

# Przetwarzanie obrazów 5

- Handout

SZ 2024/2025

Wydział Matematyki, Fizyki i Informatyki



Dr.-Ing. A. Rowińska-Schwarzsteller  
[agnieszka.rowinska-schwarzsteller@ug.edu.pl](mailto:agnieszka.rowinska-schwarzsteller@ug.edu.pl)

26. November 2024

# Spis treści

<b>9 Filtrowanie obrazów: filtry nieliniowe</b>	<b>2</b>
9.1 Filtry rankingowe ( <i>rank value filter</i> ) . . . . .	2
9.2 Inne filtry nieliniowe . . . . .	7
<b>10 Korelacja w przetwarzaniu obrazów</b>	<b>10</b>
10.1 Związek między korelacją a splotem . . . . .	10
10.2 Zastosowania korelacji . . . . .	11
<b>11 Transformacje geometryczne obrazu</b>	<b>17</b>
11.1 Przekształcenia afiniczne . . . . .	19
11.1.1 Przesunięcie . . . . .	19
11.1.2 Rotacja . . . . .	20
11.1.3 Skalowanie . . . . .	20
11.1.4 Odbicie . . . . .	21
11.2 Reprezentacja macierzy transformacji afinicznych: współrzędne jednorodne . . . . .	23
11.3 Rodzaje przekształceń analitycznych . . . . .	27
11.4 Interpolacja w geometrycznych przekształceniach obrazu . . . . .	28
11.5 Nieliniowe przekształcenia geometryczne . . . . .	30
11.5.1 Wyprowadzenie odcinkowo-liniowej transformacji warping . . . . .	31
11.5.2 Algorytm wypaczania ( <i>Wrapping</i> ) . . . . .	32

# 9

## Filtrowanie obrazów: filtry nieliniowe

Filtry liniowe mają poważną wadę, gdy używane są do wygładzania i usuwania zakłóceń: struktury obrazu, takie jak punkty, krawędzie i linie, ulegają rozmyciu, co pogarsza ogólną jakość obrazu. Nie da się tego uniknąć w przypadku filtrów liniowych i dlatego ich możliwe zastosowania do tych celów są ograniczone. W filtrach nieliniowych operacje łączenia sąsiednich pikseli to porównywanie i selekcja.

### 9.1 Filtry rankingowe (*rank value filter*)

Wszystkie wartości szarości pikseli znajdujących się pod maską filtra są sortowane w kolejności rosnącej. Ten proces sortowania jest charakterystyczny dla wszystkich filtrów rankingowych. Wynik podstawowych filtrów - minimalnego, maksymalnego i medianowego - różni się jedynie pozycją na liście, z której wybierana jest wartość szarości i zapisywana z powrotem do centralnego piksela. → grafika 9.1

Przykład zastosowania filtrów nieliniowych → grafika

#### Filtr minimalny

filtr kompresujący, filtr erozyjny → grafika 9.3

$$g'(m, n) = \min \{g(m + i, n + j) | (i, j) \in R\}$$

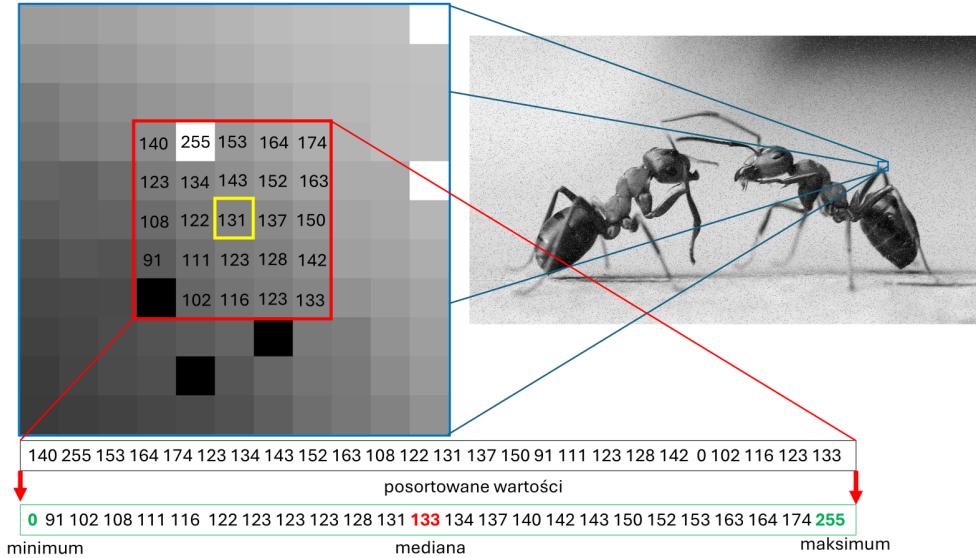
gdzie  $R$  oznacza obszar filtra w bieżącej pozycji  $(m, n)$ .

#### Filtr maksymalny

filtr dekompresujący, filtr ekspansywny → grafika 9.4

$$g'(m, n) = \max \{g(m + i, n + j) | (i, j) \in R\}$$

gdzie  $R$  oznacza obszar filtra w bieżącej pozycji  $(m, n)$ .



Grafika 9.1: Wybór wartości piksela w filtrach nieliniowych.

### Filtr medianowy

$$g'(m, n) = \text{median}(R)$$

gdzie  $R$  oznacza obszar filtra w bieżącej pozycji  $(m, n)$ , a mediana  $2K + 1$  wartości pikseli  $p_i$  jest zdefiniowana jako:  $\text{mediana}(p_1, p_2, \dots, p_{K+1}, \dots, p_{2K+1}) = p_{K+1}$ , czyli wartość środkowa, jeśli ciąg  $(p_1, \dots, p_{2K+1})$  posortowany jest według wielkości jego elementów ( $p_i \leq p_{i+1}$ ).

Przykład zastosowania filtra uśredniającego i medianowego → grafika 9.5. Media na z reguły eliminuje zakłócenia lepiej niż filtr uśredniający.

### Filtr średniodzakresowy (*Mid-Range-Filter*, *Midpoint-Filter*)

→ grafika 9.6, 9.9

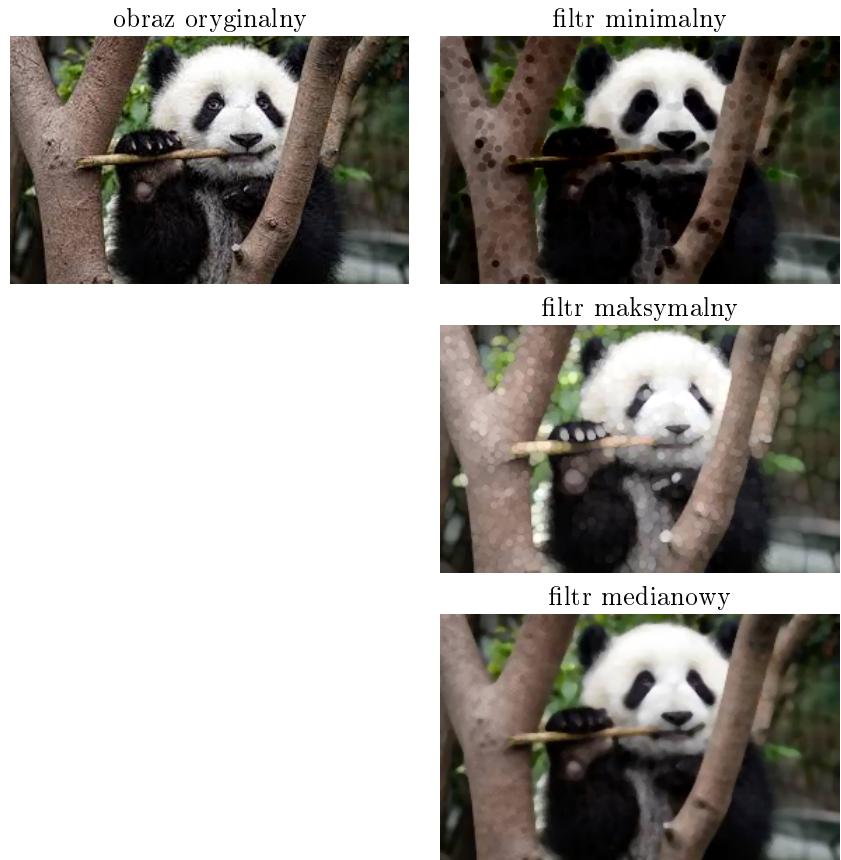
$$g'(m, n) = \frac{1}{2} \{ \min g(m+i, n+j) + \max g(m+i, n+j) | (i, j) \in R \}$$

### Filtr średniej uciętej (*k-trimmed-mean Filter*, *Alpha-trimmed-mean Filter*)

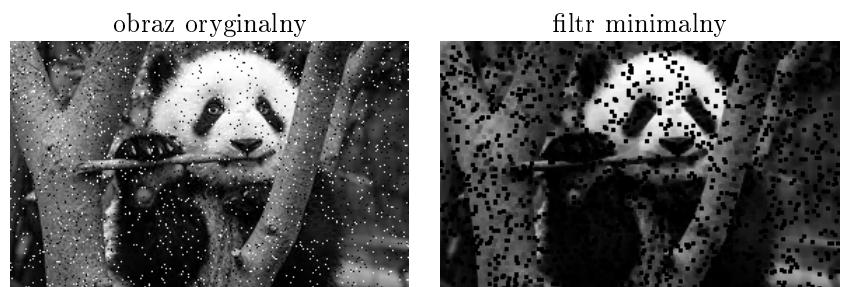
→ grafiki 9.9, 9.7

Dla ciągu  $(p_0, \dots, p_N)$  posortowanego według wielkości jego elementów ( $p_i \leq p_{i+1}$ ) reprezentującego wartości szarości w sąsiedztwie piksela  $g(m, n)$  w obszarze filtra  $R$  w bieżącej pozycji:

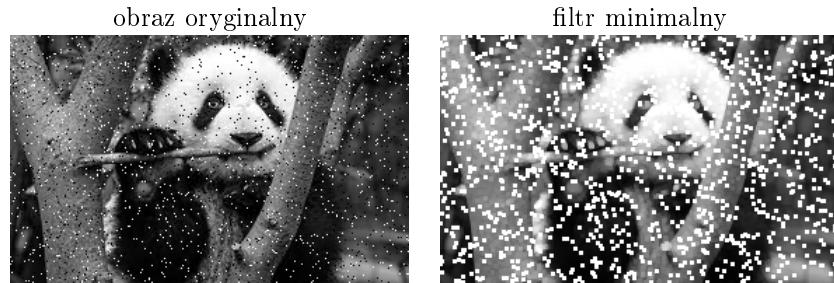
$$g'(m, n) = \frac{1}{N - 2k} \sum_{i=1+k}^{N-k} p_i$$



