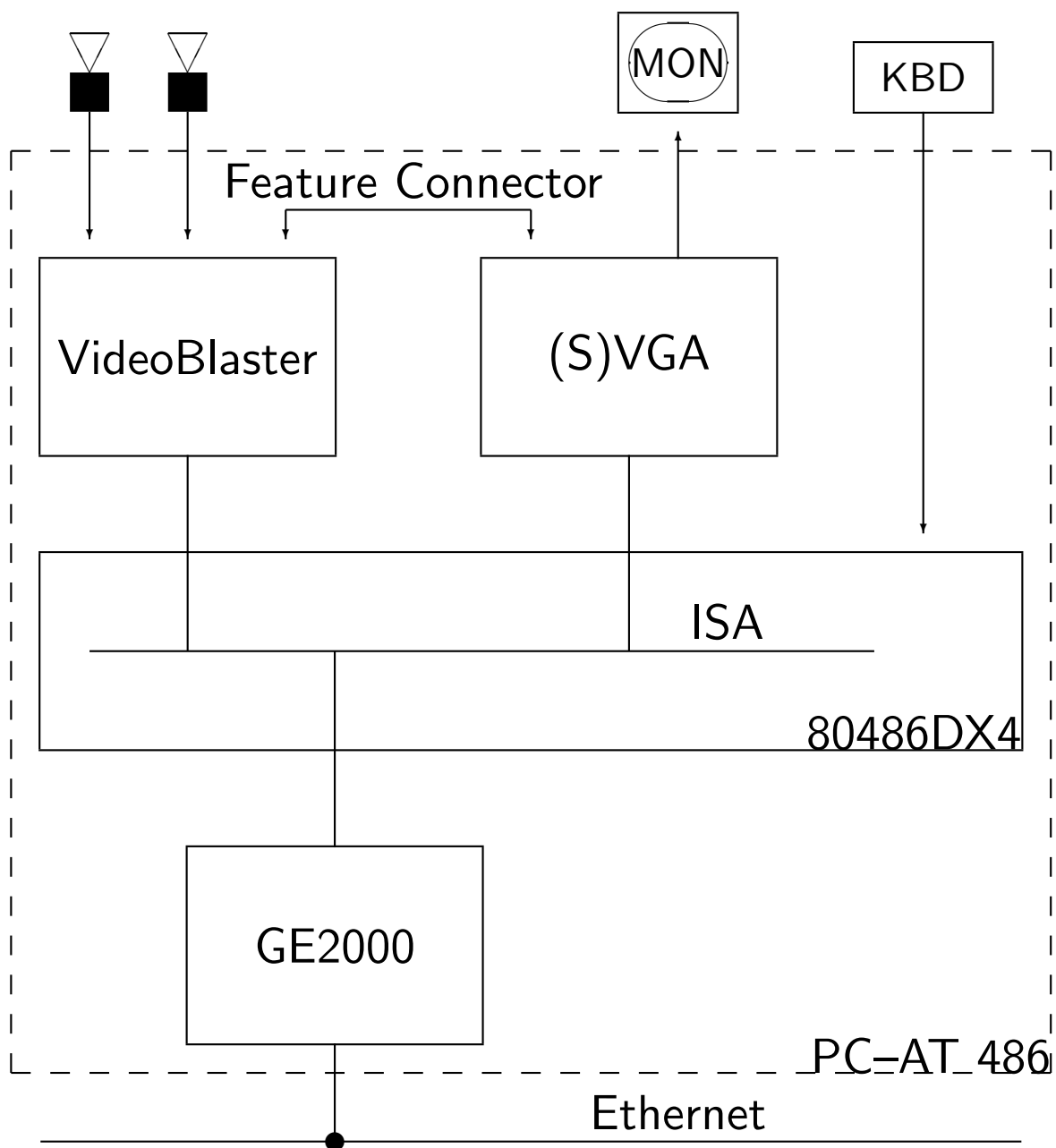
Współpraca robota z systemem wizyjnym w układzie klient-serwer



