

Grafika i wizualizacja

Wprowadzenie

Edward Półrolniczak, dr inż., Katedra Systemów Multimedialnych
// na podstawie wykładów Mariusza Borawskiego, dr hab. inż..

Grafika i wizualizacja

Wprowadzenie

- Literatura
- Zastosowania grafiki komputerowej
- Podstawowe pojęcia
- Sprzęt
 - ✓ Komputer
 - ✓ Monitor
 - ✓ Urządzenie wskazujące
 - ✓ Cyfrowy aparat fotograficzny
 - ✓ Skaner
 - ✓ Urządzenia drukujące
- Oprogramowanie
- Geometria
- Rys historyczny

Grafika i wizualizacja

Literatura

- Foley J. D., Dam A., Hughes J., Phillips R., Wprowadzenie do grafiki komputerowej, Wydawnictwo Naukowo-Techniczne, Warszawa 1995
- Borawski M., Rachunek wektorowy w przetwarzaniu obrazów, Wydawnictwo Uczelniane Politechniki Szczecińskiej, Szczecin 2007
- Michał Jankowski, *Elementy grafiki komputerowej*, WNT, Warszawa, Polska, 2006, 2

Grafika i wizualizacja

- Zastosowania

Grafika i wizualizacja

Zastosowania

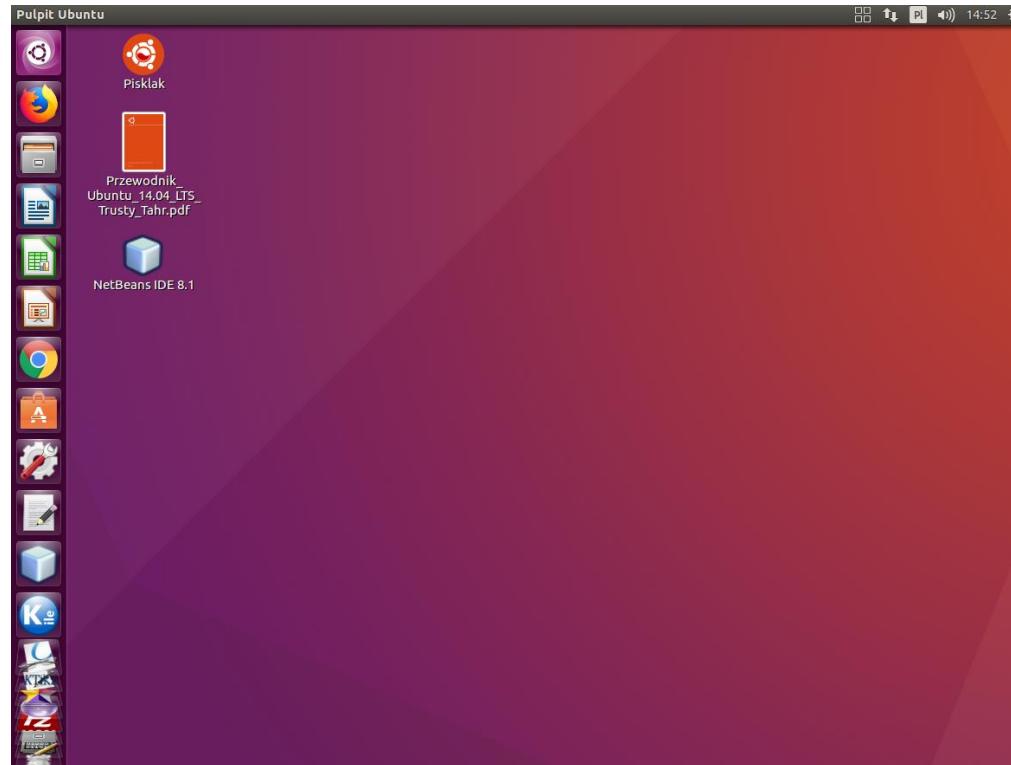
Zastosowania grafiki komputerowej:

- ✓ graficzne interfejsy użytkownika;
- ✓ wizualizacja informacji;
- ✓ wspomaganie prac inżynierskich CAD;
- ✓ wizualizacja wyników symulacji;
- ✓ rzeczywistość wirtualna i rozszerzona;
- ✓ systemy DTP (poligrafia);
- ✓ systemy informacji przestrzennej i geograficznej;
- ✓ edukacja i rozrywka;
- ✓ inne.

Grafika i wizualizacja

Graficzny interfejs użytkownika

- Graficzny interfejs użytkownika jest sposobem komunikacji człowieka z komputerem wykorzystującym elementy graficzne.



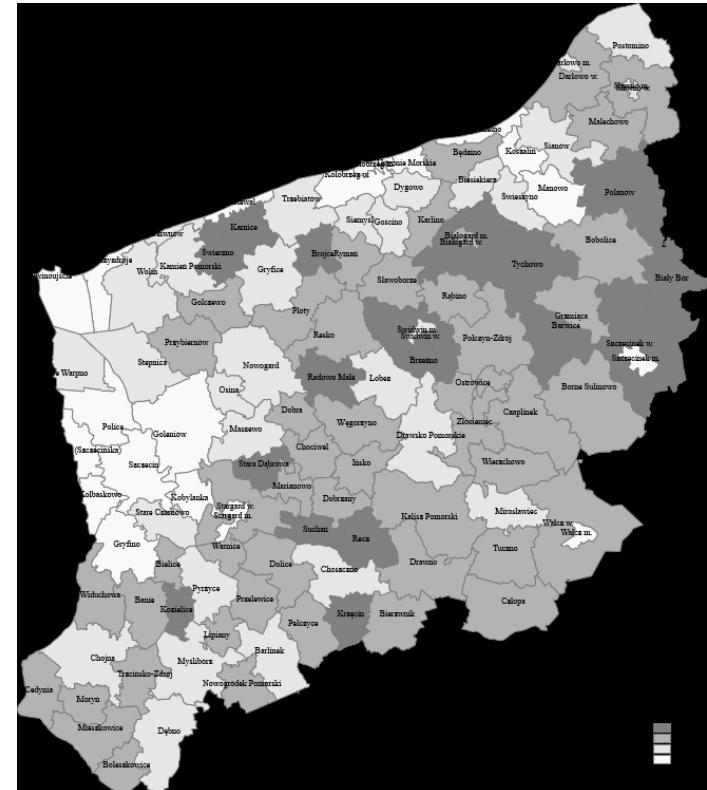
//XEROX a Jobs i
Gates
//START -
Windows 95

Grafika i wizualizacja

Wizualizacja danych

- Przykład klasyfikacji gmin pod względem stopnia ubóstwa w 2014 roku (klasa pierwsza – największy stopień ubóstwa).

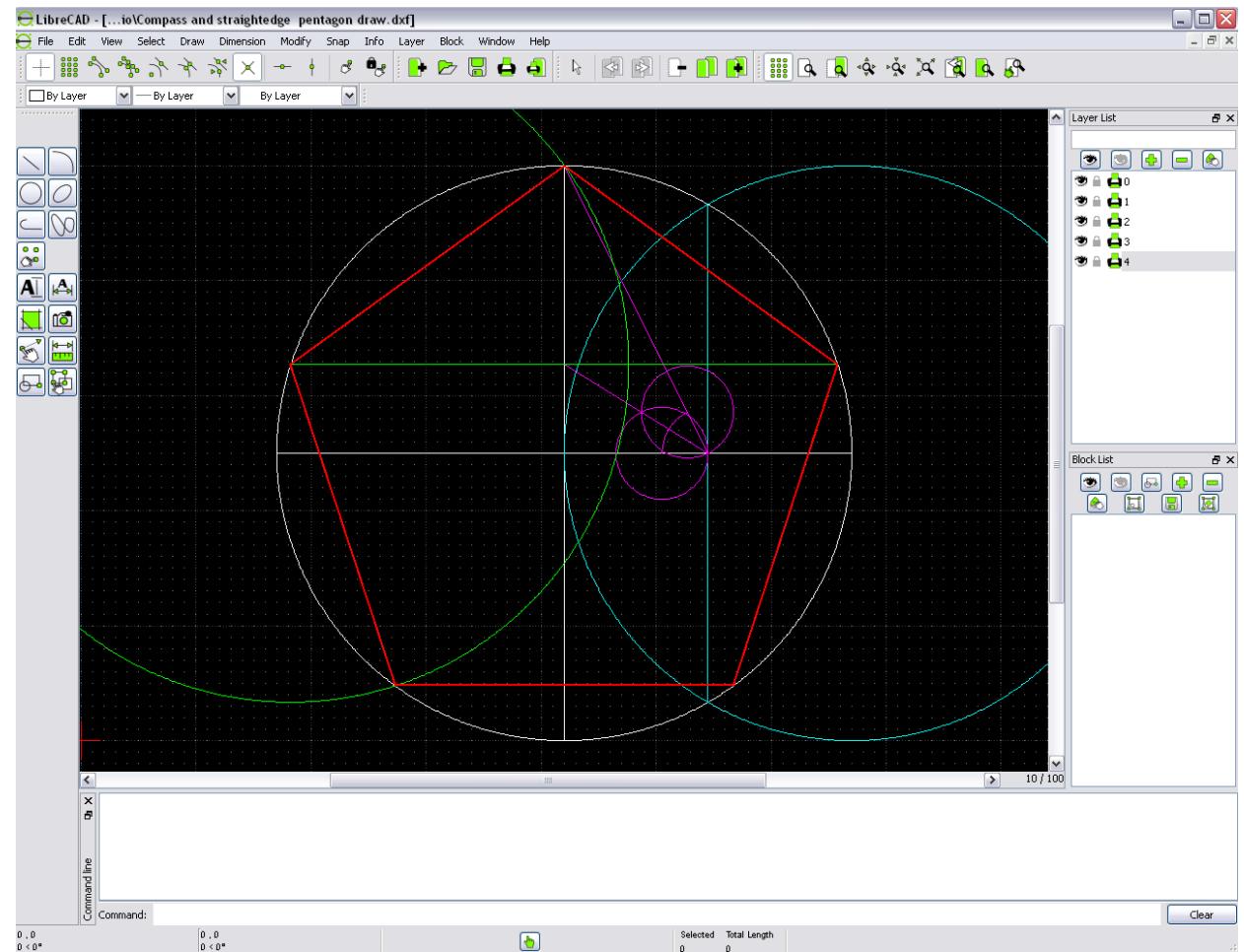
Gmina	Wartość miary	Klasa
Banie	0,450247	2
Barlinek	-0,214371	3
Barwice	0,735803	1
Będzino	0,350029	2
Białogard m.	0,23763	2
Białogard w.	0,880496	1
Biały Bór	0,906482	1
Bielice	0,181998	2
Bierzwnik	0,40511	2
Biesiekierz	-0,341896	3
Bobolice	0,345295	2
Boleszkowice	0,364966	2
Borne Sulinowo	0,210338	2
Brojce	1,18232	1
Brzeźno	1,04606	1
Cedynia	0,188426	2



Grafika i wizualizacja

Wprowadzenie

- Wspomaganie prac inżynierskich CAD.



Źródło: Raulshc – Pentagon made with compass and straightedge on LibreCAD 2.0.0beta.

Grafika i wizualizacja

Wizualizacja wyników symulacji

- Wizualizacja wyników symulacji.



Wizualizacja wyników symulacji. Źródło: AnyLogic –
[Transport Hub Simulation Model: Pedestrian and Traffic Simulation](#)

Grafika i wizualizacja

Rzeczywistość wirtualna i rozszerzona

- Rzeczywistość wirtualna i rozszerzona.

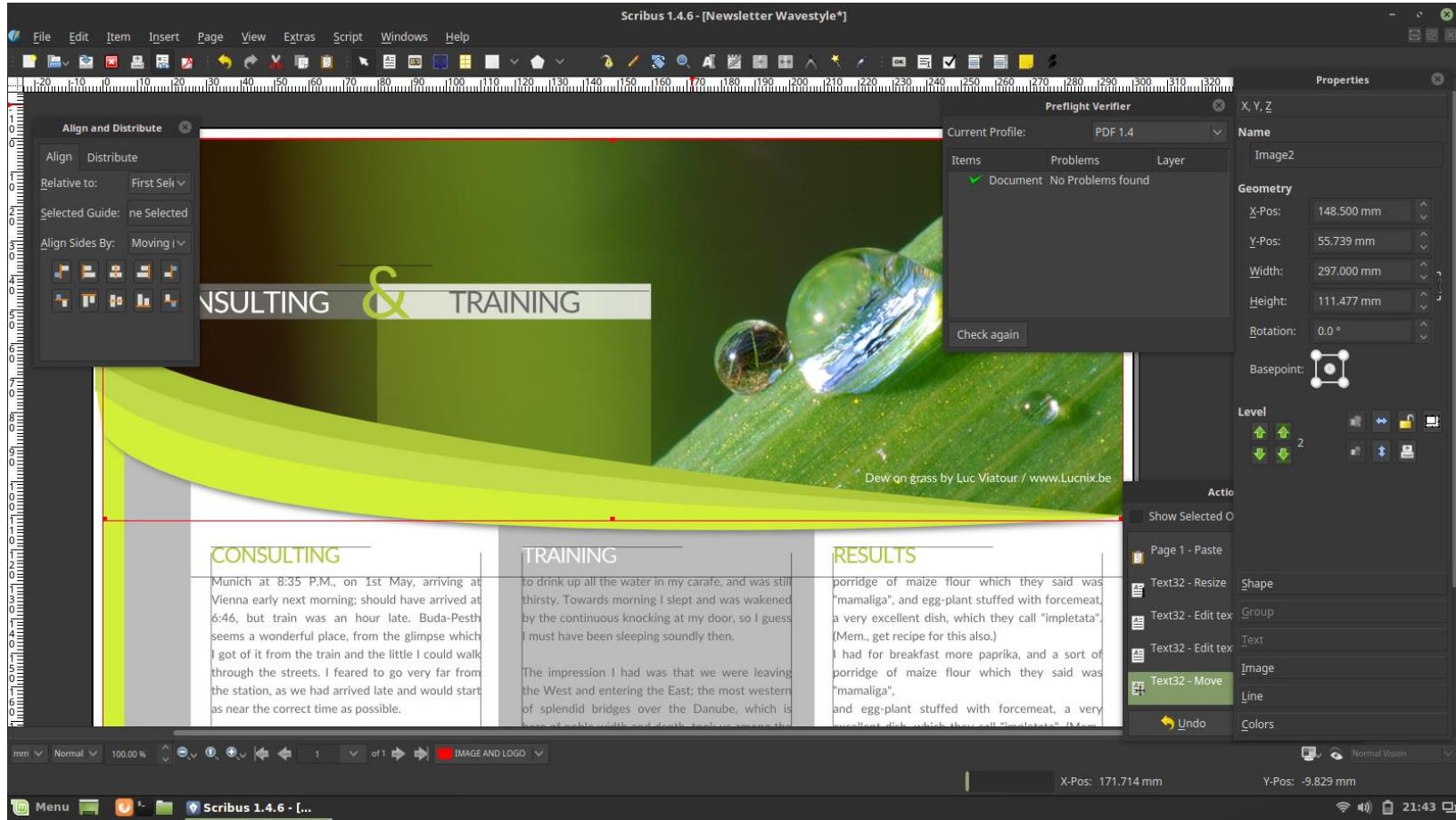


Źródło: Singularity Lab – [Virtual and augmented reality for industrial use](#)

Grafika i wizualizacja

Wprowadzenie

- Systemy DTP
(poligrafia)



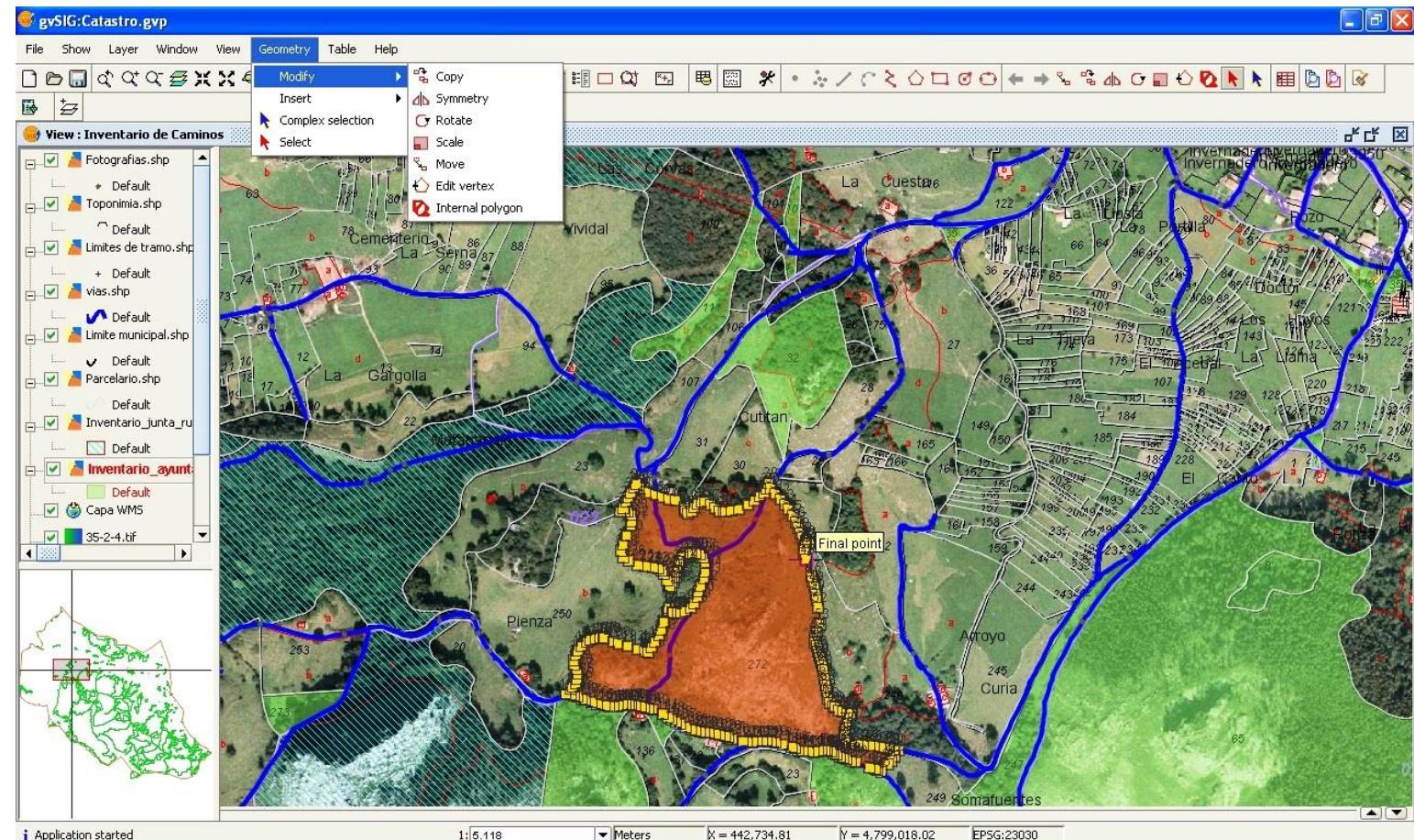
Źródło: Henrik "HerHde" Hüttemann – A Screenshot of Scribus 1.4.6 on Linux Mint 18 with Mint-Y-Theme, showing the Newsletter Wavestyle Template, containing the Image Dew on grass by Luc Viatour.

https://en.wikipedia.org/wiki/File:Scribus_1.4.6_on_Linux_Mint_18.png

Grafika i wizualizacja

Wprowadzenie

- Wizualizacja danych w systemach informacji przestrzennej i geograficznej.



Źródło: Emilio Gómez Fernández – Editing a cartography layer with the gvSIG 1.0
https://commons.wikimedia.org/wiki/File:GvSIG_-_GIS.jpg

Grafika i wizualizacja

Wprowadzenie

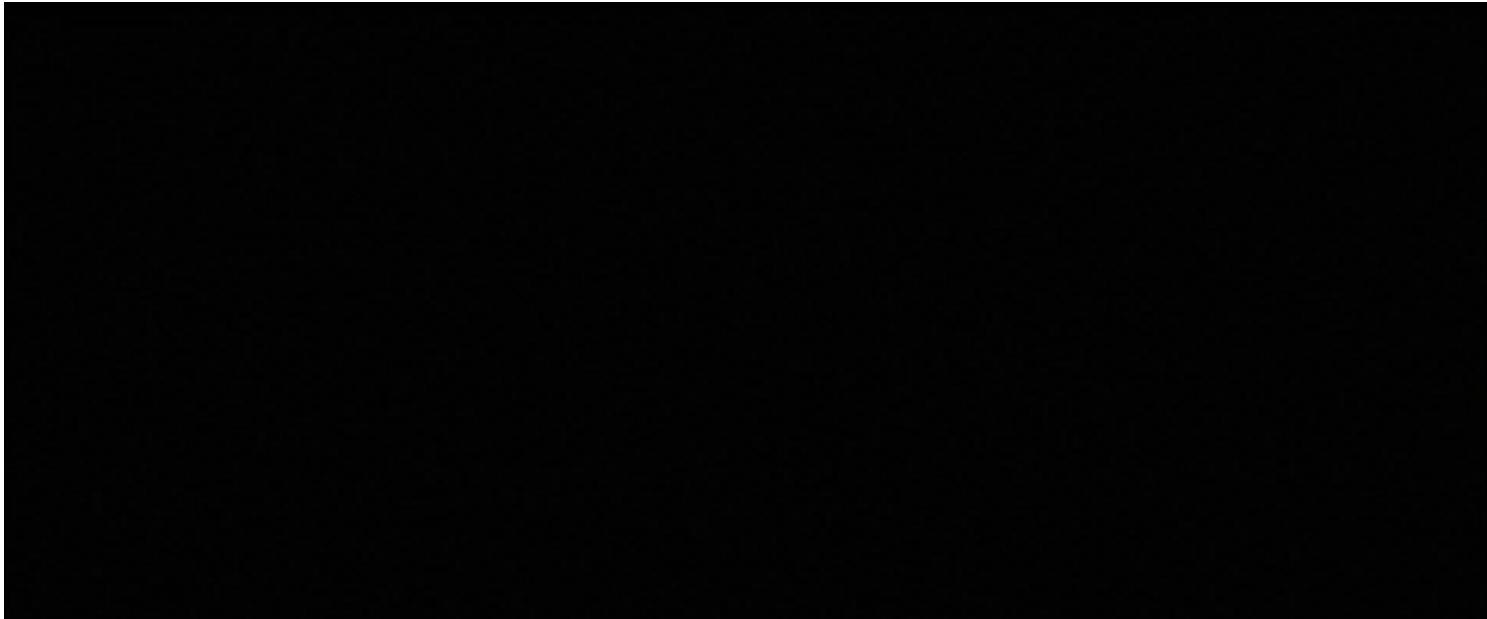
- Edukacja. Jeden z pierwszych eksperymentów edukacyjnych związanych z użyciem komputerów – rok 1974.
(od 37"25')



Grafika i wizualizacja

Wprowadzenie

- Rozgrywka (gry komputerowe, tworzenie filmów)



Źródło: © Blender Foundation, Blender Animation Studio, Andy Goralczyk <https://cloud.blender.org/spring>

Grafika i wizualizacja

- Podstawowe pojęcia

Grafika i wizualizacja

Podstawowe pojęcia

- Przetwarzanie danych – przekształcanie zbioru danych w inny zbiór danych, według określonego przepisu (algorytmu).



Grafika i wizualizacja

Podstawowe pojęcia

- Przetwarzanie sygnałów – przekształcanie sygnału w inny sygnał, lub do postaci reprezentującej sygnał, według określonego przepisu (algorytmu).



Grafika i wizualizacja

Podstawowe pojęcia

- Przetwarzanie obrazów – przekształcanie obrazu w inny obraz, lub do postaci reprezentującej obraz, według określonego przepisu (algorytmu).



Grafika i wizualizacja

Podstawowe pojęcia

- Rozpoznawanie obrazów – interdyscyplinarna dziedzina zajmująca się analizą obrazu mająca na celu określenie przynależności obiektów do klas, wykrycie prawidłowości w danych, podjęcie decyzji itp.



Grafika i wizualizacja

Podstawowe pojęcia

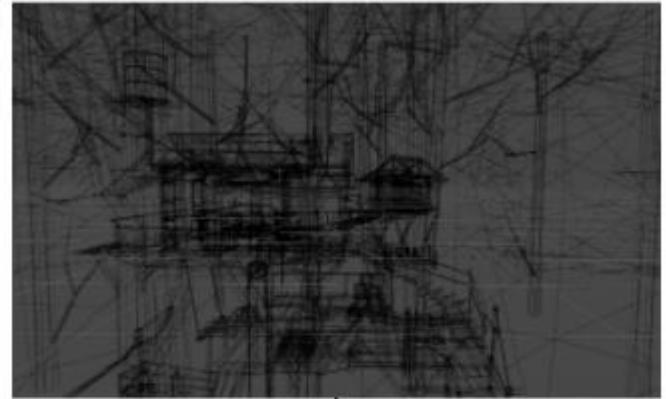
- Grafika komputerowa – dział informatyki zajmujący się tworzeniem obrazów obiektów rzeczywistych i wyimaginowanych. Pod pojęciem grafiki komputerowej rozumie się także dzieła stworzone przez człowieka przy współudziale algorytmów grafiki komputerowej, oraz ich reprezentację w pamięci komputera.



Grafika i wizualizacja

Podstawowe pojęcia

- Tree house



Silnik graficzny

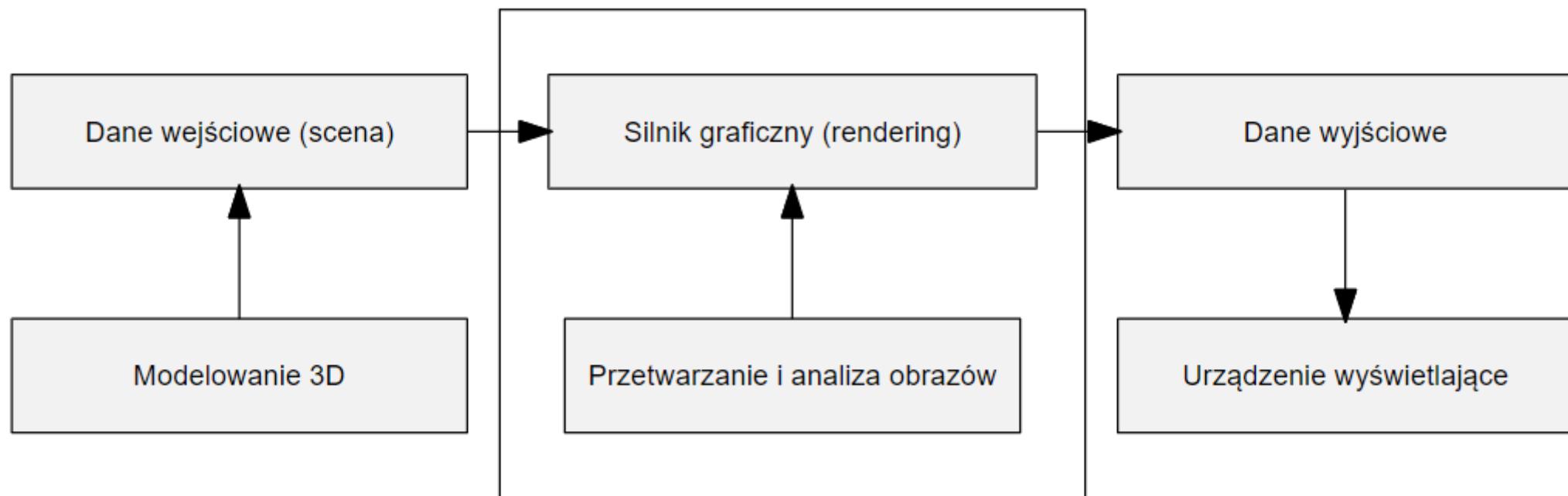


<https://sketchfab.com/3d-models/tree-house-494788a17a7e4c6d9ea62b43e2730607>

Grafika i wizualizacja

Podstawowe pojęcia

- Potok graficzny



Grafika i wizualizacja

Podstawowe pojęcia

Podział grafiki komputerowej ze względu na sposób reprezentacji danych w pamięci komputera:

- grafika rastrowa;
- grafika wektorowa.

Grafika i wizualizacja

Podstawowe pojęcia

grafika rastrowa



grafika wektorowa



Źródło: [Seal Vectors by Vecteezy](#), [Seals sitting on ice](#)

Grafika i wizualizacja

Podstawowe pojęcia

Podział grafiki komputerowej ze względu na wymagany czas realizacji obliczeń:

- grafika nieinterakcyjna – jakość stworzonego obrazu jest ważniejsza od czasu obliczeń;
- grafika interakcyjna (czasu rzeczywistego) – czas wykonania obliczeń jest ważniejszy od jakości stworzonego obrazu, istnieje narzucony maksymalny czas wykonywania obliczeń dla jednego obrazu (klatki).

Grafika i wizualizacja

Podstawowe pojęcia

Podział grafiki komputerowej ze względu na liczbę wymiarów obiektów służących do stworzenia obrazu (obrazów):

- grafika dwuwymiarowa (2D);
- grafika trójwymiarowa (3D).

Grafika i wizualizacja

Podstawowe pojęcia

grafika 2D



Źródło: [Locomotive Vectors by Vecteezy, Flat](#)

[Locomotive Illustration](#)

grafika 3D



Źródło modelu: Ashley Aslett – [WWI Trench Locomotive](#)

[Locomotive](#)

Licencja modelu: CC Attribution, brak zmian w modelu.

Grafika i wizualizacja

Podstawowe pojęcia

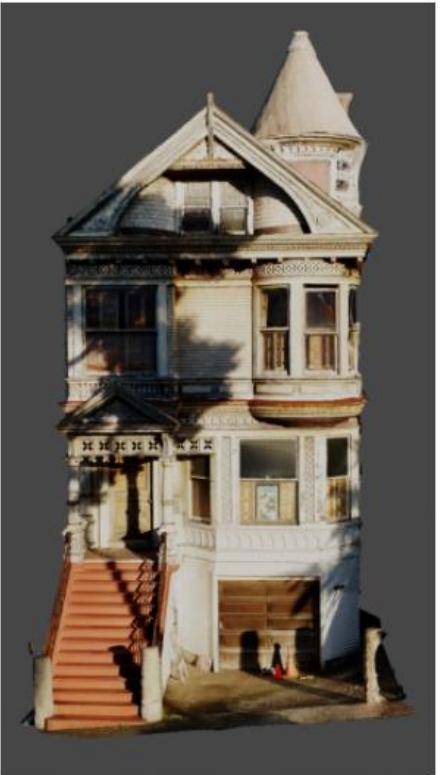
Podział grafiki komputerowej ze względu na dążenie do realizmu tworzonych obrazów:

- grafika realistyczna – przy jej tworzeniu stosuje się algorytmy zapewniające uzyskanie obrazu przypominającego jak najwierniej obraz widziany oczami człowieka;
- grafika nierealistyczna – przy jej tworzeniu stosuje się algorytmy odwzorowujące różnego rodzaju techniki malarskie lub wizje artystyczne.

Grafika i wizualizacja

Podstawowe pojęcia

grafika realistyczna



Źródło modelu: Azad Balabanian – #12 - Abandoned

Victorian House

Licencja modelu: CC Attribution, brak zmian w modelu.

grafika nierealistyczna



Grafika i wizualizacja

- Sprzęt

Grafika i wizualizacja

Sprzęt

Sprzęt wykorzystywany w grafice komputerowej:

- komputer;
- monitor;
- urządzenie wskazujące;
- cyfrowy aparat fotograficzny i kamera cyfrowa;
- skaner obrazu i skaner 3D;
- urządzenia drukujące.

Grafika i wizualizacja

Sprzęt

- Algorytmy grafiki komputerowej wymagają dużych mocy obliczeniowych. Komputer wykonujący te algorytmy musi być wydajny. Trójwymiarowa grafika wymaga wydajnych kart graficznych. W szczególności tyczy się to grafiki czasu rzeczywistego. Tworzenie i obróbka filmów wymaga szybkich i pojemnych dysków twardych.



Zdjęcie z [Unsplash'a komputera](#) wykonane przez
Caspar'a Camille Rubin'a.



Zdjęcie z [Unsplash'a karty graficznej](#) wykonane
przez Christian'a Wiediger'a.

Grafika i wizualizacja

Sprzęt

- Monitor jest urządzeniem służącym do wizualizacji danych przetworzonych przez komputer.



Zdjęcie z [Unsplash'a](#) monitora wykonane przez
Daniel'a Korpai'a.

Grafika i wizualizacja

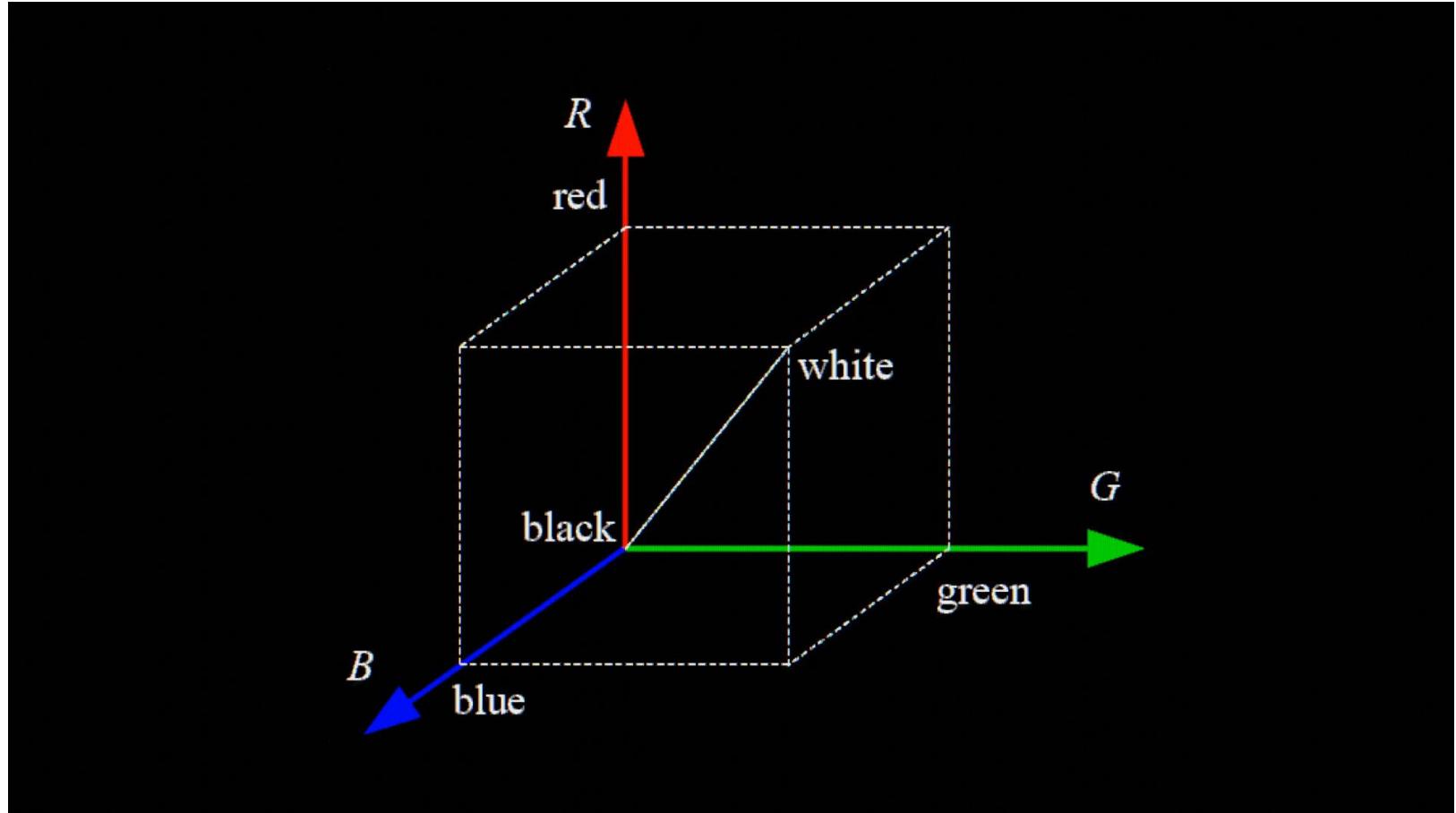
Sprzęt

- Monitor wyposażony jest w prostokątną matrycę składającą się (najczęściej) ze świecących punktów – pikseli. W czarno-białych monitorach dla każdego piksela można ustawić jasność z pewnego zakresu. W kolorowych monitorach każdy piksel składa się najczęściej z trzech elementów świecących w kolorach czerwonym, zielonym i niebieskim. Możliwe są zastosowania innych zestawów kolorów, na przykład czerwonego, zielonego, niebieskiego i białego.

Grafika i wizualizacja

Sprzęt

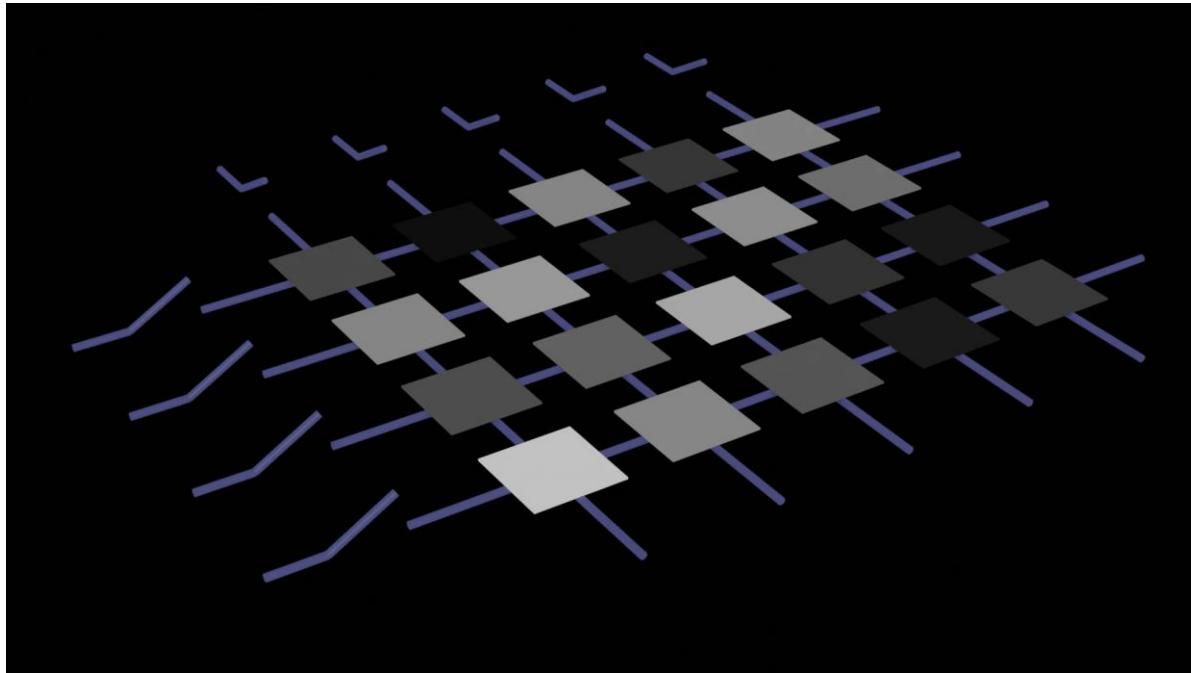
- Zasada działania monitora



Grafika i wizualizacja

Sprzęt

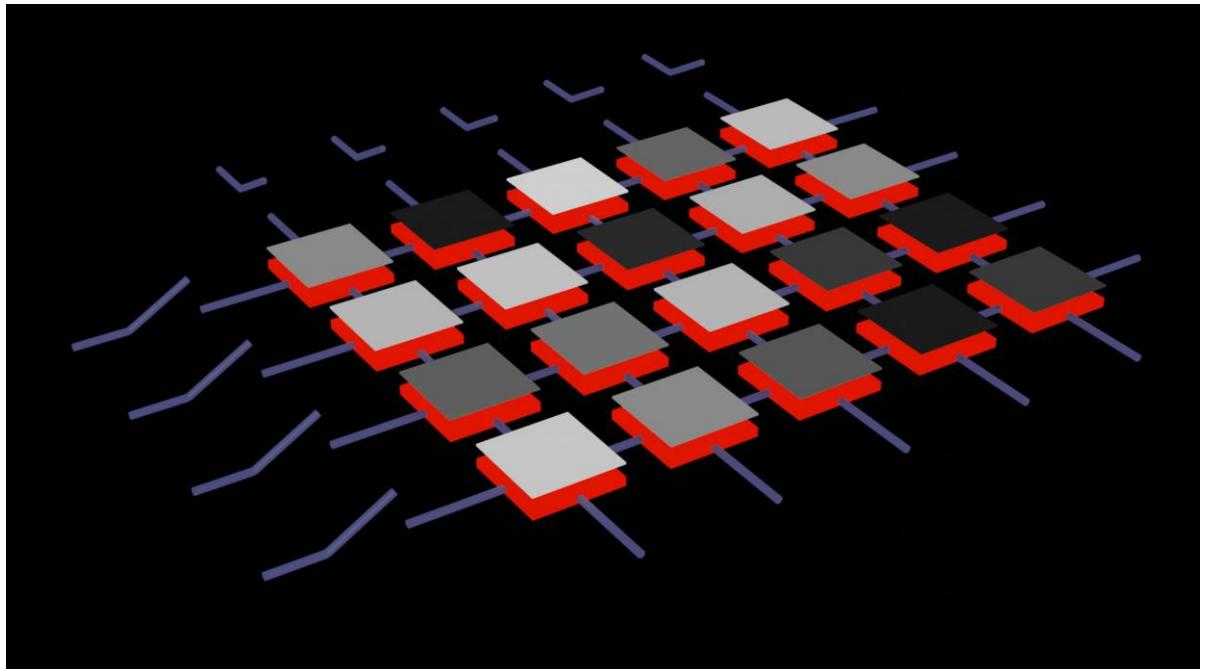
- Informacje przekazywane do pikseli matrycy są przesyłane najczęściej sekwencyjnie. W matrycach pasywnych element wyświetlający musi zachować określony stan do czasu przesłania następnej informacji.



Grafika i wizualizacja

Sprzęt

- W matrycach aktywnych piksel wyposażony jest w dodatkowy element (składający się z tranzystora i kondensatora), który zapamiętuje przekazaną informację do czasu następnej jej aktualizacji. Stosuje się takie rozwiązanie w przypadku elementów wyświetlających wymagających do pracy stałego zasilania.