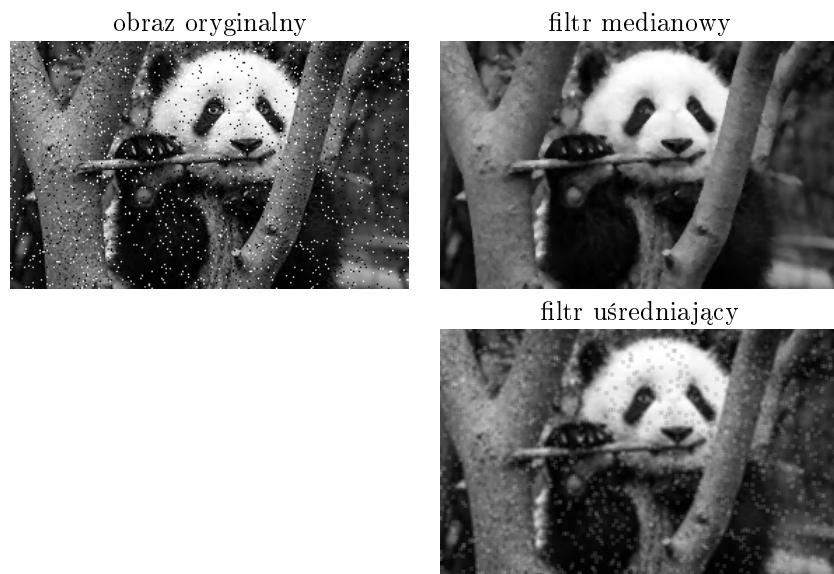
Grafika 9.2: Filtry nieliniowe



Grafika 9.3: Filtr minimalny zastosowany do zaszumionego obrazu



Grafika 9.4: Filtr maksymalny zastosowany do zaszumionego obrazu



Grafika 9.5: Porównanie filtra medianowego i filtra uśredniającego zastosowanych do zaszumionego obrazu



Grafika 9.6: Filtr średnizakresowy zastosowany do zaszumionego obrazu



Grafika 9.7: Filtr k-trimmed-mean. Źródło: <http://www.librow.com/articles/article-7>

### Filtры rankingowe dla obrazów *RGB*

Porównywanie wartości pikseli nie jest możliwe bezpośrednio dla obrazów *RGB*: nie ma sensownego liniowego porządku  $\leq$  dla wektorów RGB.

Filtrowanie nie można przeprowadzać niezależnie na poszczególnych kanałach kolorów, ponieważ może to skutkować uzyskaniem kolorów, które nie mają nic wspólnego z wartościami początkowymi.

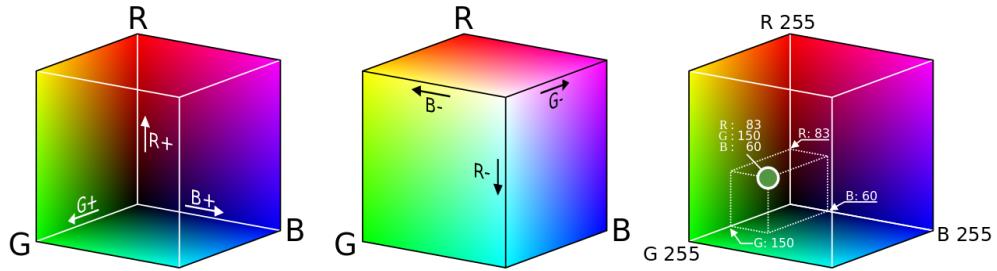
Jednym ze sposobów przeniesienia koncepcji filtrów nielinowych na obrazy kolorowe jest zdefiniowanie odległości euklidesowej dla wartości kolorów pikseli. → grafika 9.8

Niech  $c_0, \dots, c_8$  oznacza wartości pikseli w sąsiedztwie  $3 \times 3$  (sąsiedztwo ośmiosąsiedzkie).

$$c_i = (r_i, g_i, b_i), i \in \{0, \dots, 8\}.$$

**Odległość euklidesowa między wartościami  $(R, G, B)$  pikseli:**

$$\|c_i - c_j\| = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$$



Grafika 9.8: Odległości euklidesowe w przestrzeni *RGB*. Źródło: Wikimedia Commons

1. filtr minimalny

$k \in \{0, \dots, 8\}$  zostaje wybrany tak, że

$$\|c_k\| = \min_{0 \leq i \leq 8} \{\|c_i\|\}$$

tzn. dla  $c_k$  odległość do koloru czarnego  $c = (0, 0, 0)$  jest minimalna.

2. filtr maksymalny  
 $k \in \{0, \dots, 8\}$  zostaje wybrany tak, że

$$\|c_k\| = \max_{0 \leq i \leq 8} \{\|c_i\|\}$$

tzn. dla  $c_k$  odległość do koloru czarnego  $c = (0, 0, 0)$  jest maksymalna.

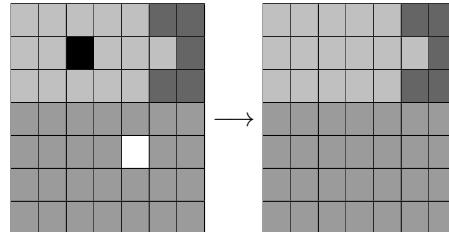
3. median  
 $k \in \{0, \dots, 8\}$  zostaje wybrany tak, że

$$\sum_{j=0}^8 \|c_k - c_j\| = \min_{0 \leq i \leq 8} \left\{ \sum_{j=0}^8 \|c_i - c_j\| \right\}$$

tzn. dla  $c_k$  suma odległości kolorów do wszystkich innych punktów sąsiedztwa jest minimalna.

## 9.2 Inne filtry nieliniowe

### Eliminacja punktów izolowanych



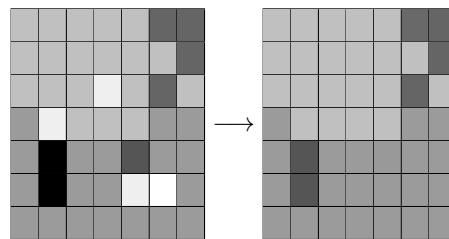
Izolowany punkt obrazu (odstający punkt) to piksel o losowej wartości, która nie ma nic wspólnego z rzeczywistym sygnałem obrazu (wadliwy punkt sensora optycznego).

$$g'(m, n) = \begin{cases} g(m, n) & : |g(m, n) - \mu| < \Theta \\ \mu & : |g(m, n) - \mu| \geq \Theta \end{cases}$$

gdzie  $\mu$  oznacza średnią wartość szarości pikseli w sąsiedztwie bez piksela centralnego.

### k-Nearest Neighbor

→ grafika 9.9



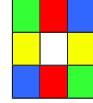
Niech wartości  $g_1, g_2, \dots, g_k$  oznaczają  $k$  wartości szarości pod maską filtra  $R$ , które są najbliższe jasności piksela  $g(m, n)$ :

$$g'(m, n) = \frac{1}{k} \sum_{i=1}^k g_i$$

W obrazach 8-bitowych, dla maski filtra  $3 \times 3$  (sąsiedztwo ośmiospójne) Davis i Rosenfeld wykazali, że optymalną wartością  $k$  jest  $k = 6$ .

### Symmetric Nearest Neighbor

→ grafika 9.9



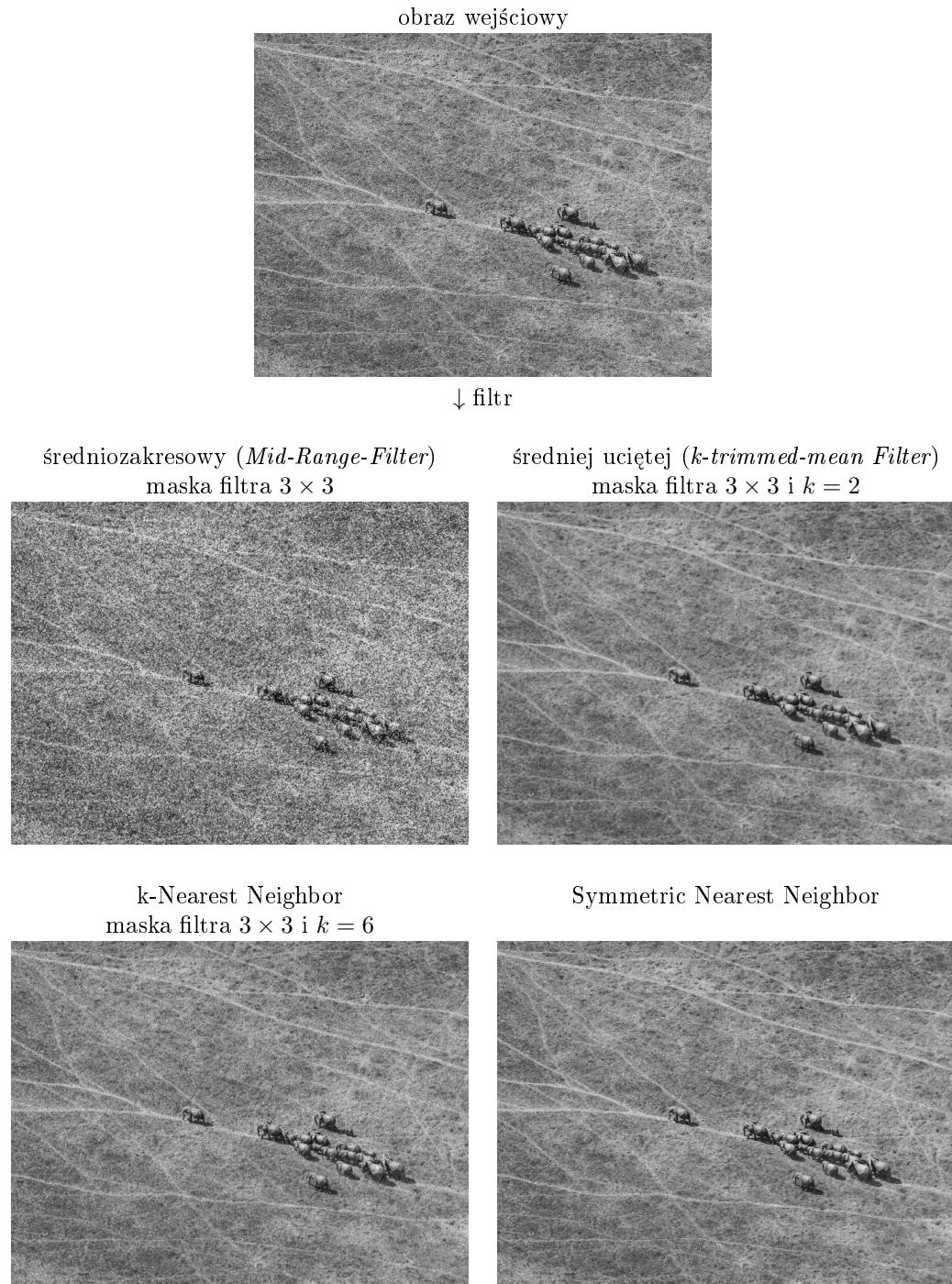
W każdej z czterech par centralnie symetrycznych pikseli w sąsiedztwie środkowego piksela wybierany jest piksel, który jest najbardziej podobny do piksela środkowego. Środkowy piksel zostaje zastąpiony średnią z czterech wybranych sąsiadów.