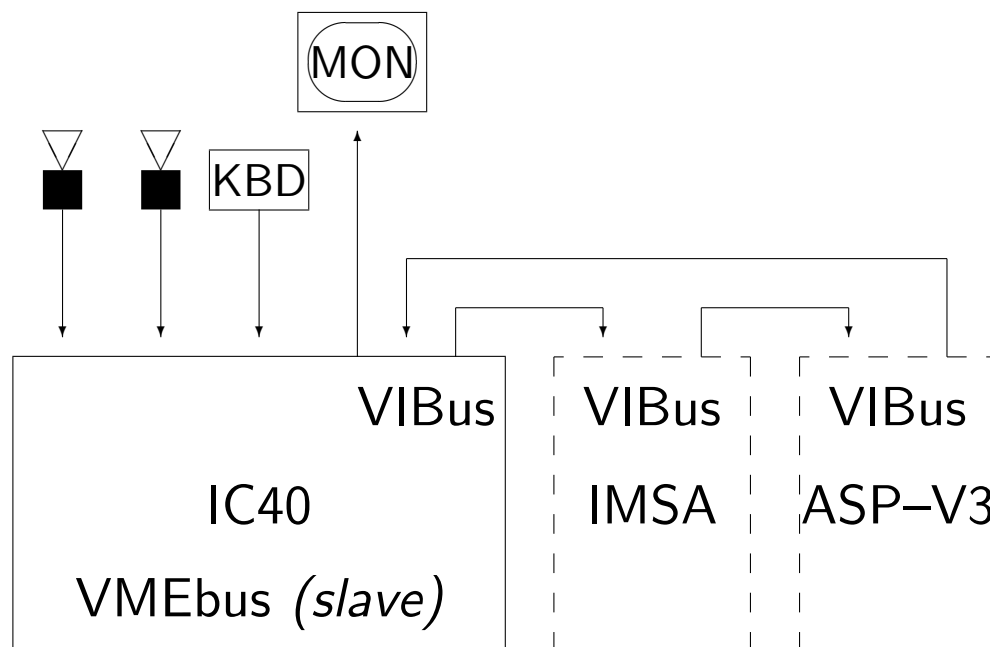
Kaskadowa konfiguracja systemu rozproszonego klient-serwer



System akwizycji na magistrali ISA



Komputer wizyjny IC40



- wyświetlanie obrazów z wejść wideo, magistrali VIBus lub pamięci obrazu, z wyjściem RGB przez programowalną paletę (CLUT),
- procesor MC68040/33MHz (27MIPS, 4.6MFLOPS),
- 4 wejścia wideo RS170/CCIR z przetwornikiem 8-bitowym (do 15MHz),
- wyjście (*master*) magistrali VIBus,
- wejście (*slave*) magistrali VIBus,
- 4 megabajty pamięci obrazu,
- bezpośredni dostęp do pamięci obrazu z magistrali VMEbus,
- dwa porty szeregowo (MC68681).

Magistrala obrazowa VIBus

Sygnaly magistrali VIBus:

D0..D23 linie danych

RAWCLOCK synchronizacja punktów (sygnał zegarowy)

CSYNC* synchronizacja zespolona (typu telewizyjnego)

VSYNC* synchronizacja pionowa (pola)

HSYNC* synchronizacja pozioma (linii)

DATAVALID maskowanie nieużytecznej części danych w linii

REQUEST maskowanie nieużytecznej części danych w polu

EVENFIELD parzystość pola (obrazy typu *interlaced*)

Moduły typu *master* dostarczają sygnałów sterujących, a moduły typu *slave* z nich korzystają.