Grafika i wizualizacja

Sprzęt

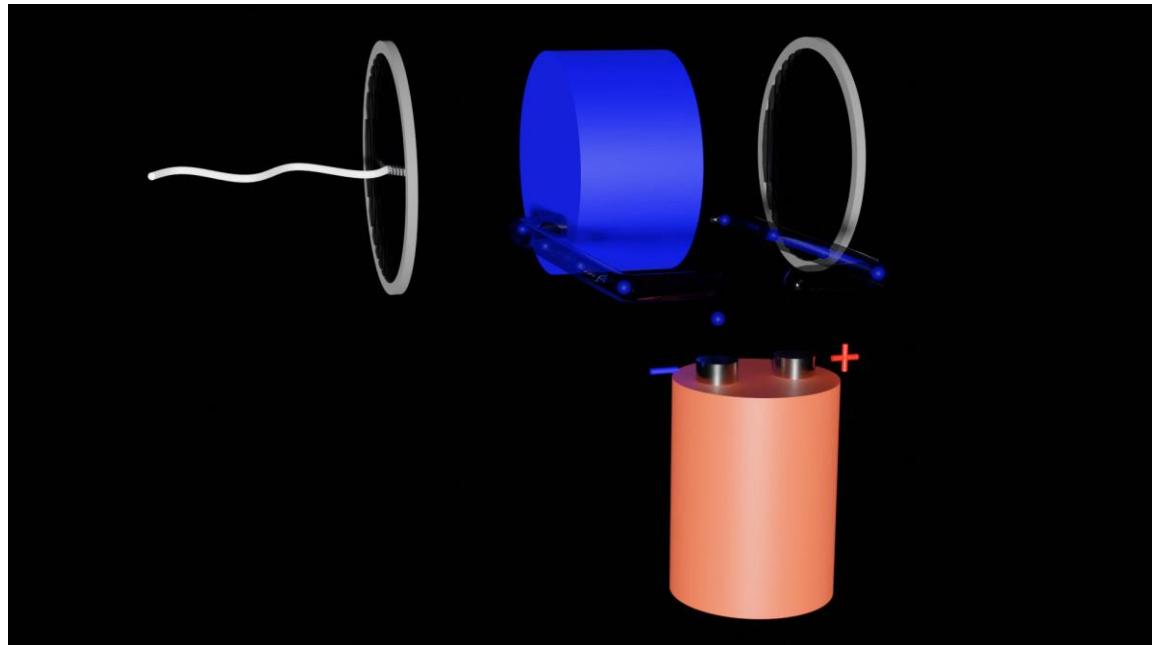
Najczęściej spotykane obecnie matryce monitorów:

- LCD;
- OLED;
- E-Ink.

Grafika i wizualizacja

Sprzęt

- W monitorach LCD (ang. liquid-crystal display) świecenie pikseli najczęściej uzyskuje się poprzez ich podświetlenie od tyłu. Światło przechodzi przez filtr polaryzacyjny, następnie przez substancję mogącą zmieniać kierunek polaryzacji pod wpływem napięcia i znów przez filtr polaryzacyjny. Dzięki temu ilość przechodzącego światła zależy od podanego napięcia.



Grafika i wizualizacja

Sprzęt

- Monitory LCD do uzyskania dobrej jakości obrazu wymagają stałego podświetlania matrycy od tyłu. Intensywność podświetlania nie zależy od jasności świecącego punktu.
- Matryca wraz z podświetleniem pobiera podobną ilość energii niezależnie od tego, czy wyświetlany jest cały biały czy cały czarny ekran. Z tego względu matryce LCD stosuje się tam, gdzie ilość pobranej energii nie ma dużego znaczenia (na przykład monitory komputerów stacjonarnych) lub gdy matryca musi zapewnić wysoką jasność obrazu (rzutniki).

Grafika i wizualizacja

Sprzęt

- LCD

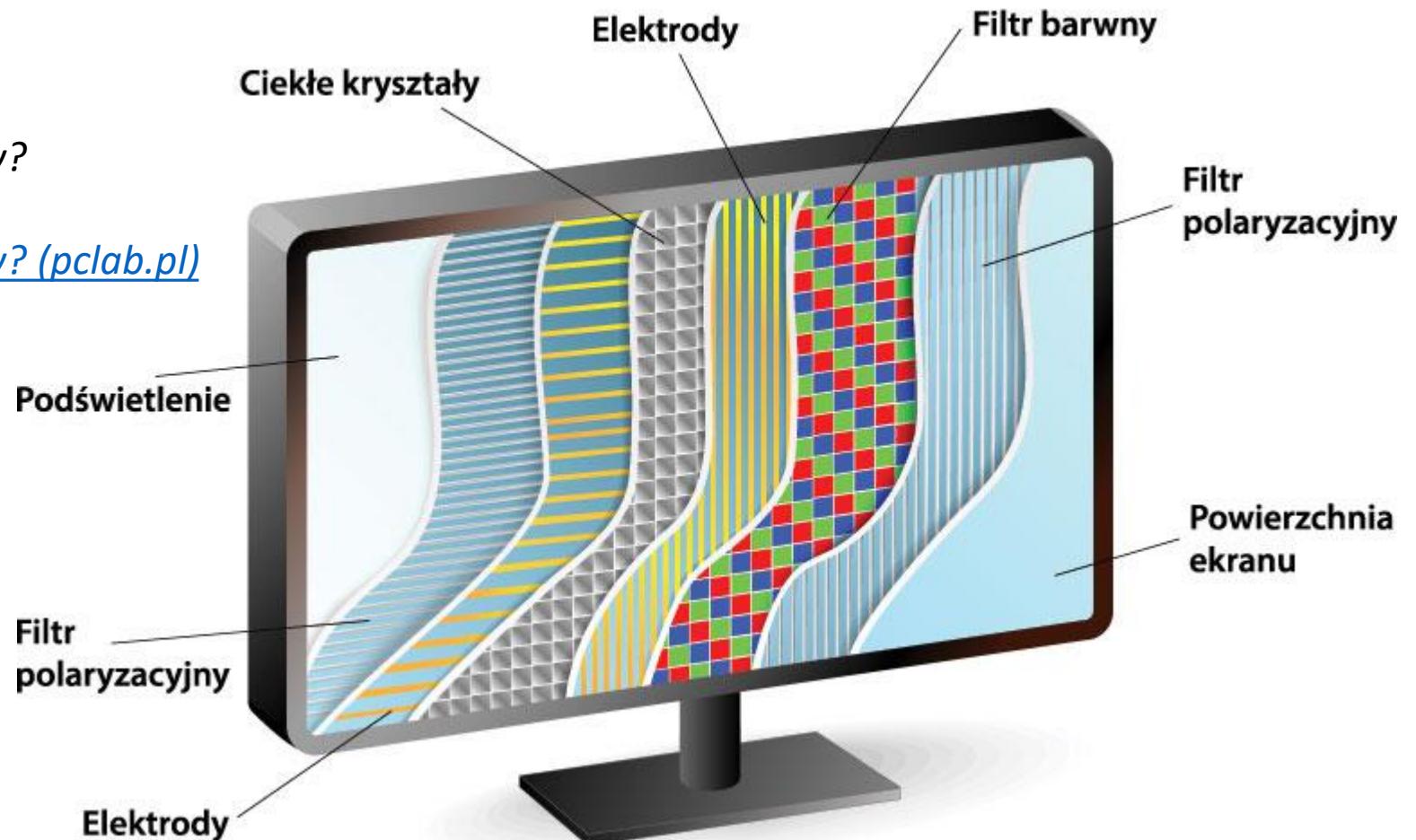
Źródło:

Jak działa monitor ciekłokrystaliczny?

Obraz w monitorze LCD

[Jak działa monitor ciekłokrystaliczny? \(pclab.pl\)](#)

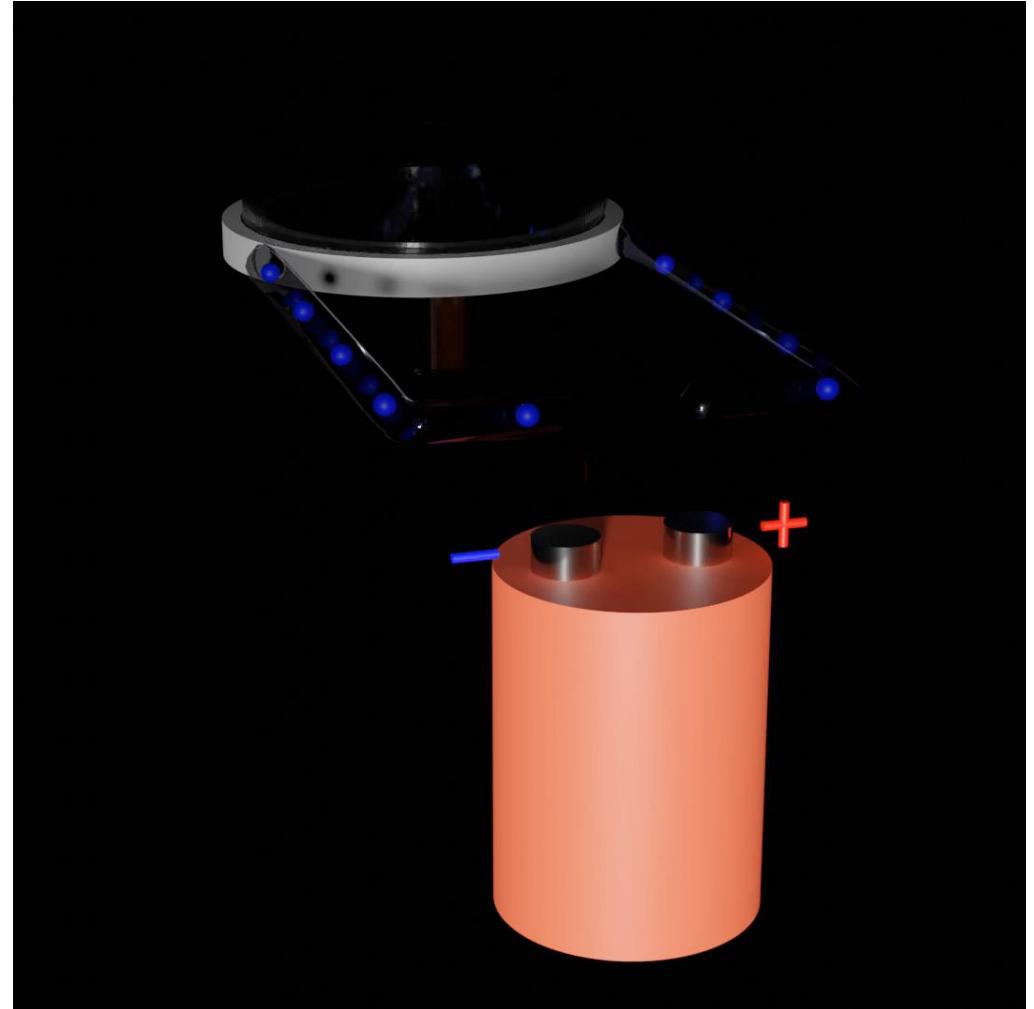
SCHEMAT EKRANU LCD



Grafika i wizualizacja

Sprzęt

- Ekrany monitorów OLED (ang. organic light-emitting diode) nie wymagają podświetlenia. Składają się one z elementów świecących pod wpływem przyłożonego napięcia.



Grafika i wizualizacja

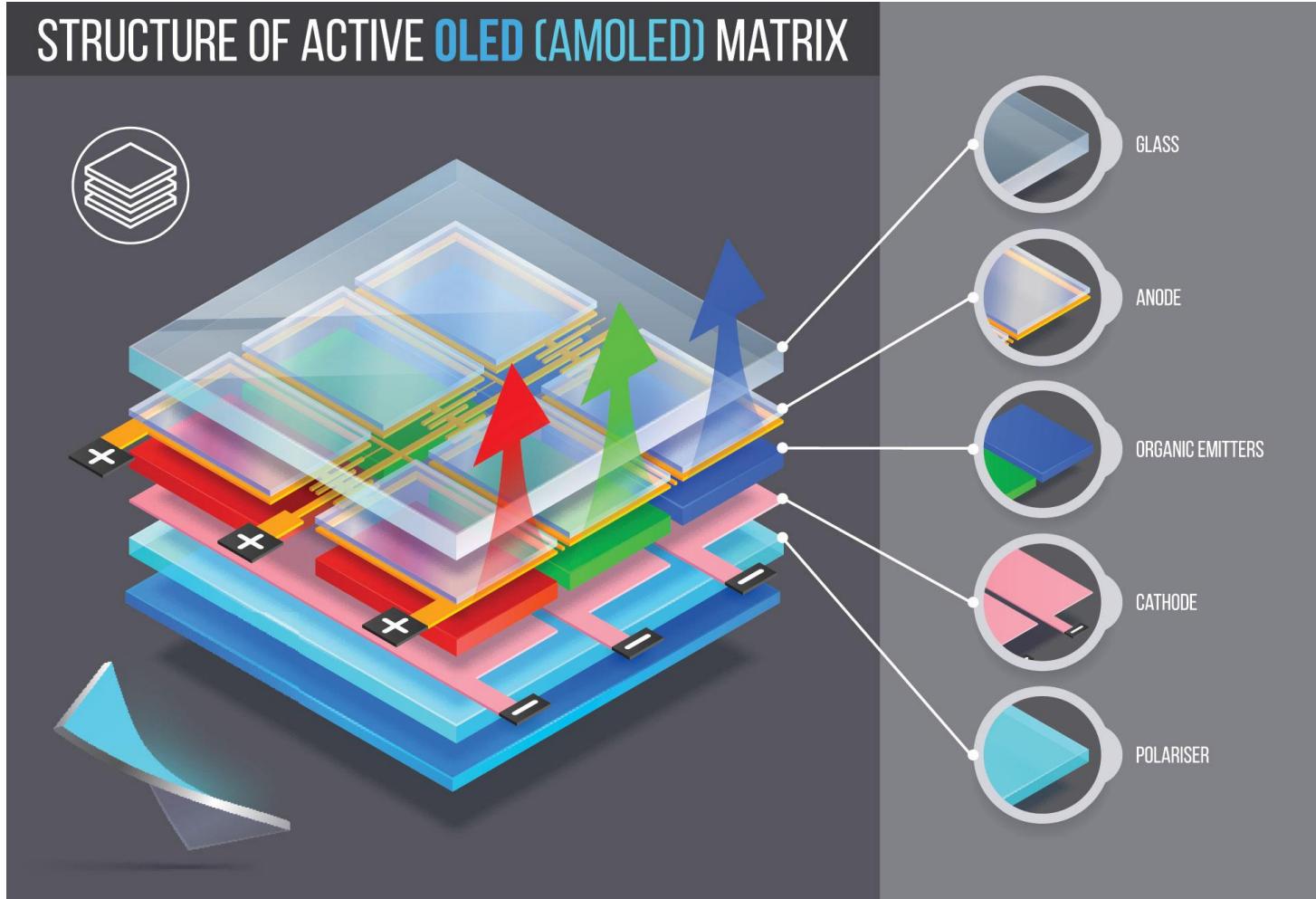
Sprzęt

- OLED

Źródło:

Monitor bez tajemnic, cz. 2.

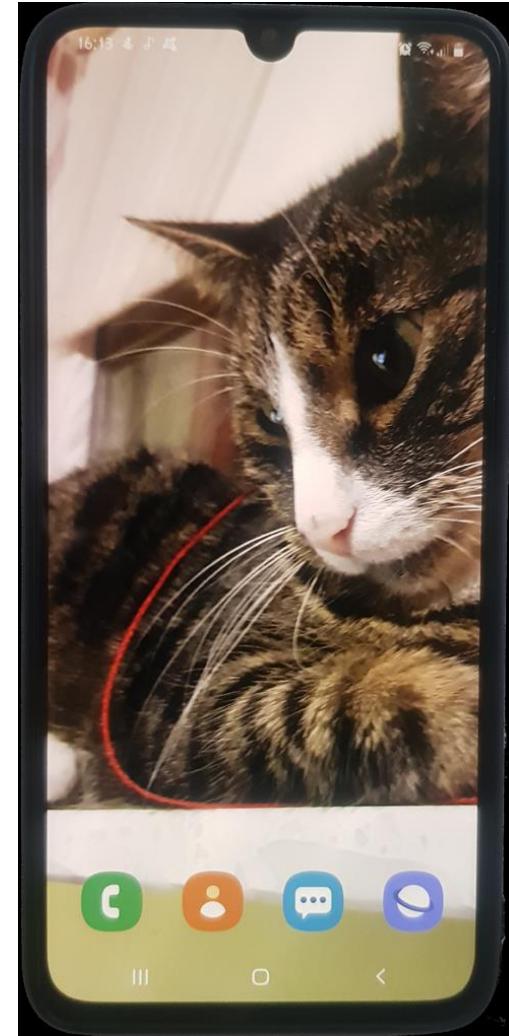
<https://pclab.pl/art84224-5.html>



Grafika i wizualizacja

Sprzęt

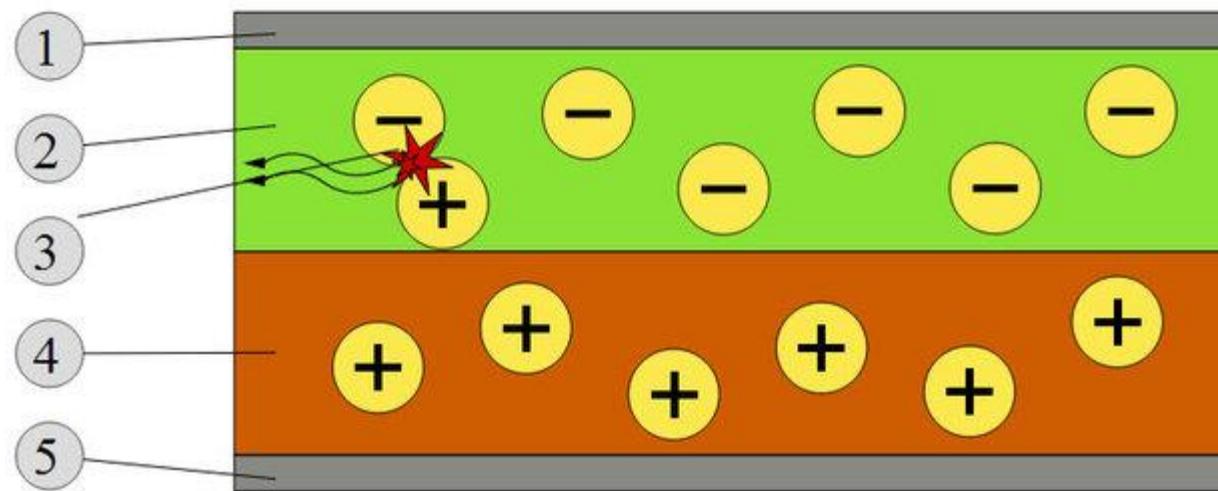
- W monitorach OLED, źródłem światła jest matryca. Pobór energii przez matrycję zależy od zawartości ekranu. Z tego względu monitory OLED zużywają mniej energii niż monitory LCD i stosowane są w urządzeniach pracujących na zasilaniu baterijnym wymagających dobrej jakości obrazu (telefony komórkowe, tablety itp.).



Grafika i wizualizacja

Sprzęt

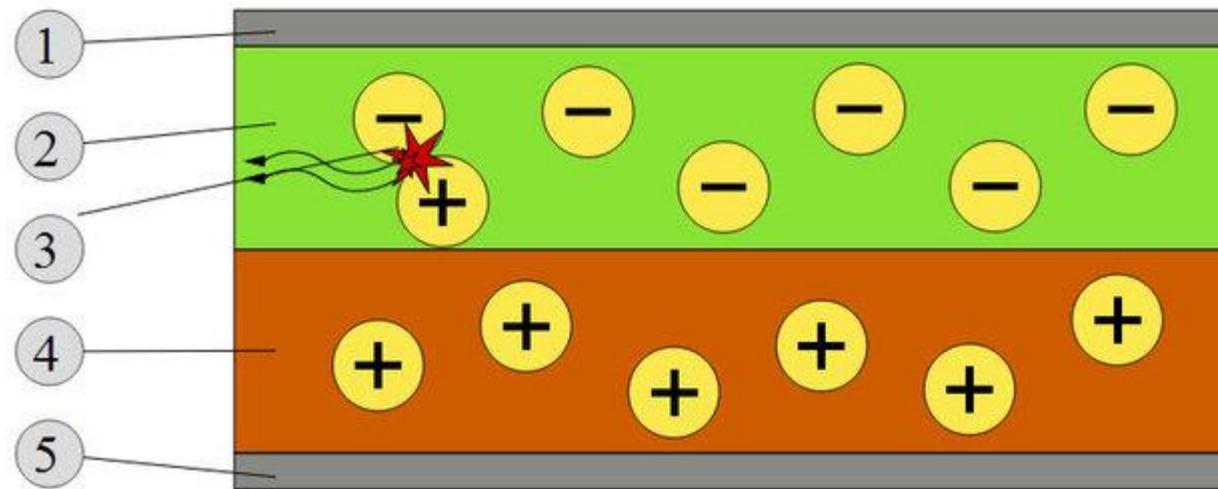
- Świecenie elementów matrycy OLED jest wynikiem tzw. procesu rekombinacji, czyli odwrotności jonizacji. W trakcie rekombinacji połączenie się jona dodatniego (cząstki z brakującym elektronem w pełnym paśmie walencyjnym) z elektronem swobodnym z pasma przewodnictwa skutkuje wyzwoleniem części energii w postaci fotonu.



Grafika i wizualizacja

Sprzęt

- Każdy półprzewodnikowy element aktywny matrycy OLED składa się z: katody (1), warstwy emisyjnej (2) zbudowanej z organicznych polimerów charakteryzujących się przewodnictwem elektronowym, warstwy przewodzącej (4) - ta składa się z organicznych elementów charakteryzującej się przewodnictwem dziurowym - oraz przezroczystej anody (5) wykonanej najczęściej z tzw. szkła przewodzącego (ITO - Indium Tin Oxide).



Grafika i wizualizacja

Sprzęt

- TN, VA i IPS to OLED, QLED czy LCD?
- Jakiego rodzaju podświetlenia występują w ekranach LCD?
- Jakie są różnice między pojęciami: ekran, matryca, monitor?

Grafika i wizualizacja

Sprzęt

TN



kontra

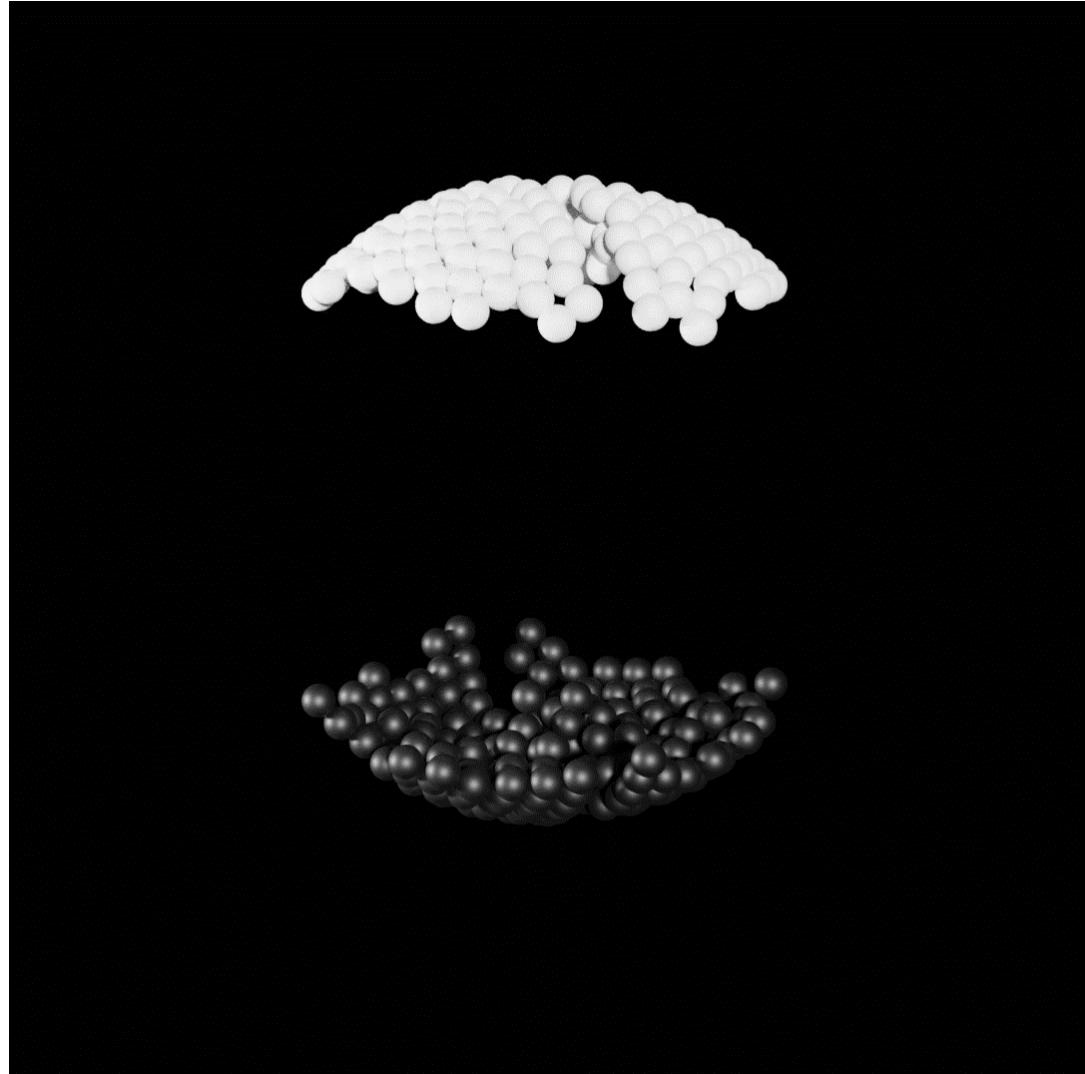
IPS



Grafika i wizualizacja

Sprzęt

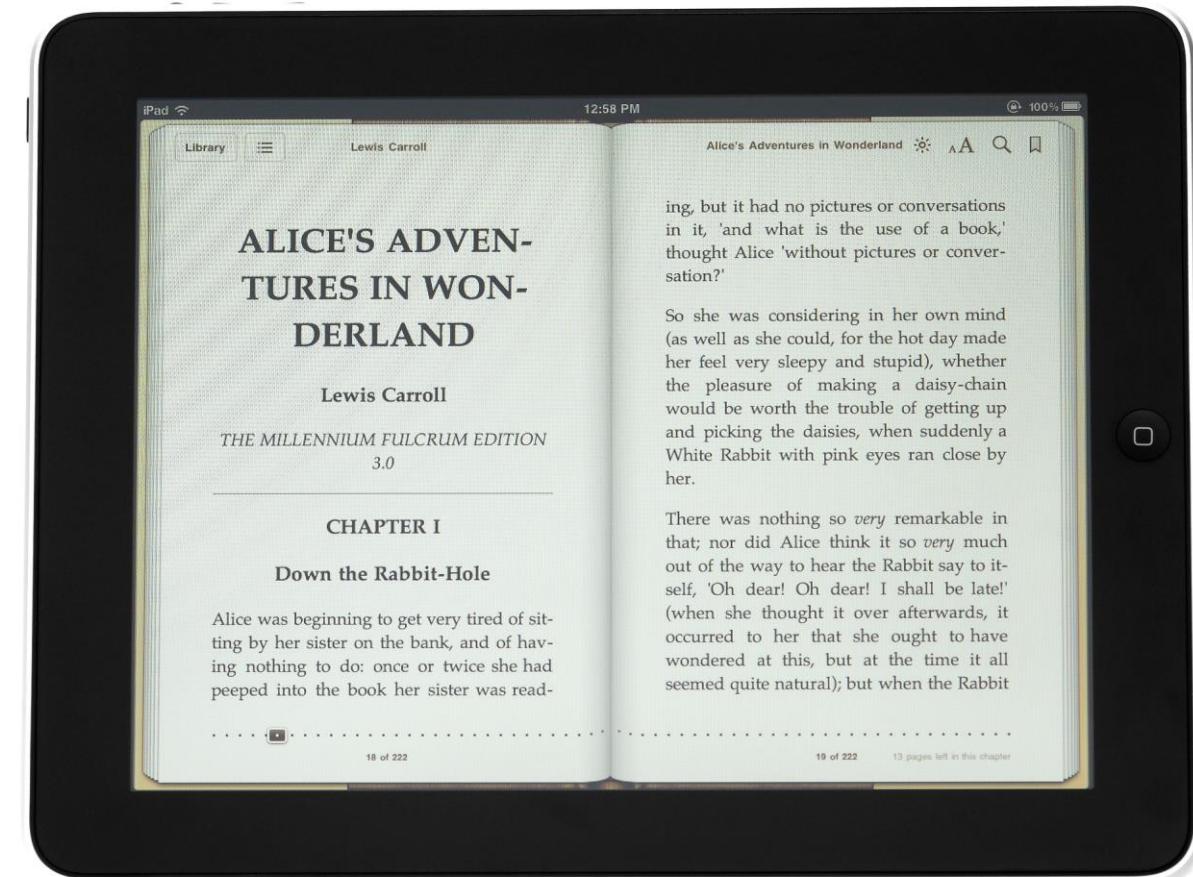
- Matryce e-Ink (ang. electronic ink) składają się z mikrokapsułek zawierających dodatnio naładowane białe cząstki i ujemnie naładowane czarne cząstki. Zmieniają one swoje położenie po przyłożeniu potencjału elektrycznego.



Grafika i wizualizacja

Sprzęt

- Matryce eInk pobierają energię tylko w momencie zmiany koloru wyświetlanego piksela. Dzięki temu są bardzo oszczędne. Jakość uzyskanego obrazu jest bardzo dobra, jednak liczba dostępnych kolorów niewielka. Z tego względu stosowane są głównie w czytnikach książek elektronicznych.



Grafika i wizualizacja

Sprzęt

Urządzenie wskazujące jest urządzeniem wejściowym komputera pozwalającym na wprowadzenie danych przestrzennych. Często ruch urządzenia wskazującego jest odzwierciedlany na ekranie za pomocą kurSORA. Przykładowe urządzenia wskazujące:

- mysz komputerowa,
- manipulator kulkowy,
- manipulator punktowy,
- urządzenia dotykowe.

Grafika i wizualizacja

Sprzęt

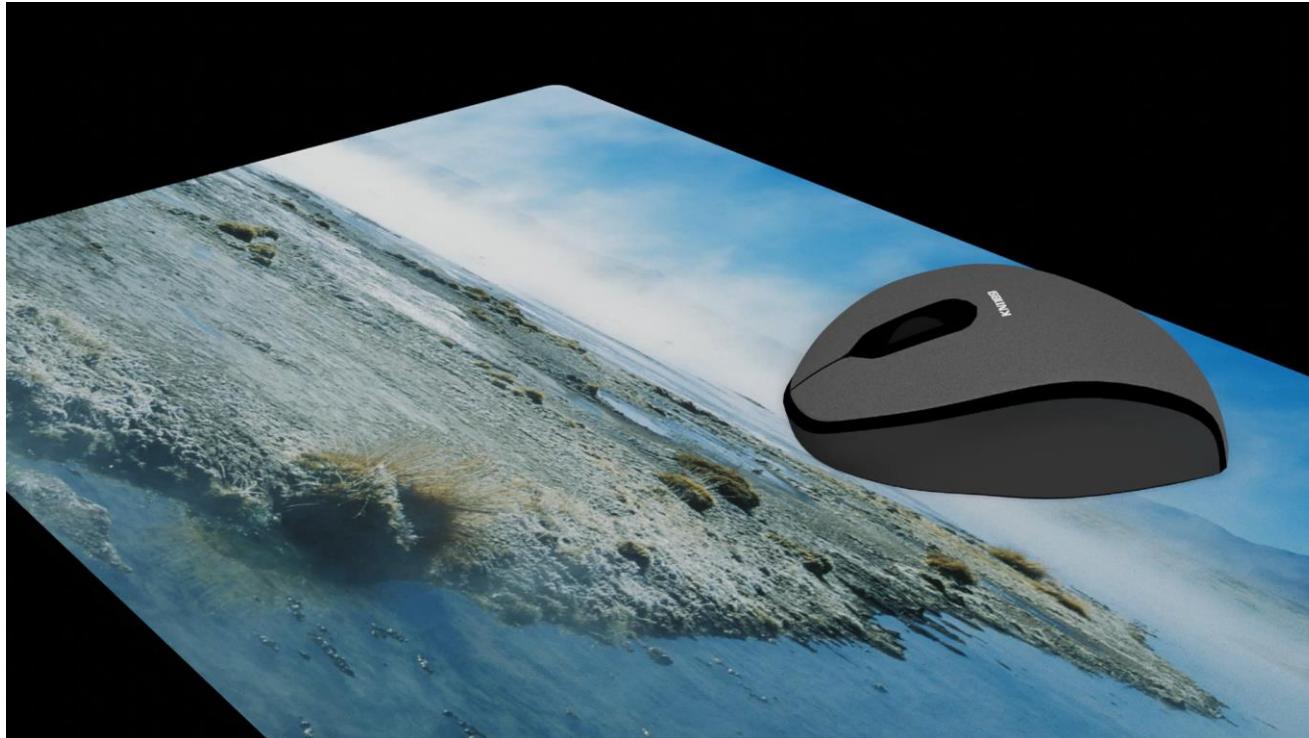
- Mysz komputerowa pozwala na wprowadzanie danych przestrzennych poprzez poruszanie nią na płaskiej powierzchni. Obecnie najpopularniejsze są myszy optyczne. W myszy tego rodzaju, przesunięcie wykrywane jest na podstawie światła odbitego od powierzchni, na której się ona znajduje.



Grafika i wizualizacja

Sprzęt

- Jednym z rozwiązań pozwalającym na wyznaczanie parametrów ruchu myszy jest wykorzystanie kamery. Zarejestrowane po sobie obrazy są porównywane i na tej podstawie określany jest wektor przesunięcia myszy.



Grafika i wizualizacja

Sprzęt

- Manipulator kulkowy (trackball, manipulator kulowy) – urządzenie wskazujące, które wykorzystuje obrotową kulę do przemieszczania kurSORA. Łatwiejsze w obsłudze od myszy dla ludzi z niektórymi problemami ruchowymi.



Zdjęcie z Pixabay'a manipulatora kulowego wykonane przez skeeza'a.

<https://pixabay.com/photos/tracking-mouse-wireless-computer-860223/>

Grafika i wizualizacja

Sprzęt

- Manipulator punktowy (trackpoint, punkt dotykowy) – miniaturowy joystick umieszczony pomiędzy klawiszami.

Zdjęcie z Pixabay'a manipulatora punktowego wykonane przez Lorenzo Cafaro.
<https://pixabay.com/photos/laptop-keyboard-notebook-trackpoint-1864126/>



Grafika i wizualizacja

Sprzęt

- Urządzenia dotykowe pozwalają na wprowadzenie informacji przestrzennych poprzez dotknięcie (zbliżenie) palca lub elementu przypominającego długopis do płaskiej powierzchni. Konstrukcyjnie można je podzielić na urządzenia, w których detekcja położenia odbywa się z wykorzystaniem specjalnie przygotowanej powierzchni do rysowania lub w elemencie przypominającym długopis.



Grafika i wizualizacja

Sprzęt

W powierzchniach (lub na granicach powierzchni) służących do rysowania umieszczane są specjalne czujniki pozwalające na wykrycie położenia obiektu dotykającego lub zbliżającego się do nich. Przykładowe czujniki:

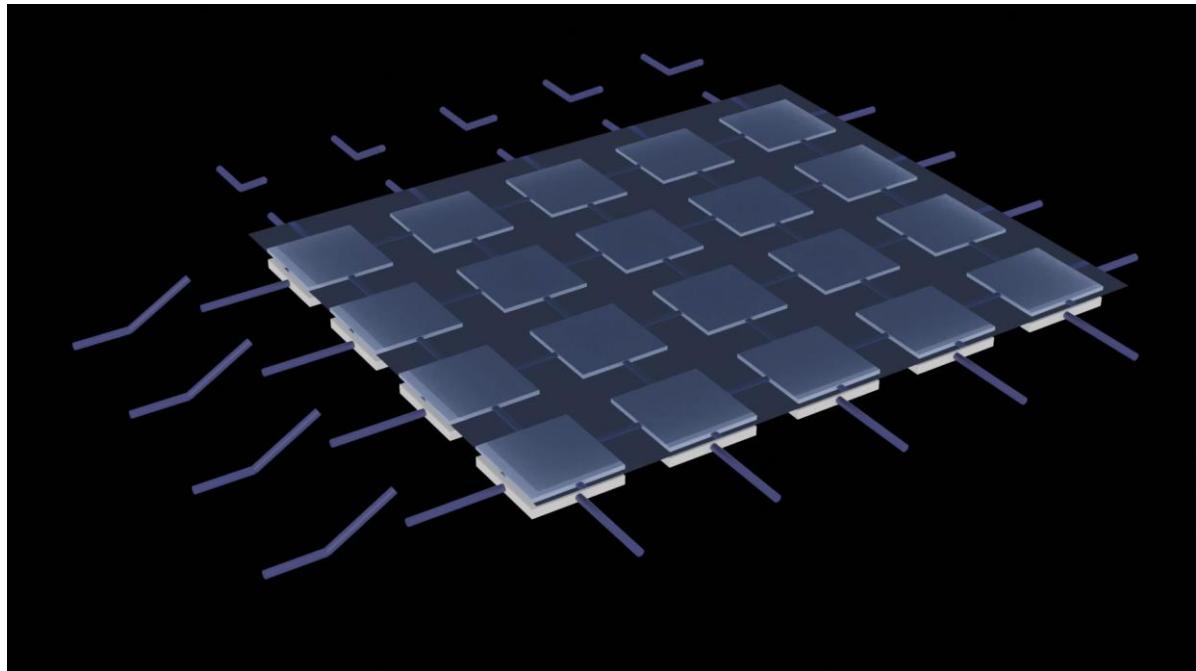
- pojemnościowe,
- rezystancyjne,
- podczerwone,

Istnieje również rozwiązanie, zwane aktywnym piórem, w którym czujniki w powierzchni do rysowania komunikują się z urządzeniem rysującym.

Grafika i wizualizacja

Sprzęt

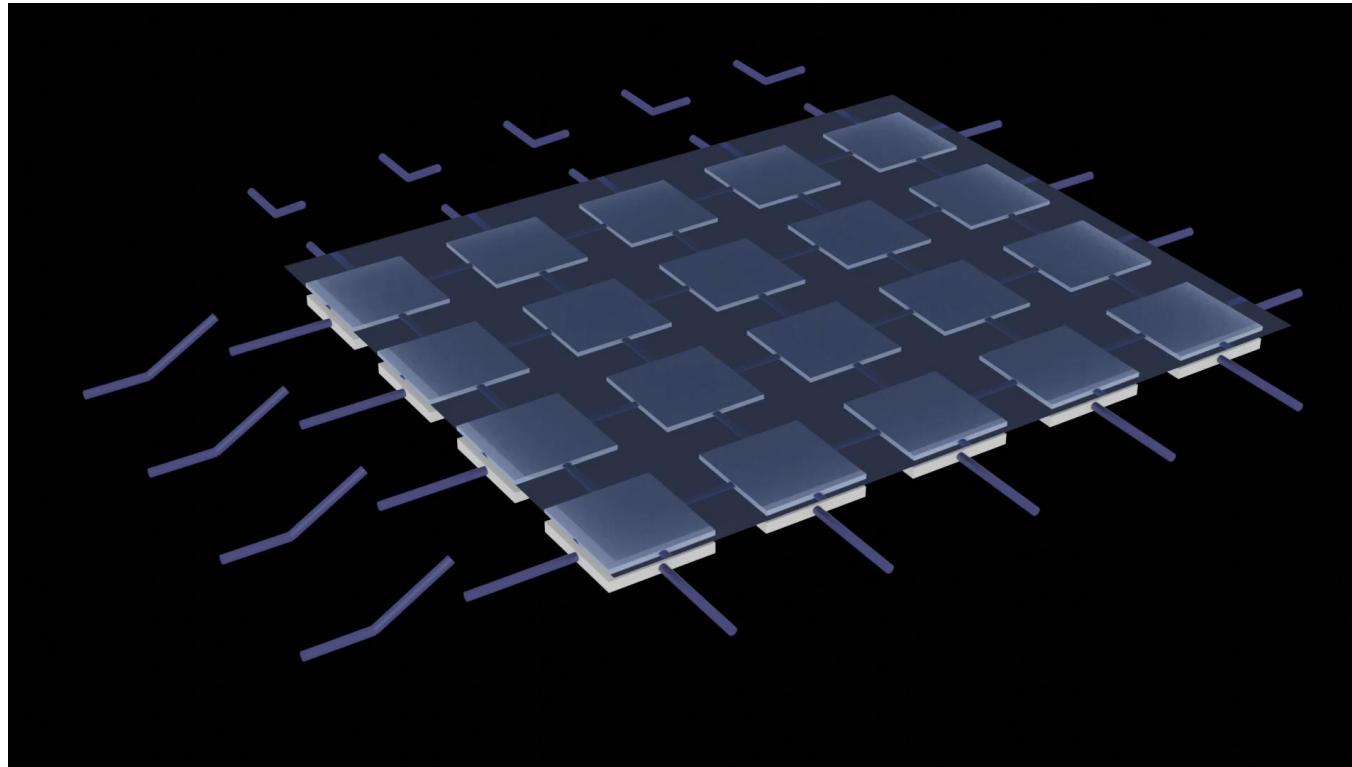
- Czujniki pojemnościowe wykrywają zmianę pojemności związaną ze zmianą pola elektrostatycznego.
- Możliwa jest detekcja położenia tylko tych obiektów, które wpływają w sposób znaczny na to pole.
- Na animacji przedstawiono przykładowe położenie czujników, ich układ połączeń oraz sposób odczytu. Istnieją rozwiązania, w których czujniki są tylko na obrzeżach powierzchni do rysowania.



Grafika i wizualizacja

Sprzęt

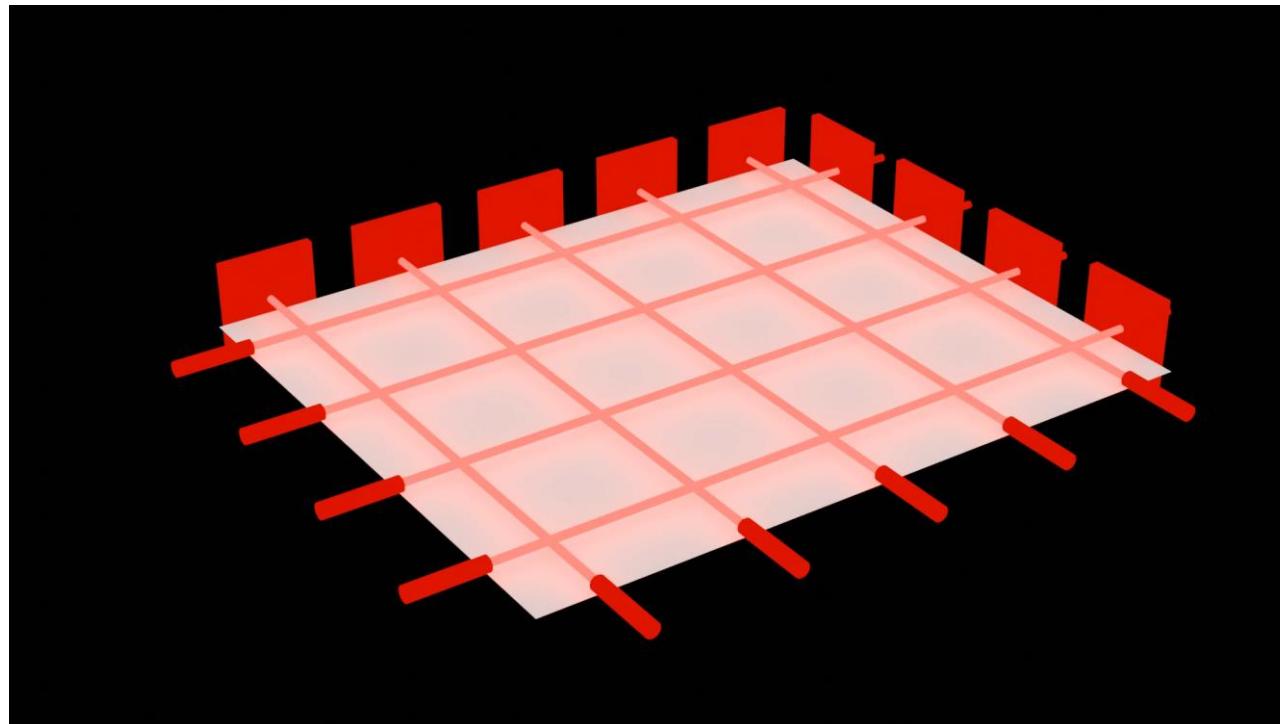
- Czujniki rezystancyjne wykrywają zmianę rezystancji związaną z odkształceniem materiału.