Niech wartości  $g_1, g_2, g_3, g_4$  oznaczają 4 wartości szarości wybranych sąsiadów  $g(m, n)$ :

$$g'(m, n) = \frac{1}{4} \sum_{i=1}^4 g_i$$

### Przykład: porównanie filtrów nieliniowych

→ grafika 9.9

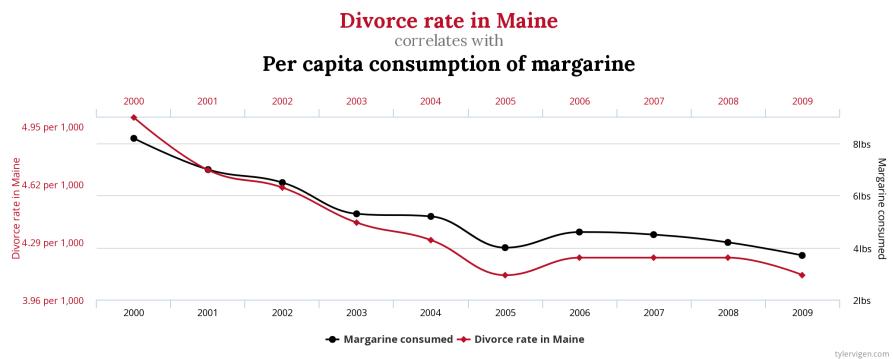


Grafika 9.9: Porównanie filtrów nieliniowych dla zaszumionego obrazu

# 10

## Korelacja w przetwarzaniu obrazów

Korelacja potocznie → grafika 10.1



Grafika 10.1: Potoczne pojęcie korelacji. Źródło:  
<https://www.tylervigen.com/spurious-correlations>

Korelacja (od łac. *współzależność*) w przetwarzaniu sygnału lub przetwarzaniu obrazu opisuje związek (podobieństwo) między dwiema lub większą liczbą funkcji czasowych lub przestrzennych.

W przypadku dyskretnych, dwuwymiarowych funkcji  $g_1$  i  $g_2$  operację korelacji  $g'(m, n)$  definiuje się jako:

$$g'(m, n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g_1(i, j) \cdot g_2(i + m, j + n)$$

### 10.1 Związek między korelacją a splotem

W przypadku dyskretnych, dwuwymiarowych funkcji  $g_1$  i  $g_2$  zachodzi zależność:

$$\begin{aligned}
 g'(m, n) = g_1(m, n) * g_2(-m, -n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g_1(m-i, n-j) \cdot g_2(-i, -j) \\
 &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g_1(i, j) \cdot g_2(i+m, j+n) \\
 &= g_1(-m, -n) * g_2(m, n)
 \end{aligned}$$

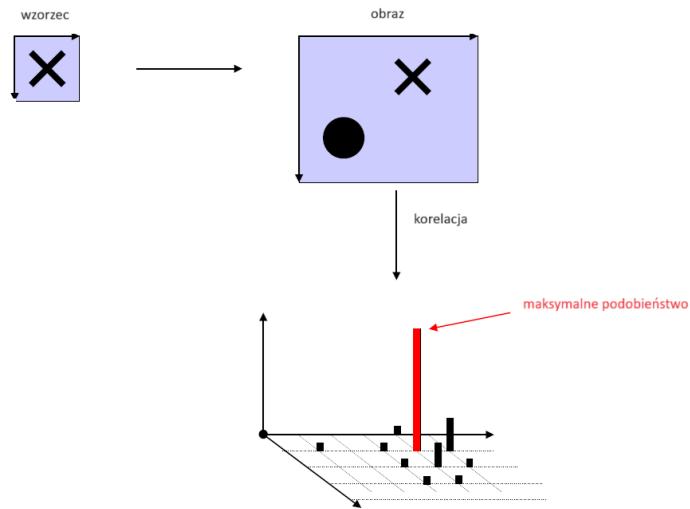
tz. korelacja obrazów  $g_1(m, n)$  i  $g_2(m, n)$  jest splotem obrazu  $g_1(m, n)$  z obrazem  $g_2(-m, -n)$ .

## 10.2 Zastosowania korelacji

Korelacja daje ilościowe oszacowanie podobieństwa dwóch funkcji i przesunięcia czasowego/przestrzennego między nimi.

### Znajdywanie wzorca na obrazie

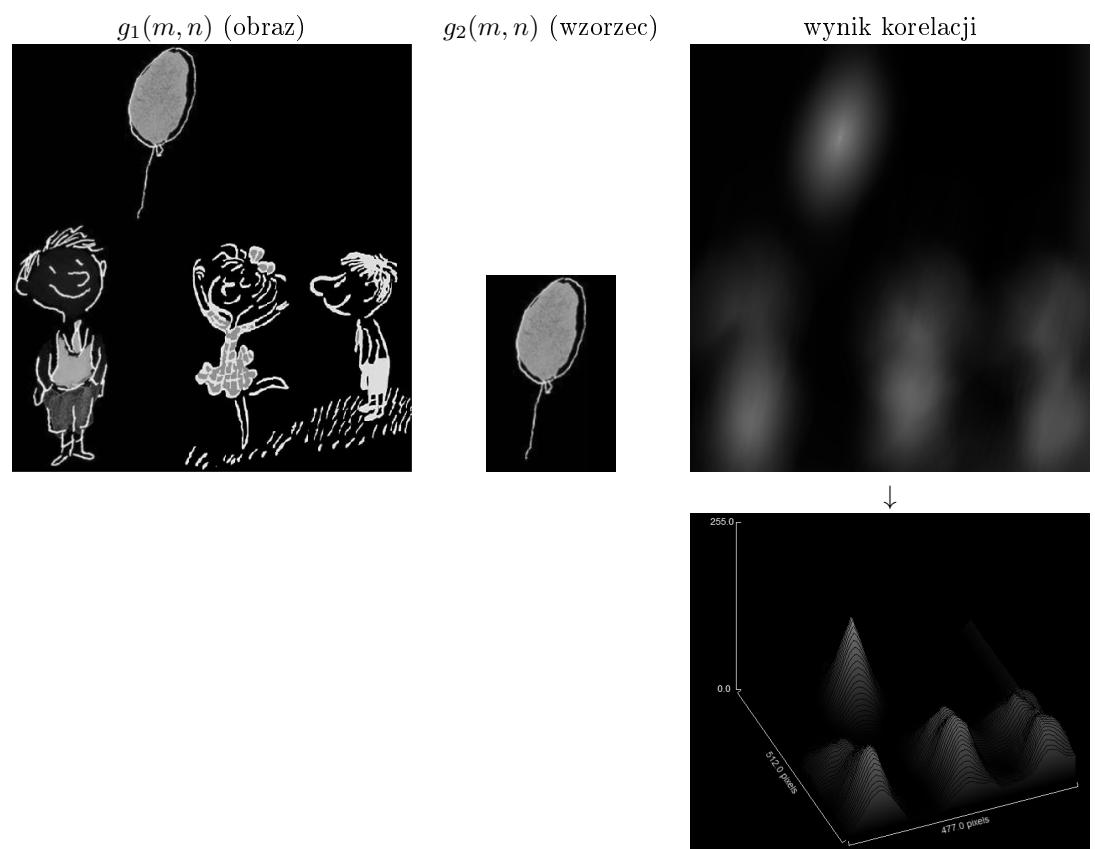
W przetwarzaniu obrazu funkcje korelowalne wykorzystuje się m.in. do dokładnego umiejscowienia wzorca (funkcji wzorcowej w sensie korelacji matematycznej) na obrazie.  
→ grafika 10.2



Grafika 10.2: Znajdywanie wzorca w obrazie

Przykład 1: wyszukiwanie wzorca w obrazie → grafika 10.3

Przykład 2: odnajdywanie wzorca w wielu obrazach (korelacja zdjęcia Einsteina z innymi obrazami, w tym z zaszumionym obrazem Einsteina i jego negatywem) →



Grafika 10.3: Korelacja obrazów w celu znalezienia wzorca

grafika 10.4

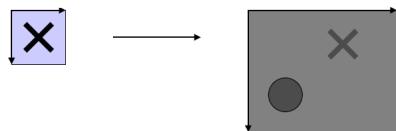
Ogólne wnioski dotyczące znajdywania wzorców poprzez korelację:

- Zalety:

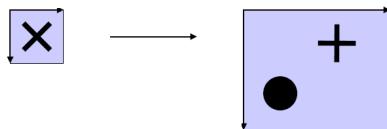
- łatwa implementacja
- wynik rozkładu podobieństwa jest gęsty

- Wady:

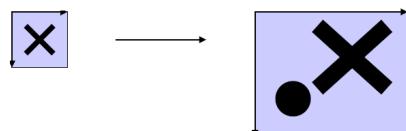
- algorytm jest powolny,
- podatny na różnice w oświetleniu,