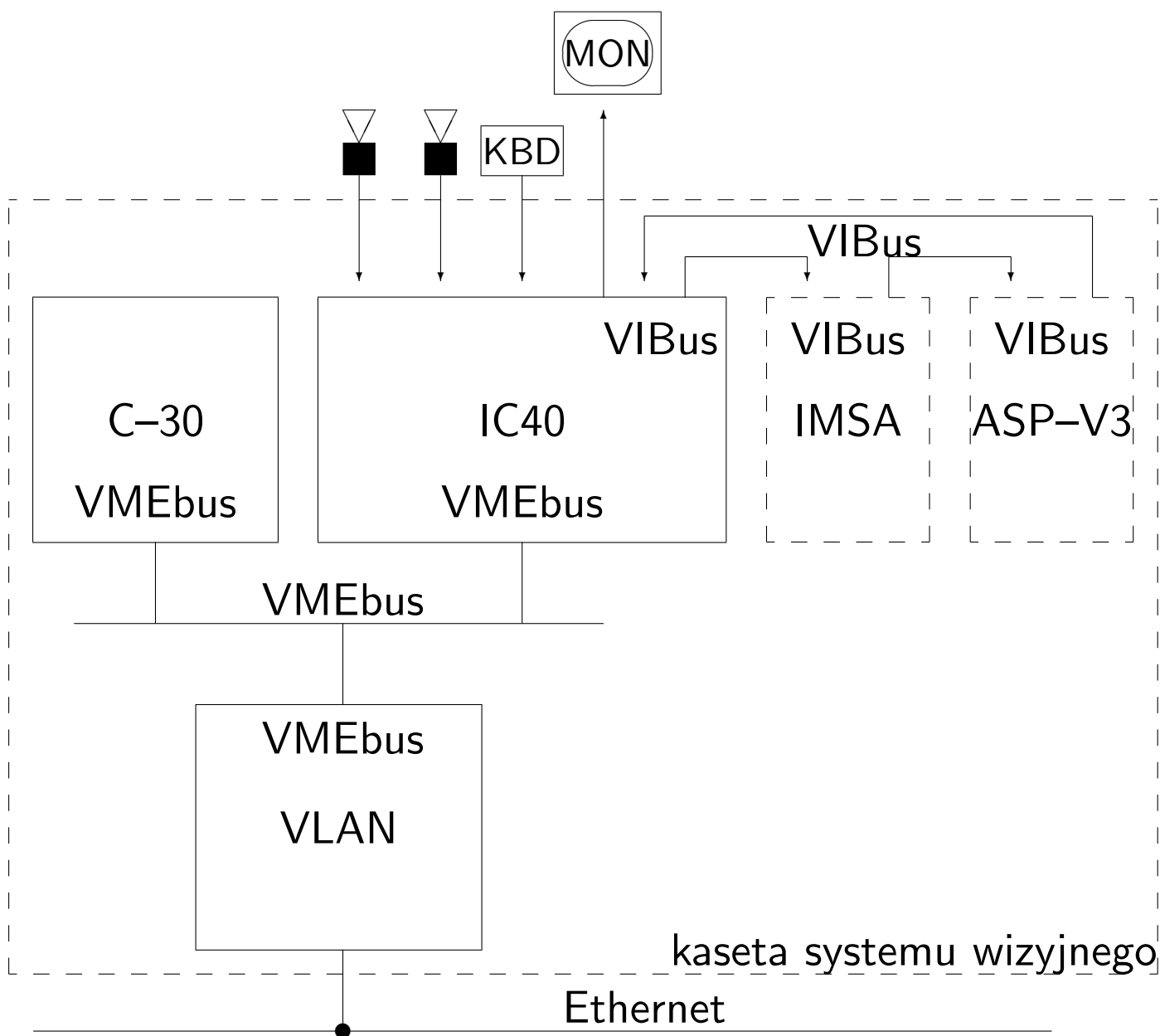
Kategorie pakietów VIBus ze względu na ich miejsce w ciągu przetwarzania:

- początkowe (*frontend*) – układy akwizycji - mają tylko złącze *master*
- przelotowe (*pipeline*) – układy filtracji - mają złącza typu *master* i *slave*
- końcowe (*backend*) – układy wizualizacji - mają tylko złącze *slave*

Topologie połączeń VIBus:

- kaskadowa (*pipelined*) - pakiety przelotowe są włączone kaskadowo pomiędzy pakiet początkowy a pakiet końcowy; każdy odcinek magistrali łączy dokładnie dwa pakiety
- równoległa (*broadcast*) - pakiet typu początkowego rozsyła dane wizyjne do wielu pakietów innych typów; magistrala ma więcej niż dwa złącza

System wizyjny na magistrali VMEbus



Magistralowy system rozproszony