Grafika i wizualizacja

Sprzęt

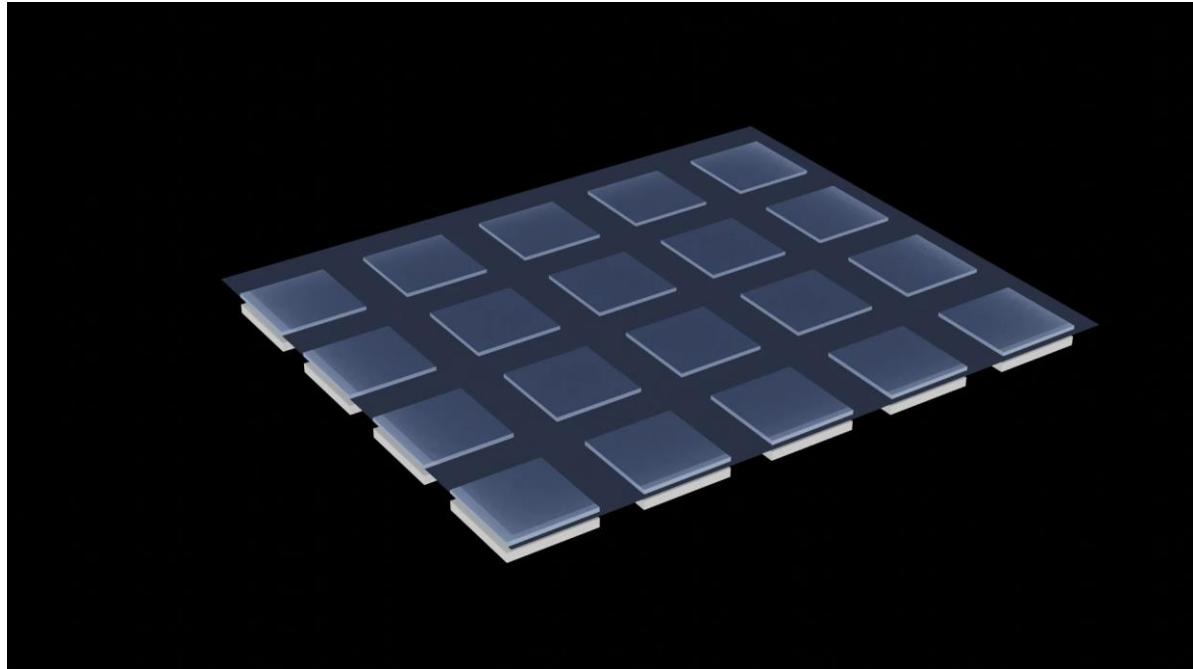
- Czujniki podczerwieni (fotodetektory) odbierają sygnały z nakierowanych na nie podczerwonych diod LED. Obiekt dotykający powierzchni zasłania diody LED, co pozwala na określenie jego położenia.



Grafika i wizualizacja

Sprzęt

- Powierzchnie wykorzystujące aktywne pióra pozwalają na uzyskanie większej dokładności jednak współpracują tylko z dedykowanym urządzeniem piszącym. Dzięki wymianie informacji z aktywnym piórem, oprócz położenia, określana może być siła nacisku, pochylenie, czy też fakt wcisnięcia określonego przycisku. Przykładem aktywnych piór mogą być urządzenia japońskiej firmy Wacom, które do komunikacji wykorzystują pole elektromagnetyczne.



Grafika i wizualizacja

Sprzęt

Rodzaje urządzeń wykorzystujących czujniki zamontowane w powierzchniach:

- panel dotykowy,
- tablet graficzny,
- ekran dotykowy.

Grafika i wizualizacja

Sprzęt

- Panel dotykowy (touchpad) – urządzenie zastępujące mysz w laptopach. Zamienia ruch palców po jego powierzchni na ruch wskaźnika myszy.



Grafika i wizualizacja

Sprzęt

- Tablet graficzny (digitizer) – powierzchnia do rysowania z dedykowanym do niej urządzeniem rysującym. Powierzchnia do rysowania w niektórych modelach może wyświetlać obraz, ale nie stanowi podstawowego ekranu komputera.



Grafika i wizualizacja

Sprzęt

- Ekran dotykowy – wyświetlacz umożliwiający określenie punktu (punktów), w których został dotknęty.



Grafika i wizualizacja

Sprzęt

- Pióro cyfrowe (digital pen, smart pen) – jest urządzeniem przypominającym długopis wyposażonym w elektronikę. Zwykle posiada takie funkcje jak: czułość na dotyk, rejestracja pisma odręcznego wewnętrznej pamięci, wymiana danych z komputerem.

Źródło: Warfieldian – Photo of the Live Scribe Echo pen
<https://en.wikipedia.org/wiki/File:Livescribeecho.png>



Grafika i wizualizacja

Sprzęt

Przykładowe technologie wykorzystywane w piórach cyfrowych:

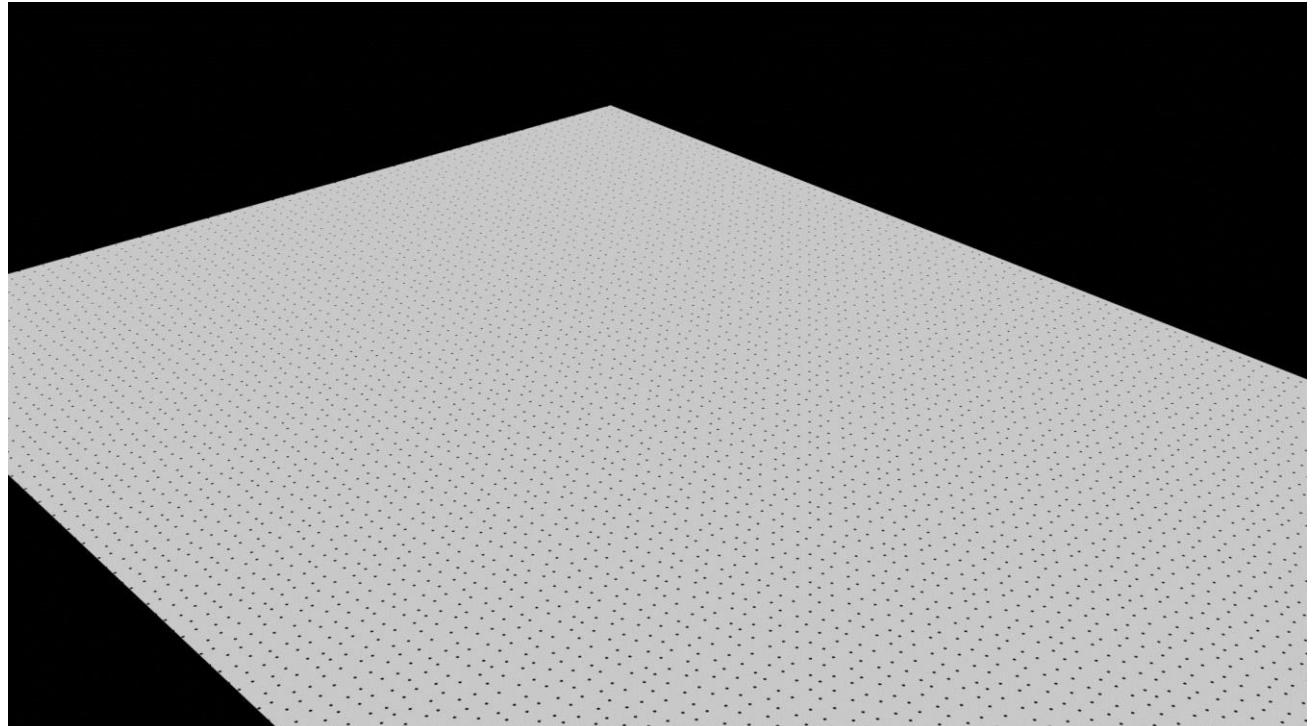
- wykrywanie ruchu za pomocą akcelerometru;
- pozycjonowanie z wykorzystaniem technologii aktywnego pióra,
- wykrywanie ruchu kulki,
- pozycjonowanie z wykorzystaniem specjalnego wzoru na kartce.

Grafika i wizualizacja

Sprzęt

- Przykład pozycjonowania cyfrowego pióra z wykorzystaniem aparatu cyfrowego i nadruku w postaci kropek. Nadruk ten stosowany jest przez firmę Anoto w piórach cyfrowych livescribe.

Źródło modelu: *wass – model pióra Pen_Day1*
<https://sketchfab.com/3d-models/pen-day1-64e9d2ab4c4d4459a8a53b5e2aa74afd>



Grafika i wizualizacja

Sprzęt

Sprzęt, który również należałoby dalej omówić:

- Cyfrowy aparat fotograficzny
- Skaner
- Urządzenia drukujące

I zagadnienia pozostałe do omówienia:

Oprogramowanie

Geometria

Rys historyczny

Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Grafika i wizualizacja

Obraz

Edward Półrolniczak, dr inż., Katedra Systemów Multimedialnych
// na podstawie wykładów Mariusza Borawskiego, dr hab. inż..

Grafika i wizualizacja

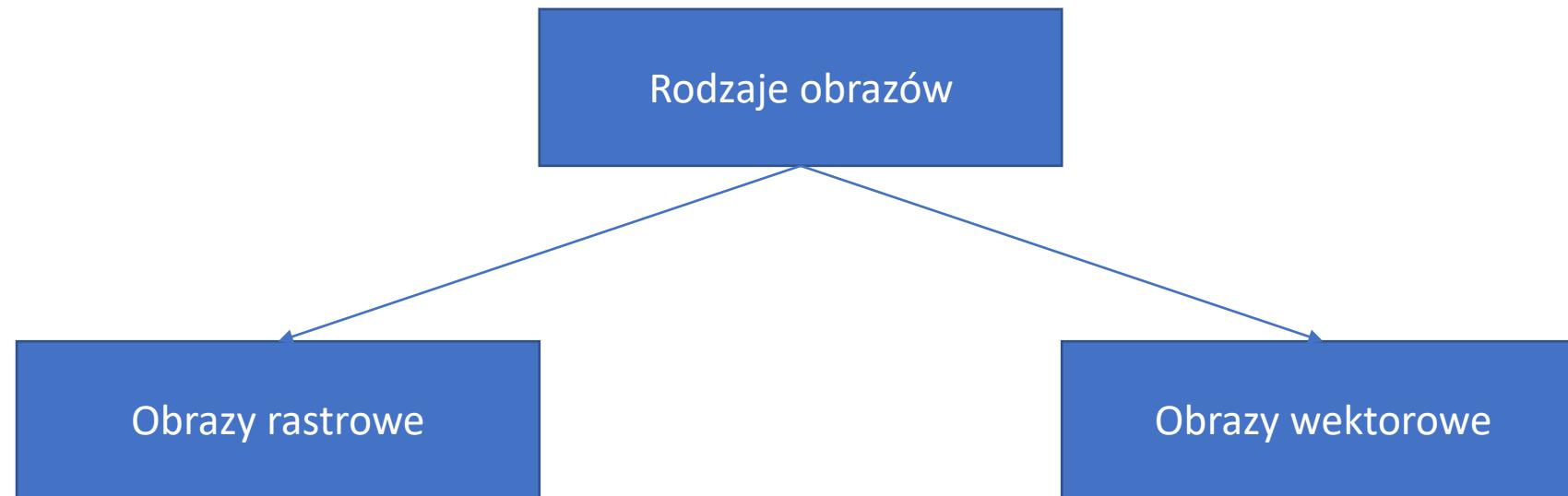
Wprowadzenie

- Rodzaje obrazów
- Obraz binarny
- Obraz monochromatyczny
- Obraz kolorowy
- Grafika wektorowa
- Rejestracja obrazu

Grafika i wizualizacja

- rodzaje obrazów

podział



Grafika i wizualizacja

- obraz wektorowy

Obraz wektorowy i rastrowy różnią się podejściem do reprezentacji obrazu.

W obrazie wektorowym to co widzimy jest opisane za pomocą figur geometrycznych: punktów, okręgów, prostokątów, krzywych, itp. W przypadku punktów zapamiętuje się ich współrzędne, dla okręgów – współrzędne środka i promień, itd. Sposób zapamiętywania figur zależy od przyjętej konwencji w danym formacie zapisu obrazu wektorowego.

Grafika i wizualizacja

- obraz rastrowy

Obraz rastrowy stanowi tablicę liczb opisującą kolory pikseli (punktów) na ekranie lub wydruku.

AB	23	4F	15	A2	43
11	53	42	35	D1	EE
01	18	4A	7F	AE	11
28	FF	0A	AA	46	712
BA	21	94	B3	34	B5

Grafika i wizualizacja

- rodzaje obrazów rastrowych

Obrazy rastrowe:

BINARNE



MONOCHROM.



KOLOROWE



INNE:

- multi/hiperspektralne

Grafika i wizualizacja

- obraz binarny

Obraz binarny jest tablicą dwuwymiarową liczb których poszczególne bity reprezentują kolory (0 – pierwszy kolor, 1 – drugi kolor). Przykładowo dla tablicy liczb:

F8	1F
F7	EF
F8	1F

oraz kolorów ◎  i 

Uzyskujemy:

1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1

Grafika i wizualizacja

- obraz binarny

Obraz binarny jest tablicą dwuwymiarową liczb których poszczególne bity reprezentują kolory (0 – pierwszy kolor, 1 – drugi kolor). Przykładowo dla tablicy liczb:

F8	1F
F7	EF
F8	1F

oraz kolorów:  i 

Uzyskujemy:

1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1

Grafika i wizualizacja

- obraz binarny

Obraz binarny:

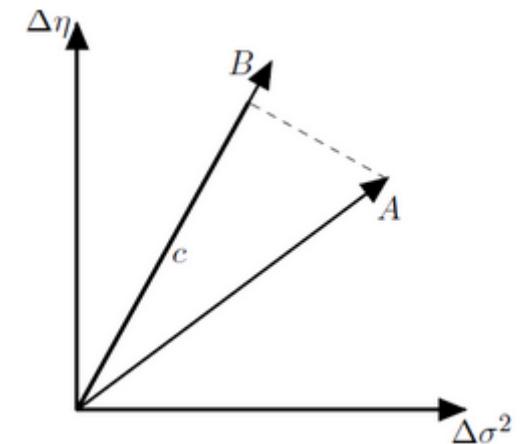
- Zajmują niewiele miejsca w pamięci. Dla obrazu o rozmiarach $n \times m$ (na przykład 640×480) ilość potrzebnej pamięci to:

$$Ms = nm/8 = 640 \cdot 480 / 8 = 38400B$$

*dzielimy przez 8 bo na jednym bajcie przechowywanych jest 8 pikseli

Powyższy wzór nie bierze pod uwagę ilości pamięci potrzebnej do zapamiętania definicji kolorów.

- Na ogół dobrze kompresują się bezstratnie.
- Wykorzystywane są do przechowywania obrazów dwukolorowych.



Grafika i wizualizacja

-

Obraz monochromatyczny jest tablicą dwuwymiarową liczb reprezentujących różne poziomy jasności jednej barwy. Przykładowo dla poziomów jasności przykładowych kolorów otrzymamy:



FA	86	22	43
C8	2D	69	3B
D3	B4	5A	E6
6D	6B	0B	E1



FA	86	22	43
C8	2D	69	3B
D3	B4	5A	E6
6D	6B	0B	E1

Grafika i wizualizacja

- obraz monochromatyczny

Obraz momochromatyczny:

Dla obrazu o rozmiarach $n \times m$ (na przykład 640×480) i k bitach (8 bitach) przeznaczonych do zapisu kolorów, ilość potrzebnej pamięci to:

$$Ms = n \cdot m \cdot k / 8 = 640 \cdot 480 \cdot 8 / 8 = 307200B.$$

Liczba dostępnych kolorów przy k -bitowym zapisie (tutaj 8-bitowym) wynosi 2^k czyli 2^8 czyli 256.

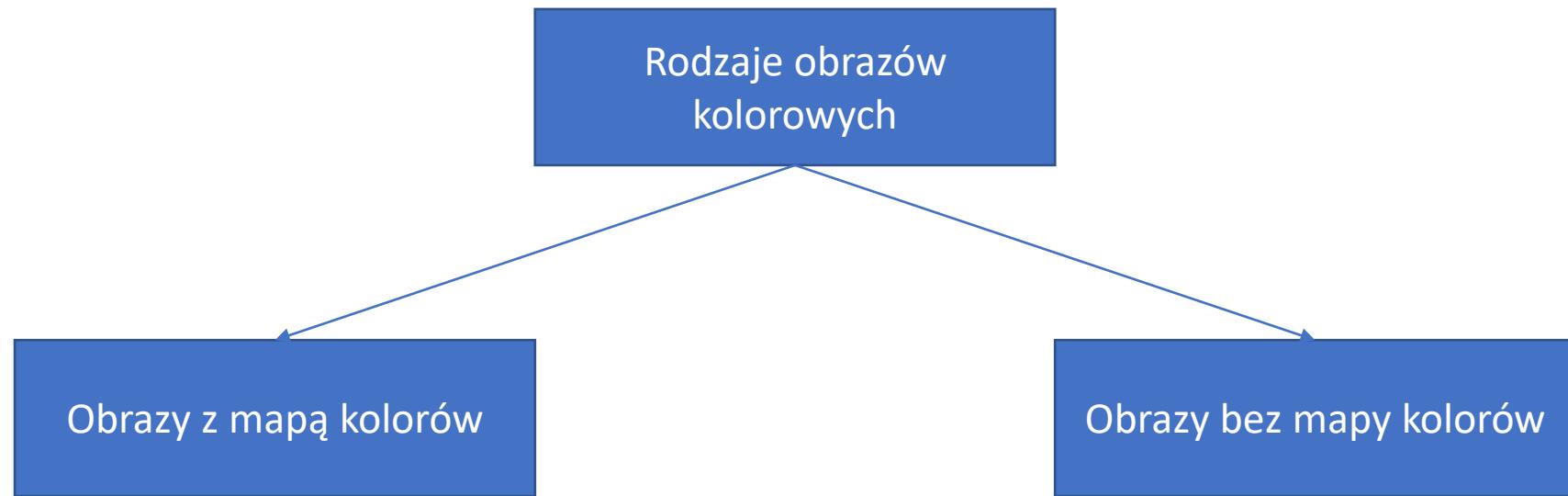
Przykłady obrazów monochromatycznych:



Grafika i wizualizacja

- obrazy kolorowe

podział



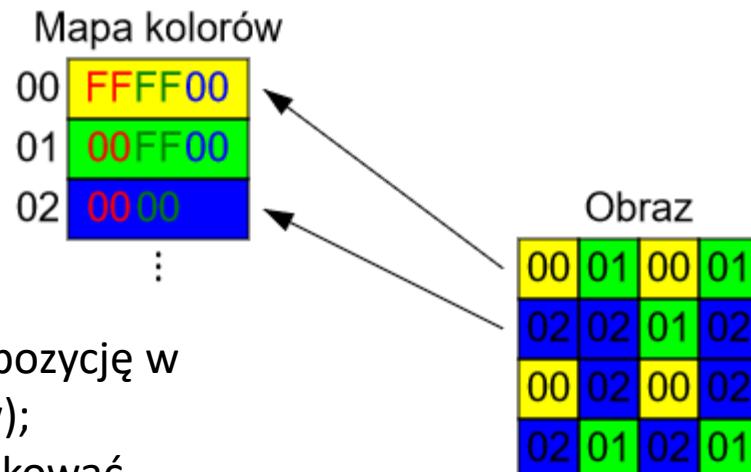
Grafika i wizualizacja

- obraz kolorowy z mapą kolorów

Obraz kolorowy z mapą kolorów jest tablicą dwuwymiarową liczb, które są numerami kolorów określonych w tablicy zwanej mapą kolorów. Przykładowo dla mapy kolorów:

-  ,  ,  , ...

uzyskuje się obraz:



*Wartości obrazu wskazują na pozycję w tablicy kolorów (mapie kolorów);
Dzięki temu można nieco zredukować wielkość obrazu, tym bardziej im częściej kolory się powtarzają

W przypadku zmiany mapy kolorów zmienia się nie tyle obraz co jego wygląd – choć finalnie można twierdzić, że zmienia się obraz.

Grafika i wizualizacja

- obraz kolorowy z mapą kolorów

Obraz kolorowy z mapą kolorów:

Dla obrazu o rozmiarach $n \times m$ (na przykład 640×480) i k bitach (8 bitach) przeznaczonych do zapisu kolorów, ilość potrzebnej pamięci to:

$$Ms = n \cdot m \cdot k \cdot 8 = 640 \cdot 480 \cdot 8 / 8 = 307200B.$$

Liczba dostępnych kolorów przy k -bitowym zapisie (tutaj 8-bitowym) wynosi 2^k czyli 2^8 czyli 256.

Rozmiar mapy kolorów przy l -bitowym zapisie (np. kolor 24-bitowy -> $l=24$) wynosi:

$$Mm = 2^k \cdot l / 8 = 2^8 \cdot 24 / 8 = 768B .$$

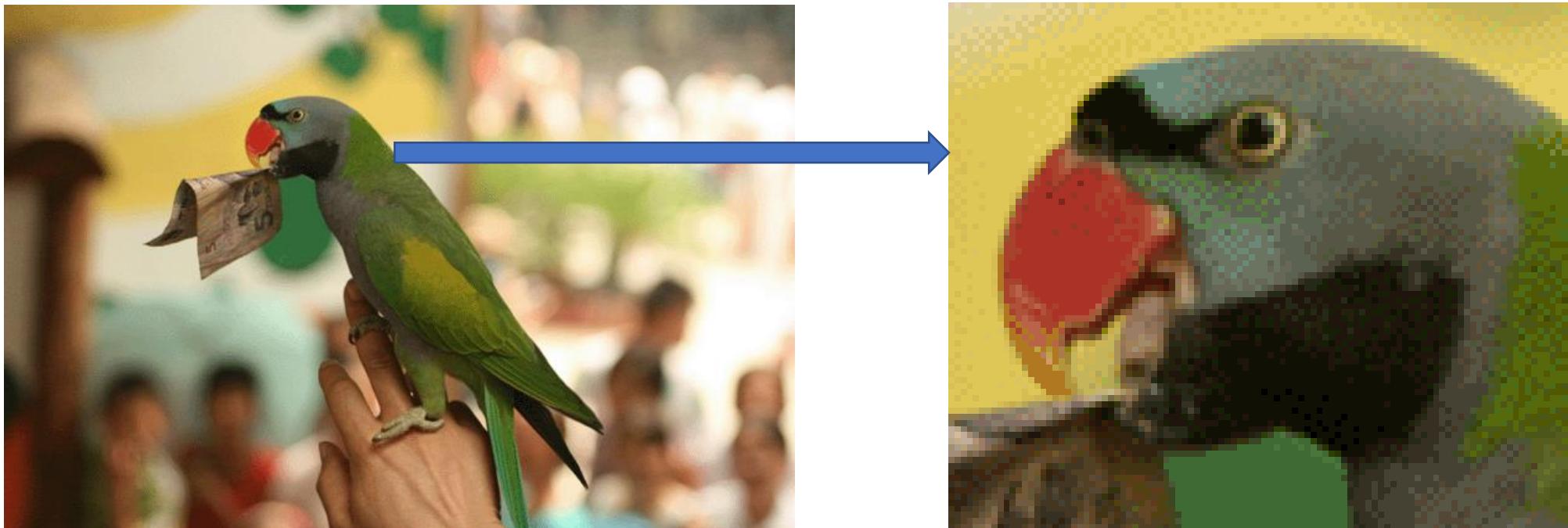
// /8, aby uzyskać liczbę bajtów B

// bity oznaczamy małą literą b

Grafika i wizualizacja

- obraz kolorowy z mapą kolorów

Przykład obrazu z mapą kolorów (w przybliżeniu widać, że jest to jednak 256 różnych kolorów):



Grafika i wizualizacja

- obraz kolorowy bez mapy kolorów

Obraz kolorowy (bez mapy kolorów) jest tablicą dwuwymiarową liczb, które bezpośrednio definiują kolory.

Oznacza to, że wiele wartości kolorów się powtarza.

47 89 63	FA 8C 10	21 AF 51
10 DC 12	00 63 09	DC 85 9D
32 48 D9	7E 04 80	8E 44 D7

Grafika i wizualizacja

- obraz kolorowy bez mapy kolorów

Obraz momochromatyczny:

Dla obrazu o rozmiarach $n \times m$ (na przykład 640x480) i k bitach (24 bitach) przeznaczonych do zapisu kolorów, ilość potrzebnej pamięci to:

$$Ms = n \cdot m \cdot k / 8 = 640 \cdot 480 \cdot 24 / 8 = 921600B.$$

Liczba dostępnych kolorów przy k -bitowym zapisie (tutaj 24-bitowym) wynosi 2^k czyli 2^{24} czyli 16 777 216.

Grafika i wizualizacja

-

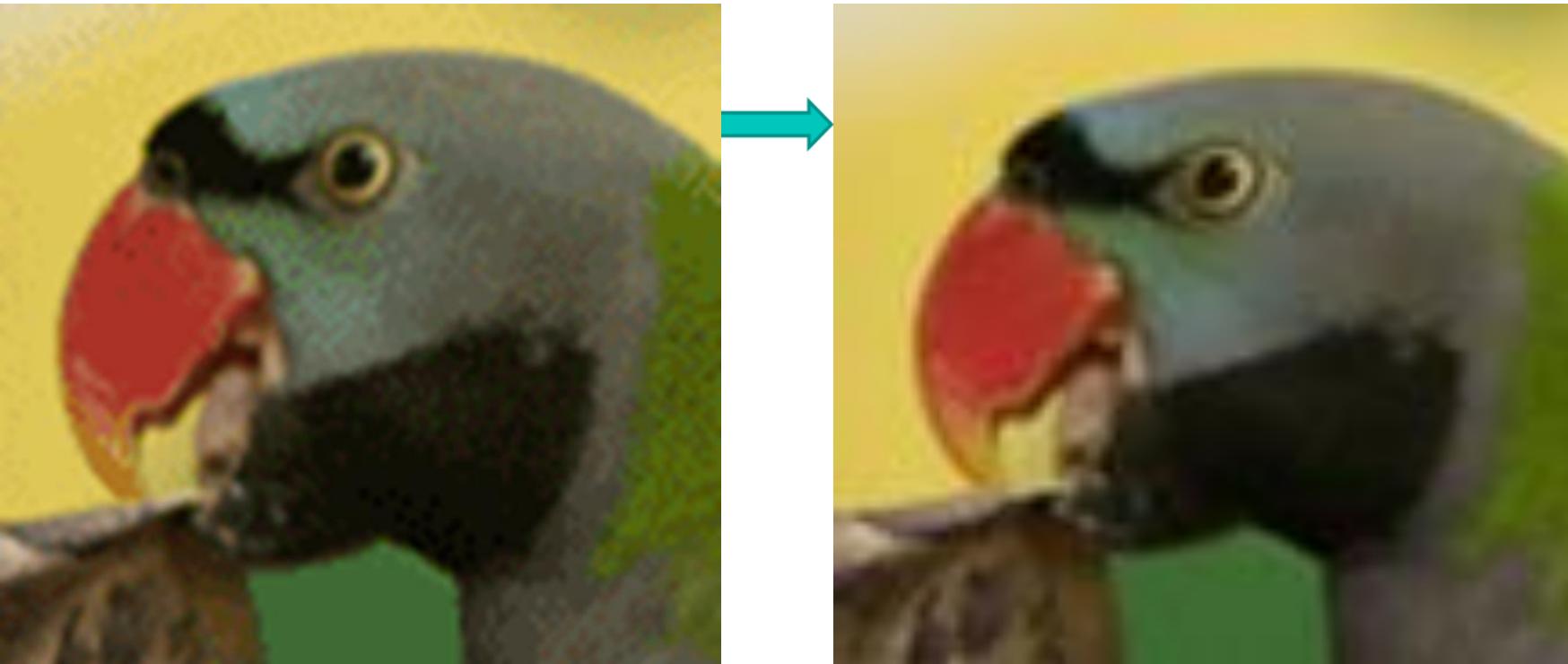
Przykład obrazu bez mapy kolorów (kolory bezpośrednio definiują piksele, tu kolorów jest jednak 17mln – co widać porównując z obrazem z mapą 256 kolorów):



Grafika i wizualizacja

-

Porównanie obrazu kolorowego 17mln (po prawej) z obrazem z mapą 256 kolorów (po lewej):



Format BMP

Format pliku BMP (plik obrazu bitmapowego / format pliku mapy bitowej niezależnej od urządzenia (DIB) / mapa bitowa) - format pliku obrazu grafiki rastrowej używany do przechowywania cyfrowych obrazów bitmapowych , niezależnie od urządzenia wyświetlającego (takiego jak karta graficzna), zwłaszcza w systemach operacyjnych Microsoft Windows i OS/2.

Format BMP

Zawiera w sobie prostą kompresję bezstratną RLE (która nie musi być użyta), informację o użytych kolorach. Obsługuje tryby RGB oraz RGBA.

Format pliku BMP umożliwia przechowywanie dwuwymiarowych obrazów cyfrowych, zarówno monochromatycznych , jak i kolorowych, w różnych głębiach kolorów i opcjonalnie z kompresją danych , kanałami alfa i profilami kolorów .

Specyfikacja Windows Metafile (WMF) obejmuje format plików BMP.

Format BMP

Struktura pliku

Nazwa struktury	Opcjonalna?	Rozmiar	Przeznaczenie	Komentarz
Nagłówek pliku	Nie	14 Bajtów	Przechowuje najważniejsze informacje o pliku.	Jest zbędny po załadowaniu obrazka do pamięci.
Nagłówek DIB	Nie	Różny rozmiar (istnieje 7 różnych wersji tego nagłówka)	Przechowuje najważniejsze informacje o obrazie i definiuje format piksela.	Występuje bezpośrednio po nagłówku pliku.
Dodatkowa maska bitowa	Tak	3 lub 4 DWORD ^[1] (12 lub 16 bajtów)	Definiuje format piksela.	Występuje tylko wtedy gdy nagłówek DIB jest w formacie BITMAPINFOHEADER.
Tablica kolorów	Zależy od liczby kolorów	Różny rozmiar	Definiuje kolory użyte przez obraz (tablicę pikseli)	Występuje wtedy gdy głębia kolorów ≤ 8
Gap1	Tak	Różny rozmiar	Struktura dostosowująca	Artefakty znajdujące się, aż do offsetu pliku dla tablicy pikseli znajdującego się w nagłówku pliku.
Tablica pikseli	Nie	Różny rozmiar	Definiuje wartości poszczególnych pikseli	Format piksela zdefiniowany jest w nagłówku DIB lub w dodatkowej masce bitowej. Każdy wiersz tablicy pikseli jest wyrównany do wielokrotności 4 bajtów.
Gap2	Tak	Różny rozmiar	Struktura dostosowująca	Artefakty znajdujące się, aż do offsetu danych ICC znajdującego się w nagłówku DIB.
Profil kolorów ICC	Tak	Różny rozmiar	Definiuje profil kolorów dla systemu zarządzania kolorem	Może zawierać ścieżkę do zewnętrznego pliku przechowującego profil kolorów. Po załadowaniu do pamięci jako "nie-spakowany DIB", znajduje się między tablicą kolorów a Gap1 ^[2] .

Format BMP

Nagłówek pliku

Blok znajduje się na początku pliku i jest używany do jego identyfikacji. Używa się go, aby sprawdzić czy jest to plik BMP i czy nie jest on uszkodzony. Pierwszymi dwoma bajtami w formacie BMP są znaki "B", oraz "M" w kodowaniu ASCII. Wszystkie wartości całkowite są przechowywane w formacie little-endian, tj. najmniej znaczący bajt umieszczony jako pierwszy.

Offset#	Rozmiar	Przeznaczenie
0000 _h	2 bajty	początek nagłówka używany do identyfikacji pliku BMP, DIB. Najczęściej wynosi 0x42 0x4D w systemie szesnastkowym , odpowiada to znakom BM w kodzie ASCII. Możliwe wartości: <ul style="list-style-type: none">• BM – Windows 3.1x, 95, NT, ... itd.• BA – OS/2 struktura tablicy bitmapy• CI – OS/2 struktura kolorowej ikony• &CP – OS/2 stały wskaźnik koloru• IC – OS/2 struktura ikony• *PT – OS/2 wskaźnik
0002 _h	4 bajty	Oznacza rozmiar całego pliku w bajtach
0006 _h	2 bajty	zarezerwowany; zależy od aplikacji, która tworzy obraz
0008 _h	2 bajty	zarezerwowany; zależy od aplikacji, która tworzy obraz
000A _h	4 bajty	Offset , tzn. początkowy adres bitów w tablicy pikseli.

Format BMP

Nagłówek DIB (Device-Independent Bitmap)

Ta struktura przekazuje aplikacji informacje o obrazie, który będzie wyświetlany. Nagłówek jest dopasowany do wewnętrznych mechanizmów systemów Windows i OS/2 i posiada 7 różnych wariantów. Wszystkie z nich zawierają pole DWORD (32 bitowe), określające rozmiar nagłówka, dzięki niemu można określić typ nagłówka. Duża liczba wersji nagłówka jest spowodowana tym, że Microsoft rozszerzał informacje w nim zawarte kilka razy. Format odczytywanych plików zależy od wersji systemu, na którym uruchomiona jest aplikacja.

Rozmiar	Nazwa nagłówka	Wspierane systemy operacyjne	Dodane funkcjonalności (rosnąco)	w plikach stworzonych przez program
12	BITMAPCOREHEADER OS21XBITMAPHEADER	OS/2 i wszystkie wersje Windows począwszy od Windows 3.0		
64	BITMAPCOREHEADER2 OS22XBITMAPHEADER	OS/2	Dodaje pół-tonowanie. Dodaje kompresje RLE oraz Huffman 1D.	
40	BITMAPINFOHEADER	wszystkie wersje Windows począwszy od Windows 3.0	Likwiduje kompresje RLE-24 oraz Huffman 1D. Dodaje obsługę 16bpp oraz 32bpp pikseli. Dodaje opcjonalną maskę bitów RGB.	Adobe Photoshop
52	BITMAPV2INFOHEADER	Brak danych.	Usuwa opcjonalną maskę bitów RGB. Dodaje obowiązkową maskę bitów RGB.	
56	BITMAPV3INFOHEADER	Brak danych.	Dodaje obowiązkową maskę bitową dla kanału alfa.	Adobe Photoshop
108	BITMAPV4HEADER	wszystkie wersje Windows począwszy od 95/NT4	Dodaje typ przestrzeni barw oraz korekcje gamma.	
124	BITMAPV5HEADER	Windows 98/2000 i nowsze.	Dodaje profile kolorów ICC.	

Format BMP

Nagłówek DIB

Nowe, rozszerzone nagłówki mogą być używane z niektórymi funkcjami GDI i zapewniają większą funkcjonalność. Odkąd GDI wspomaga funkcję ładowania bitmap, większość aplikacji korzysta z tych funkcji. Jedną z konsekwencji jest to, że format odczytywanych plików zależy od wersji systemu, na którym uruchomiona jest aplikacja.

Format BMP

Przykładowy nagłówek DIB

Nagłówek BITMAPINFOHEADER składa się z 40 bajtów

Offset względem początku pliku	Rozmiar w bajtach	Nazwa	Opis zawartości
14	4	biSize	Wielkość nagłówka informacyjnego. Długość stąd do końca nagłówka – 40, czasem mniej
18	4	biWidth	Szerokość obrazu w pikselach
22	4	biHeight	Wysokość obrazu w pikselach
26	2	biPlanes	Liczba warstw kolorów, zwykle 1
28	2	biBitCount	Liczba bitów na piksel
30	4	biCompression	Algorytm kompresji
34	4	biSizelImage	Rozmiar samego rysunku
38	4	biXPelsPerMeter	Rozdzielcość pozioma
42	4	biYPelsPerMeter	Rozdzielcość pionowa
46	4	biClrUsed	Liczba kolorów w paletie
50	1	biClrImportant	Liczba ważnych kolorów w paletie (gdy 0 to wszystkie są ważne); pole to stosuje się przy animacji bitmap poprzez rotację kolorów. Oznacza od którego koloru paleta ma być podmieniana.
51	1	biClrRotation	Flaga sygnalizująca czy ma następować rotacja palety (domyślnie 0-brak rotacji). Jeśli flaga jest ustawiona na 1 następuje podmiana palety.
52	2	biReserved	

Format BMP

Paleta kolorów

Po części nagłówkowej pliku BMP zazwyczaj znajduje się paleta. Można tu wyodrębnić 2 przypadki:

- 1) W trybach 8-bitowych paleta zbudowana jest z opisów kolorów, gdzie każdy kolor opisany jest za pomocą 4 bajtów (B, G, R, nieużywany lub alpha). Każdą ze składowych koloru należy podzielić przez 4.
- 2) W trybach o większej liczbie kolorów niż 256 paleta nie występuje, a kolor piksela w obrazie zapisywany jest przy pomocy numeru kolejnego koloru który jest wyliczany ze wzoru:

$$\text{Nr Koloru} = R + 256 * G + 65536 * B,$$

gdzie:

R – wartość składowej czerwonej

G – wartość składowej zielonej

B – wartość składowej niebieskiej

Format BMP

Tabela pikseli

Piksele obrazu zapisywane są w liniach obrazu od dołu do góry.

W obrazach TrueColor punkty zapisywane są w kolejności B, G, R (trzy bajty), w obrazach o mniejszej liczbie kolorów zapisywane są one na mniejszej ilości bajtów – przykładowo 16-bitowa bitmapa (High Color) zapisuje na dwóch bajtach [5bitów B][6bitów G][5bitów R].

Dla plików z liczbą kolorów mieszczącą się w liczbie ośmiobitowej, kolor zapisywany jest jako numer koloru (wskaźnik) w palecie.

Należy zwrócić uwagę, że (ze względów historycznych) w wierszu danych pliku BMP jest zawsze zapisana wielokrotność 4 bajtów. Jeśli wiersz danych ma długość (w bajtach) inną, niż podzielna przez 4, to dopisuje się bajty o wartości 0, tak aby w danym wierszu ilość bajtów była wielokrotnością 4!

*co za marnotrawstwo miejsca

Format GIF

GIF (ang. Graphics Interchange Format) – format pliku graficznego z kompresją bezstratną (opis niżej), stworzony w 1987 roku przez firmę CompuServe.

Pliki tego typu są powszechnie używane na stronach WWW, gdyż pozwalają na tworzenie prostych animacji ze zdefiniowanym kolorem tła jako przezroczystym i określonym opóźnieniem przy odtwarzaniu poszczególnych klatek animacji.

Format GIF

Panuje powszechnie przekonanie, iż format GIF pozwala jedynie na użycie do 256 kolorów w jednym pliku obrazu, jednak w rzeczywistości odnosi się to do liczby kolorów mogących występować w pojedynczym bloku obrazu, a nie w całym pliku zawierającym obraz. Dlatego też, używając większej liczby bloków, możliwe jest zastosowanie formatu GIF do przechowywania na przykład fotografii bez utraty jakości / liczby kolorów.

Format GIF

Przekonanie o ww. ograniczeniu najprawdopodobniej wynika z tego, że ogromna większość programów do obróbki grafiki rastrowej potrafi obsługiwać jedynie jeden blok na cały obraz (a ówczesny sprzęt potrafił wyświetlać właśnie maksymalnie 256 kolorów na ekranie) albo błędnie interpretuje wiele bloków jako poszczególne klatki animowanego GIF-a.

Ograniczenie to wynika więc z niepełnej obsługi formatu, a nie z ograniczeń istniejących w nim samym. W pliku GIF, zawierającym kilka bloków obrazu, każdy z nich może zawierać 256 kolorów albo 255 kolorów i jeden kolor „przezroczysty”

Format GIF

Ze względu na to, iż do kompresji w formacie GIF może być używany algorytm LZW, na którym ciążyły patenty w kilku krajach świata, w 1995 został opracowany konkurencyjny format PNG używający do kompresji algorytmu deflate.

Według organizacji Software Freedom Law Center 1 października 2006 wygasły wszystkie ograniczenia patentowe dotyczące formatu GIF i format ten można uznać za uwolniony.

Format JPG

JPEG (ang. Joint Photographic Experts Group) to algorytm stratnej kompresji grafiki rastrowej, wykorzystany w formacie plików graficznych o tej samej nazwie. Istnieje kilka sporo modyfikacji algorytmu JPEG, ale głównie używany algorytm kompresji JPEG można przedstawić za pomocą kilku kroków:

- Konwersja z przestrzeni RGB do przestrzeni YCbCr (luminancja i dwie warstwy chrominancji)
- Ewentualna redukcja chrominancji (Chroma Subsampling)
- Podział obrazu na bloki 8x8 (każda warstwa przetwarzana jest osobno)
- Wykonanie dyskretnej transformaty kosinusowej (DCT)
- Kwantyzacja poprzez zastąpienie wartości zmiennoprzecinkowych, wartościami całkowitymi.
- Ewentualne dzielenie przez macierze kwantyzacji w celu zwiększenia kompresji
- Porządkowanie współczynników DCT w sposób liniowy poprzez zyzkakowanie
- Kompresja bezstratna danych algorytmem Huffmana.