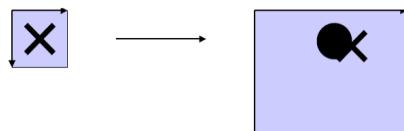
- rotacja nie jest niezmiennikiem,



- skalowanie nie jest niezmiennikiem,

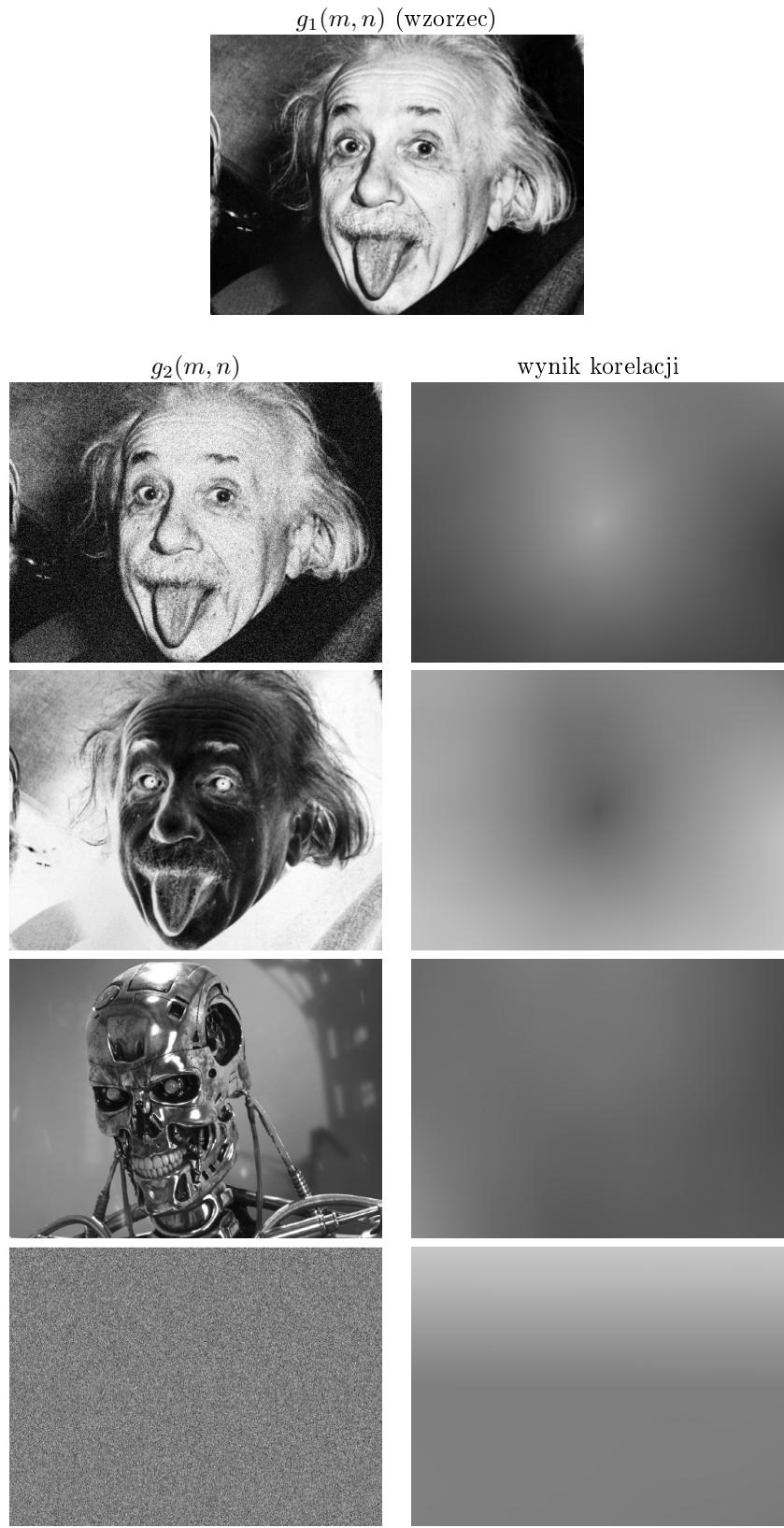


- nie znajduje okluzji.



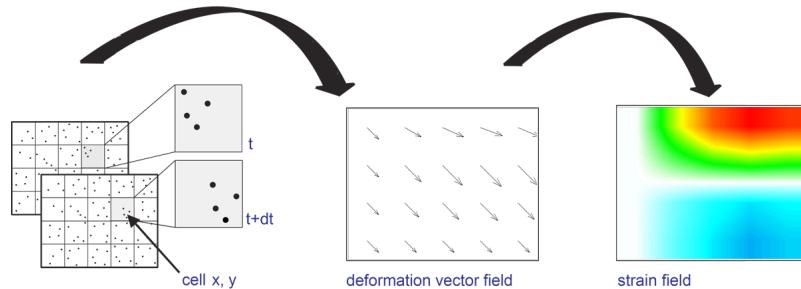
### Digital Image Correlation (DIC)

Śledzenie zmian w serii obrazów: analiza przemieszczenia wzorów w dyskretnych podzbiorach lub elementach fasetowych całego obrazu. Maksymalna korelacja w każdym



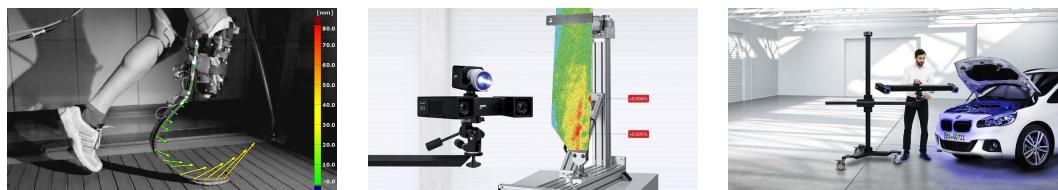
Grafika 10.4: Korelacja obrazów w celu znalezienia wzorca

okno odpowiada przemieszczeniu, co daje długość wektora i kierunek dla każdego okna.  
→ grafika 10.5



Grafika 10.5: Digital Image Correlation (DIC). Źródło: <https://www.lavision.de/en/>

DIC jest metodą bezkontaktowego pomiaru odkształceń. Stosowany jest głównie do testowania materiałów i komponentów mechanicznych. Zastosowanie w przemyśle → grafika 10.6

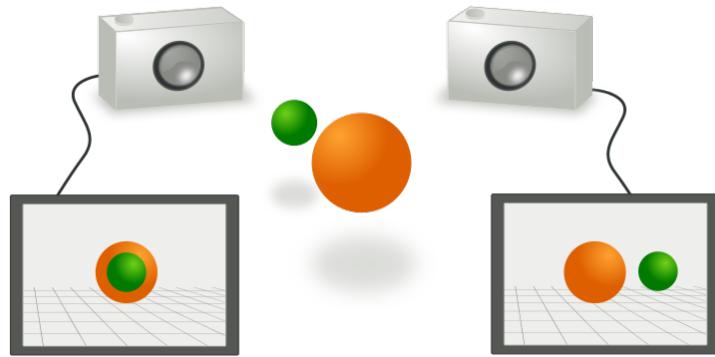


Grafika 10.6: Wykorzystanie korelacji w technice pomiarowej. Źródło: GOM Metrology.

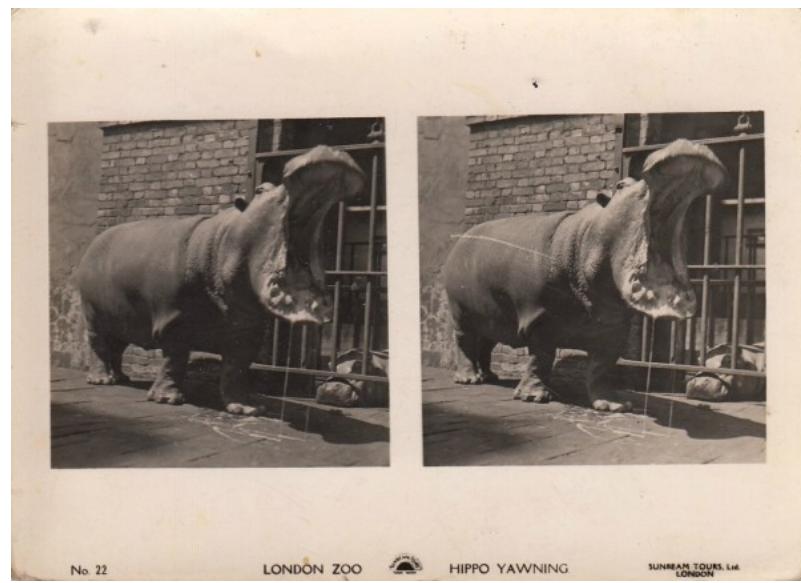
### Ocena par obrazów stereo / fotogrametria (odtwarzanie kształtów, rozmiarów i wzajemnego położenia obiektów)

Aby określić informacje 3D na podstawie obrazów z kamer, muszą być dostępne co najmniej dwa obrazy obiektu lub sceny. Podczas nagrywania te dwa obrazy mają względem siebie określoną pozycję i orientację. → grafika 10.7

Aby móc obliczyć współrzędną przestrzenną punktu, musi istnieć jednoznaczne przyporządkowanie obiektów na parze obrazów stereo. Aby je znaleźć wybiera się z reguły mały fragment jednego obrazu (wzór) i koreluje go dwuwymiarowo z drugim obrazem. Uzyskane współrzędne punktu obiektu można przekształcić na współrzędne przestrzenne za pomocą metod fotogrametrycznych. → grafika 10.8



Grafika 10.7: Stereogrametria. Źródło: wikipedia

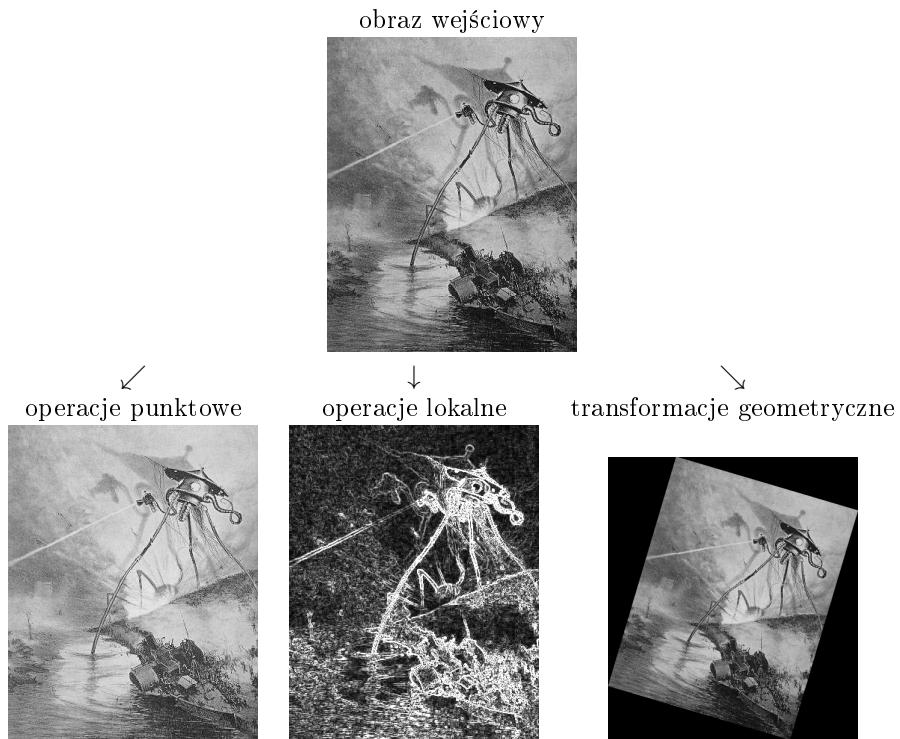


Grafika 10.8: Para obrazów stereo. Źródło: <https://www.kotzendes-einhorn.de/blog/2012-09/stereoskopie-vintage-3d/>

# 11

## Transformacje geometryczne obrazu

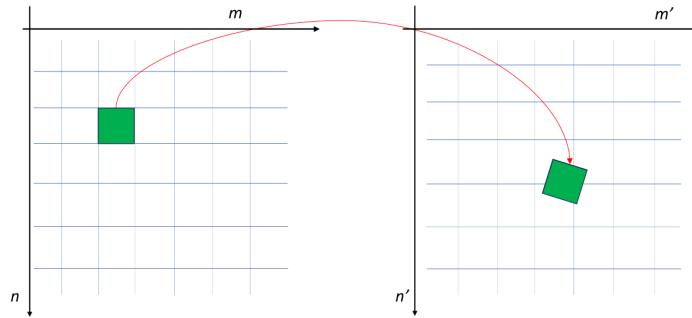
Transformacje geometryczne obrazu to algorytmy należące do grupy operacji wstępnego przetwarzania obrazu. → grafika 11.1.



Grafika 11.1: Operacje wstępne przetwarzania obrazu: operacje punktowe (np. oparte na histogramie), operacje lokalne (filtrowanie obrazów) i transformacje geometryczne

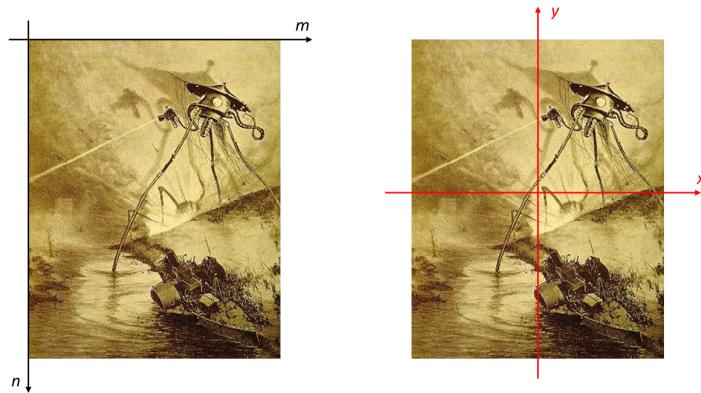
Transformacje geometryczne są powszechnymi operacjami w grafice komputerowej.

- Podczas analizy obrazu transformacje geometryczne służą do korygowania zniekształceń (np. spowodowanych przez obiektyw aparatu) lub są powszechnie stosowane jako operacja normalizacyjna przed rozpoznaniem wzoru lub pomiarem obiektu.
- Gdy obraz jest przekształcany geometrycznie, każda wartość piksela obrazu źródłowego  $g(m, n)$  jest odwzorowywana na nową pozycję  $(m', n')$  w obrazie docelowym. Warunki sąsiedzkie pozostają zachowane!
- Wyjściowa pozycja przekształconego piksela zazwyczaj nie pokrywa się dokładnie z pozycją siatki na obrazie wyjściowym (grafika 11.2):
  - Współrzędne celu  $(m', n')$  są liczbami zmiennoprzecinkowymi.
  - Aby móc wyświetlić przekształcony piksel na obrazie docelowym, należy przeprowadzić interpolację.



Grafika 11.2: Transformacja geometryczna piksela źródłowego do obrazu wyjściowego

- Geometryczne przekształcenia obrazu zmieniają nie tylko położenie piksela, ale często także obszar obrazu (liczbę punktów siatki), który piksel powinien wypełnić w obrazie docelowym.
  - Liniowe transformacje geometryczne (afiniczne) mają stały (niezależny od położenia) współczynnik powiększenia obszaru. Przykłady: translacja, obrót, skalowanie.
  - Nieliniowe transformacje geometryczne (*warping transformations*) mają zmienny (zależny od położenia) współczynnik powiększenia obszaru. Przykłady: obrazowanie perspektywiczne, zniekształcanie soczewki
- Częsta konwencja w programach przetwarzających obraz: transformacja układu współrzędnych (grafika 11.3)

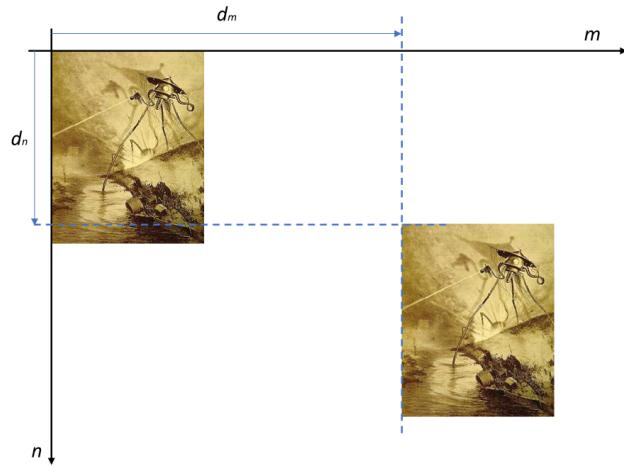


Grafika 11.3: Transformacja układu współrzędnych w programach przetwarzania obrazu

## 11.1 Przekształcenia afoniczne

### 11.1.1 Przesunięcie

→ grafika 11.4



Grafika 11.4: Przesunięcie

Położenie  $(m', n')$  piksela po przesunięciu o  $d_m$  w dodatnim kierunku  $m$  i  $d_n$  w dodatnim kierunku  $n$ :

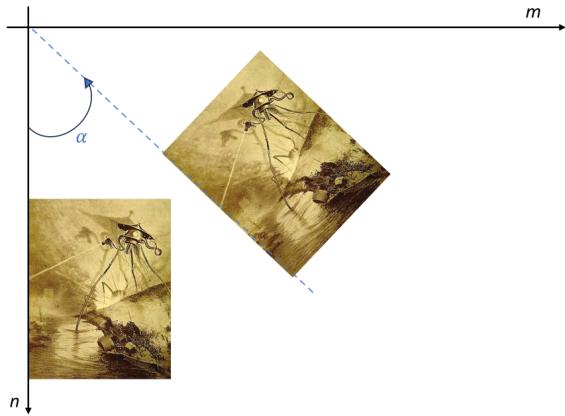
$$\begin{aligned} m' &= m + d_m \\ n' &= n + d_n \end{aligned}$$

Równanie transformacji dla przesunięcia w notacji wektorowej:

$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_m \\ 0 & 1 & d_n \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

### 11.1.2 Rotacja

→ grafika 11.5



Grafika 11.5: Rotacja obrazu

Położenie  $(m', n')$  piksela po obrocie wokół początku układu współrzędnych  $(0, 0)$  o kąt  $\alpha$ :

$$\begin{aligned} m' &= m \cos \alpha + n \sin \alpha \\ n' &= -m \sin \alpha + n \cos \alpha \end{aligned}$$

Równanie transformacji dla rotacji w notacji wektorowej:

$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

### 11.1.3 Skalowanie

→ grafika 11.6

Położenie  $(m', n')$  piksela po przeskalowaniu względem początku układu współrzędnych  $(0, 0)$ , przy współczynniku skalowania w poziomie  $S_m$  i współczynniku skalowania w pionie  $s_n$ :

$$\begin{aligned} m' &= mS_m \\ n' &\equiv nS_n \end{aligned}$$



Grafika 11.6: Skalowanie obrazu

Równanie transformacji dla skalowania w notacji wektorowej:

$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} S_m & 0 & 0 \\ 0 & S_n & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

#### 11.1.4 Odbicie

##### Odbicie poziome

→ grafika 11.7

Położenie  $(m', n')$  piksela po poziomym odbiciu względem prostej  $m_0$ :

$$\begin{aligned} m' &= 2m_0 - m \\ n' &= n \end{aligned}$$

Równanie transformacji dla odbicia poziomego w notacji wektorowej:

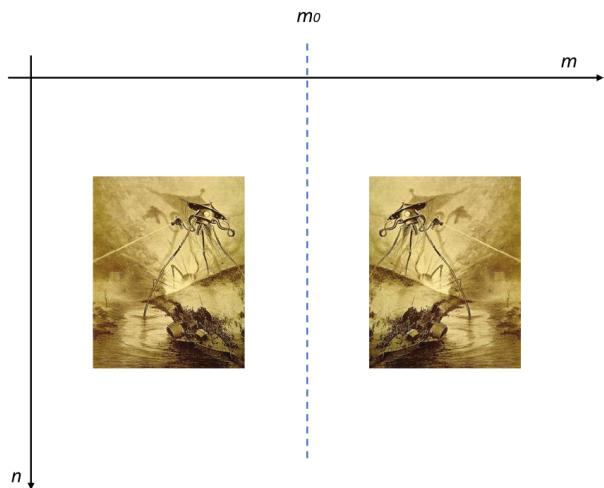
$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 2m_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