Format JPG

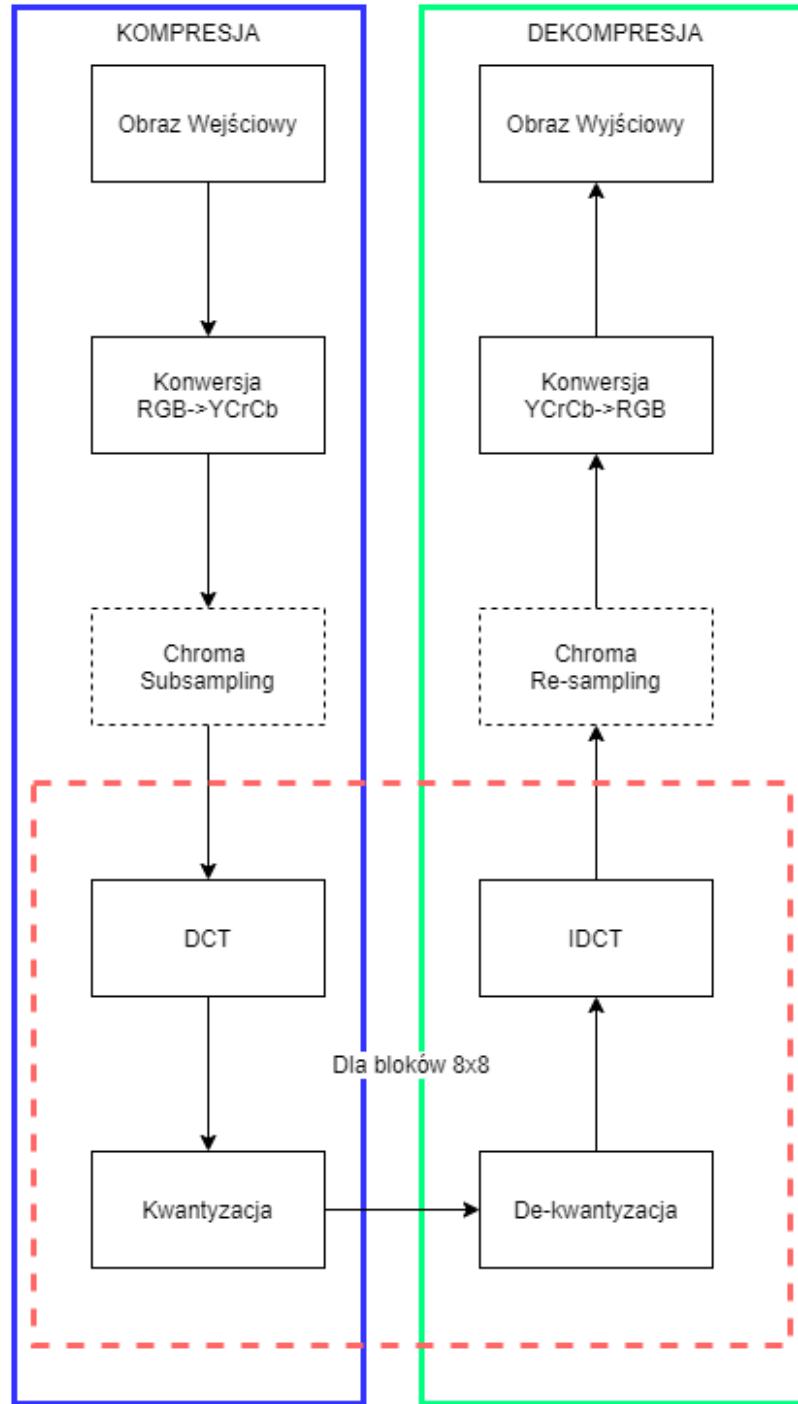
W przestrzeni Internetu można znaleźć obecnie dobre materiały, które ten algorytm wyjaśniają:

JPEG 'files' & Colour (JPEG Pt1)- Computerphile:

[Część 1 \[ENG\]](#)

JPEG DCT, Discrete Cosine Transform (JPEG Pt2)- Computerphile:

[Część 2 \[ENG\]](#).



Format JPG

Konwersja z przestrzeni RGB do przestrzeni YCbCr (luminancja i dwie warstwy chrominancji)

Na początku należy zmienić przestrzeń koloru w jakiej reprezentowany jest obraz. Jest to model przestrzeni kolorów używany do cyfrowego przesyłania oraz przechowywania obrazów i wideo.*

Wykorzystuje do tego trzy typy danych:

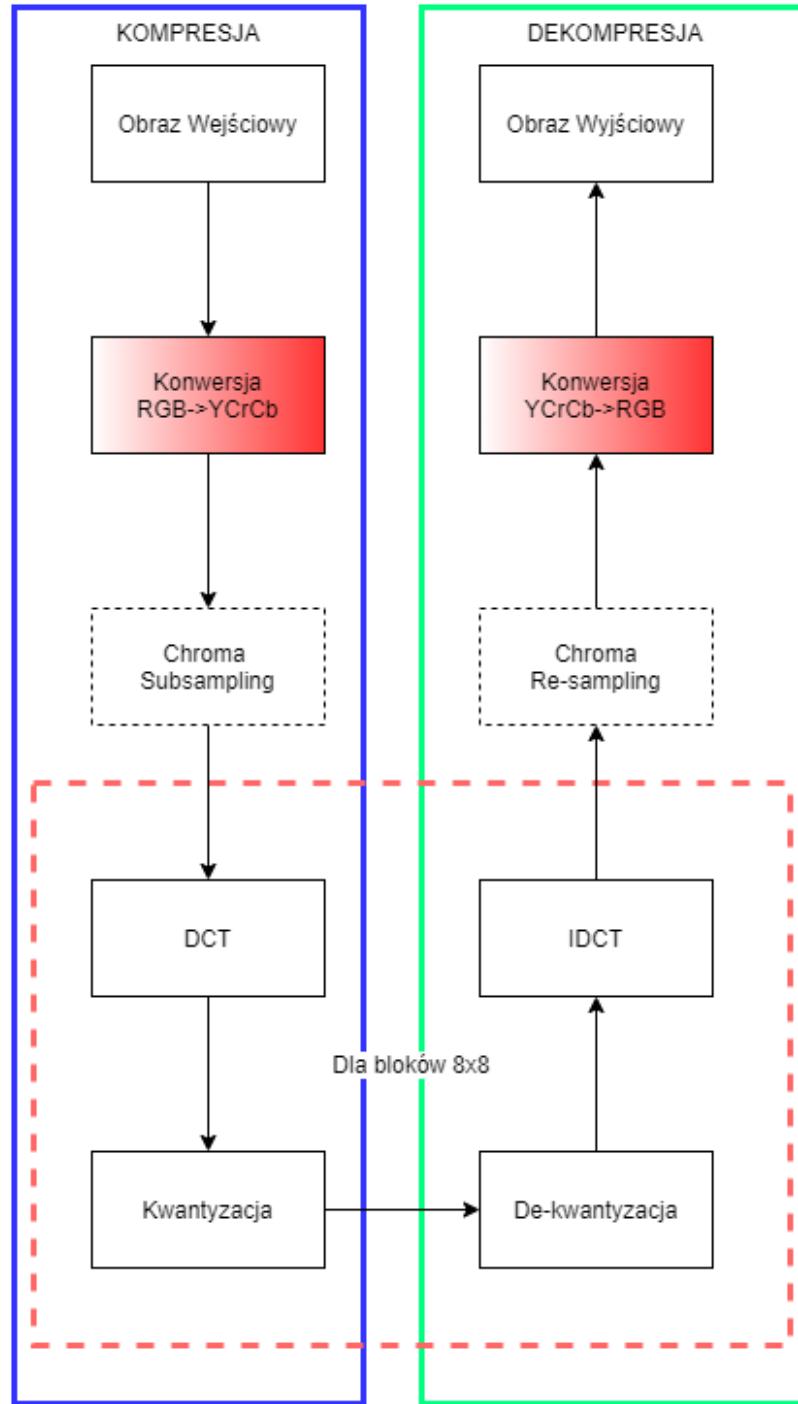
Y – składową luminancji,

Cb – składową różnicową chrominancji Y-B, stanowiącą różnicę między luminancją a niebieskim,

Cr – składową chrominancji Y-R, stanowiącą różnicę między luminancją a czerwonym.

Kolor zielony jest uzyskiwany na podstawie tych trzech wartości.

*Właściwy format to YCbCr; tutaj zamienione składowe ze względu na przykład w Python/OpenCV

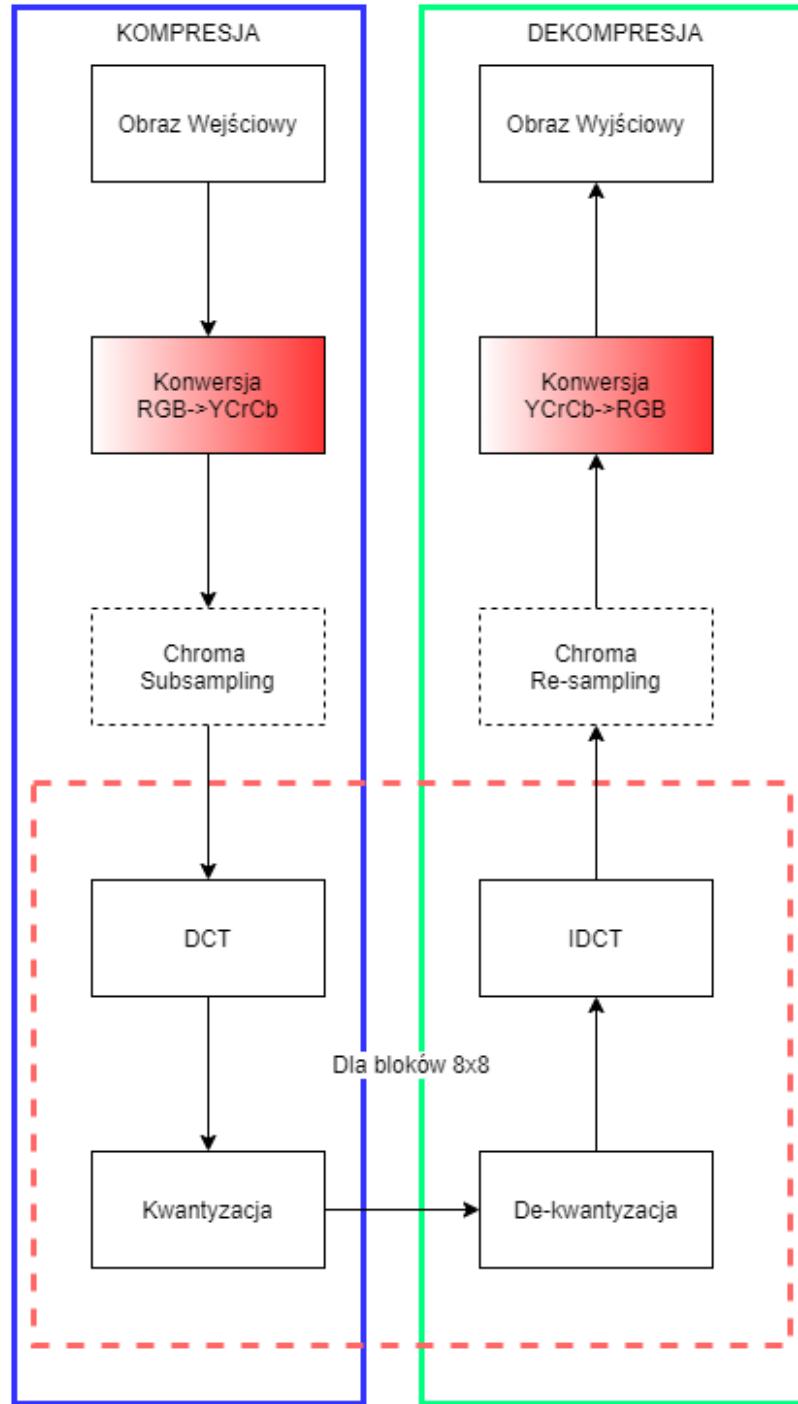


Format JPG

Do przejścia pomiędzy tymi przestrzeniami można wykorzystać OpenCV. Pamiętać należy jednak, że domyślnie obraz dla OpenCV jest zapisany albo w typie uint8 (lub float).

```
YCrCb=cv2.cvtColor(RGB,cv2.COLOR_RGB2YCrCb).astype(int)
```

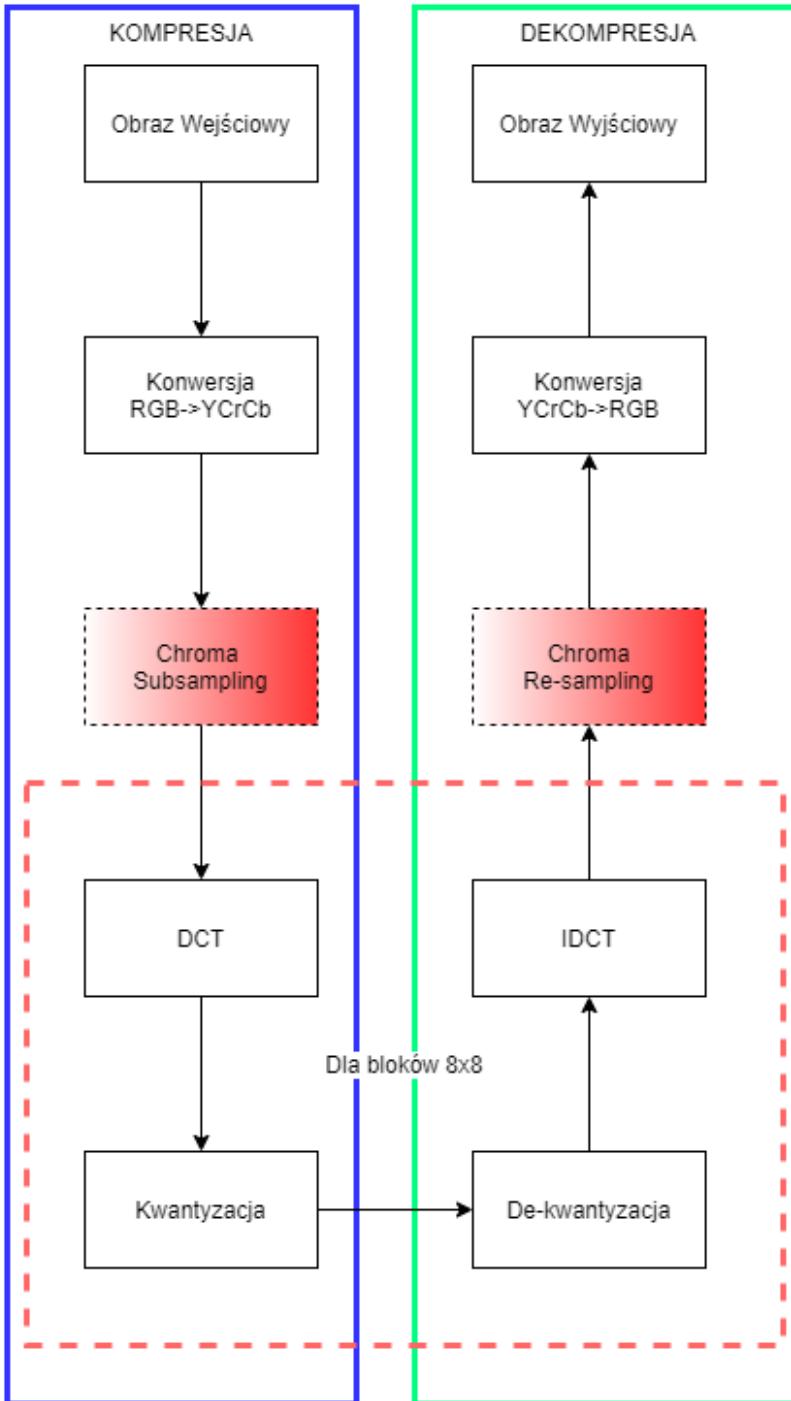
```
RGB=cv2.cvtColor(YCrCb.astype(np.uint8),cv2.COLOR_YCrCb2RGB)
```



Format JPEG

Redukcja chrominancji (Chroma Subsampling)

Kolejnym etapem jest redukcja chrominancji (ang. Chroma Subsampling). Jest to jedna z operacji kompresji polegająca na zmniejszeniu informacji na temat koloru w naszym obrazie. Wykorzystuje się tutaj fakt, że ludzkie oko jest bardziej wrażliwe na zmianę jasności (Luminancja), niż koloru (Chrominancja).

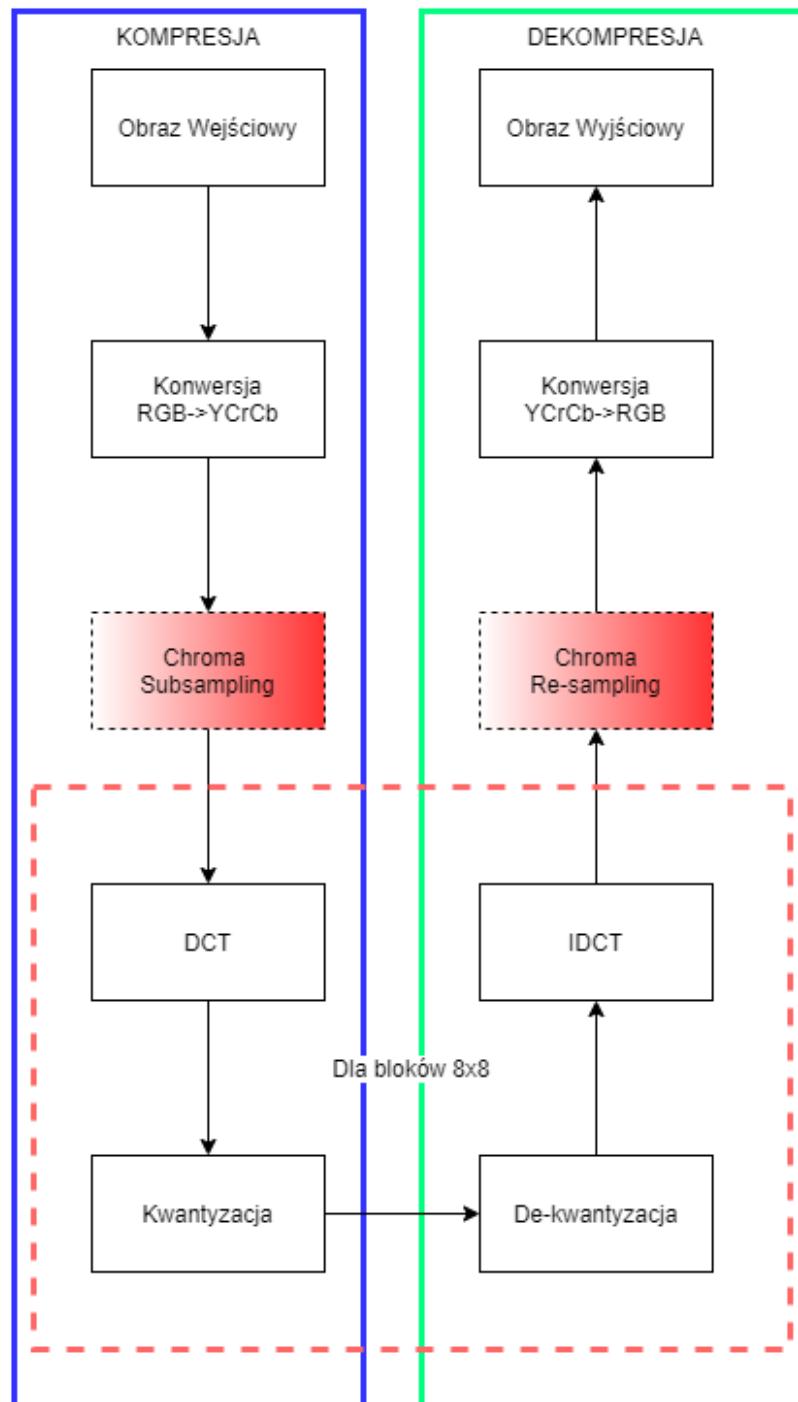
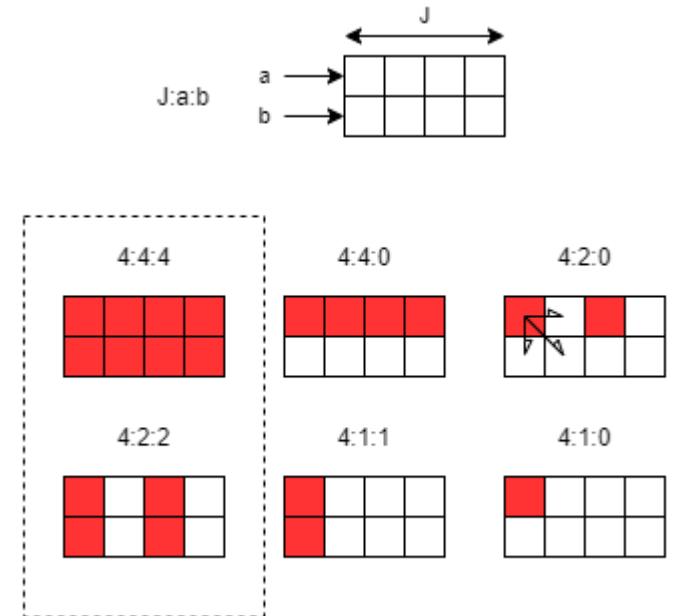


Format JPEG

Redukcja chrominancji (Chroma Subsampling)

Subsampling zatem wykonywany jest na warstwach Cb oraz Cr (dla każdej z nich osobno).

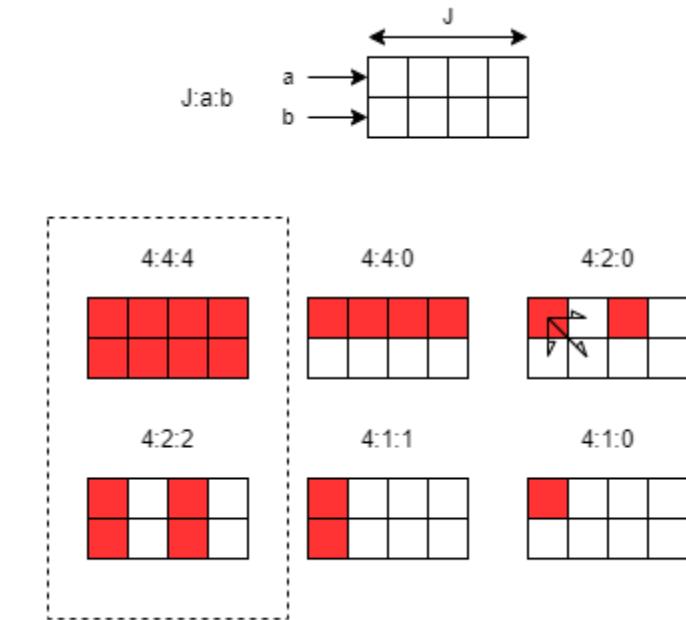
Jego rodzaj opisywany jest przez 3 wartości opisane w ciągu J:a:b (np. 4:4:4 oraz 4:2:2).



Format JPG

Redukcja chrominancji (Chroma Subsampling)

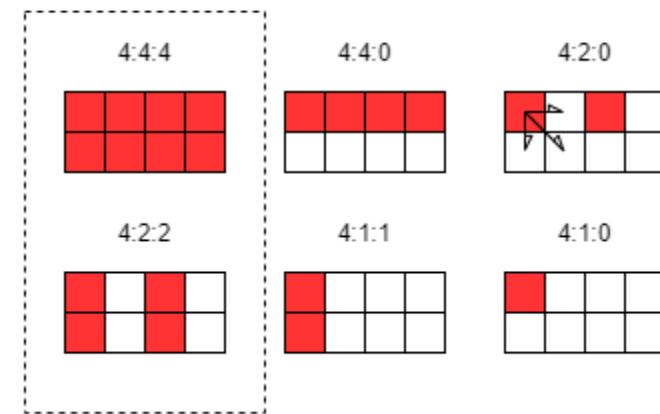
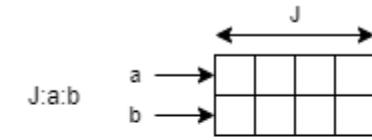
Pierwszy parametr J określa jak dużo pikseli w poziomie będzie analizowanych, zwykle jest to wartość 4. Kolejny parametr a określa ile wartości z pierwszego analizowanego przez nas wiersza będzie zapamiętywanych (pozostałe ulegają pominięciu), natomiast parametr b opisuje liczbę zachowanych wartości w drugim wierszu.



Format JPG

Redukcja chrominancji (Chroma Subsampling)

Na przykładzie graficznym zapisywane zostają wartości zaznaczone jako czerwone kwadraty. Jak widzimy w przypadku 4:4:4 wszystkie wartości zostają zapisane, więc nic nie ulega zmianie. W pozostałych przypadkach zawsze tracimy jakieś informacje.

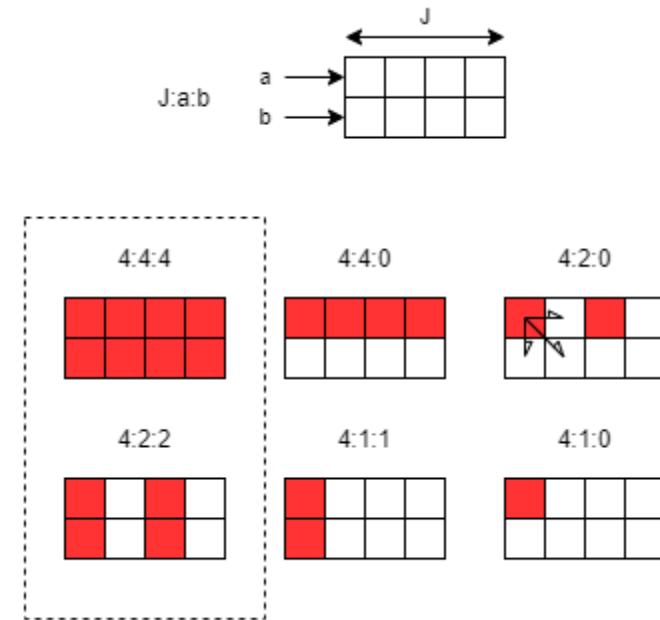


Format JPEG

Redukcja chrominancji (Chroma Subsampling)

Działanie funkcji odwrotnej polega na odtworzeniu całej warstwy naszego obrazu na podstawie zapisanych przez nas wcześniej informacji. Jeżeli nie mamy dostępnej informacji o wartości piksela to zastępujemy (w zależności gdzie się on znajduje) inną najbliższą wartością z tego lub poprzedniego wiersza.

Przykładowo dla 4:4:0 należy uzupełniać drugi wiersz wartościami z pierwszego, dla 4:2:0 jedna wartość będzie propagowana na 4 piksele, dla 4:1:1 nowa wartość będzie zastępowała cały wiersz, a dla 4:1:0 nowa wartość będzie rozchodziła się dla wszystkich 8 pikseli.



Format JPG

Podział obrazu na bloki 8x8 (każda warstwa przetwarzana jest osobno)

Dalsze operacje wykonywane są niezależnie od warstwy na blokach 8x8 (64 wartości) stąd przed kolejnymi krokami należy dokonać takiego podziału lub zaprojektować metody tak, aby wykonywały operacje na takich blokach.

Uwaga techniczna dla niektórych implementacji!

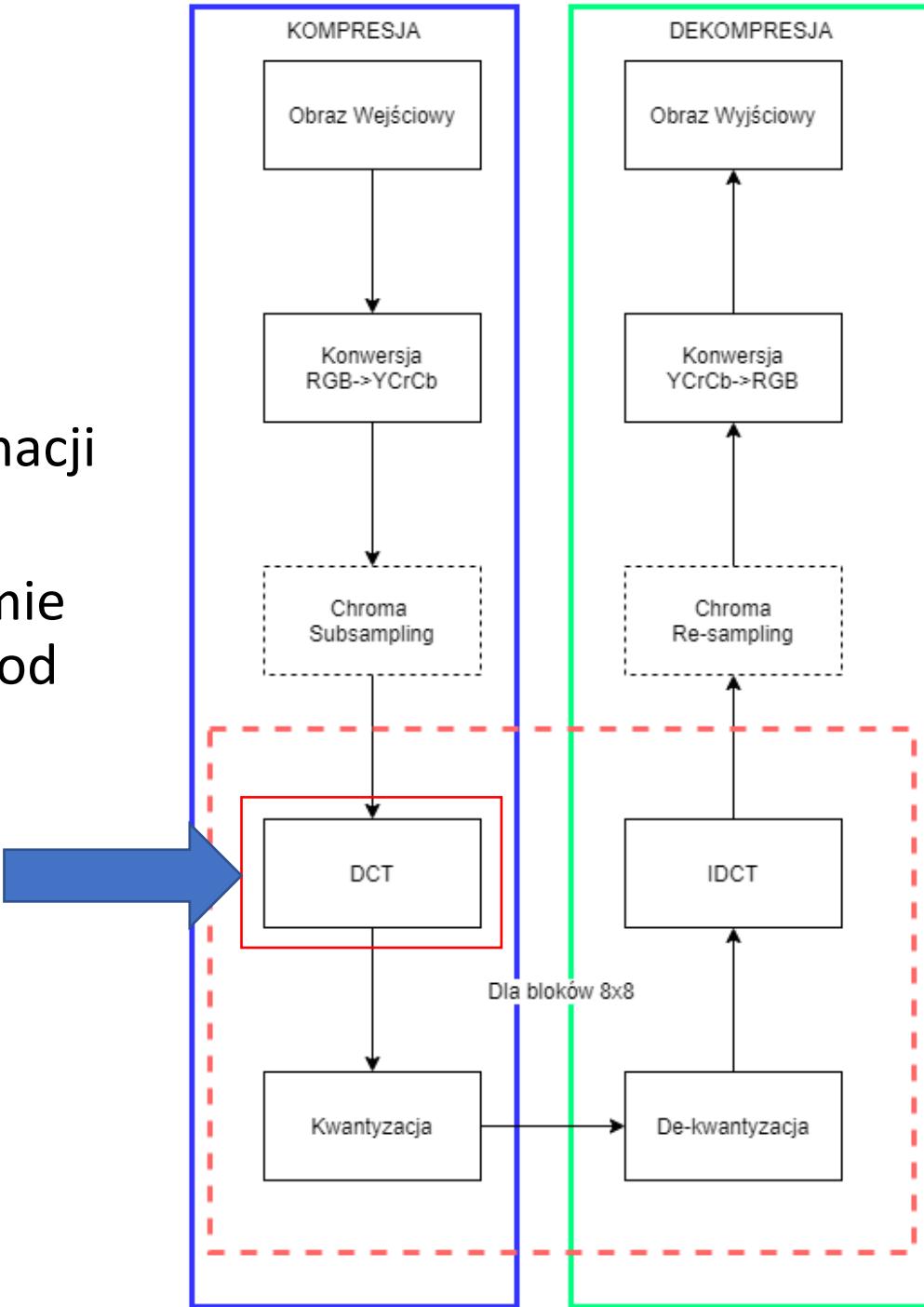
Dyskretna transformacja cosinusowa (obecna w kolejny kroku, już dla bloku) powinna otrzymywać dane wyśrodkowane na 0. W przypadku typów w rodzaju uint8 wartości są wyśrodkowane na 128, dlatego przed operacją należy od wartości każdej warstwy należy odjąć wartość 128 oraz dodać tą wartość po odtworzeniu.

Format JPEG

Wykonanie dyskretnej transformaty kosinusowej (DCT)

Pierwszym etapem kompresji na bloku danych jest wyliczenie współczynników dla dyskretnej transformacji cosinusowej (DCT).

W punkcie (0,0) macierzy DCT (podobnie jak w widmie Fourierowskim) znajduje się składowa stała, a dalej od niej znajdują się składowe o coraz wyższych częstotliwościach.

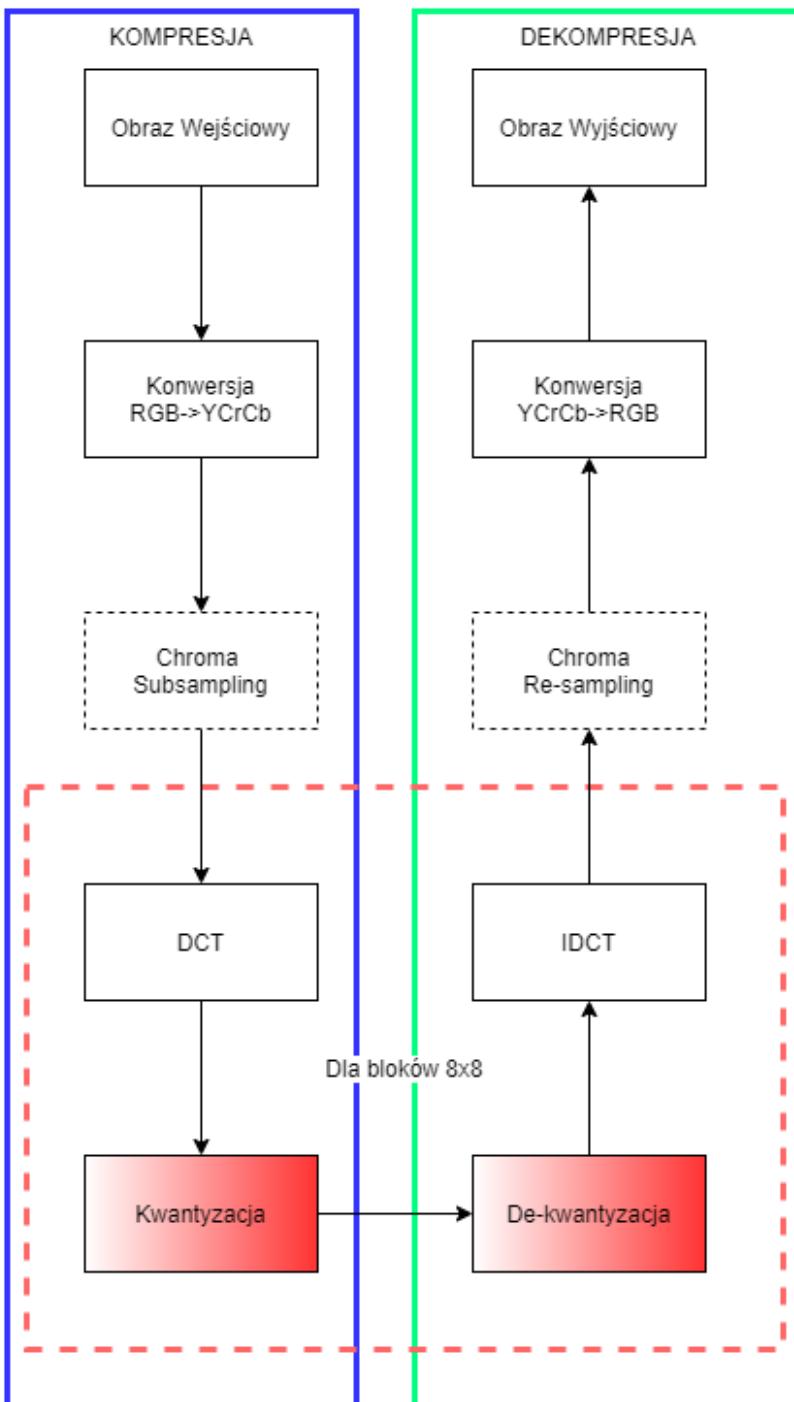


Format JPG

Kwantyzacja poprzez zastąpienie wartości zmiennoprzecinkowych, wartościami całkowitymi.

Ewentualne dzielenie przez macierze kwantyzacji w celu zwiększenia kompresji

Podstawową formą kwantyzacji wartości z DCT jest zwykła zmiana z formatu zmiennoprzecinkowego na liczby całkowite (.astype(int)). Powoduje to pewną utratę informacji, ale jest ona stosunkowo niewielka. W tym wypadku de-kwantyzacja nie będzie wymagała żadnych dodatkowych operacji. Na tym etapie można zobaczyć niewielkie artefakty po rekonstrukcji.



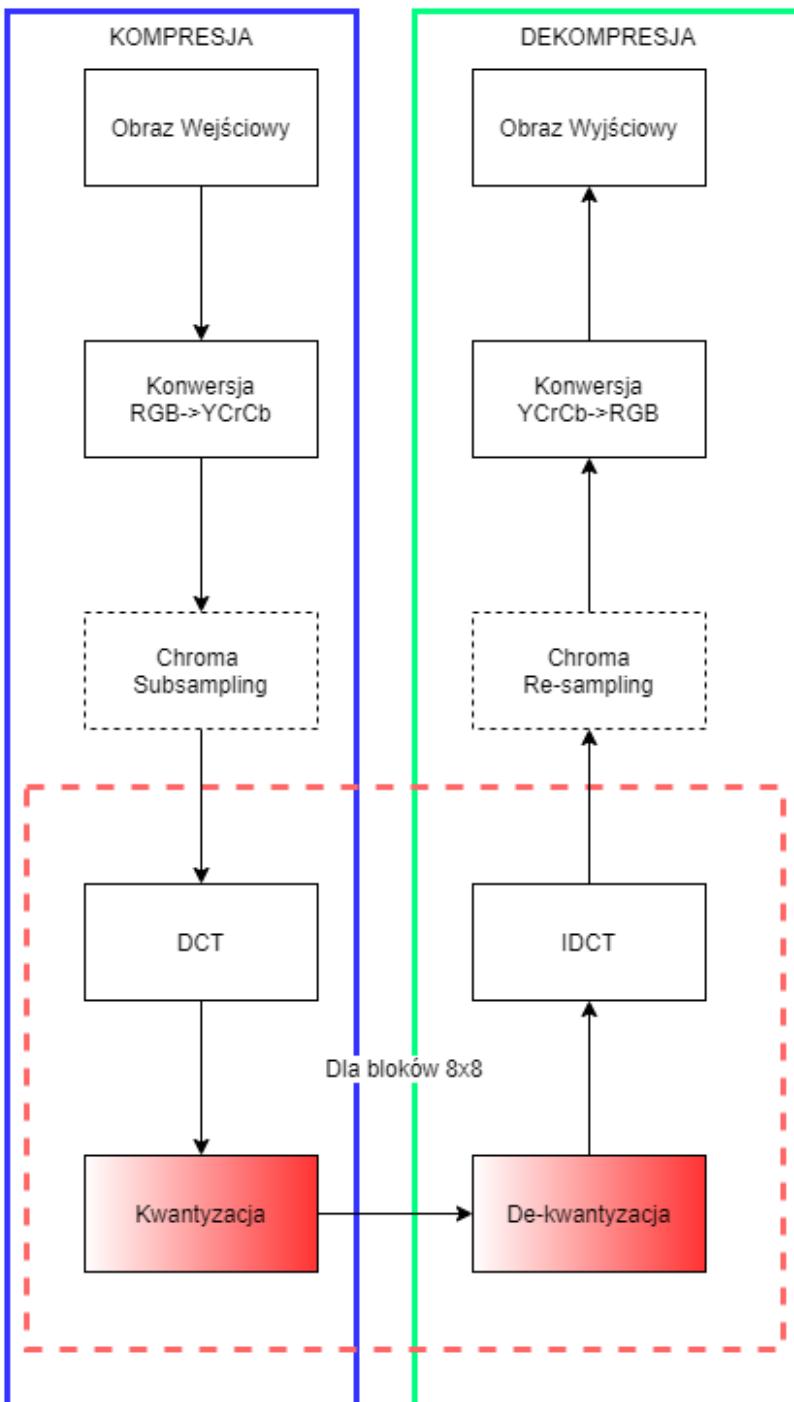
Format JPG

Kwantyzacja poprzez zastąpienie wartości zmiennoprzecinkowych, wartościami całkowitymi.

Ewentualne dzielenie przez macierze kwantyzacji w celu zwiększenia kompresji

Można zaoszczędzić więcej pamięci zmniejszając wpływ komponentów DCT o większych częstotliwościach na rekonstruowany obraz poprzez podzielenie wszystkich wartości dla komponentów DCT przez tzw. tablice kwantyzacji.

Dalej podano dwie takie tablice pochodzące ze standardu JPEG dla jakości 50% jedna pierwsza z nich jest przeznaczona do kompresji Luminancji (Y) druga dla Chrominancji (Cb oraz Cr). Podczas kwantyzacji macierz współczynników DCT dzielona jest przez tablicę kwantyzacji, a podczas de-kwantyzacji wykonywana jest operacja mnożenia przed podaniem ich działaniu IDCT.



Format JPEG

Kwantyzacja poprzez zastąpienie wartości zmiennoprzecinkowych, wartościami całkowitymi.

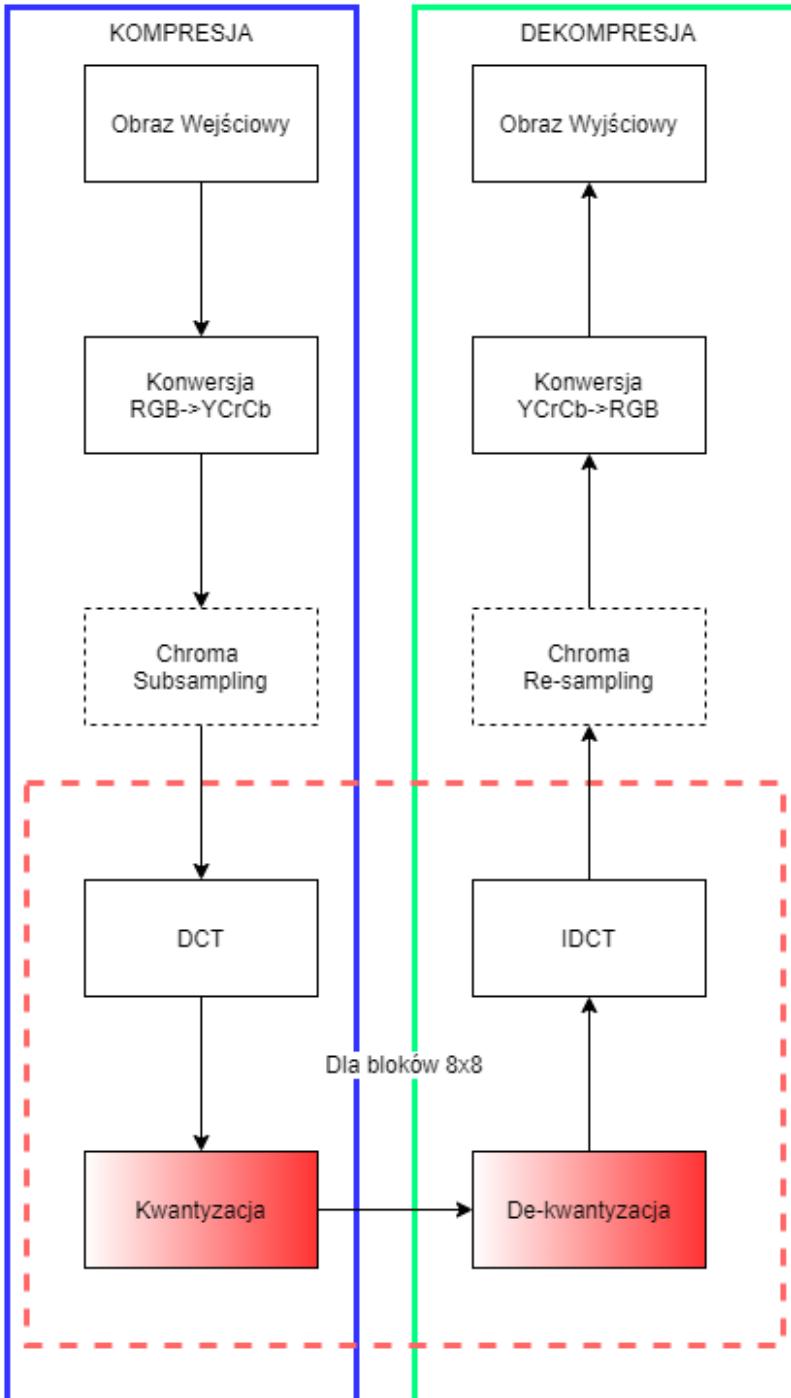
Ewentualne dzielenie przez macierze kwantyzacji w celu zwiększenia kompresji

Tablica kwantyzacji dla luminancji (jakość 50%):

```
QY = np.array([
    [16, 11, 10, 16, 24, 40, 51, 61],
    [12, 12, 14, 19, 26, 58, 60, 55],
    [14, 13, 16, 24, 40, 57, 69, 56],
    [14, 17, 22, 29, 51, 87, 80, 62],
    [18, 22, 37, 56, 68, 109, 103, 77],
    [24, 36, 55, 64, 81, 104, 113, 92],
    [49, 64, 78, 87, 103, 121, 120, 101],
    [72, 92, 95, 98, 112, 100, 103, 99],
])
```

Tablica kwantyzacji dla chrominancji (jakość 50%):

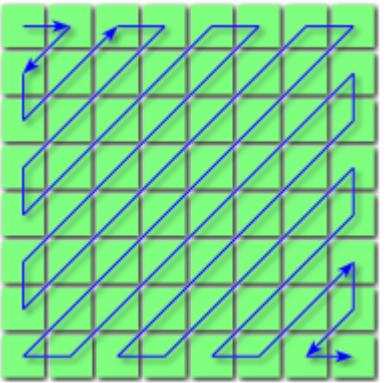
```
QC = np.array([
    [17, 18, 24, 47, 99, 99, 99, 99],
    [18, 21, 26, 66, 99, 99, 99, 99],
    [24, 26, 56, 99, 99, 99, 99, 99],
    [47, 66, 99, 99, 99, 99, 99, 99],
    [99, 99, 99, 99, 99, 99, 99, 99],
    [99, 99, 99, 99, 99, 99, 99, 99],
    [99, 99, 99, 99, 99, 99, 99, 99],
    [99, 99, 99, 99, 99, 99, 99, 99],
])
```



Format JPG

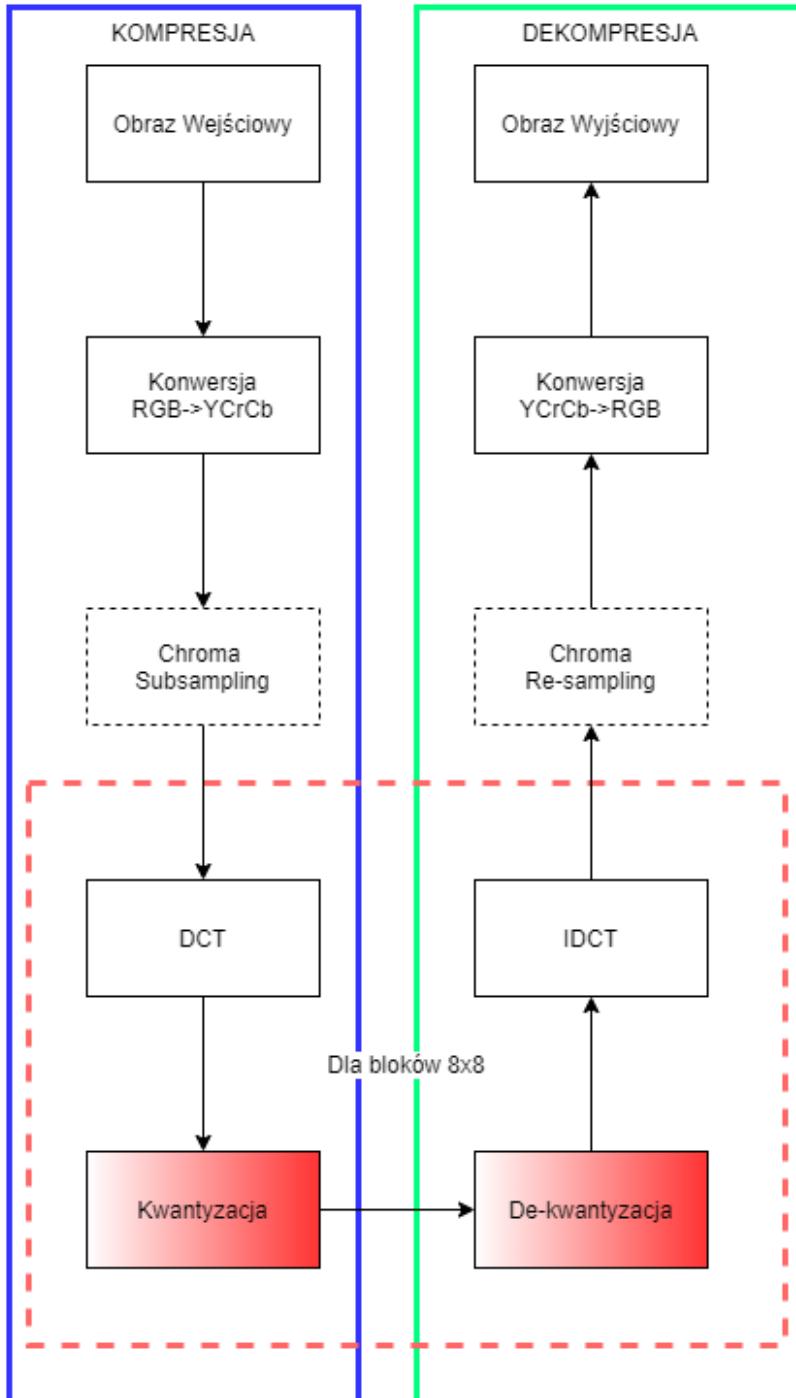
Porządkowanie współczynników DCT w sposób liniowy poprzez zyzkakowanie

Funkcja zyzkakująca (ang. ‘zigzagging’) zamienia macierz 8x8 na wektor 64 elementowy według poniższego schematu:



źródło:https://commons.wikimedia.org/wiki/File:JPEG_ZigZag.svg

Operacja ta ma znaczenie w przygotowaniu danych do kompresji. W wyniku kwantyzacji danych macierz DCT będzie zawierać dużo zer w wyższych partiach częstotliwości. W wyniku zyzkowania utworzony wektor będzie posiadał zera zgrupowane na jego końcu – daje to dobrą podstawę do zastosowania kompresji bezstratnej (Huffman).

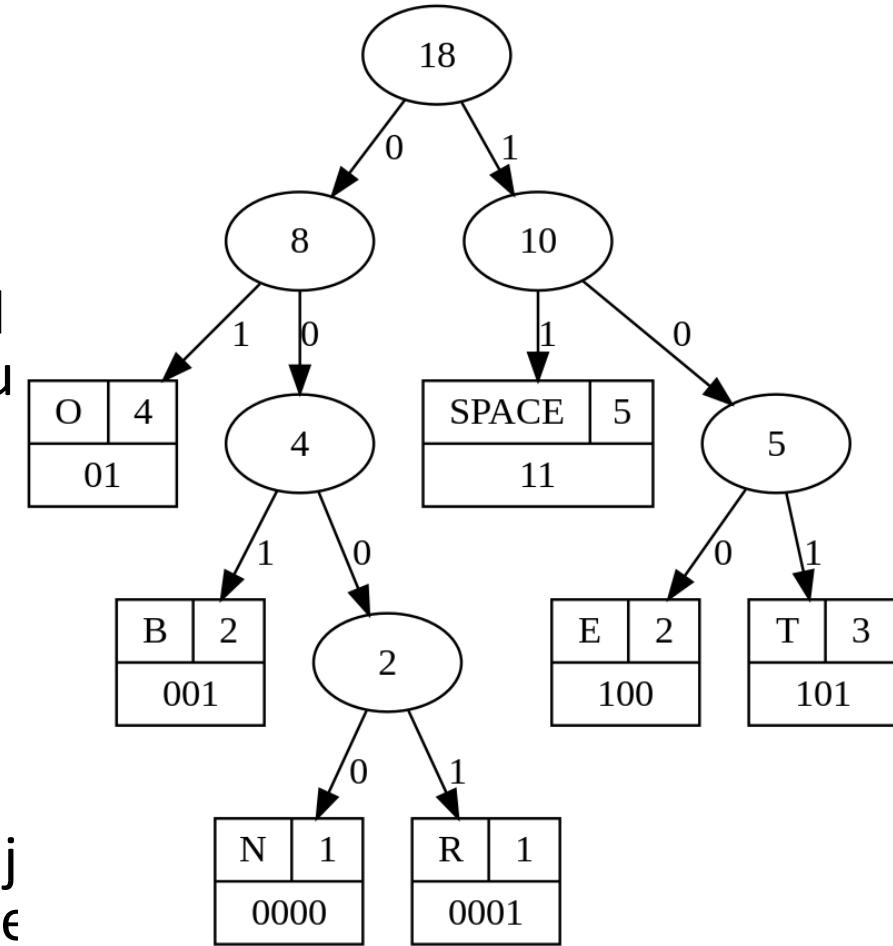


Format JPG

Kompresja bezstratna danych algorytmem Huffmana

Jedna z najprostszych i łatwych w implementacji metod kompresji bezstratnej. Została opracowana w 1952 roku przez Amerykanina Davida Huffmana.

Algorytm Huffmana nie należy do najefektywniejszych obliczeniowo systemów bezstratnej kompresji danych, dlatego też praktycznie nie używa się go samodzielnie. Często wykorzystuje się go jako ostatni etap w różnych systemach kompresji, zarówno bezstratnej, jak i stratnej np. MP3 lub JPEG. Pomimo że nie jest doskonały, stosuje się go ze względu na prostotę oraz brak ograniczeń patentowych. Jest to przykład wykorzystania algorytmu zachłanego.



Drzewo Huffmana wygenerowane z frazy „TO BE OR NOT TO BE”

Format JPG

Kompresja bezstratna danych algorytmem Huffmana

W wyniku zastosowania tego etapu uzyskuje się pełną kompresję, de facto stratno-bezstratną.

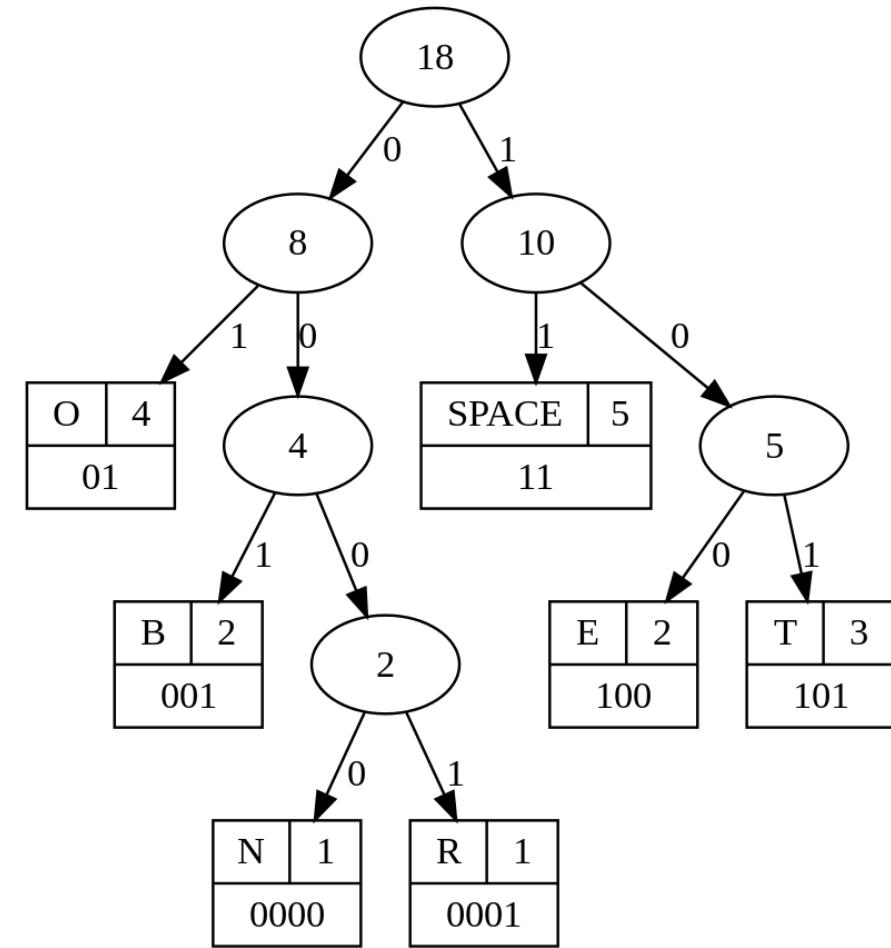
Strata następuje na etapach:

- ?

Kompresja bezstratna w momencie:

- ?

Do powyższego opisu należałoby dodać opis techniczny konstrukcji formatu graficznego.



Drzewo Huffmana wygenerowane
z frazy „TO BE OR NOT TO BE”

Grafika i wizualizacja

- grafika wektorowa

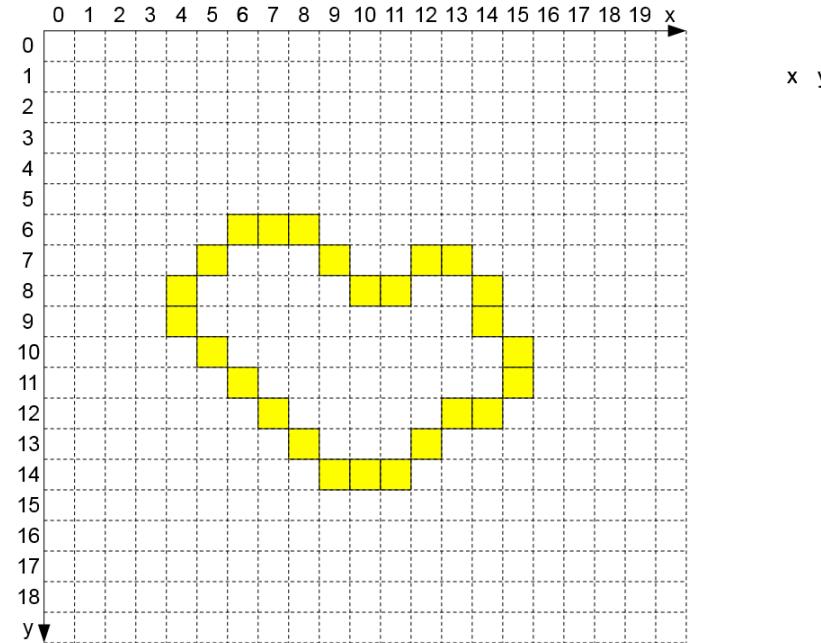
Formaty zapisów grafiki wektorowej:

- tablica nieuporządkowanych punktów;
- tablica uporządkowanych punktów;
- parametry obiektów;
- równania parametryczne.

Grafika i wizualizacja

- grafika wektorowa

Tablica nieuporządkowanych punktów.

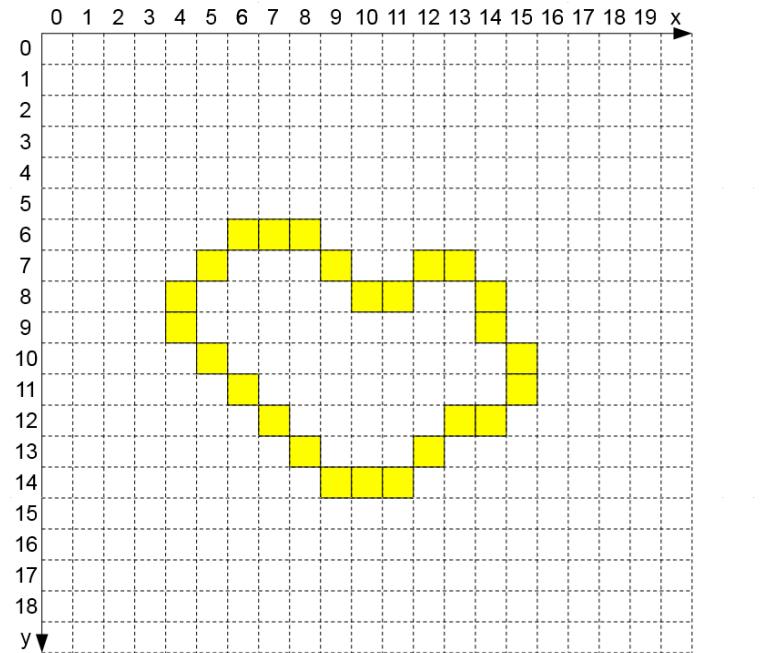


Grafika demonstracyjna w formie animowanego GIFa

Grafika i wizualizacja

- grafika wektorowa

Tablica uporządkowanych punktów.

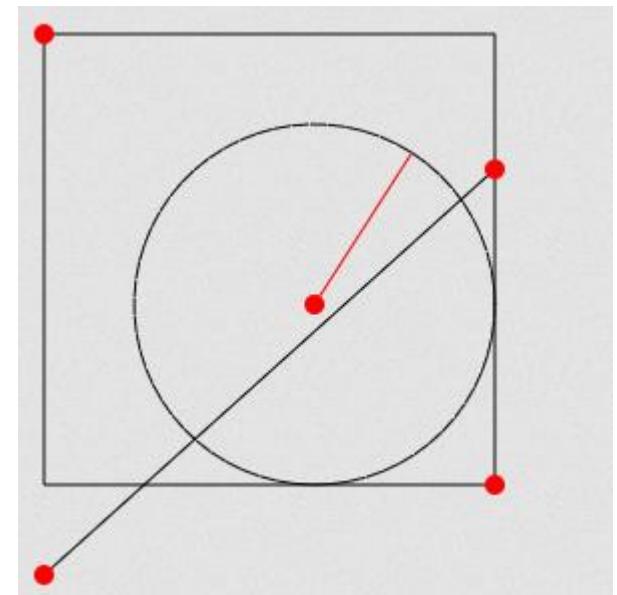


Grafika i wizualizacja

- grafika wektorowa

Parametry obiektów. Przykłady obiektów (rysunek obok):

- odcinek – współrzędne początku $x_0=2, y_0=1.5$ i końca $x_1=-3, y_1=-3$;
- prostokąt – współrzędne lewego górnego $x_0=-3, y_0=3$ i prawego dolnego rogu prostokąta $x_1=2, y_1=-2$;
- okrąg – współrzędne środka $x_0=0, y_0=0$ i promień okręgu $r=2$.



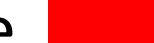
Grafika i wizualizacja

- grafika wektorowa

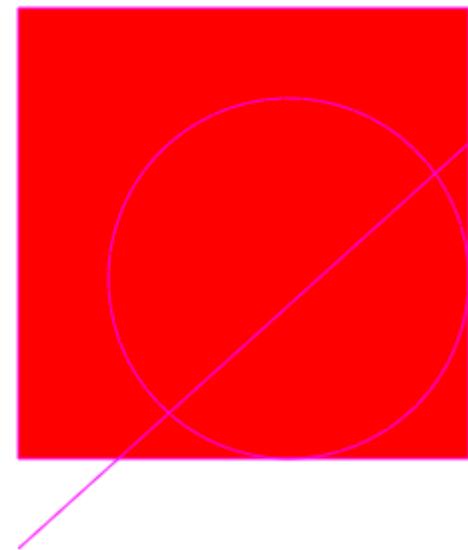
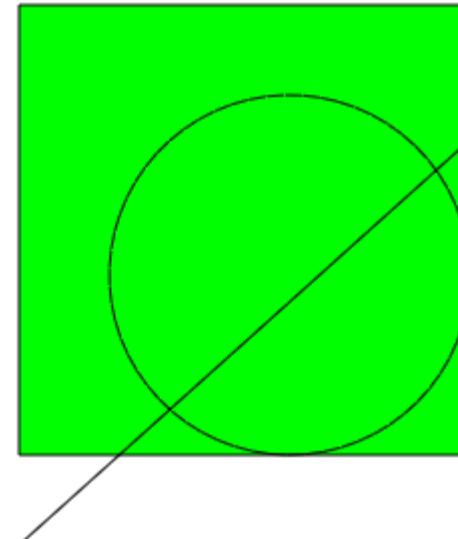
W grafice wektorowej obiektom mogą być bezpośrednio przypisane kolory. Stosuje się także rozwiązanie w którym do figur przypisuje się obiekty definiujące kolor obrzeża, wypełnienia, teksturę itp. Zaletą takiego rozwiązania jest to, że wiele figur może mieć przypisany jeden obiekt, co pozwala na wspólną zmianę parametrów.

Przykładowo dla obiektu:

Obrzeże  , wypełnienie 

obrzeże  , wypełnienie 

uzyskuje się wygląd figur:



Grafika i wizualizacja

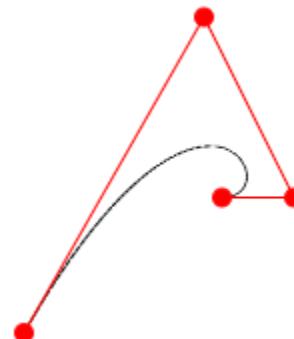
- grafika wektorowa

Równania parametryczne. Opisuję kształt krzywych lub powierzchni za pomocą kilku parametrów. Parametry dobierane są tak, aby odwzorować określony kształt. Przykładem równania parametrycznego jest równanie krzywej Beziera trzeciego stopnia:

$$P(t) = A(1-t)^3 + 3Bt(1-t)^2 + 3Ct^2(1-t) + 3Dt^3.$$

gdzie A, B, C i D są punktami sterującymi przebiegiem krzywej, $t \in (0;1)$.

Dla punktów A(-1,-2), B(1,1.5), C(2,-0.5), D(1.2,-0.5) uzyskuje się następującą krzywą:

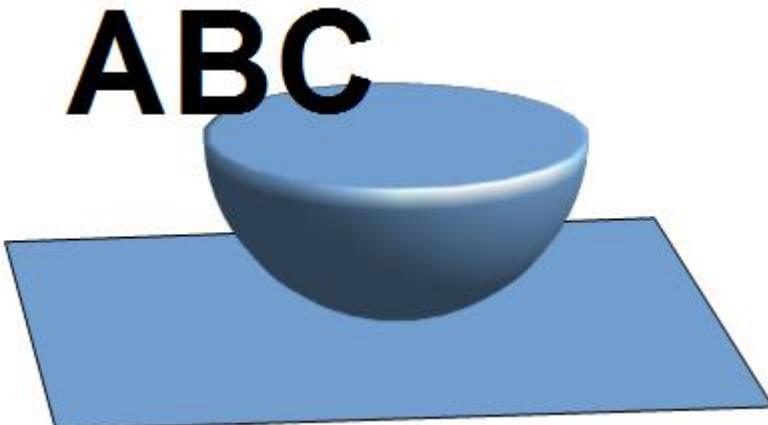


Grafika i wizualizacja

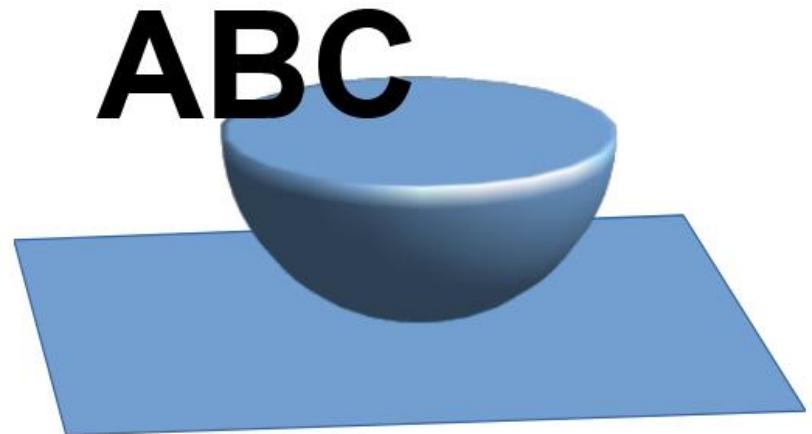
- - grafika rastrowa a wektorowa

Grafika wektorowa i rastrowa różnią się sposobem przechowywania danych.

grafika rastrowa



grafika wektorowa



W przybliżeniu obrazów widać różnicę na krawędziach.

Grafika i wizualizacja

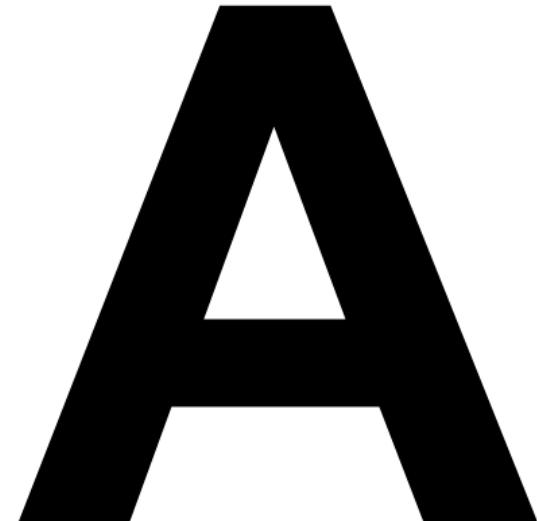
- grafika rastrowa a wektorowa

Powiększanie grafiki rastrowej i wektorowej. Skalowalność jest jedną z zalet grafiki wektorowej.

grafika rastrowa



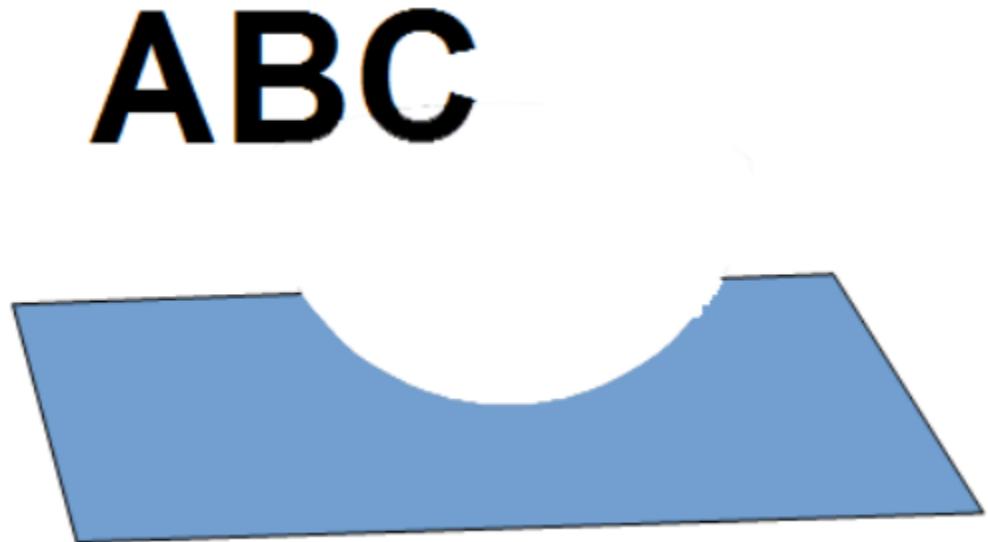
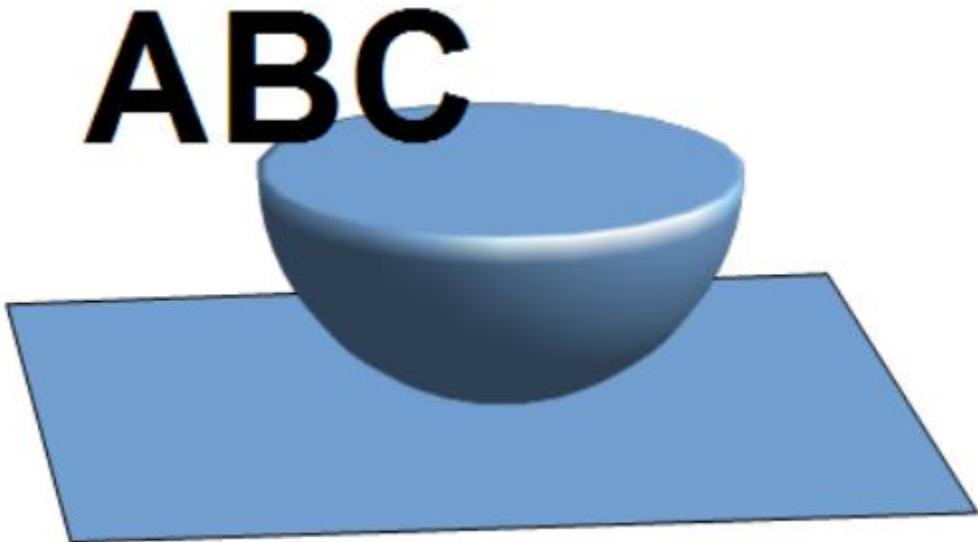
grafika wektorowa



Grafika i wizualizacja

- grafika rastrowa a wektorowa

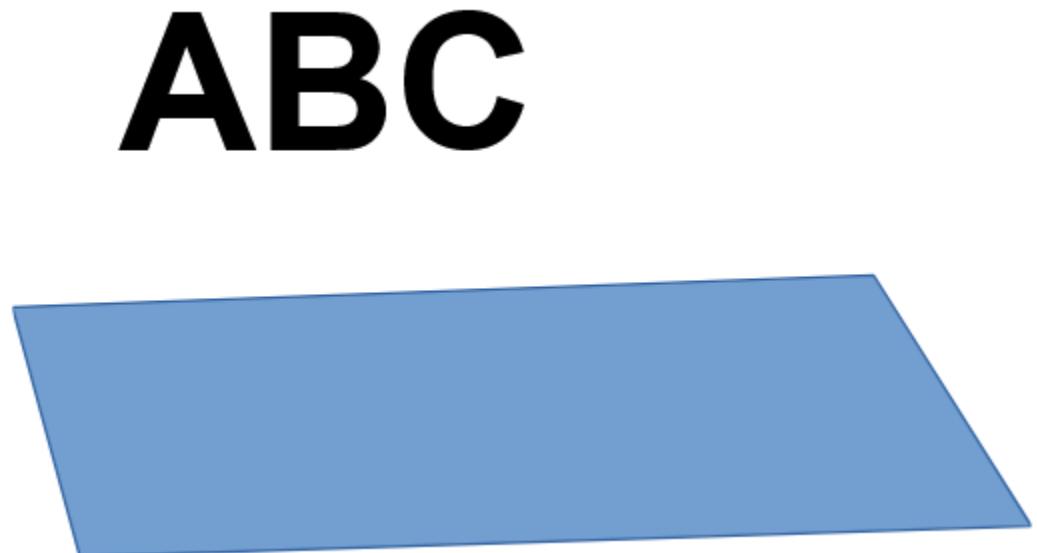
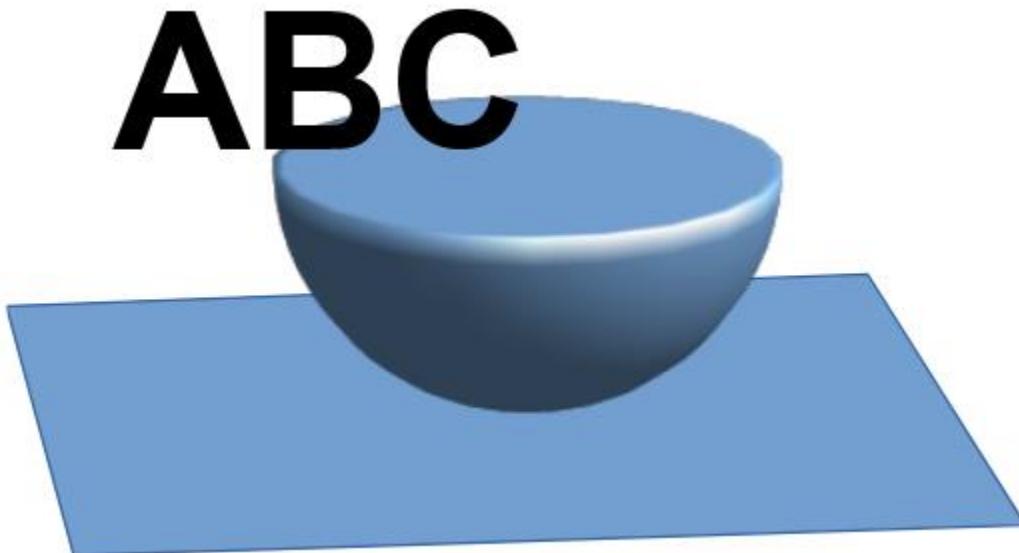
Grafika rastrowa – usunięcie obiektu.



Grafika i wizualizacja

- grafika rastrowa a wektorowa

Grafika wektorowa – usunięcie obiektu. Wnioski?

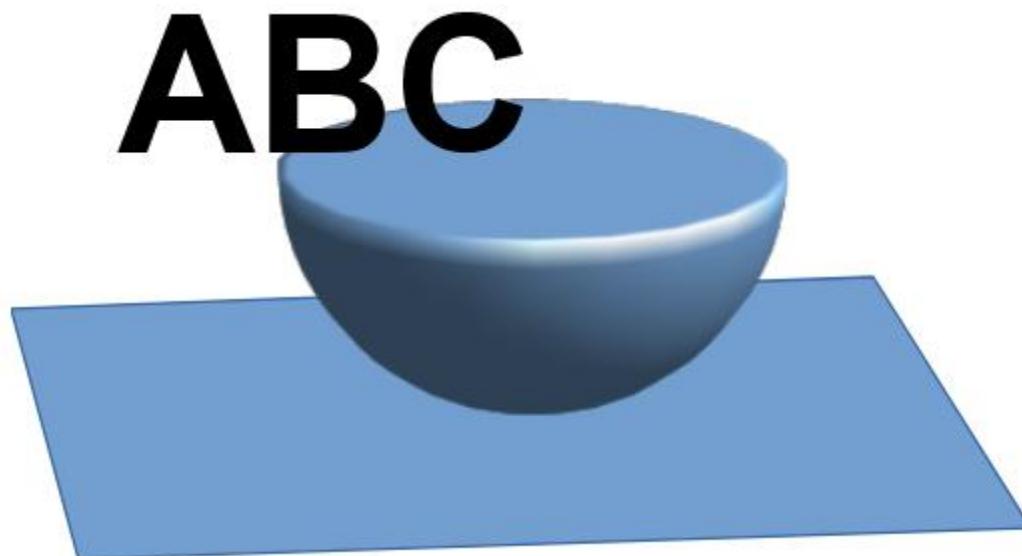


Grafika i wizualizacja

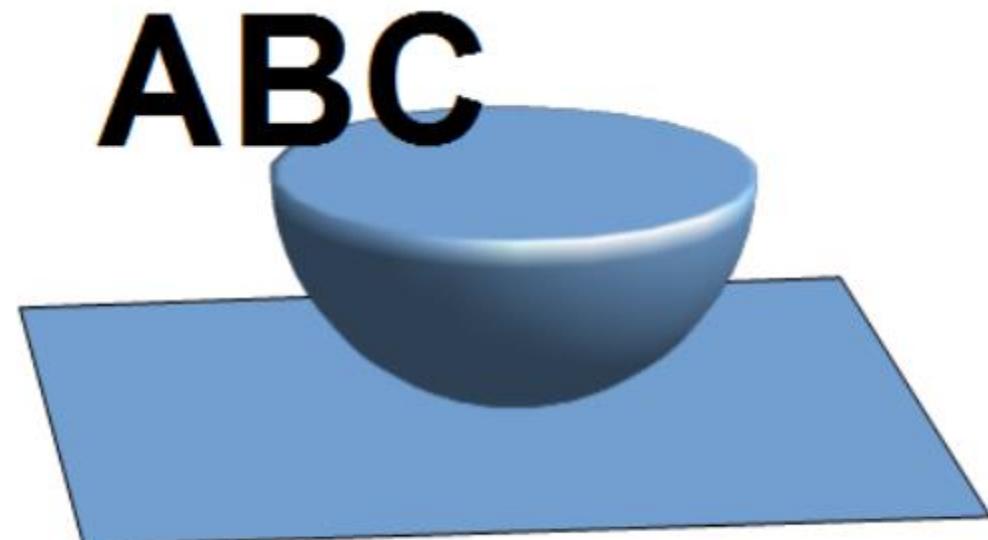
- grafika rastrowa a wektorowa

Konwersja grafiki wektorowej do rastrowej. Coś zostaje utracone/uzyskane?

grafika wektorowa



grafika rastrowa



Grafika i wizualizacja

Konwersja grafiki rastrowej do wektorowej. Coś zostaje utracone/uzyskane?

grafika wektorowa



grafika rastrowa



Grafika i wizualizacja

- formaty plików grafiki rastrowej

Formaty plików grafiki rastrowej:

BMP – mapa bitowa, bez kompresji lub z kompresją stratną. 1, 4, 8, 24 bity na piksel;

GIF – zapis z paletą barw. Liczba barw ograniczona do 256 kolorów. Bezstratna kompresja LZW. Możliwość zapisu kilku obrazów na raz dla celów animacji;

TIFF – zapis bezstratny. Różne możliwość zapisu. Najczęściej kompresja LZW;

TGA – zapis bitowej 8, 16, 24, 32 bitowej, zapis przezroczystości;

JPG – zapis z kompresją stratną wykorzystująca dyskretną transformację kosinusów;

JPG2000 – zapis z kompresją stratną wykorzystującą dyskretną transformację falkową;

PNG – unowocześniona kompresja GIF, wydajniejsza kompresja. Brak ograniczenia co do zapisu z paletą barw, obsługa przezroczystości.

Grafika i wizualizacja

- formaty plików grafiki wektorowej

Formaty plików grafiki wektorowej:

WMF – wewnętrzny format systemu Windows;

PS, EPS – język opisu strony opracowany przez firmę Adobe;

HPGL – język sterowania ploterami firmy HP;

DXF – przemysłowy standard opracowany przez firmę Autodesk (AutoCAD);

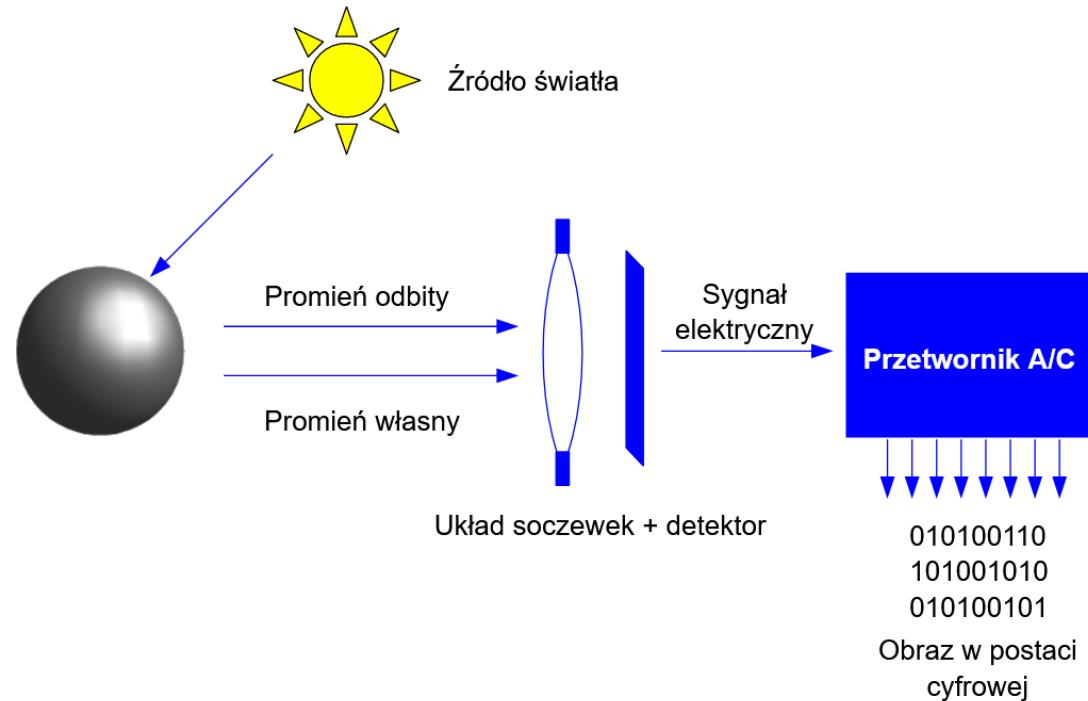
CDR – format stosowany w programie CorelDraw;

SVG – format przeznaczony dla grafiki wektorowej na stronach WWW, wspiera animacje.

Grafika i wizualizacja

- rejestracja obrazu

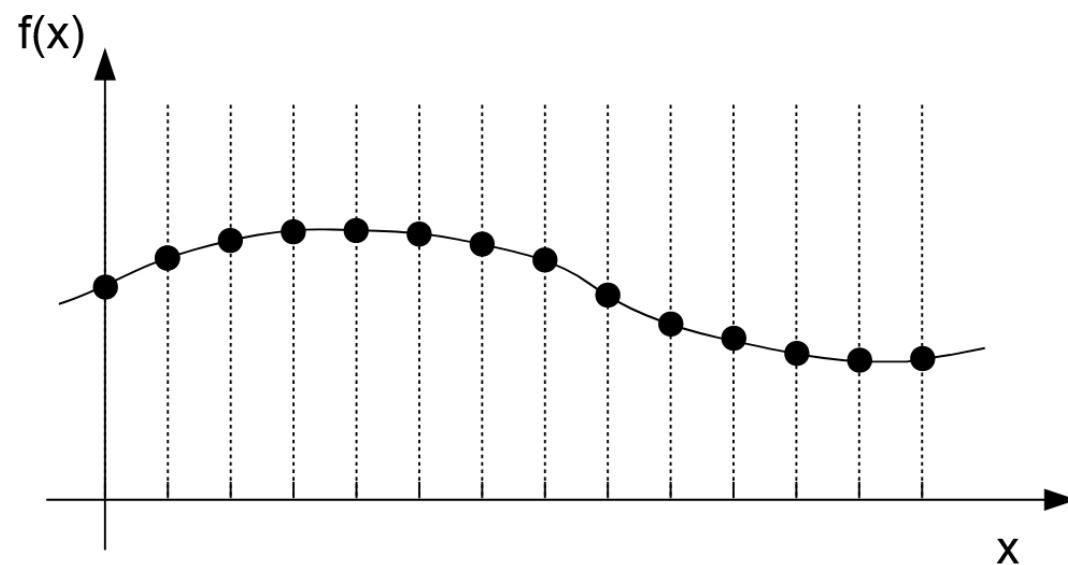
Proces rejestracji obrazu.



Grafika i wizualizacja

- rejestracja obrazu

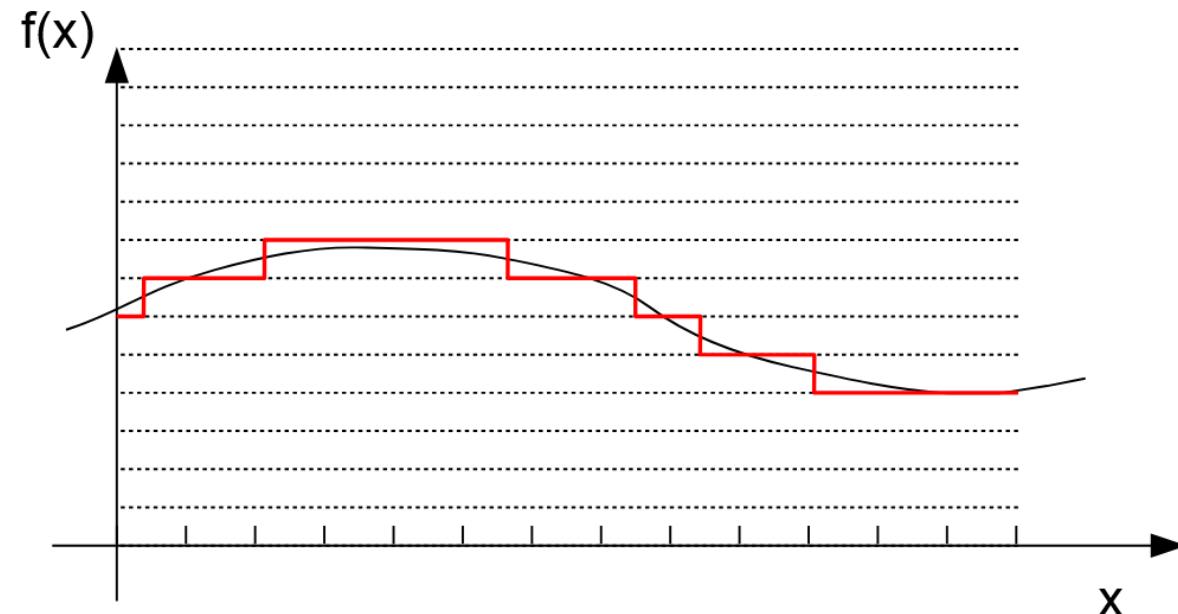
Próbkowanie w przetwarzaniu sygnałów – proces zmiany sygnału ciągłego na sygnał o dyskretnym czasie (opisany wartościami w dyskretnych chwilach czasu).