##### Odbicie pionowe

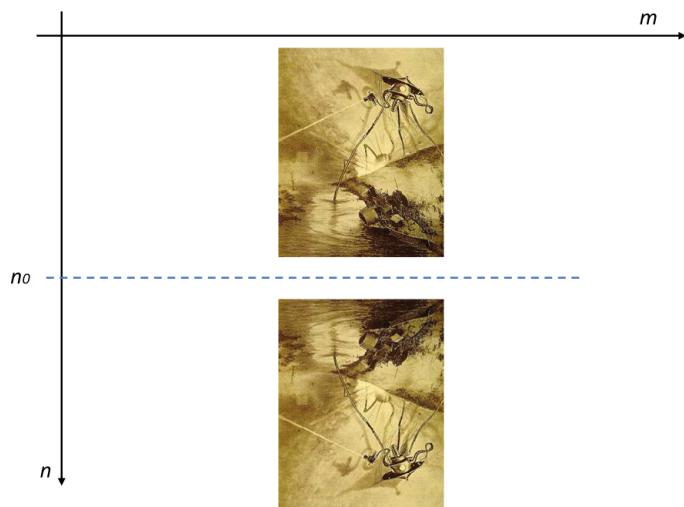
→ grafika 11.8

Położenie  $(m', n')$  piksela po pionowym odbiciu względem prostej  $n_0$ :

$$\begin{aligned} m' &= m \\ n' &= 2n_0 - n \end{aligned}$$



Grafika 11.7: Poziome odbicie obrazu



Grafika 11.8: Pionowe odbicie obrazu

Równanie transformacji dla odbicia poziomego w notacji wektorowej:

$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 2n_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

## 11.2 Reprezentacja macierzy transformacji afnicznych: współrzędne jednorodne

- Dwuwymiarowe przekształcenia liniowe można przedstawić za pomocą macierzy  $3 \times 3$ .
- Transformację punktu  $(m, n)$  w nowy punkt  $(m', n')$  poprzez dowolną sekwencję translacji, obrotów i skalowania opisuje wzór:

$$\begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix} = \begin{pmatrix} \textcolor{green}{a} & \textcolor{green}{b} & \textcolor{blue}{c} \\ \textcolor{green}{d} & \textcolor{green}{e} & \textcolor{blue}{f} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m \\ n \\ 1 \end{pmatrix}$$

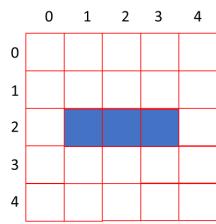
rotacja, skalowanie  
przesunięcie

- Dwie lub więcej transformacji afnicznych, które przeprowadzane są jedna po drugiej, można połączyć w jedną transformację (konkatynację) i opisać przy użyciu jednej macierzy transformacji.

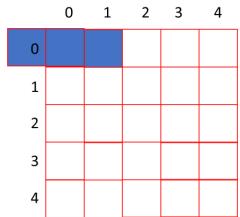
**Przykład: rotacja i skalowanie linii**

1. przesunięcie

$$d_m = -2, d_n = -2$$

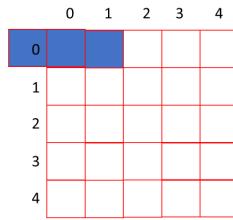


$$\begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

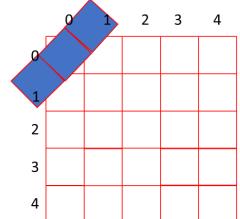


2. rotacja

$$\alpha = 45^\circ$$

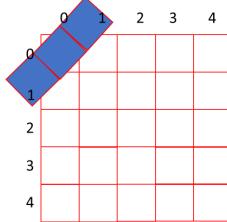


$$\begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

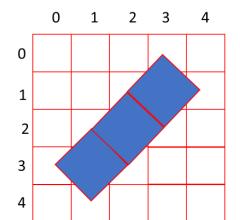


3. skalowanie

$$S_m = \sqrt{2}, S_n = \sqrt{2}$$

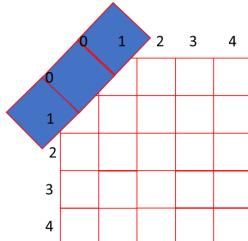


$$\begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



4. przesunięcie

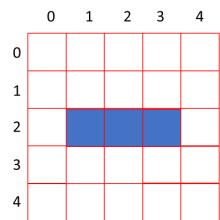
$$d_m = 2, d_n = 2$$



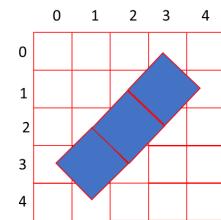
$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Złożenie transformacji (konkatenacja):

$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & -2 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

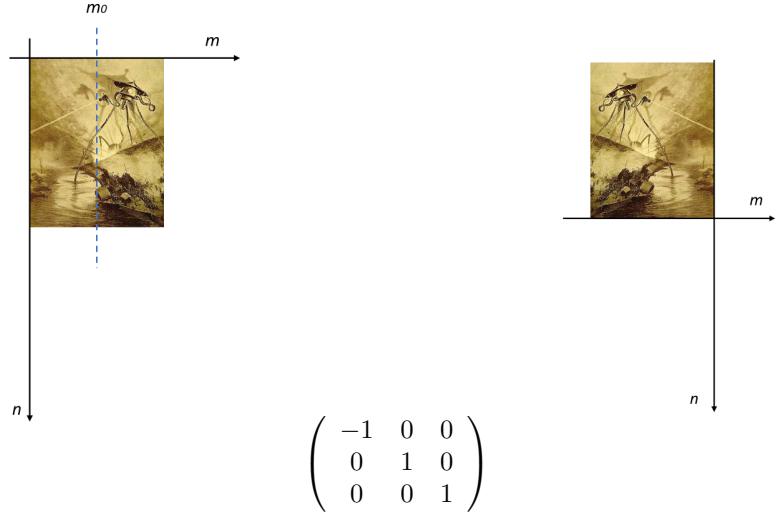


$$\begin{pmatrix} 1 & -1 & -2 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

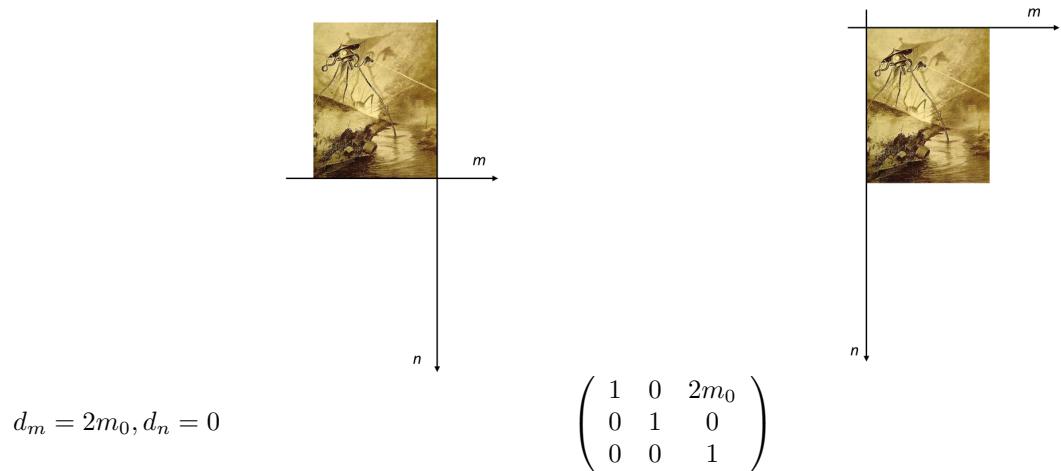


**Przykład: odbicie w poziomie wzdłuż linii środka obrazu**

1. skalowanie



2. przesunięcie



Złożenie transformacji (konkatenacja):

$$\begin{pmatrix} 1 & 0 & 2m_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 2m_0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### 11.3 Rodzaje przekształceń analitycznych

euklidesowe

$$\begin{aligned} m' &= am + bn + d_n, & a = s \cos \alpha \\ n' &= -bm + an + d_m, & b = s \sin \alpha \end{aligned}$$

affine, 1<sup>st</sup> order polynomial

$$\begin{aligned} m' &= a_0 + a_1m + a_2n \\ n' &= b_0 + b_1m + b_2n \end{aligned}$$

bilinear

$$\begin{aligned} m' &= a_0 + a_1mn + a_2m + a_3n \\ n' &= b_0 + b_1mn + b_2m + b_3n \end{aligned}$$

perspective

$$\begin{aligned} m' &= (a_0 + a_1m + a_2n)/(c_0m + c_1n + 1) \\ n' &= (b_0 + b_1m + b_2n)/(c_0m + c_1n + 1) \end{aligned}$$

2<sup>nd</sup> order polynomial, biquadratic

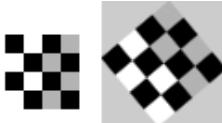
$$\begin{aligned} m' &= a_0 + a_1m + a_2n + a_3m^2 + a_4n^2 + a_5mn \\ n' &= b_0 + b_1m + b_2n + b_3m^2 + b_4n^2 + b_5mn \end{aligned}$$

general polynomial

$$\begin{aligned} m' &= \sum_i \sum_j a_{ij} m^i n^j \\ n' &= \sum_i \sum_j b_{ij} m^i n^j \end{aligned}$$

#### Właściwości przekształceń analitycznych

euklidesowe: linie pozostają liniami, kwadraty pozostają kwadratami



affine: linie pozostają liniami, równoległe pozostają równoległe, prostokąt → równoległobok



bilinear: linie stają się krzywymi

perspective: linie pozostają liniami, równoległe → linie ze wspólnym punktem zbiegu



biquadratic: linie stają się krzywymi

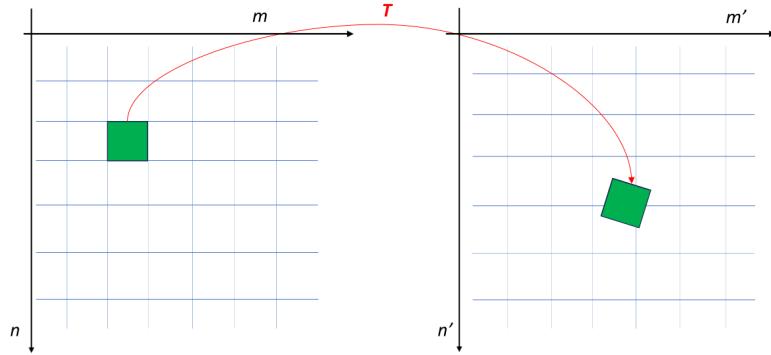


**Liczba punktów kontrolnych  $\geq$  liczba stopni swobody/2**

- euklidesowe: przesunięcie + rotacja + skalowanie globalne min .2
- affine: przesunięcie + rotacja + skalowanie  $S_m + S_n$  min .3
- bilinear: min .4
- perspective: min .4
- biquadratic: min .6

## 11.4 Interpolacja w geometrycznych przekształceniach obrazu

- Podczas geometrycznego przekształcania obrazów zwykle implementowane jest mapowanie od celu do źródła (*target-to-source, backwards*).
- W obrazie wyjściowym każdemu pikselowi przypisana jest wartość obliczona po przez interpolację sąsiedztwa pikseli w obrazie wejściowym (*resampling*).