Grafika i wizualizacja

- rejestracja obrazu

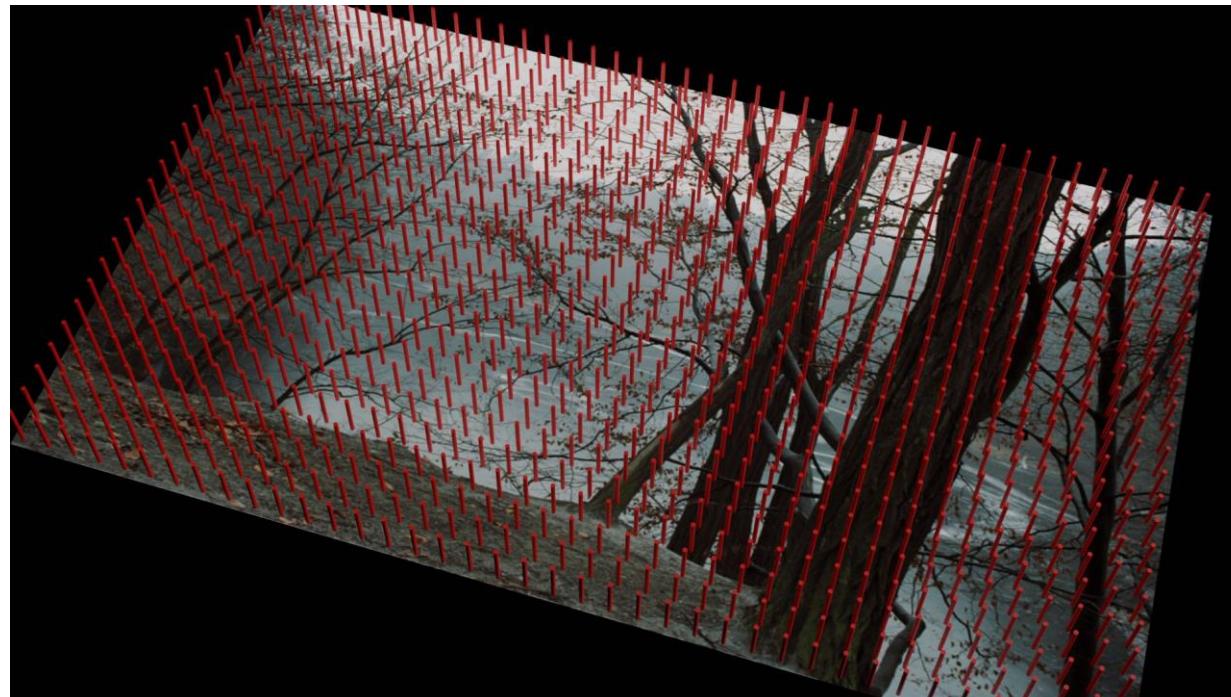
Kwantyzacja – proces zamiany wartości rzeczywistej na dyskretną.



Grafika i wizualizacja

- rejestracja obrazu

O ile poprzednie rysunki poglądowe odzwierciedlały raczej próbkowanie i kwantyzację sygnału jednowymiarowego (np. dźwięk) - obraz rastrowy powstaje w wyniku próbkowania ciągłej dwuwymiarowej funkcji.



Grafika i wizualizacja

- rejestracja obrazu

Aliasing – efekt związany z próbkowaniem sprawiający, że dwa różne sygnały (obrazy) po próbkowaniu są takie same. Może powodować:

- zmianę częstotliwości sygnału (obrazu) okresowego,
- zanikanie szczegółów.

Rozróżnia się:

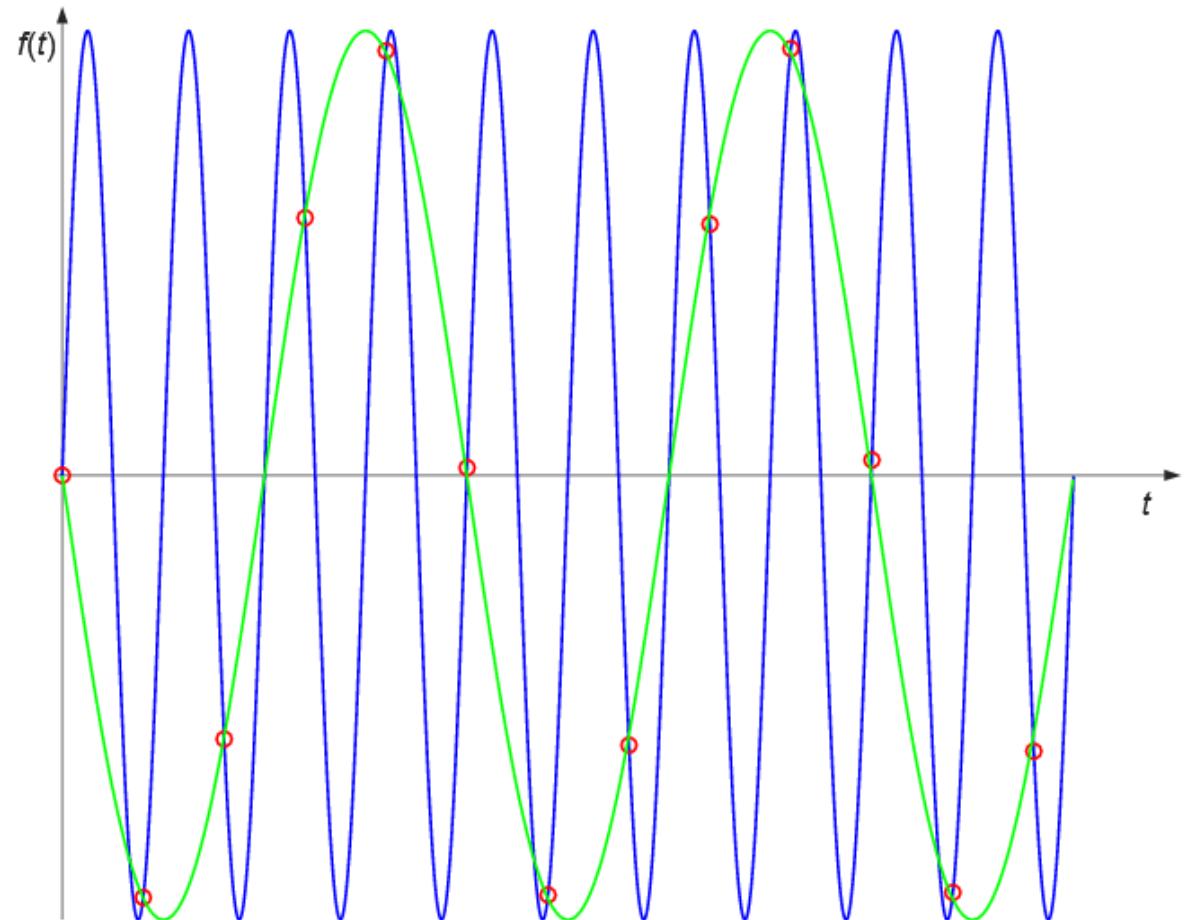
- aliasing związany z próbkowaniem w czasie,
- aliasing związany z próbkowaniem w przestrzeni.

Grafika i wizualizacja

- rejestracja obrazu

Aliasing związany z próbkowaniem sygnału w czasie.

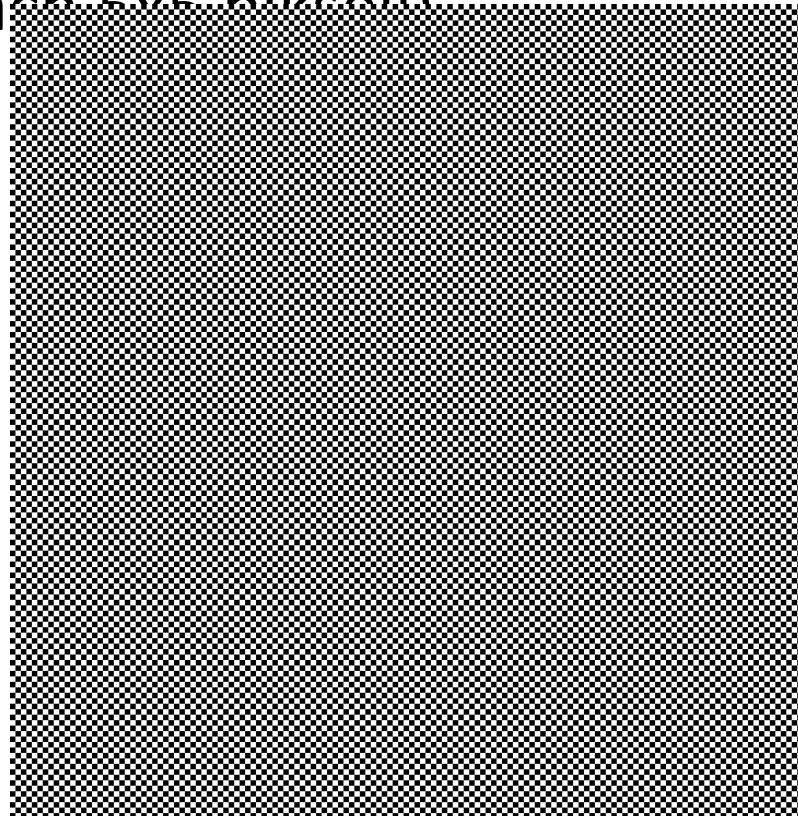
Aby uniknąć błędnej interpretacji częstotliwości, częstotliwość próbkowania należy przyjąć dwukrotnie wyższą niż przewidywana najwyższa częstotliwość w sygnale (patrz twierdzenie o próbkowaniu, twierdzenie Nyquista-Shannona).



Grafika i wizualizacja

- rejestracja obrazu

Obraz szachownicy o rozmiarach 750x750 pikseli (złożonej ze 150x150 kwadratów o rozmiarach 5x5 pikseli)



Grafika i wizualizacja

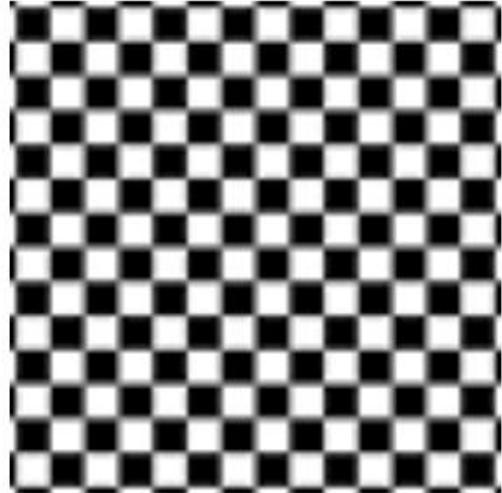
- rejestracja obrazu

Ten sam obraz szachownicy co poprzednio po zmianie rozmiaru obrazu.

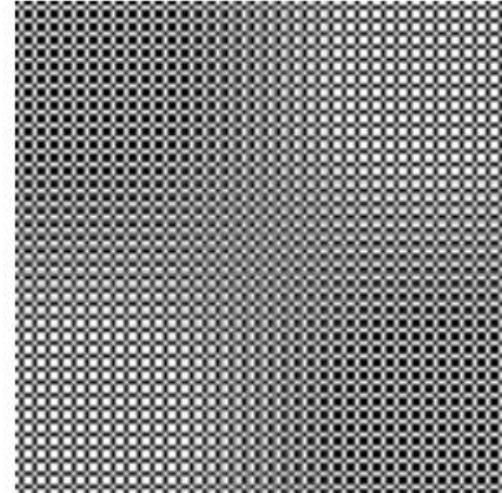
75x75 pikseli



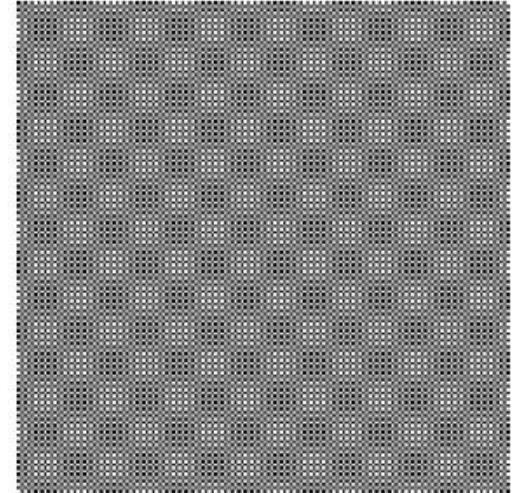
233x233 piksele



83x83 piksele



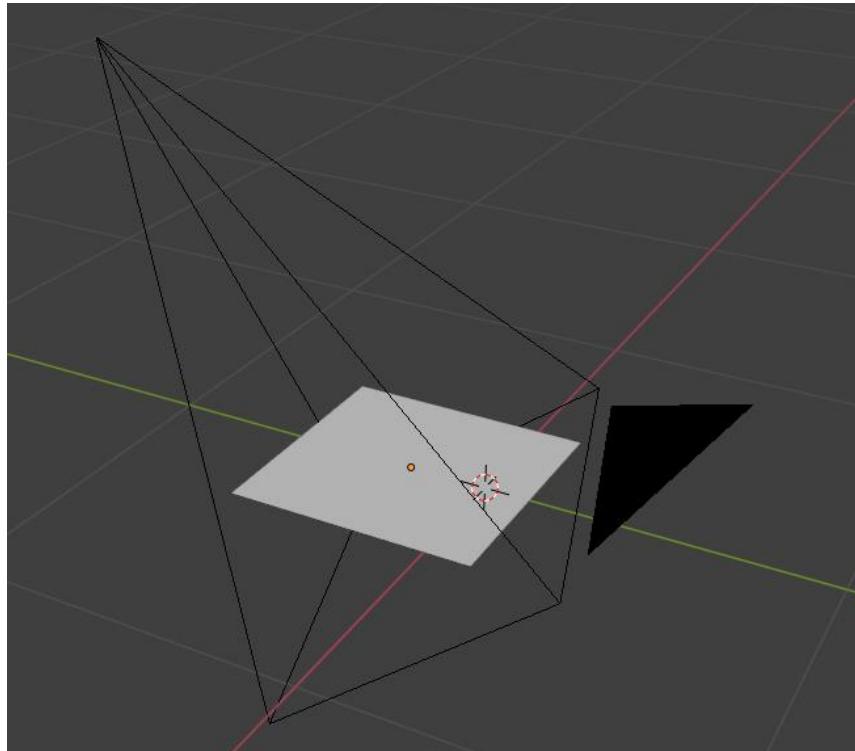
113x113 piksele



Grafika i wizualizacja

- rejestracja obrazu

Kamera skierowana na obiekt pokryty tekstonią szachownicy.

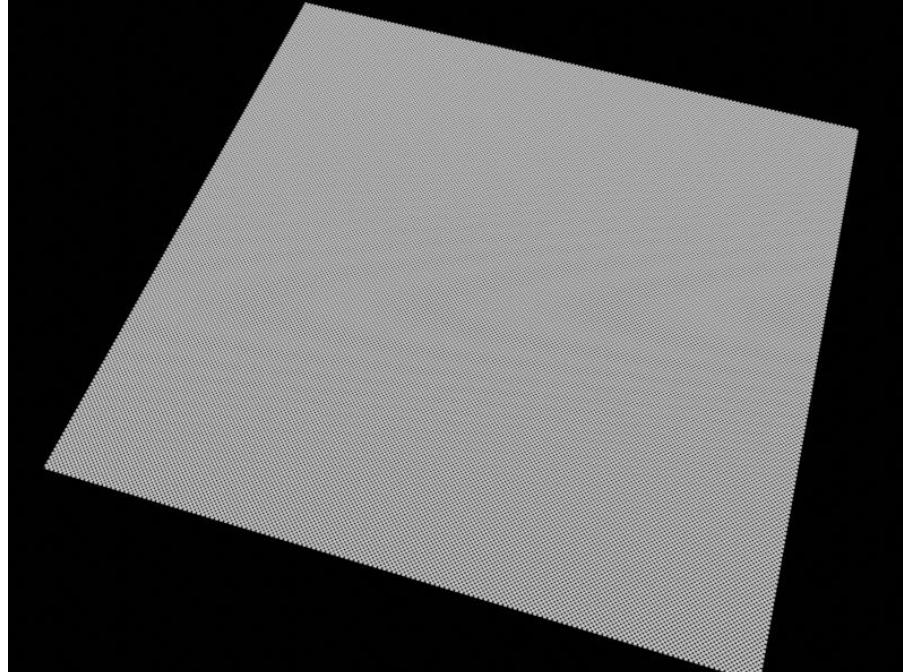


Grafika i wizualizacja

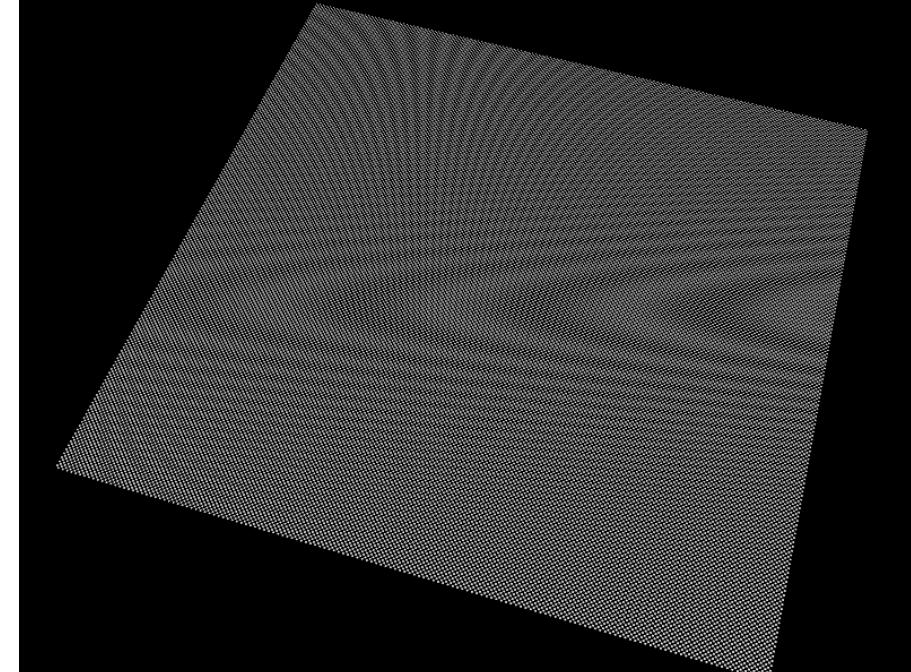
- rejestracja obrazu

Obraz z kamery.

Obraz bezpośrednio z kamery



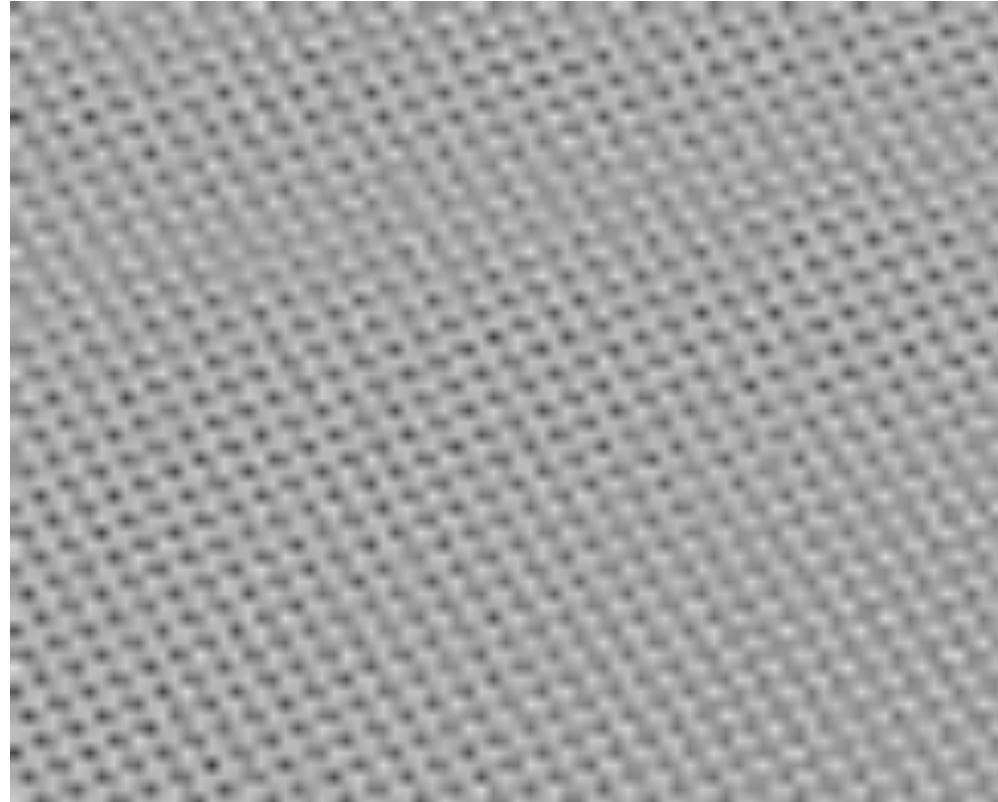
Obraz po zmianie jasności i kontrastu



Grafika i wizualizacja

- rejestracja obrazu

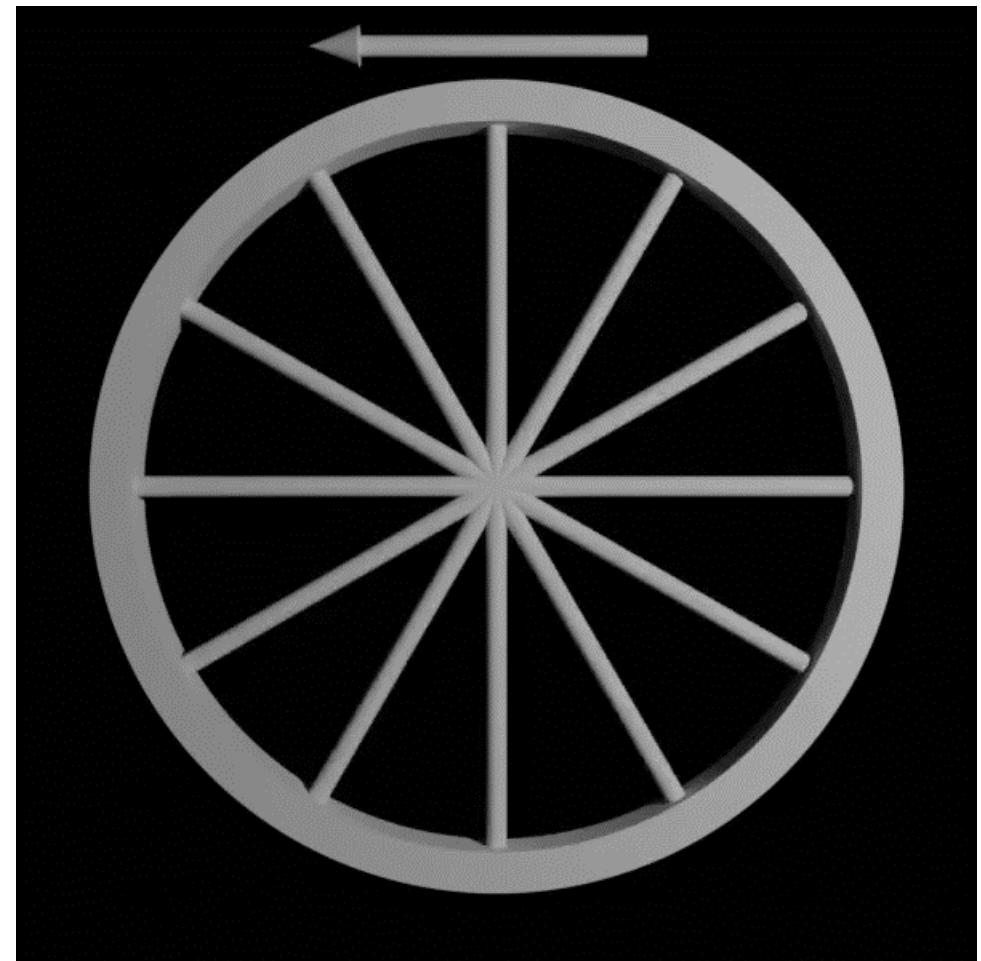
Powiększenie obrazu z kamery (aliasing).



Grafika i wizualizacja

- rejestracja obrazu

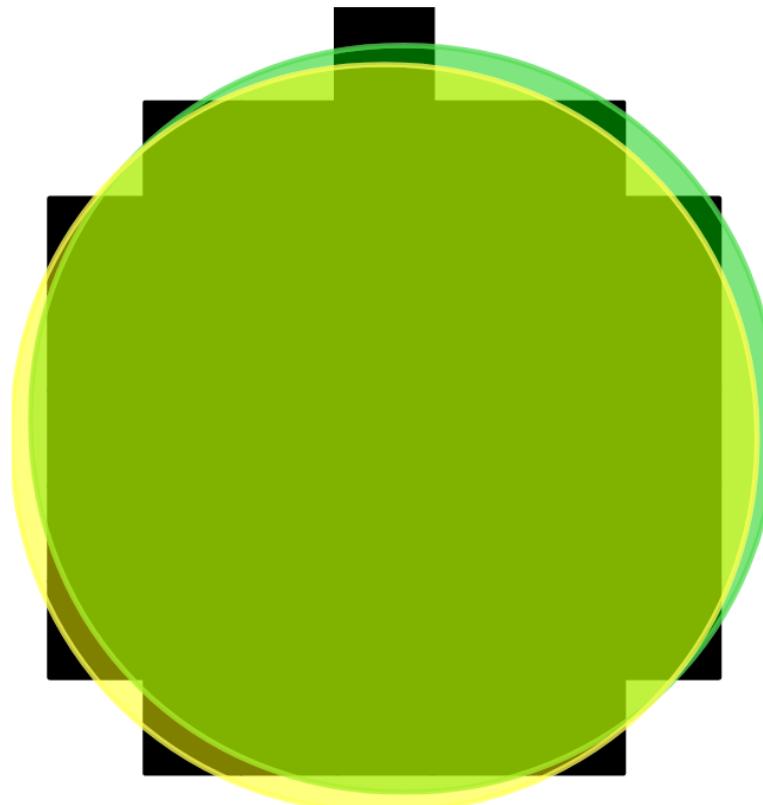
Obraz kręcącego się koła. Strzałką zaznaczono prawidłowy kierunek obrotu.



Grafika i wizualizacja

- rejestracja obrazu

Dwa różne koła z których powstaje ten sam obraz rastrowy (aliasing).



Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Grafika i wizualizacja

Barwa Kolor

Mariusz Borawski, dr hab. inż., Edward Półrolniczak, dr inż.,
Katedra Systemów Multimedialnych,
Wydział Informatyki, ZUT w Szczecinie

Grafika i wizualizacja

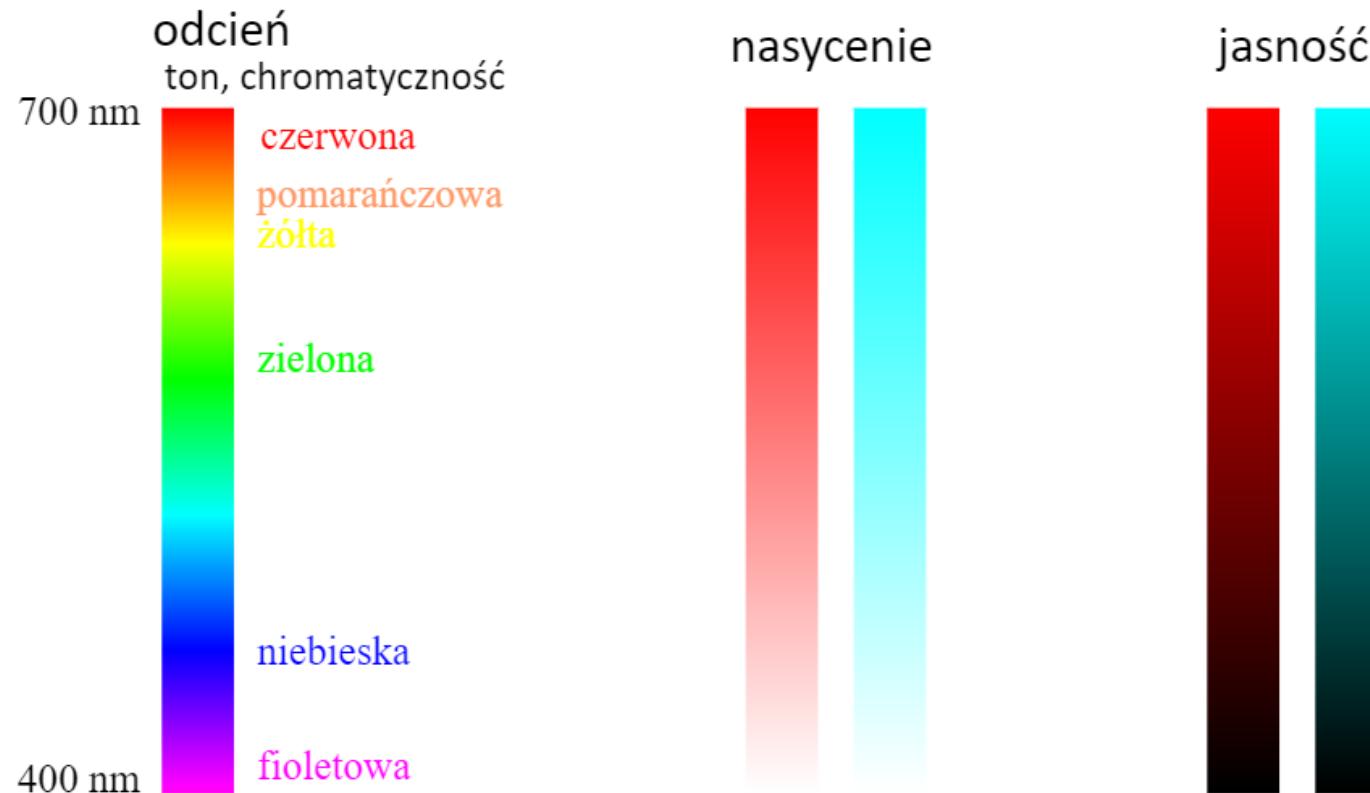
Zagadnienia

- Percepcja barwy
- Modele barw
- System zarządzanie kolorem

Grafika i wizualizacja

- Percepcja barwy

Kolor (barwa) – wrażenie, które jest odczuwane za pośrednictwem zmysłu wzroku powstające pod wpływem światła.



Grafika i wizualizacja

- Percepcja barwy

W grafice komputerowej (i w rozumieniu potocznym) terminów kolor i barwa można używać zamiennie. W poligrafii i malarstwie pojęcie barwy jest związane z wrażeniem wzrokowym, a pojęcie koloru z własnościami użytej farby.

Grafika i wizualizacja

Zagadnienia

ZNACZENIE BARWY (PDF)

Grafika i wizualizacja

- Percepcja barwy

Barwy (kolory) proste (widmowe, spektralne, monochromatyczne) – wrażenie psychofizjologiczne, które jest odczuwane za pośrednictwem zmysłu wzroku powstające pod wpływem światła. Można je uzyskać przez rozszczepienie światła białego na pryzmacie. Barwami prostymi są kolejne barwy tęczy:

- fioletowa – fale o długości ~ 400–450 nm;
- niebieska – fale o długości ~ 450–490 nm;
- cyjan – fale o długości ~ 490–520 nm;
- zielona – fale o długości ~ 520–560 nm;
- żółta – fale o długości ~ 560–590 nm;
- pomarańczowa – fale o długości ~ 590–635 nm;
- czerwona – fale o długości ~ 635–700 nm.

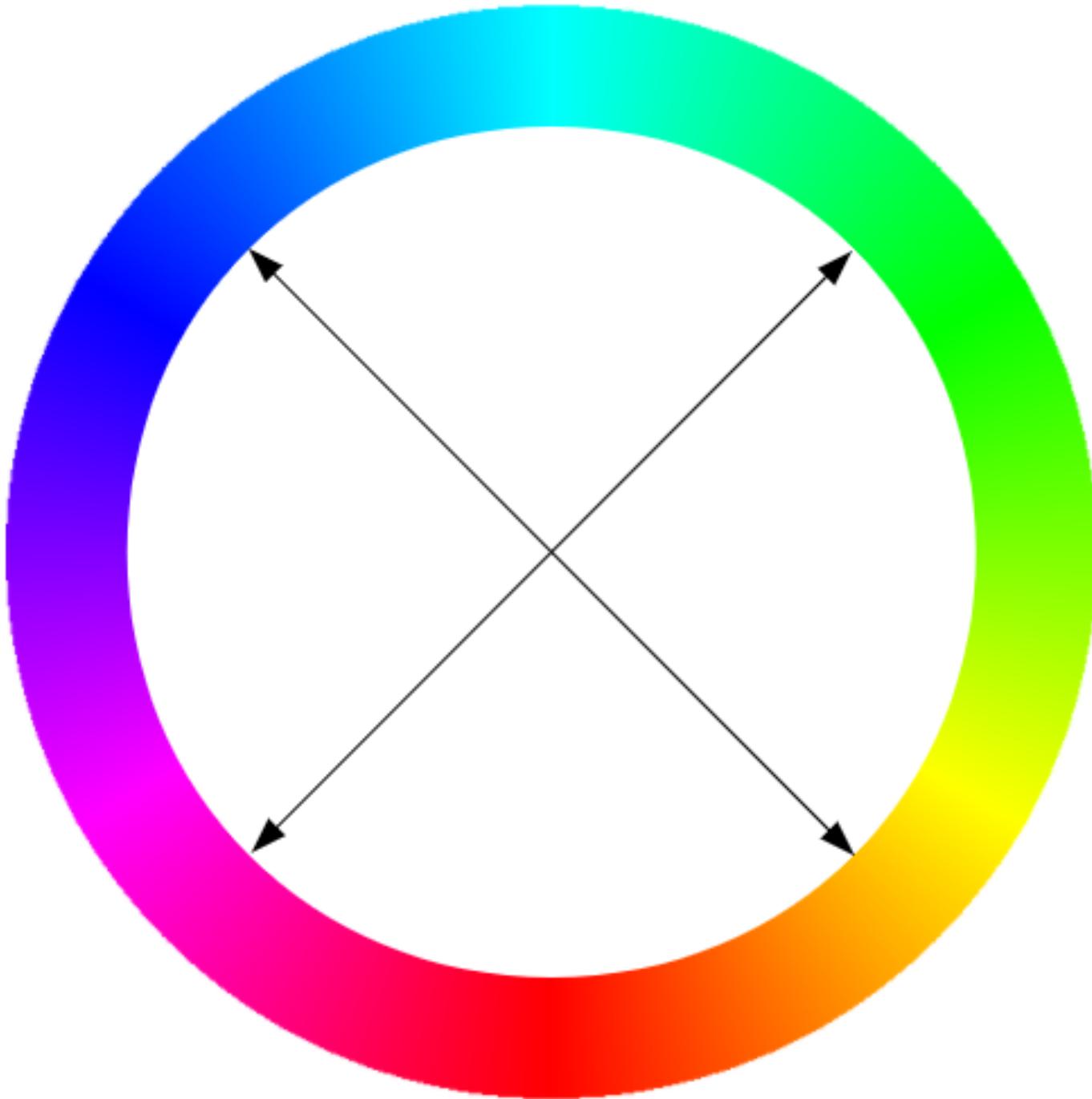


Grafika i wizualizacja

- Percepcja barwy

Barwy achromatyczne (niekolorowe) – odcienie szarości od bieli do czerni.





Grafika i wizualizacja - Percepcja barwy

Barwy dopełniające – barwy które po połączeniu ze sobą dadzą barwę achromatyczną.

Grafika i wizualizacja

- Percepcja barwy

Percepcja wzrokowa – proces interpretacji otaczającego świata na podstawie informacji z aparatu widzenia przez układ wzrokowy.

Etapy przetwarzania informacji:

- rejestracja obrazu,
- wstępne przetworzenie,
- postrzeganie,
- identyfikacja i rozpoznawanie obiektów.

Grafika i wizualizacja

- Percepcja barwy

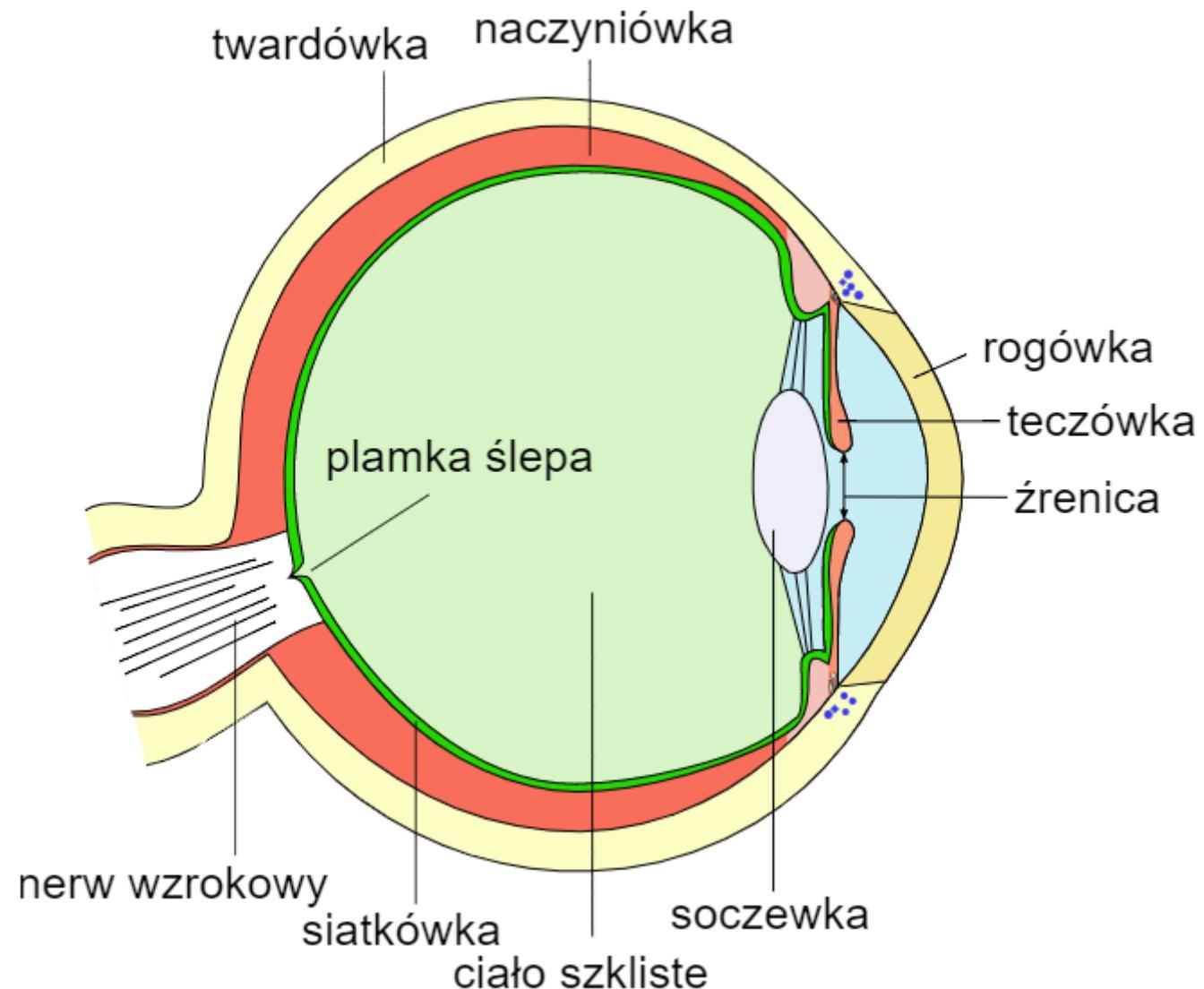
Najważniejsze elementy układu wzrokowego:

- oczy – rejestracja obrazu i częściowe jego przetworzenie,
- nerw wzrokowy – przekazanie informacji do mózgu,
- kora wzrokowa – przetworzenie obrazu,
- kora skojarzenia wzrokowego – określenie orientacji przestrzennej, identyfikacja i rozpoznawanie obiektów.

Grafika i wizualizacja

- Percepcja barwy

Budowa oka



Grafika i wizualizacja

- Percepcja barwy

Aby możliwa była percepcja barwy przez człowieka konieczne są trzy czynniki:

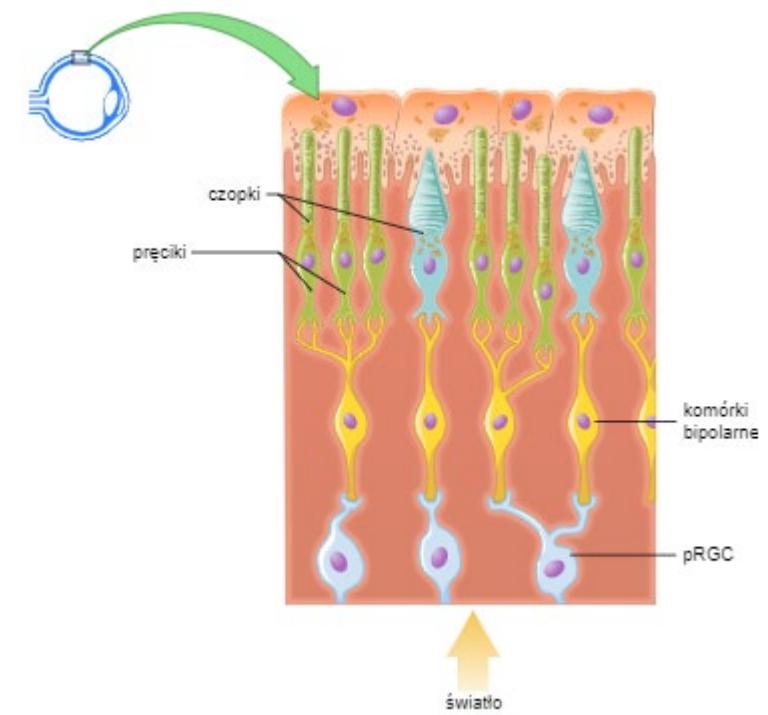
- emisja światła,
- pobudzenie receptorów odpowiedzialnych za barwę,
- przetworzenie w korze mózgowej informacji z receptorów.

Grafika i wizualizacja

- Percepcja barwy

Fotoreceptory:

- pRGC (ang. Photosensitive Retinal Ganglion Cells) – najprawdopodobniej czułe są na światło fioletowo-niebieskie,
- czopki – odpowiadają za „widzenie” dzienne,
- pręciki – odpowiadają za „widzenie” nocne.