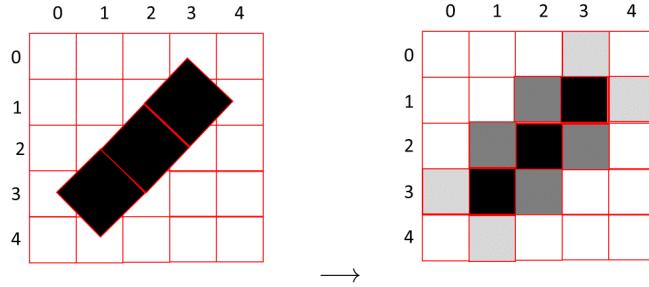


$$\begin{pmatrix} m \\ n \\ 1 \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix}$$

$\mathbf{T}^{-1}$  jest macierzą odwrotną transformacji wejścia do wyjścia  $\mathbf{T}$

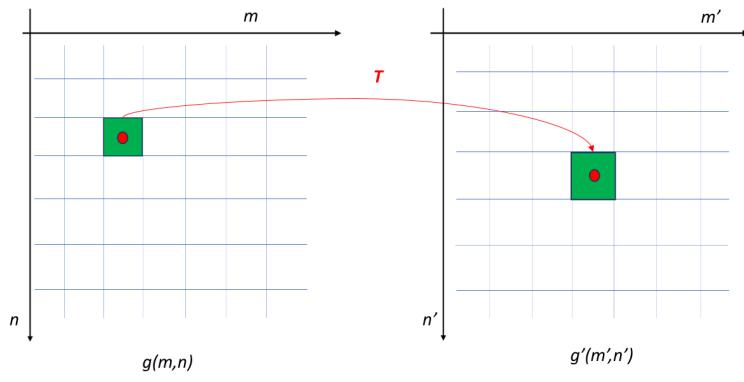
- Przekształcony piksel pokrywa (całkowicie lub częściowo) jedną lub więcej komórek siatki na obrazie wyjściowym.
- Prosta metoda interpolacji wartości szarości na obrazie wyjściowym opiera się na obliczeniu wartości szarości dla wszystkich pokrytych pikseli obrazu wyjściowego, przy czym są one ważone proporcjonalnie do ich pokrytych udziałów w

powierzchni.

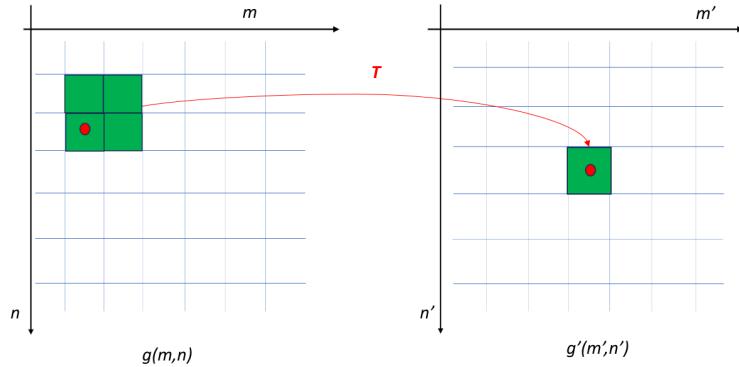


- W praktyce najczęściej stosowanymi metodami interpolacji w transformacji obrazu są (rozdział "Odtwarzanie obrazów"):
  - interpolacja nearestneighbor,
  - interpolacja dwuliniowa (bilinear),
  - interpolacja dwusześcienna (bicubic).

### Interpolacja nearestneighbor



### Interpolacja dwuliniowa



$$\begin{aligned}\tilde{m} &= \lfloor m \rfloor \\ \tilde{n} &= \lfloor n \rfloor\end{aligned}$$

$$\begin{aligned}a &= m - \tilde{m} \\ b &= n - \tilde{n}\end{aligned}$$

$$\begin{aligned}g'(m', n') &= (1-a)(1-b)g(\tilde{m}, \tilde{n}) + a(1-b)g(\tilde{m}+1, \tilde{n}) + \\ &\quad +(1-a)bg(\tilde{m}, \tilde{n}+1) + abg(\tilde{m}+1, \tilde{n}+1)\end{aligned}$$

### 11.5 Nieliniowe przekształcenia geometryczne

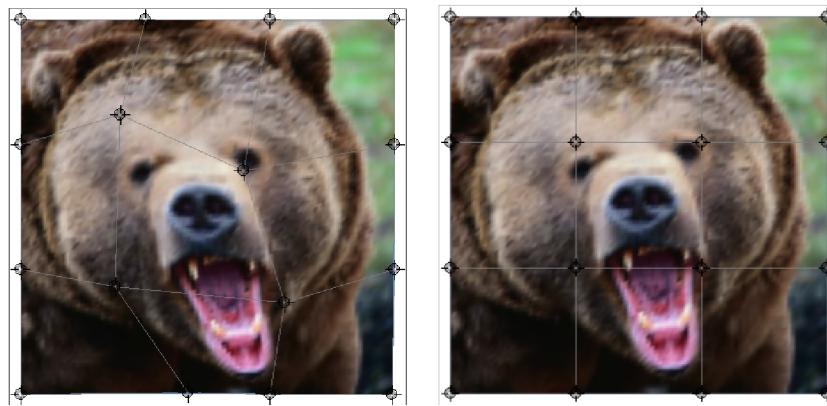
→ grafika 11.9



Grafika 11.9: Nieliniowe przekształcenie obrazu. Źródło: materiały firmy Panasonic.

W przypadku transformacji nieliniowych lokalne zniekształcenie obrazu jest zmienne (współczynnik powiększenia obszaru w zależności od lokalizacji). Przykłady: obrazy perspektywiczne, sferyczne i cylindryczne.

- W przypadku wypaczenia (*Warping*) odpowiednie punkty na obrazie docelowym są wyznaczane dla zbioru wybranych punktów kontrolnych (punktów orientacyjnych, punktów odniesienia) na obrazie wejściowym. → grafika 11.10

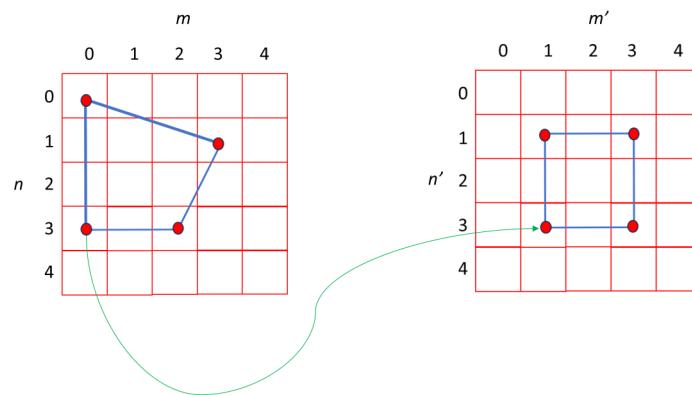


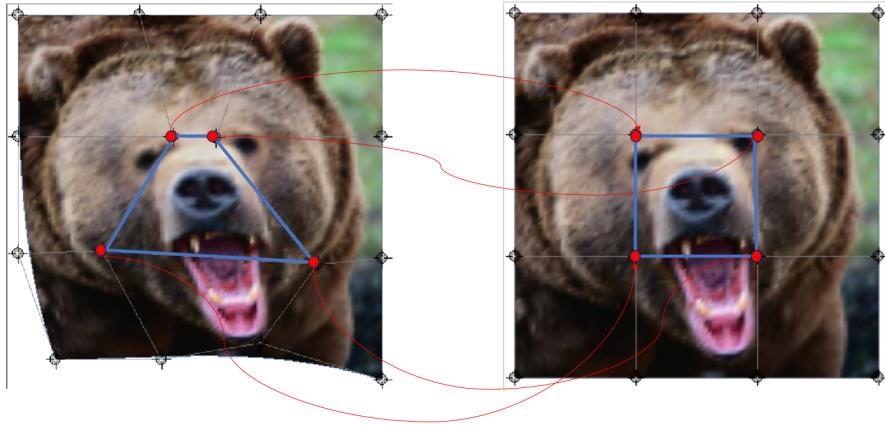
Grafika 11.10: Warping.

Grafiki wykonane w programie <https://demonstrations.wolfram.com/ImageWarping/>

- Aby skorygować geometryczne zniekształcenia obrazu, można utworzyć siatkę punktów kontrolnych, które należy nałożyć na regularną (niezniekształconą) siatkę.
- Cała transformacja obrazu jest podzielona na regiony, przy czym każdy region reprezentuje kwadrat w siatce. → grafika 11.11

### 11.5.1 Wyrowadzenie odcinkowo-liniowej transformacji warping





Grafika 11.11: Siatki warpingu.

Grafiki wykonane w programie <https://demonstrations.wolfram.com/ImageWarping/>

Dowolny czworokąt można odwzorować na inny czworokąt (np. kwadrat) za pomocą transformacji dwuliniowej (bilinearnej):

$$\begin{aligned} m' &= a_1m + a_2n + a_3mn + a_4 \\ n' &= b_1m + b_2n + b_3mn + b_4 \end{aligned}$$

Przy odpowiednio wybranych 4 punktach przyporządkowania uzyskuje się układ ośmiu równań dla 8 niewiadomych  $a_1, a_2, \dots, b_4$ .

#### **Przykład: korekta zniekształconego wzoru szachownicy**

→ grafika 11.12

### **11.5.2 Algorytm wypaczania (Wrapping)**

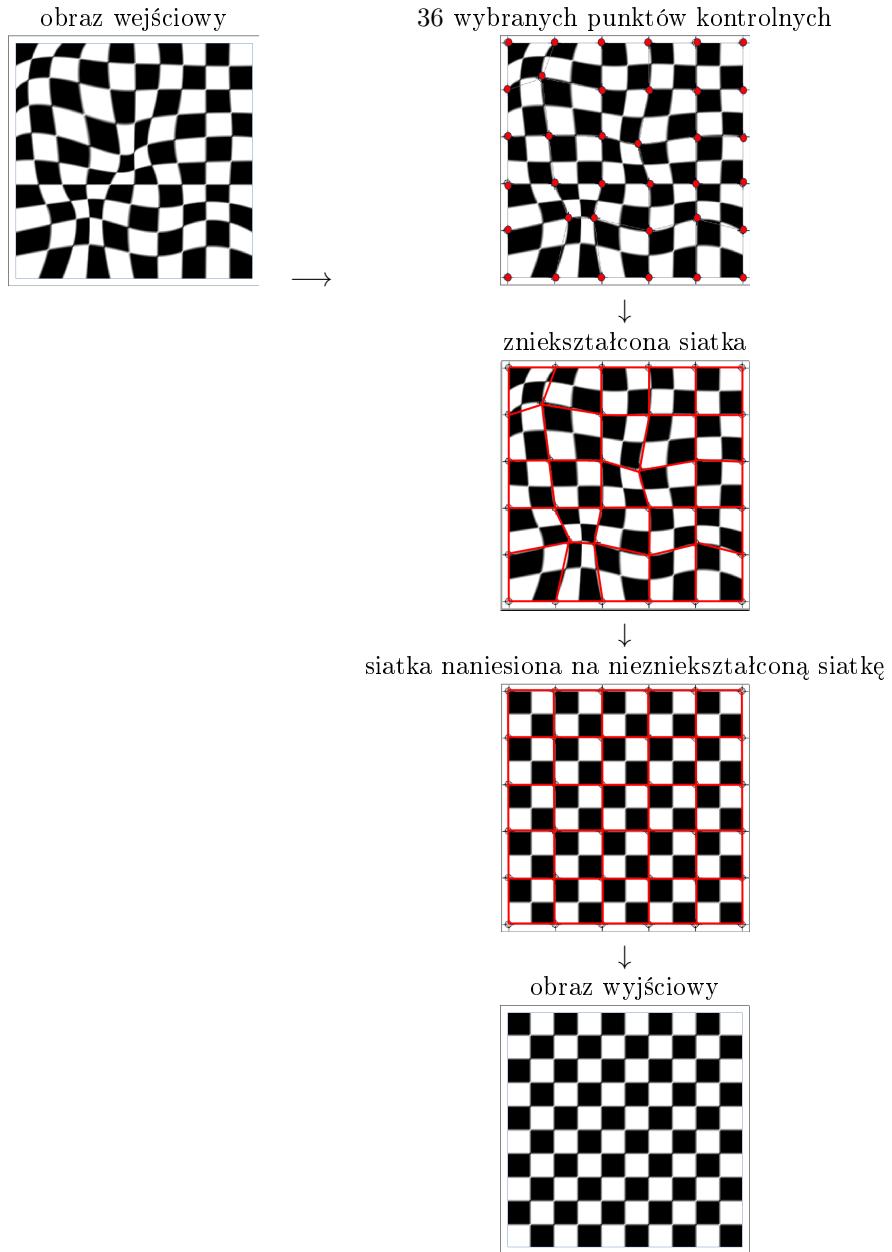
#### **1. Wyznaczenie punktów dopasowania**

- automatycznie lub odręcznie

#### **2. Znalezienie najlepszej transformacji dla punktów dopasowania**

- rozwiązywanie analityczne (np. przedstawiciele klasy przekształceń afinicznych) lub
- transformacja odcinkowo-liniowa dla każdej lokalizacji piksela  $(m', n')$  na obrazie wyjściowym (*Gridding*)

#### **3. Wyznaczenie wartości szarości pikseli w obrazie wyjściowym**



Grafika 11.12: Korekta zniekształconego obrazu szachownicy.

- poprzez interpolację w lokalizacji źródłowej tzn. na obrazie wejściowym (nearest neighbor, bilinear, bicubic)

## 1. Wyznaczenie punktów dopasowania

- odręcznie
  - OK. do edycji obrazu, ale nie do (zautomatyzowanego) przetwarzania obrazów
  - niedokładne
    - opcjonalne: znalezienie dopasowania poprzez korelację
  - zastosowanie: kalibracja
- automatyczne
  - możliwe, jeśli na 2 obrazach znajdują się korespondujące ze sobą punkty
  - *Matching*
- 2 podproblemy:
  - znalezienie odpowiednich punktów na obrazie ("salient points")
  - dokładne odnalezienie ich na drugim obrazie (*Template Matching*)

## 2. Znalezienie najlepszej transformacji dla punktów dopasowania

Metody:

1. obliczenie transformacji  $\mathbf{T}$  z punktów dopasowania:
  - A. jeśli  $\mathbf{T}$  powinno należeć do określonej klasy (np. Euklidesa): znalezienie najlepszej transformacji w sensie metody najmniejszych kwadratów (*Least-Square, LS*)
  - B. swobodny warp: odcinkowo-liniowa interpolacja pomiędzy punktami dopasowania
2. zastosowanie  $\mathbf{T}^{-1}$  do każdego (jednorodnego) piksela obrazu wyjściowego

### Przykład: najlepsza transformacja $\mathbf{T}$ z Least-Square-Fit

Niech danych będzie więcej punktów dopasowania niż potrzeba do określenia  $\mathbf{T}$ .  
Np. złożenie rotacji i jednokładności:

$$\begin{pmatrix} m' \\ n' \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \begin{pmatrix} m \\ n \end{pmatrix} \Leftrightarrow \begin{pmatrix} m' \\ n' \end{pmatrix} = \begin{pmatrix} m & n \\ n & -m \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

- Dla wielu punktów dopasowania  $(m_1, n_1), \dots, (m_k, n_k)$ :

$$\underbrace{\begin{pmatrix} m'_1 \\ \vdots \\ m'_k \\ n'_1 \\ \vdots \\ n'_k \end{pmatrix}}_{\mathbf{x}'} = \underbrace{\begin{pmatrix} m_1 & n_1 \\ \vdots & \vdots \\ m_k & n_k \\ n_1 & -m_1 \\ \vdots & \vdots \\ n_k & -m_k \end{pmatrix}}_{\mathbf{Z}} \underbrace{\begin{pmatrix} a \\ b \end{pmatrix}}_{\mathbf{t}} \Leftrightarrow \mathbf{X}' = \mathbf{Z}\mathbf{t}$$

→ (z reguły) nadokreślony liniowy układ równań dla wektora parametrów  $\mathbf{t} = (a, b)^T$   
 rozwiązywanie metodą najmniejszych kwadratów (*Least-Square*) np. za pomocą polecenia JAMA<sup>1</sup>

```
Matrix t = Z.solve(Xp);
```

- Wyznaczenie macierzy transformacji geometrycznej  $\mathbf{T}$  z wektora  $\mathbf{t}$ :

$$\mathbf{t} = \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow \mathbf{T} = \begin{pmatrix} a & b & 0 \\ -b & a & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Obliczenie transformacji  $\mathbf{T}^{-1}$   
 → polecenia JAMA<sup>2</sup> dla każdego piksela docelowego  $(m', n') = (i, j)$ :

```
Matrix T_inverse = T.inverse();
double [][] valXP = {{i}, {j}, {1.0}};
Matrix XP = new Matrix(valXP);
Matrix X = T_inverse.times(XP);
double x = X.get(0,0);
double y = X.get(1,0);
```

### 3. Wyznaczenie wartości szarości pikseli w obrazie wyjściowym

Backwards-Warp:

piksel  $(m', n')$  obrazu wyjściowego jest kolorowany zgodnie z pozycją  $(m, n)$  na obrazie wejściowym → grafika 11.13

$$\begin{pmatrix} m \\ n \\ 1 \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} m' \\ n' \\ 1 \end{pmatrix}$$

Wartość szarości piksela wejściowego interpolowana jest w obszarze obrazu wejściowego (np. poprzez interpolację dwuliniową).

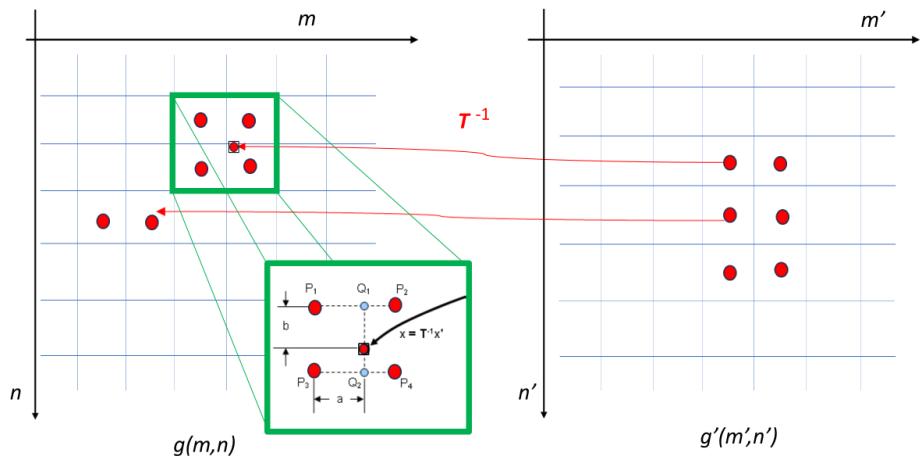
### Zastosowania algorytmu wypaczania (Warp)

- kalibracja kamery
  - korekta dystorsji beczkowej i poduszkowej układu optycznego → grafika 11.14
- przemysłowe badania obiektów odkształcalnych
- fotografia lotnicza obejmująca krajobraz (nakładające się obszary)
  - ... znalezienie różnic (np. ekologia)

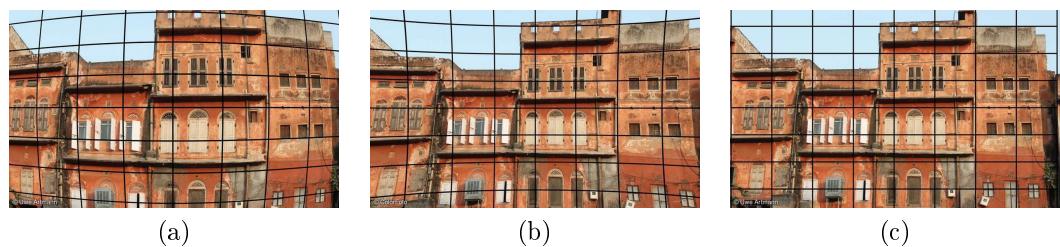
---

<sup>1</sup>JAMA : A Java Matrix Package <https://math.nist.gov/javanumerics/jama/>

<sup>2</sup>JAMA : A Java Matrix Package <https://math.nist.gov/javanumerics/jama/>



Grafika 11.13: Backwards-Warp



Grafika 11.14: Dystorsja beczkowa (a) i poduszkowa (b), obraz po korekcie (c). Źródło: <https://www.connect-living.de/>

- odwrotna perspektywa
- przetwarzanie obrazów medycznych
- ...