Grafika i wizualizacja

- Percepcja barwy

Plamka ślepa:

- obszar siatkówki w którym nerw wzrokowy opuszcza oko. W miejscu tym siatkówka pozbawiona jest fotoreceptorów,
- plamki ślepe umieszczone są tak aby brak informacji z jednego oka był kompensowany informacją z drugiego oka,
- brakująca informacja jest interpolowana na podstawie obrazu z drugiego oka i danych z fotoreceptorów otaczających plamkę ślepą.

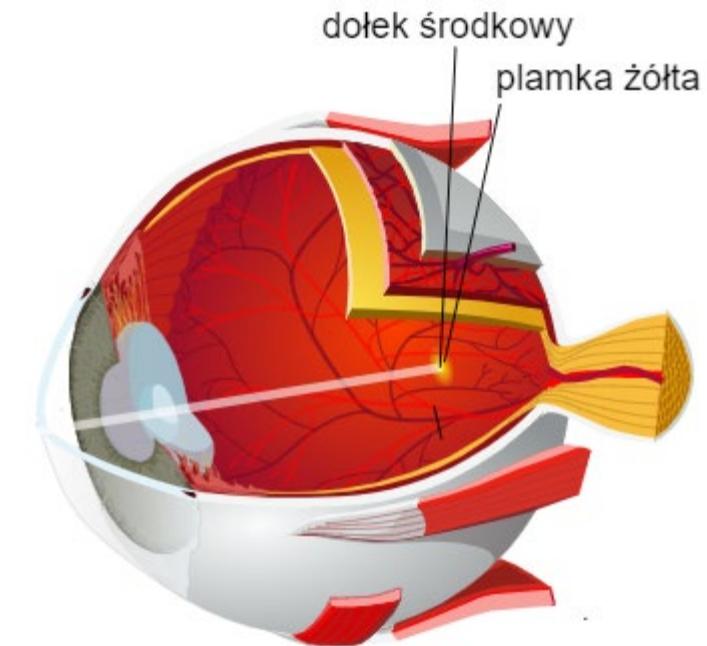
Grafika i wizualizacja

- Percepcja barwy

Plamka żółta jest obszarem siatkówki z dominacją czopków.

Odpowiedzialna jest za widzenie kolorów w wysokiej rozdzielczości.

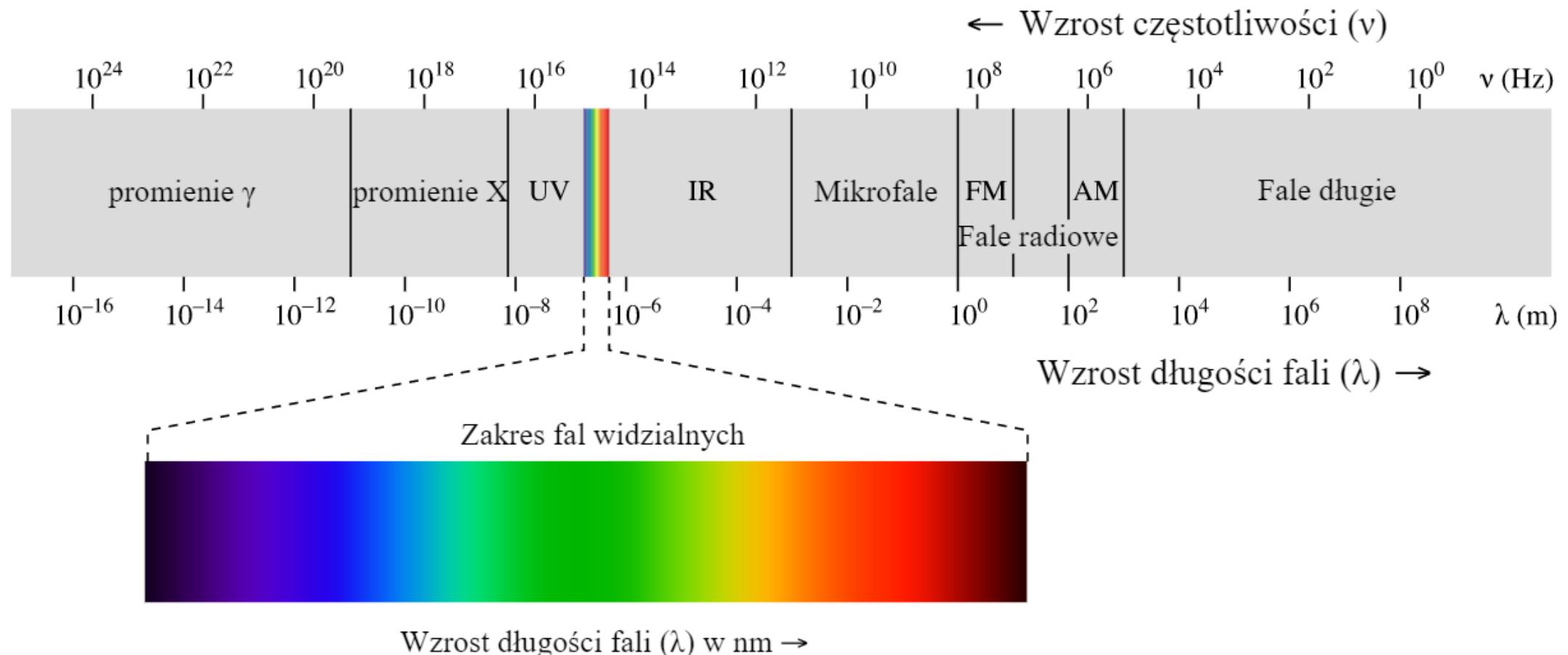
Dołek środkowy siatkówki oka (dołek środkowy) – centralna część plamki żółtej o maksymalnym zagęszczaniu czopków i braku pręcików. Jest to obszar najbardziej ostrego widzenia odpowiadający 2° obszaru widzenia.



Grafika i wizualizacja

- Percepcja barwy

Okno optyczne widzenia oka



Grafika i wizualizacja

- Percepcja barwy

pRGC (ang. Photosensitive Retinal Ganglion Cells):

- czułe są na światło fioletowo-niebieskie,
- biorą udział w odruchu żrenicznym,
- prawdopodobnie biorą udział w regulacji rytmu okołodobowego,
- prawdopodobnie biorą udział w odbiorze wizualnym obrazu.

Grafika i wizualizacja

- Percepcja barwy

Pręciki:

- mają bardzo wysoką czułość,
- ich największe zagęszczenie jest wokół plamki żółtej,
- maksymalna czułość przesunięta jest nieco w kierunku światła niebieskiego.

Grafika i wizualizacja

- Percepcja barwy

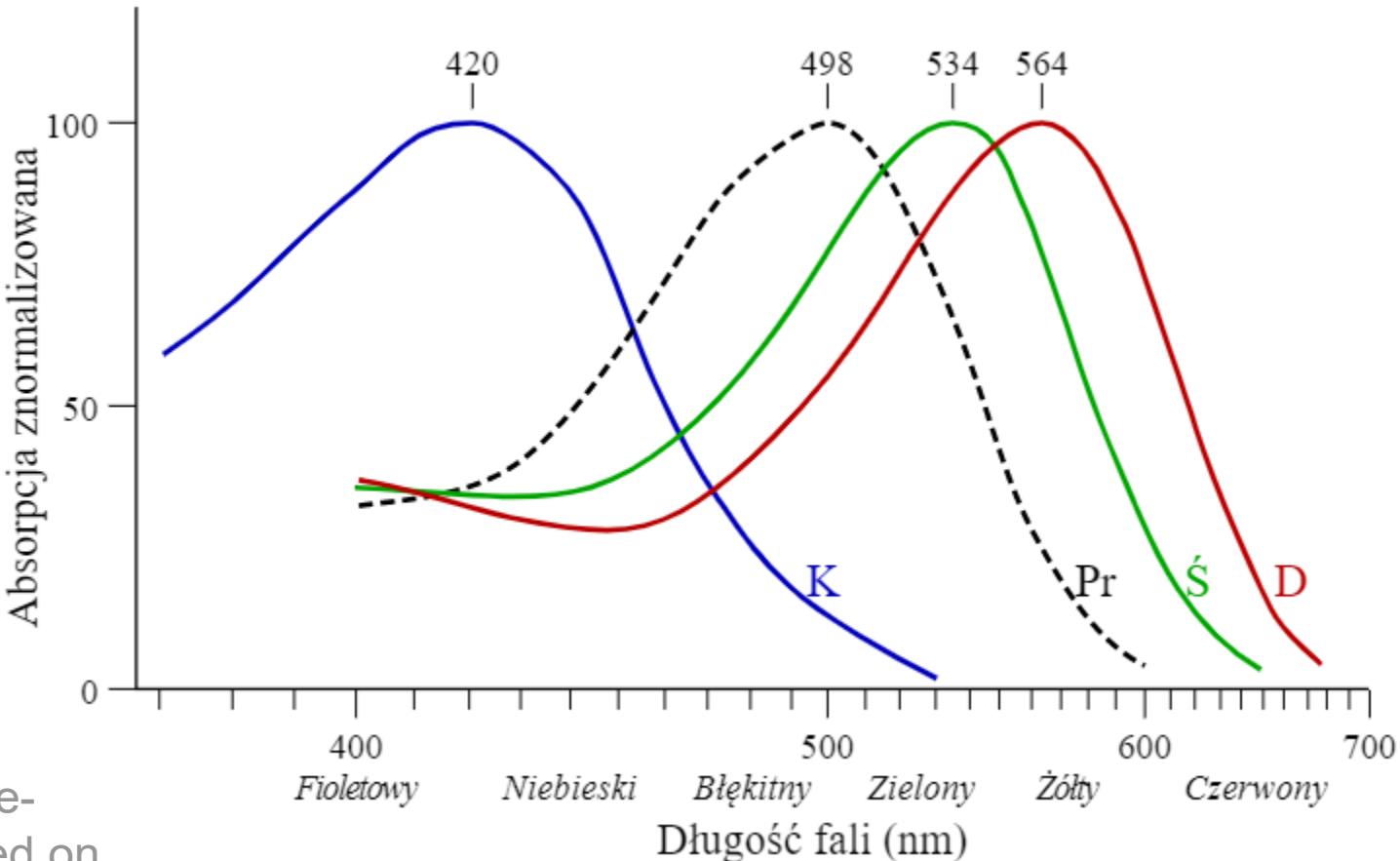
Czopki:

- rozróżnia się trzy rodzaje czopków odpowiedzialnych za widzenie kolorów czerwonego, zielonego i niebieskiego,
- czopki odpowiedzialne za widzenie koloru czerwonego najbardziej czułe są na promieniowanie o $\lambda = 560$ nm, oznaczane są czasami symbolem D – długofalowe (ang. L),
- czopki odpowiedzialne za widzenie koloru zielonego najbardziej czułe są na promieniowanie o $\lambda = 530$ nm, oznaczane są czasami symbolem S – średniofalowe (ang. M),
- czopki odpowiedzialne za widzenie koloru niebieskiego najbardziej czułe są na promieniowanie o $\lambda = 420$ nm, oznaczane są czasami symbolem K – krótkofalowe (ang. S),
- najmniej jest czopków niebieskich,
- największe skupisko czopków jest w środkowej części siatkówki w plamce żółtej.

Grafika i wizualizacja

- Percepcja barwy

Czułość fotoreceptorów



Źródło: Vectorized version of the GFDL image Cone-response.png uploaded by User:Maxim Razin based on work by User:DrBob and User:Zeimus, polish version
Masur – [Krzywe spektrum absorpcji światła](#)

Grafika i wizualizacja

- Percepcja barwy

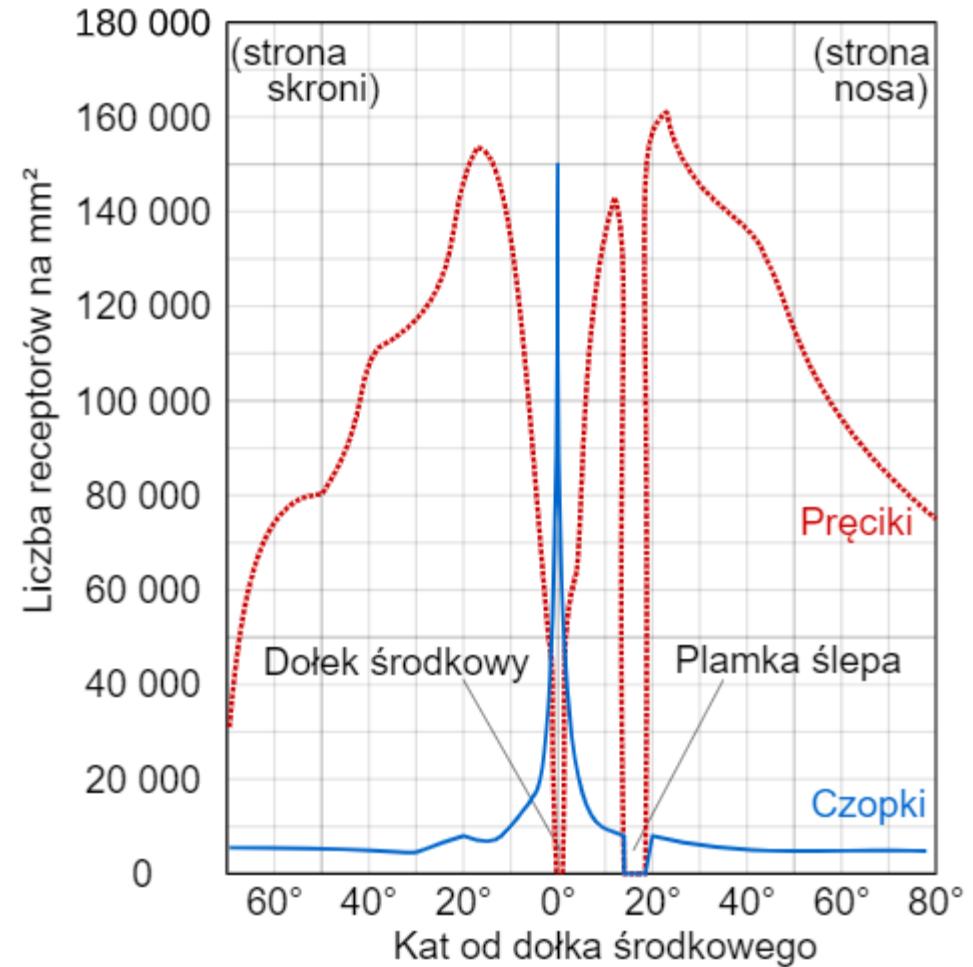
Widzenie:

- skotopowe: pręciki działają przy bardzo słabym świetle nie dając wrażeń barwnych,
- fotopowe: czopki działają przy silnym świetle, dając wrażenie barwne,
- mezopowe: oba receptory działają w zakresie ograniczonym (np. o zmierzchu).

Grafika i wizualizacja

- Percepcja barwy

Rozmieszczenie pręcików i czopków



Źródło: Cmglee – [Density of rod \(dotted line\) and cone \(solid line\) \[...\] based on 'Foundations of Vision' by Brian A. Wandell](#)

Grafika i wizualizacja

- Percepcja barwy

Rozmieszczenie czopków w pobliżu dołka środkowego osoby z normalnym widzeniem (po lewej) i z protanopią – nierozróżnianiem barwy czerwonej (po prawej)

Grafika i wizualizacja

- Percepcja barwy

Obraz pręcików i czopków w mikroskopie elektronowym



Źródło: Helga Kolb – [Adapted from 'Photoreceptors' by Helga Kolb](http://webvision.med.utah.edu/book/part-ii-anatomy-and-physiology-of-the-retina/photoreceptors/) <http://webvision.med.utah.edu/book/part-ii-anatomy-and-physiology-of-the-retina/photoreceptors/>

Grafika i wizualizacja

- Percepcja barwy

Porównanie prećików i czopków

Pręciki	Czopki
widzenie nocne	widzenie dzienne
bardzo duża czułość	niewielka czułość
rozróżnianie odcienni szarości	rozróżnianie kolorów
przeciętnie 90 milionów	przeciętnie 4,5 miliona
czułość na światło rozproszone	czułość tylko na światło bezpośrednie
niska rozdzielcość	wysoka rozdzielcość
wolna reakcja na światło	szymbka reakcja na światło

Grafika i wizualizacja

- Percepcja barwy

Teorie widzenia barwnego (opisują różne etapy przetwarzania informacji wizualnej):

- Trichromatyczna,
- przetwarzania przeciwnego.

Grafika i wizualizacja

- Percepcja barwy

Teoria trichromatyczna (teoria Younga-Helmholtza):

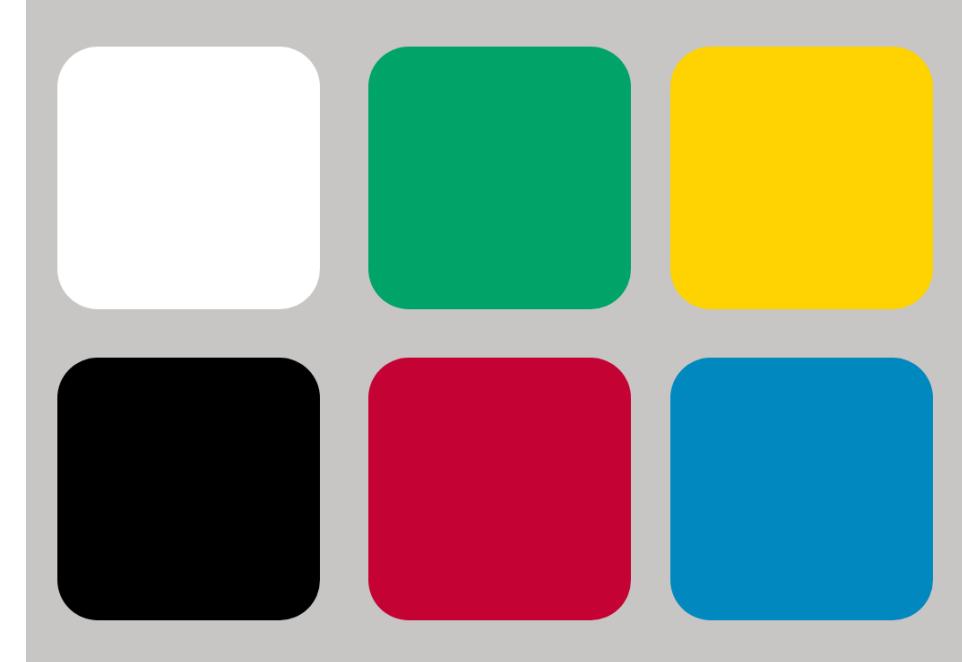
- zaproponowana została w XIX w. przez Thomasa Younga i Hermanna von Helmholtza,
- określa, że widzenie barwne powstaje na skutek łączenia informacji z trzech rodzajów receptorów w oku czułych na różne długości fal świetlnych.

Grafika i wizualizacja

- Percepcja barwy

Teoria przetwarzania przeciwnego:

- zaproponowana w 1872 roku przez Ewalda Heringa,
- określa, że układ wizualny interpretuje kolory przecistawiając sobie sygnały z receptorów,
- przeciwnostawne pary kolorów: czerwony – zielony , niebieski – żółty, czarny – biały.

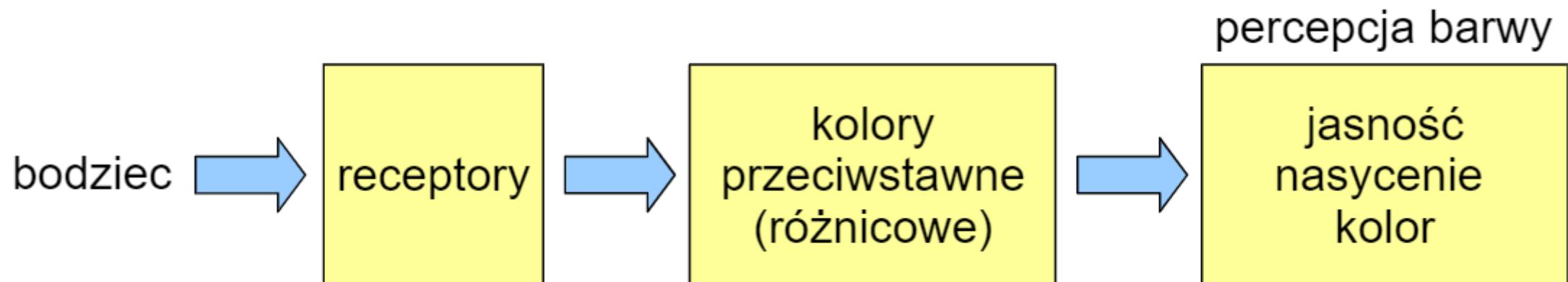


Źródło: User:Spooky – [The coordinates were taken from the “NCS Navigator” applet available from the NCS website](#)

Grafika i wizualizacja

- Percepcja barwy

Proces przetwarzania przeciwnego



Grafika i wizualizacja

- Percepcja barwy

Przesłanki prawdziwości teorii przetwarzania przeciwnego:

- wszystkie cztery kolory przeciwnostawne odbierane są jako wyjątkowo czyste,
- nie można wyobrazić sobie połączenia kolorów przeciwnostawnych (na przykład żółto-niebieskiego),
- zaburzenia w postrzeganiu kolorów często przejawiają się utratą rozróżniania kolorów przeciwnostawnych,
- kontrast równoczesny,
- kontrast następczy.

Grafika i wizualizacja

- Percepcja barwy

Kontrast równoczesny (kontrast współczesny, indukcja przestrzenna) – zwiększenie kontrastu między barwami przyległymi.

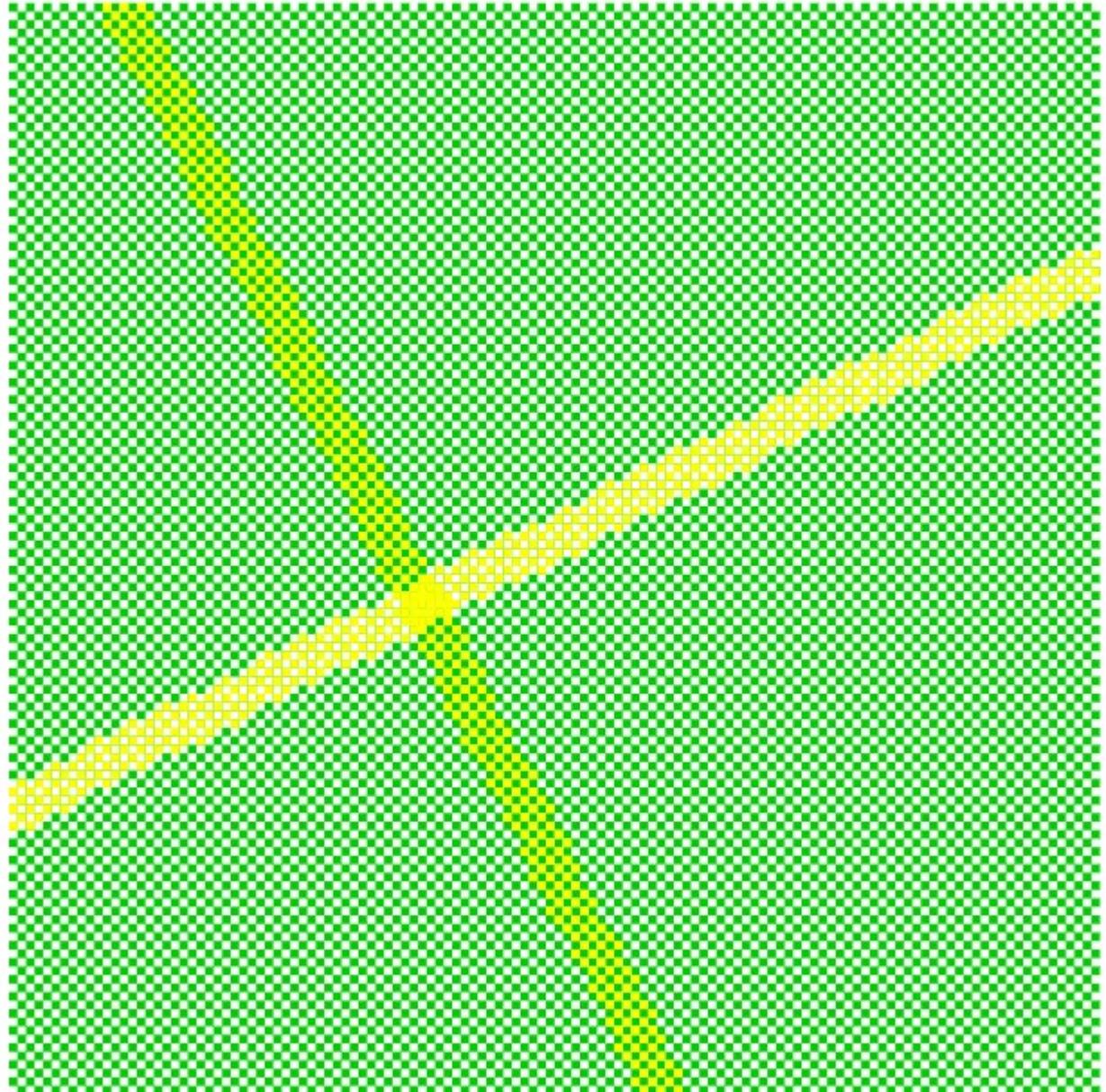
Umieszczenie szarej plamy barwnym tle powoduje, że na obrzeżach zmienia ona zabarwienie na bliskie barwom dopełniającym: na tle czerwonym będzie zielonkawa, na zielonym czerwonawa, na niebieskim żółtawa.

Podobne zjawisko obserwuje się w przypadku plamy barwnej na barwnych tła, np. żółta na tle czerwonym przybiera zabarwienie zielonożółte.

Grafika i wizualizacja

- Percepcja barwy

Kontrast równoczesny

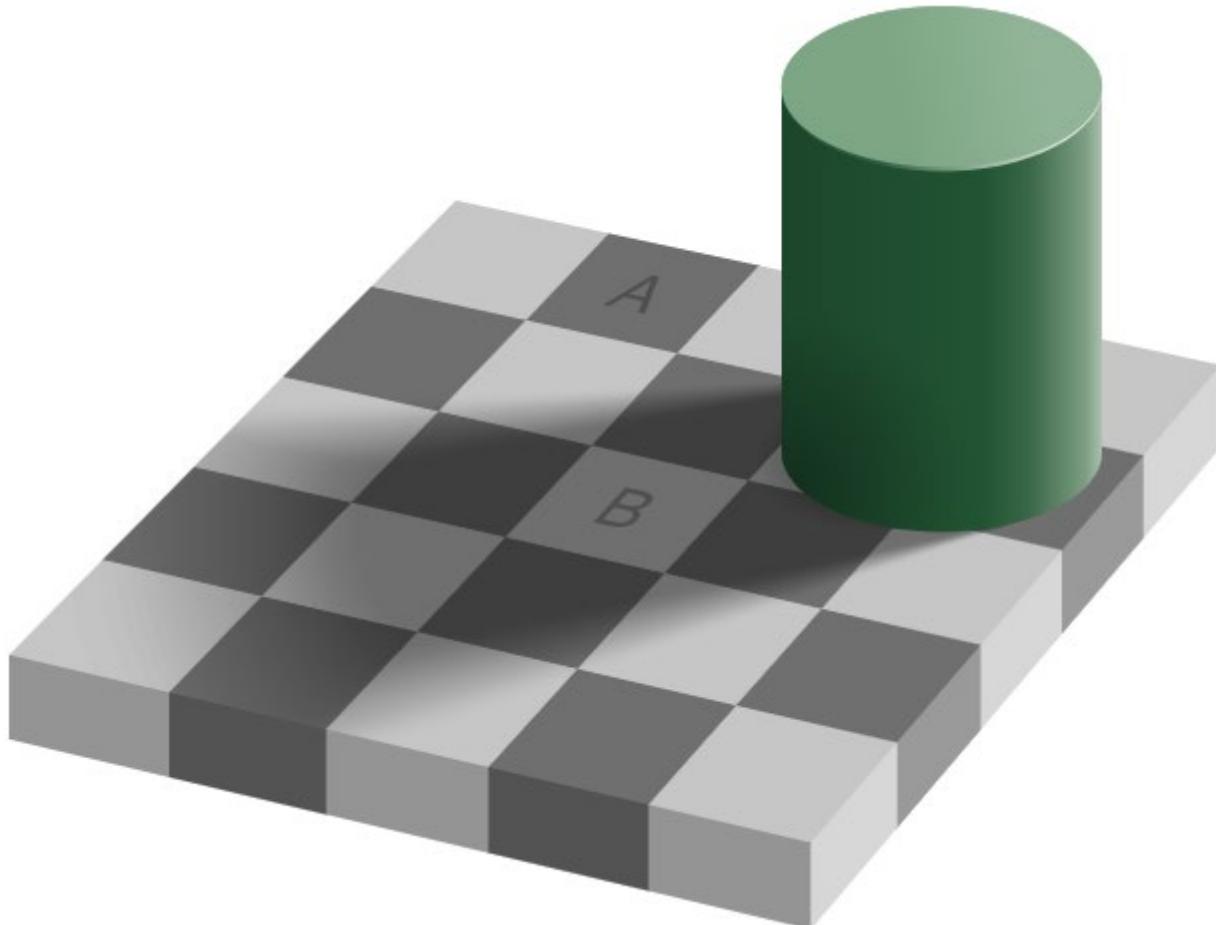


Grafika i wizualizacja

- Percepcja barwy

Kontrast równoczesny

(kwadrat A ma dokładnie ten sam odcień co B)



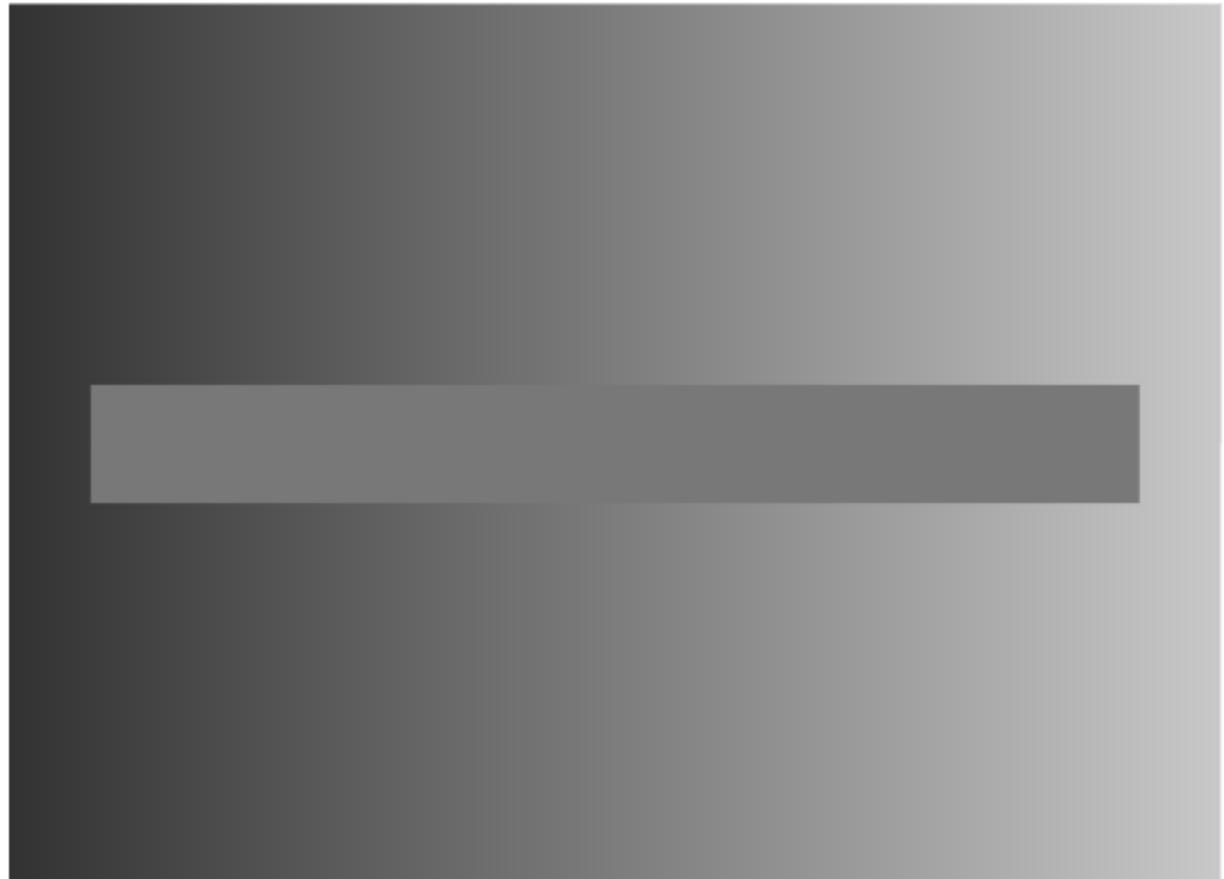
Źródło: Original: Edward H. Adelson, vectorized by Pbroks13 – [The checker shadow illusion, originally published by en: Edward H. Adelson.](#)

[Square A is exactly the same shade of grey as square B](#)

Grafika i wizualizacja

- Percepcja barwy

Kontrast równoczesny



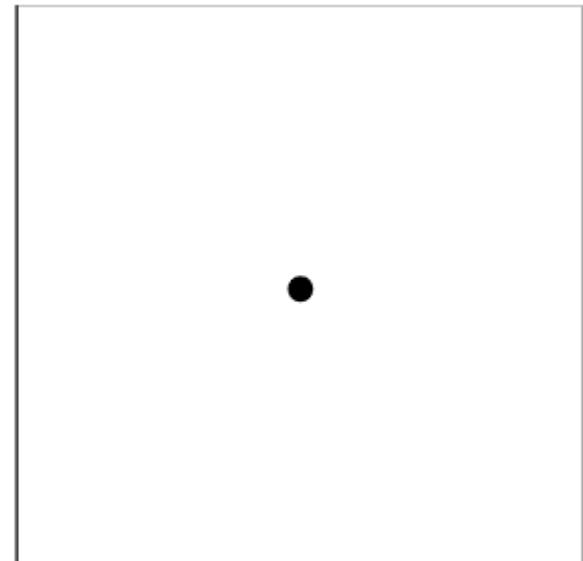
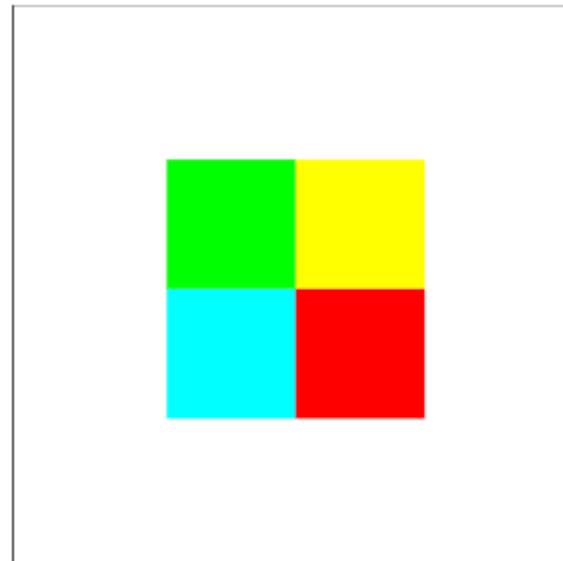
Źródło: Dodek – [Simultaneous Contrast Illusion. The horizontal grey bar is the same shade throughout](#)

Grafika i wizualizacja

- Percepcja barwy

Kontrast następczy – po zamianie kolorowego obrazu na białą planszę widzimy go przez pewien czas w kolorach dopełniających. Powstaje w wyniku "zmęczenia" receptorów, które po zaniknięciu bodźca słabiej reagują niż receptory, które nie były pobudzone.

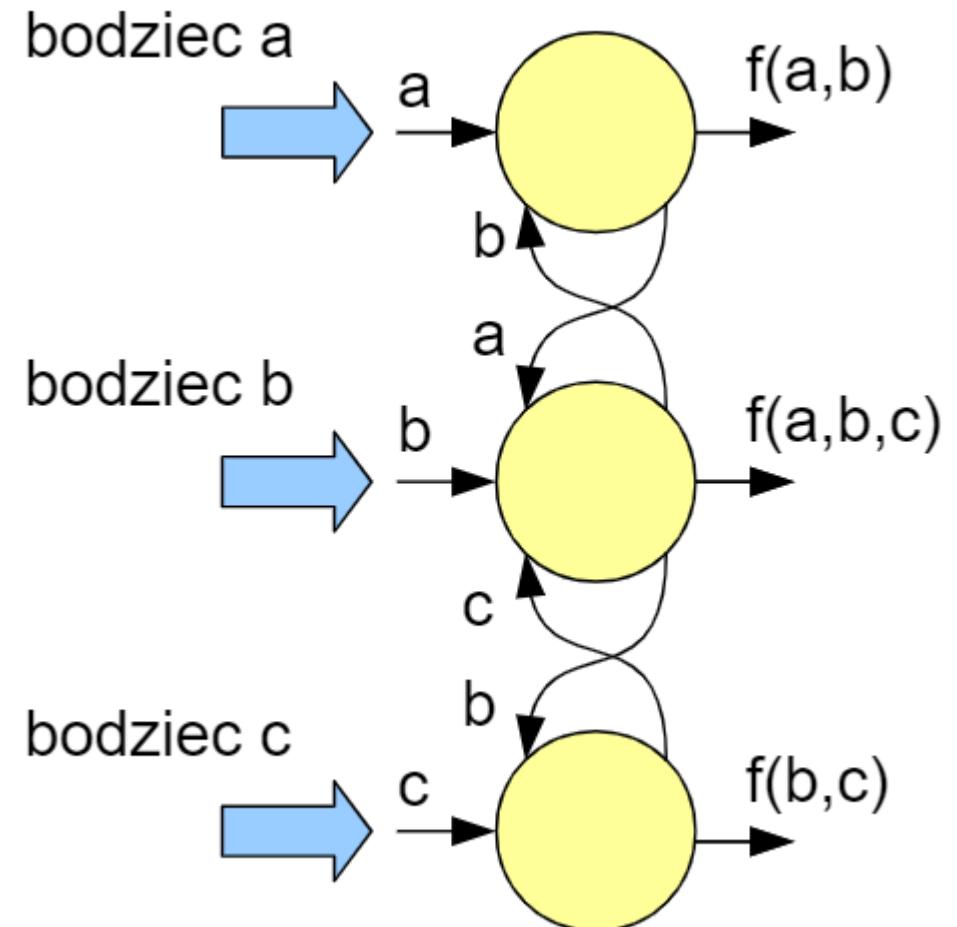
Należy wpatrywać się przez 30 sekund w środek kwadratu, a potem przenieść wzrok na kropkę.



Grafika i wizualizacja

- Percepcja barwy

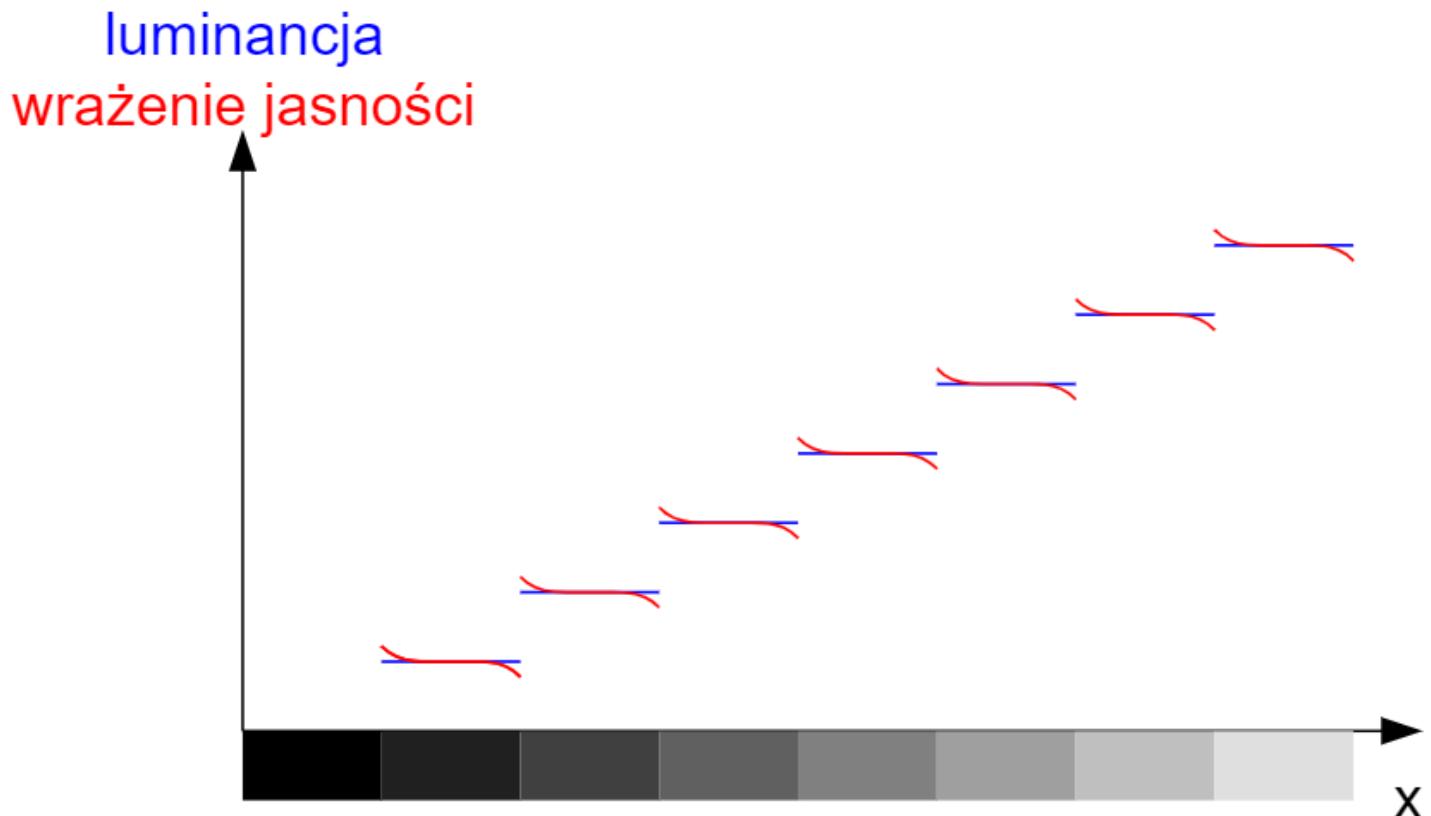
Hamowanie oboczne jest to zdolność neuronów do zmniejszania poziomu swojej reakcji pod wpływem sąsiadów. Przykładowo neuron pobudzony bodźcem b przesyła dalej informację o pobudzniu pomniejszoną o wartość zależną od bodźców a i c.



Grafika i wizualizacja

- Percepcja barwy

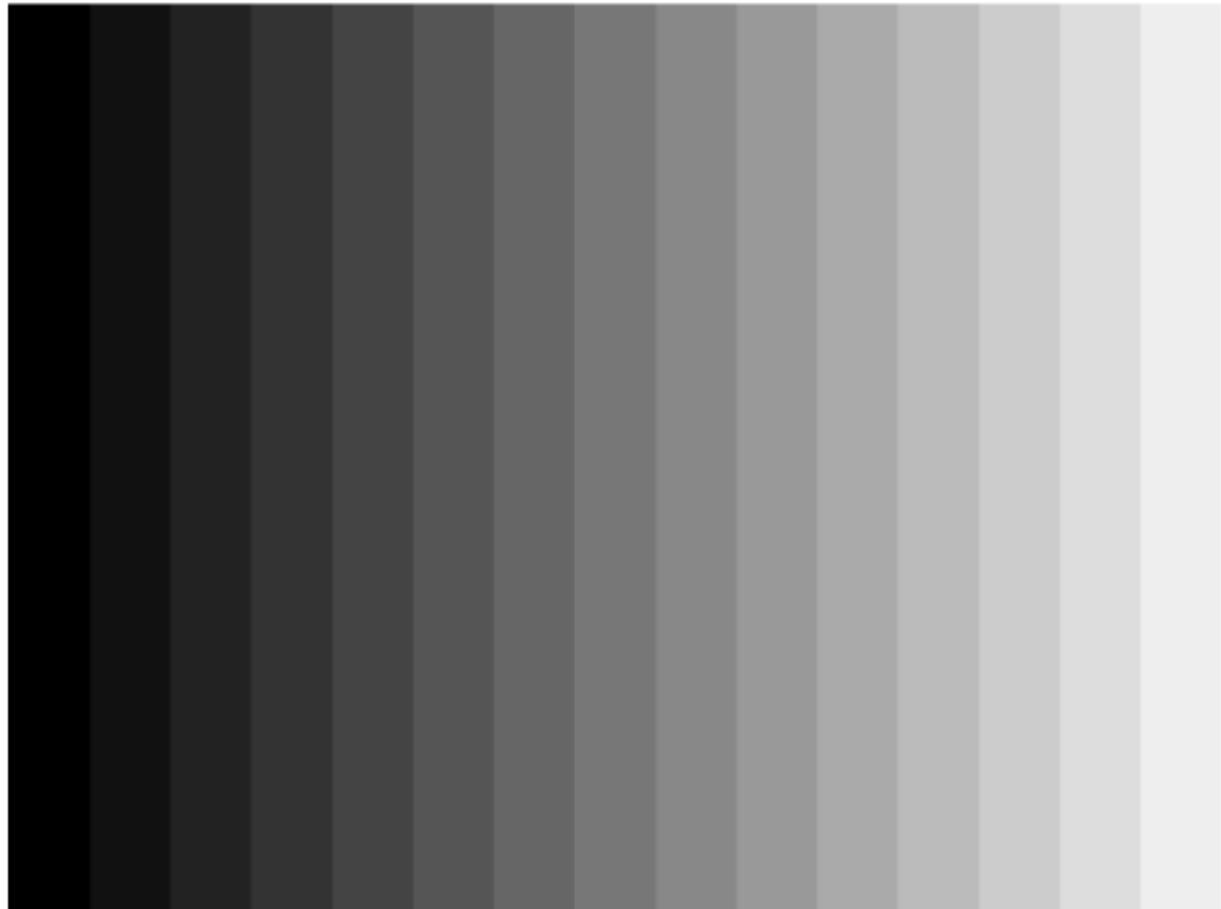
Reakcja receptorów odbierających barwę jaśniejszą jest słabiej tłumiona, a ich sąsiadów, receptorów odbierających barwę ciemniejszą – silniej tłumiona.



Grafika i wizualizacja

- Percepcja barwy

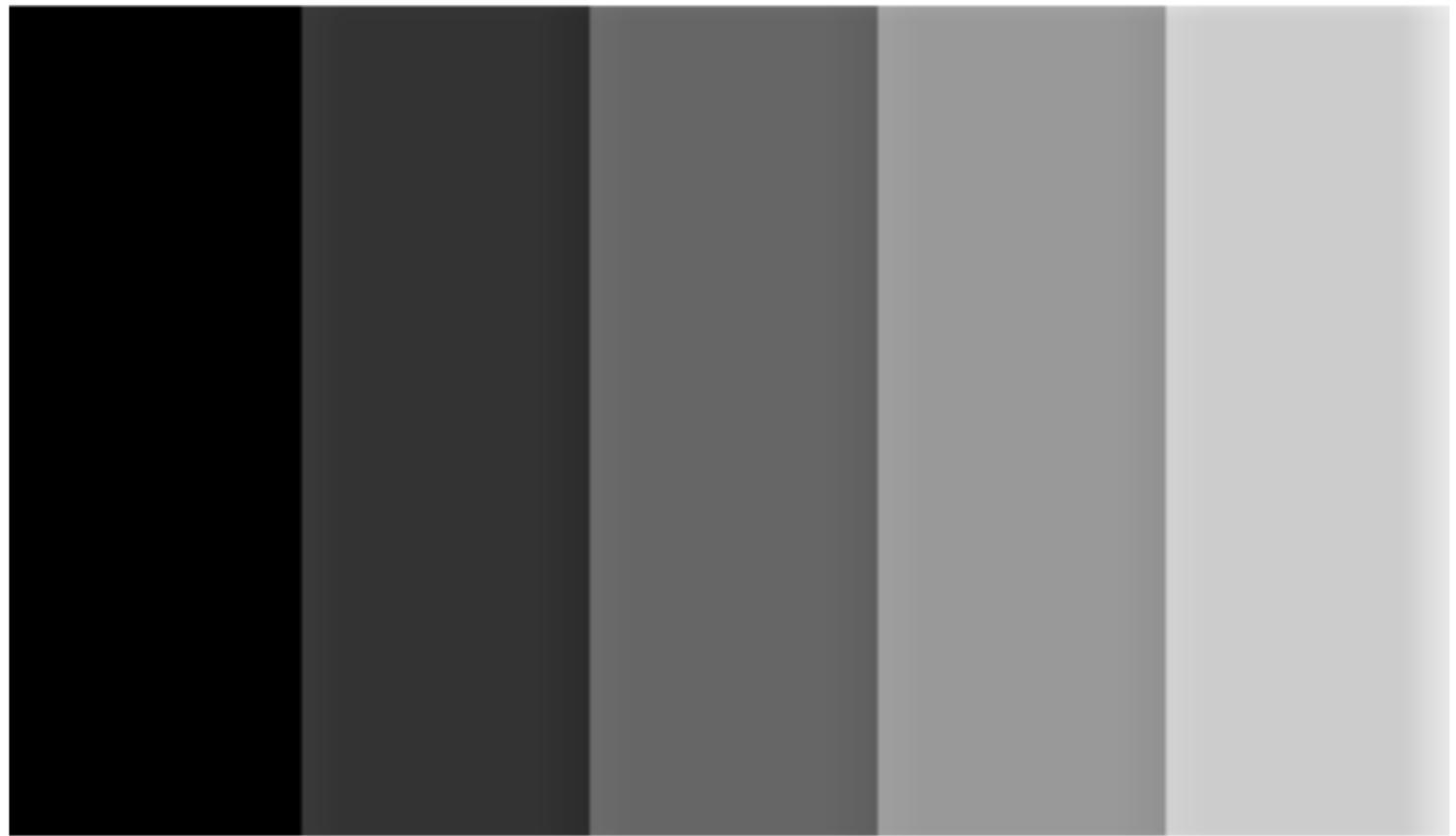
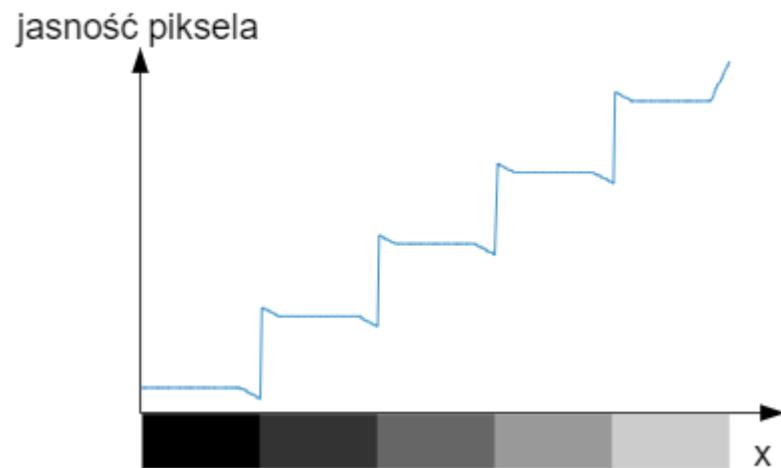
Efektem hamowania obecnego są pasma Macha. Pasma Macha to jaśniejsze i ciemniejsze pasma na krawędziach dwóch stykających się obszarów o jednolitych kolorach.



Grafika i wizualizacja

- Percepcja barwy

Symulacja pasm Macha



Grafika i wizualizacja

- Percepcja barwy

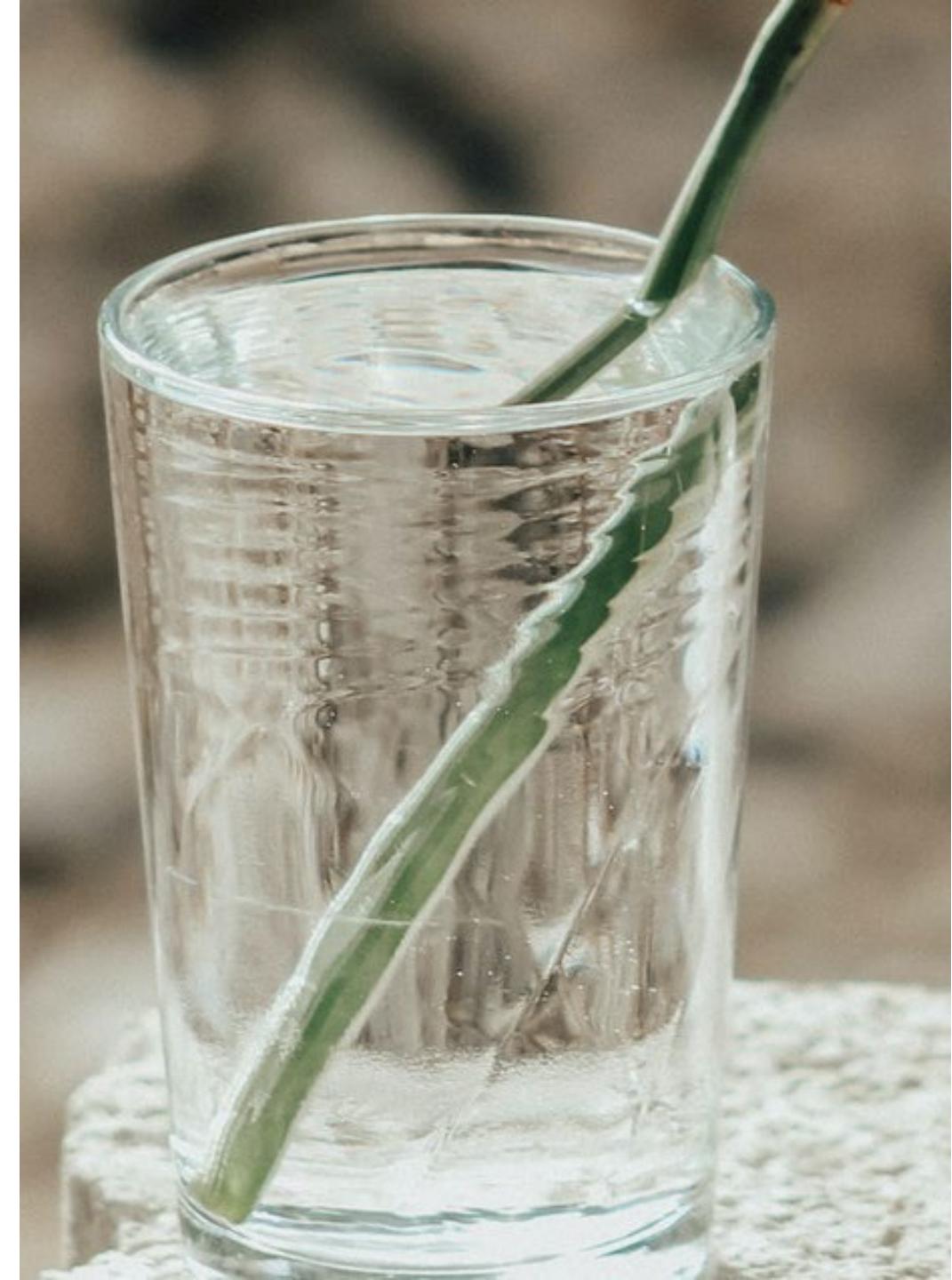
Złudzenia optyczne – błędna interpretacja obrazu przez mózg. Klasy złudzeń optycznych zaproponowana przez Richarda Gregory'ego:

- złudzenia fizyczne,
- złudzenia fizjologiczne,
- złudzenia poznawcze.

Grafika i wizualizacja

- Percepcja barwy

Złudzenia fizyczne – złudzenia wywoływanie przez zjawiska fizyczne, które powodują, że informacja jest przez mózg błędnie interpretowana. Przykładem może oświetlenie obracającego się obiektu światłem stroboskopowym zsynchronizowanym z jego obrotem. Powoduje to wrażenie, że obiekt jest nieruchomy. Innym przykładem jest patyk załamujący się w wodzie.



Fragment zdjęcia z [Pexels przedstawiającego szklankę](#) wykonane przez [daboul](#).

Grafika i wizualizacja

- Percepcja barwy

Przykładem fizycznego złudzenia optycznego jest fatamorgana.

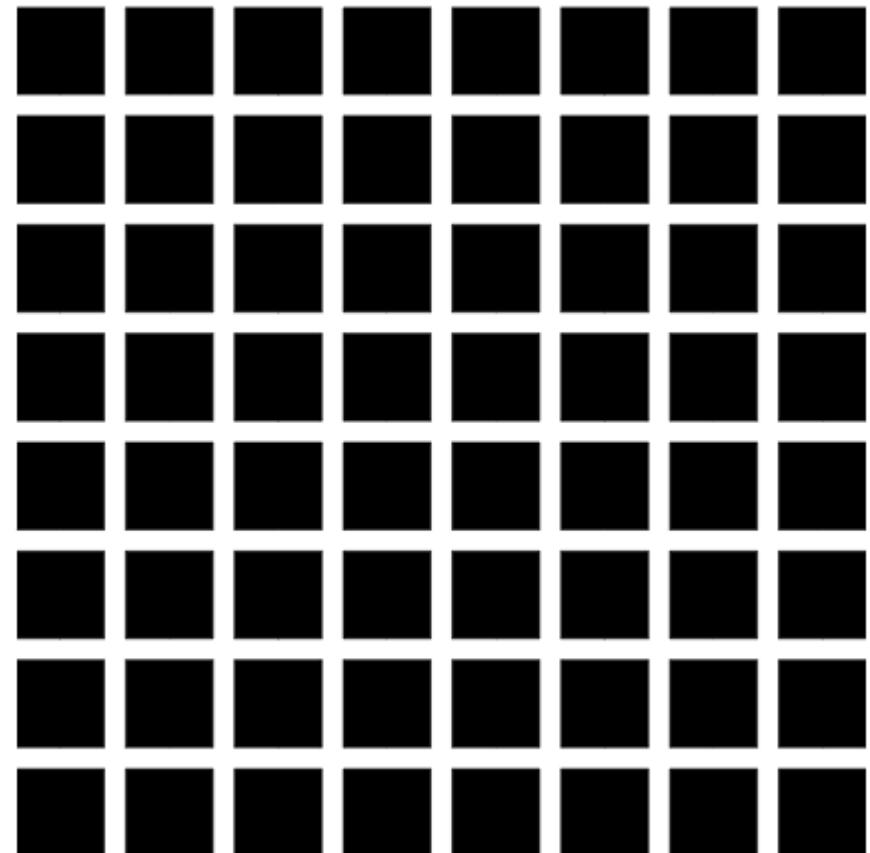


Zdjęcie z [Unsplash'a](#) [fatamorgany](#) wykonane
przez [Mathieu Perrier](#)

Grafika i wizualizacja

- Percepcja barwy

Złudzenia fizjologiczne – złudzenia związane z pracą receptorów oraz bezpośrednio z nimi związkanych neuronów. Przykładem złudzeń fizjologicznych są omawiane wcześniej pasma Macha. Innym przykładem może być siatka Hermanna przedstawiona przez Ludimara Hermanna w 1870. Na skrzyżowaniach białych linii pojawiają się szare kropki.



Grafika i wizualizacja

- Percepcja barwy

Złudzenia poznawcze – złudzenia powstające w wyniku błędnej interpretacji obrazu.

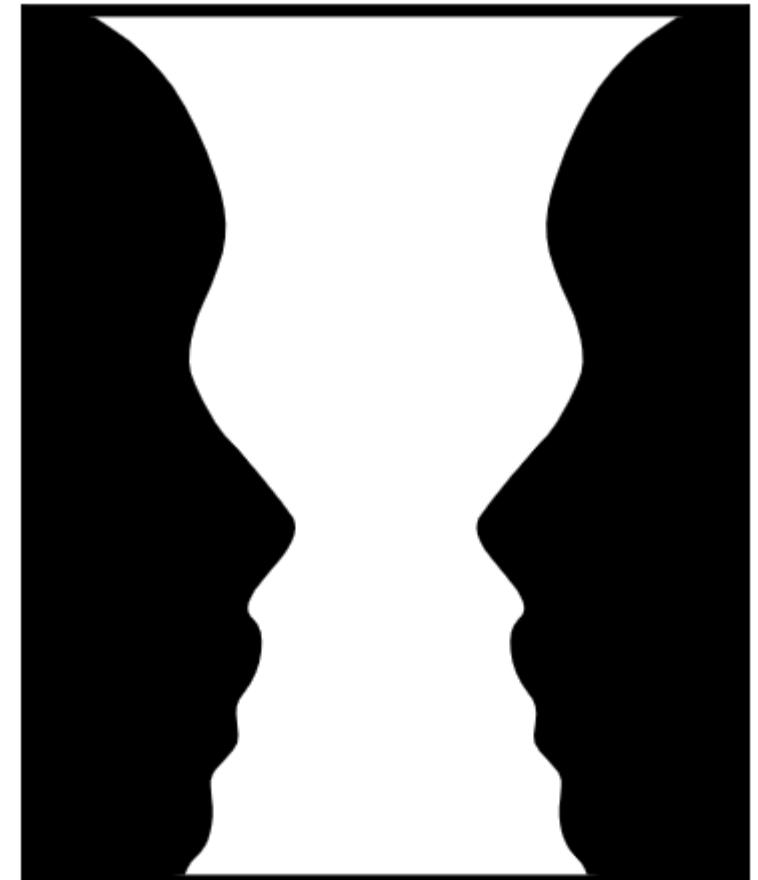
Wyróżnia się:

- złudzenia dwuznaczności;
- złudzenia geometryczno-optyczne;
- figury niemożliwe;
- pareidolia.

Grafika i wizualizacja

- Percepcja barwy

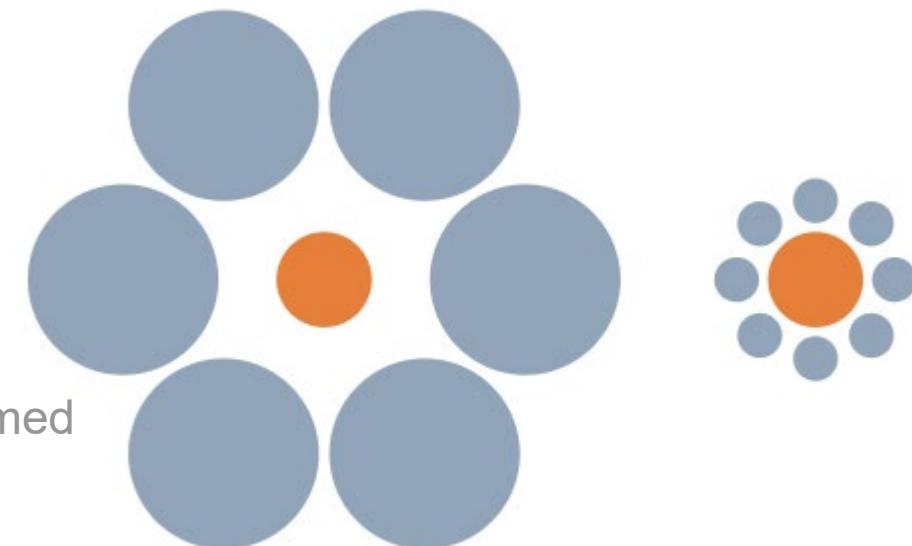
Złudzenia dwuznaczności – obrazy, które mogą być różnie interpretowane, co powoduje, "przełączanie się" między różnymi interpretacjami. Przykładem może być wazon Rubina opracowany około 1915 r. przez duńskiego psychologa Edgara Rubina. Obraz może być interpretowany jako wazon lub dwie twarze.



Grafika i wizualizacja

- Percepcja barwy

Złudzenia geometryczno-optyczne – złudzenia deformujące kształt, wielkość obiektów oraz zmieniające ich położenie. Przykładem jest złudzenie Ebbinghausa odkryte przez niemieckiego psychologa Hermanna Ebbinghausa dotyczące błędnego określania wielkości. Dwa pomarańczowe koła wydają się różnej wielkości mimo, że faktycznie są dokładnie takie same.



Źródło: No machine-readable author provided. Phrood~commonswiki assumed
(based on copyright claims) –
[Optical illusion: The two orange circles are the same size](#)

Grafika i wizualizacja

- Percepcja barwy

Pokój Amesa – złudzenia geometryczno-optyczne wynalezione w 1946 roku przez amerykańskiego naukowca Adelberta Ames Jr. Efekt specjalny używany w filmach

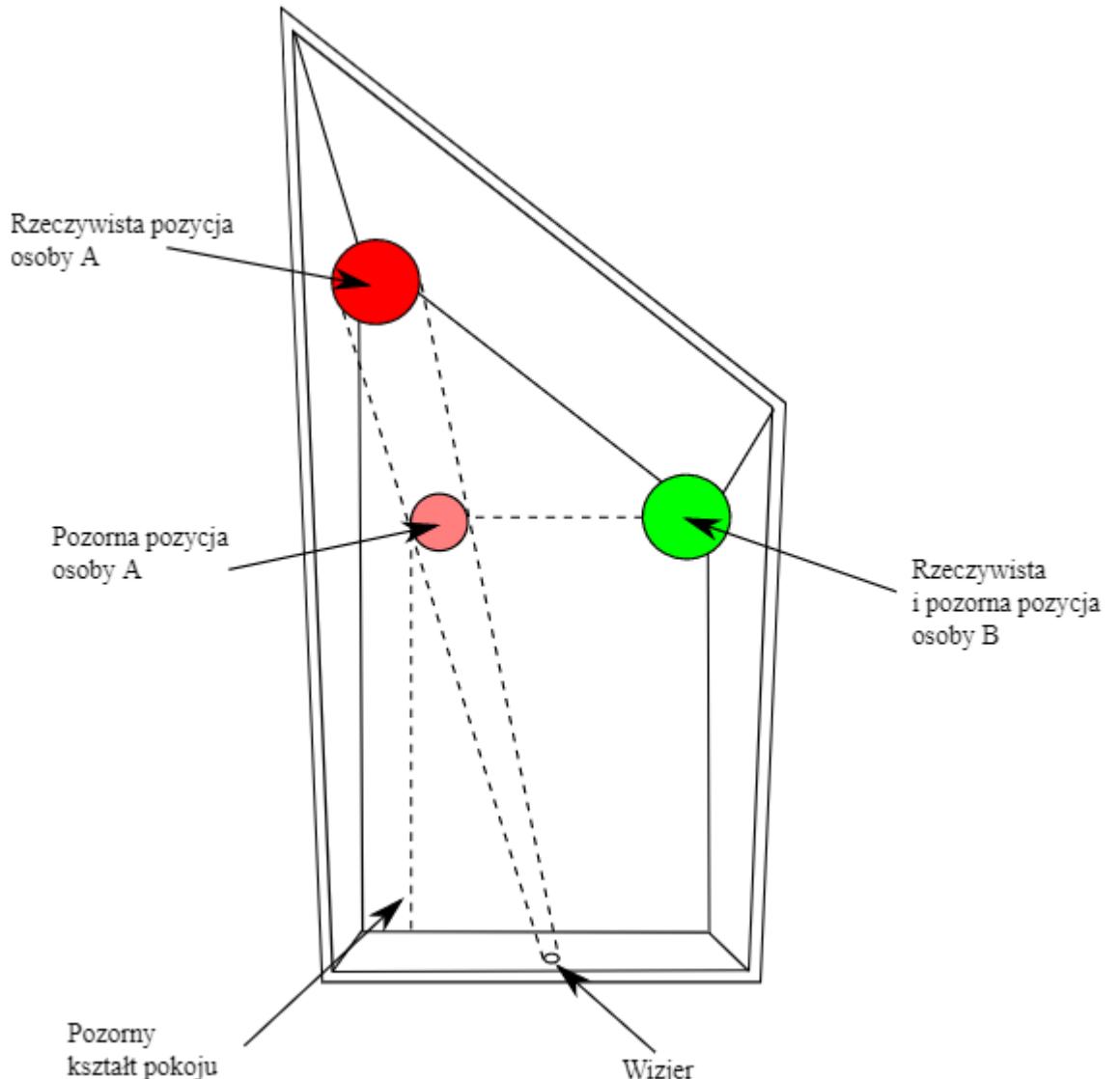


https://upload.wikimedia.org/wikipedia/commons/thumb/1/11/Ames_room.ogv/640px--Ames_room.ogv.jpg

Grafika i wizualizacja

- Percepcja barwy

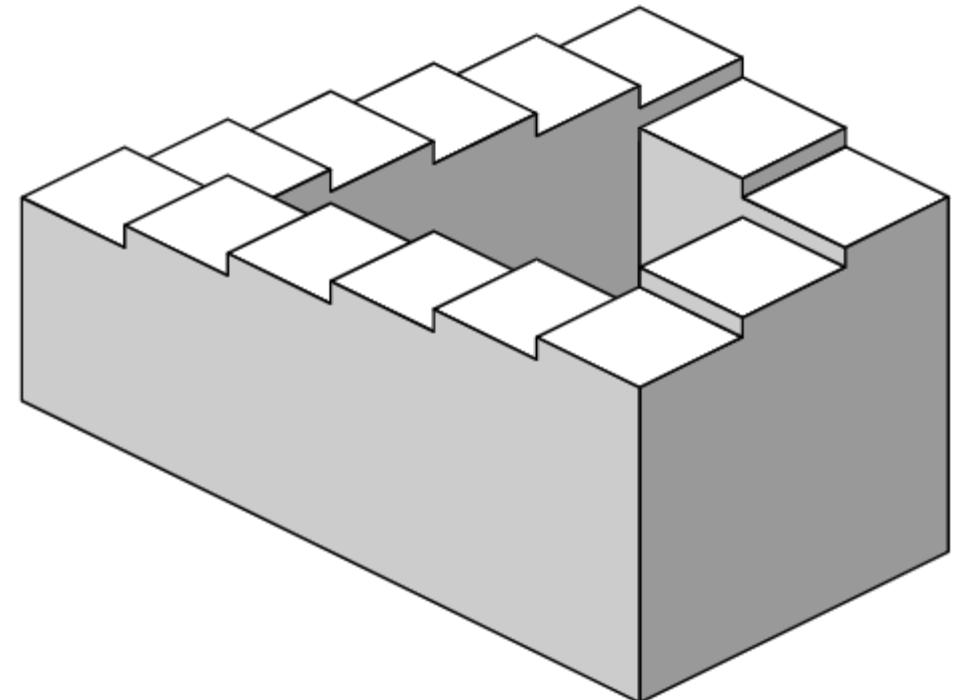
Iluzję pokoju Amesa wywołuje specjalny układ ścian pomieszczenia oraz odpowiednie rozmiary przedmiotów i rysunki na powierzchniach.



Grafika i wizualizacja

- Percepcja barwy

Figury niemożliwe – wizualizacje figur przestrzennych niemożliwych do stworzenia. Przykładem figury niemożliwej są schody Penrose'a (niemożliwe schody) stworzone niezależnie przez szwedzkiego artystę Oscara Reutersvärda (1937 r.) i Brytyjczyków Lionela Penrose'a oraz jego syna Rogera Penrose'a (1958 r.).



Grafika i wizualizacja

- Percepcja barwy

Trójkąt Penrose'a – figura niemożliwa stworzone niezależnie przez szwedzkiego artystę Oscara Reutersvärda (1936 r.) i Brytyjczyków Lionela Penrose'a oraz jego syna Rogera Penrose'a (lata 50-te).



Źródło: Bjørn Christian Tørrissen – [Impossible Triangle sculpture, put together by Brian MacKay](#)

Grafika i wizualizacja

- Percepcja barwy

Pareidolia – nieprawidłowa interpretacja kształtów powodująca "widzenie" dobrze znanych obserwatorowi obiektów



Grafika i wizualizacja

- Modele barw

Model barw – określony trój- lub więcej wymiarowy system współrzędnych z pewnym wydzielonym obszarem przestrzeni w którym leżą wszystkie rozpatrywane barwy.

Grafika i wizualizacja

- Modele barw

Prawo Helmholtza – dodanie dowolnych dwóch kolorów prostych równoważne jest wzrokowo dodaniu światła białego i światła czystego:

$$L_1 + L_2 = L_b + L_c ,$$

gdzie L_1 , L_2 są to luminancje promieniowania monochromatycznych, L_b – luminancja światła białego, a L_c – luminancja światła czystego.

Przy czym przez światło czyste należy rozumieć kolor o maksymalnym nasyceniu.

Grafika i wizualizacja

- Modele barw

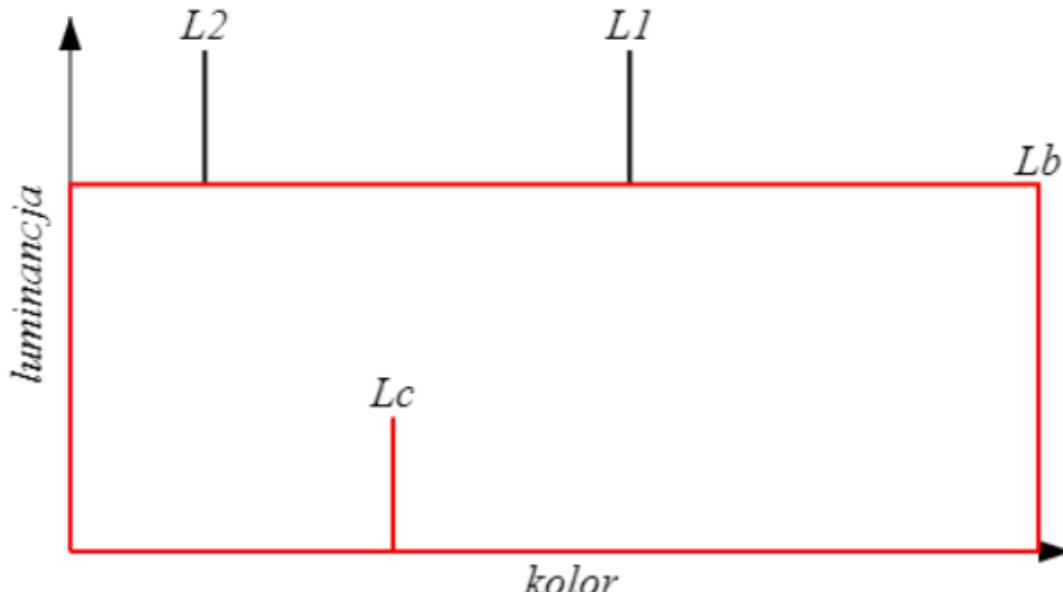
L_1 : min  max **0088FF**

L_2 : min  max **FFD500**

L_b : **BBBBBB**

L_c : **004400**

Kolor: **BBFFBB**



Prawo Helmholtza

Grafika i wizualizacja

- Modele barw

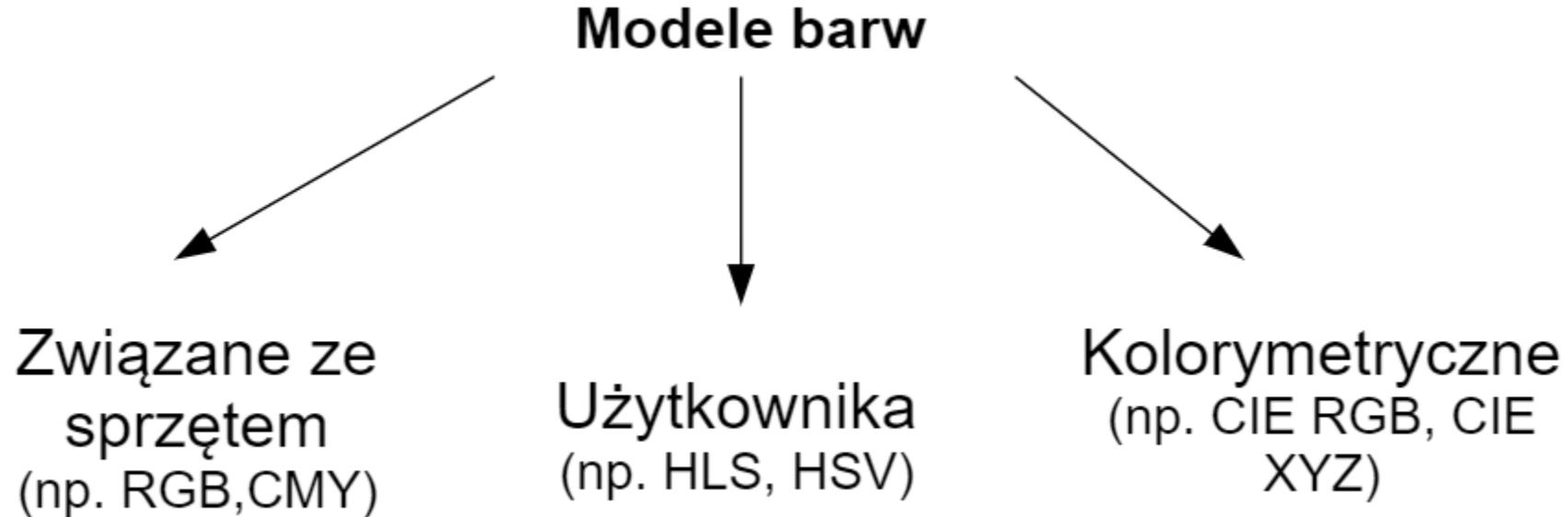
I prawo Grassmanna (prawo trójchromatyczności) – każda dowolna barwa może być odwzorowana za pomocą trzech barw niezależne kolorymetrycznie. W rachunku wektorowym niezależność kolorymetryczną utożsamiamy z liniową niezależnością wektorów.

II prawo Grassmanna (prawo ciągłości) – w mieszaninie złożonej z dwóch barw stopniowa zmiana barwy jednego ze składników pociąga za sobą zmianę barwy mieszaniny.

III prawo Grassmanna (prawo addytywności) – wrażenie barwy mieszaniny kolorów zależy od wrażenia barw z których powstały. Oznacza to, że jeżeli jeden kolor w mieszaninie kolorów zastąpimy innym o tym samym wrażeniu, a innym składzie widmowym to wrażenie koloru mieszaniny nie ulegnie zmianie.

Grafika i wizualizacja

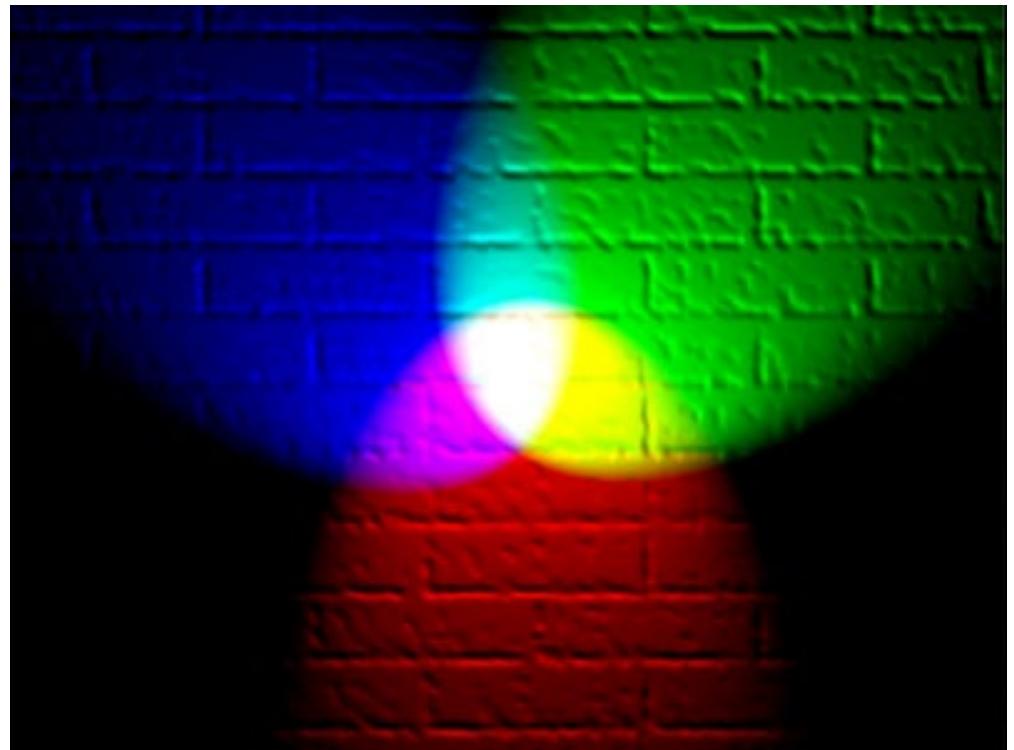
- Modele barw



Grafika i wizualizacja

- Modele barw

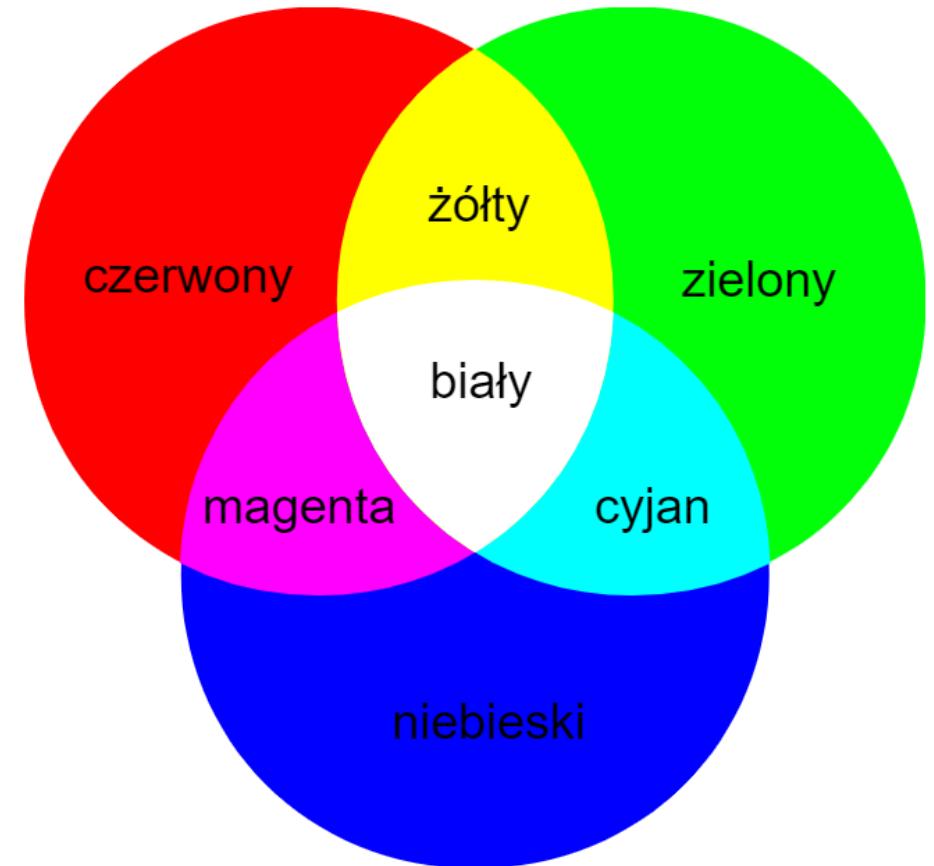
Model barw RGB – model addytywny. Każda barwa powstaje poprzez zsumowanie w różnych proporcjach światła w kolorze czerwonym (R – ang. red), zielonym (G – ang. green) oraz niebieskim (B – ang. blue).



Grafika i wizualizacja

- Modele barw

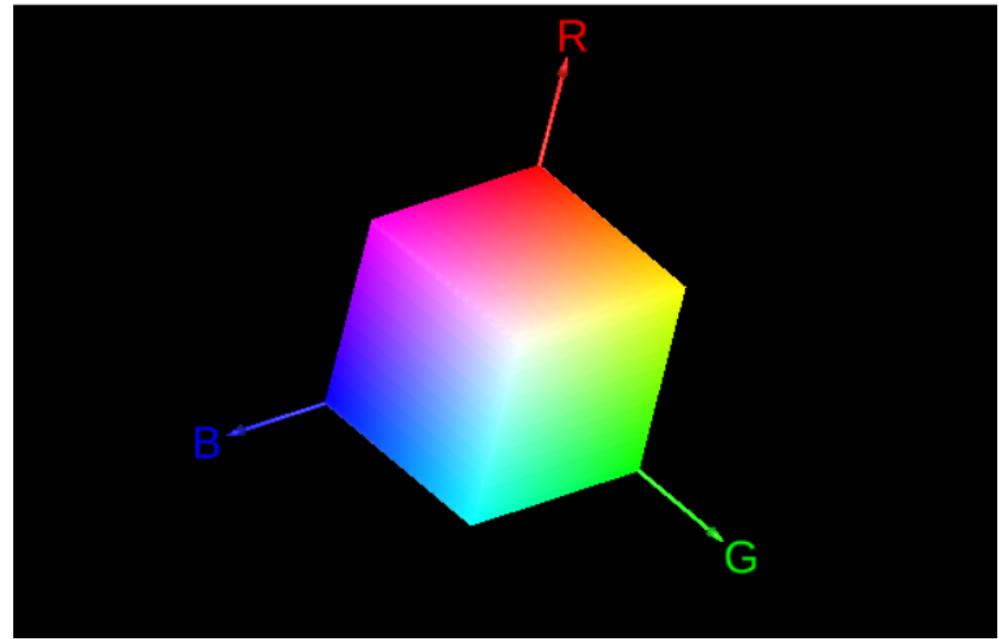
Model barw RGB



Grafika i wizualizacja

- Modele barw

Model barw RGB



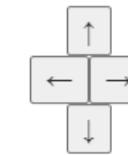
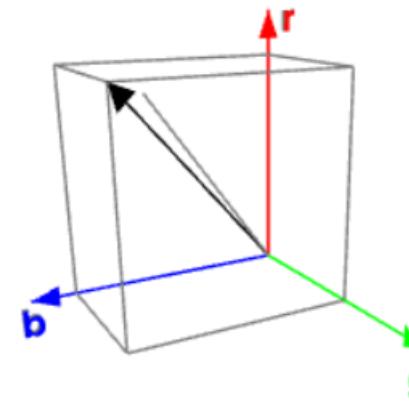
Grafika i wizualizacja

- Modele barw

W modelu barw RGB osiami układu współrzędnych są wektory reprezentujące kolory czerwony, zielony i niebieski. Kolor określa się podając wartości składowych czerwonej, zielonej i niebieskiej.

min max R = 255
min max G = 171
min max B = 255

przestrzeń barw RGB kolor

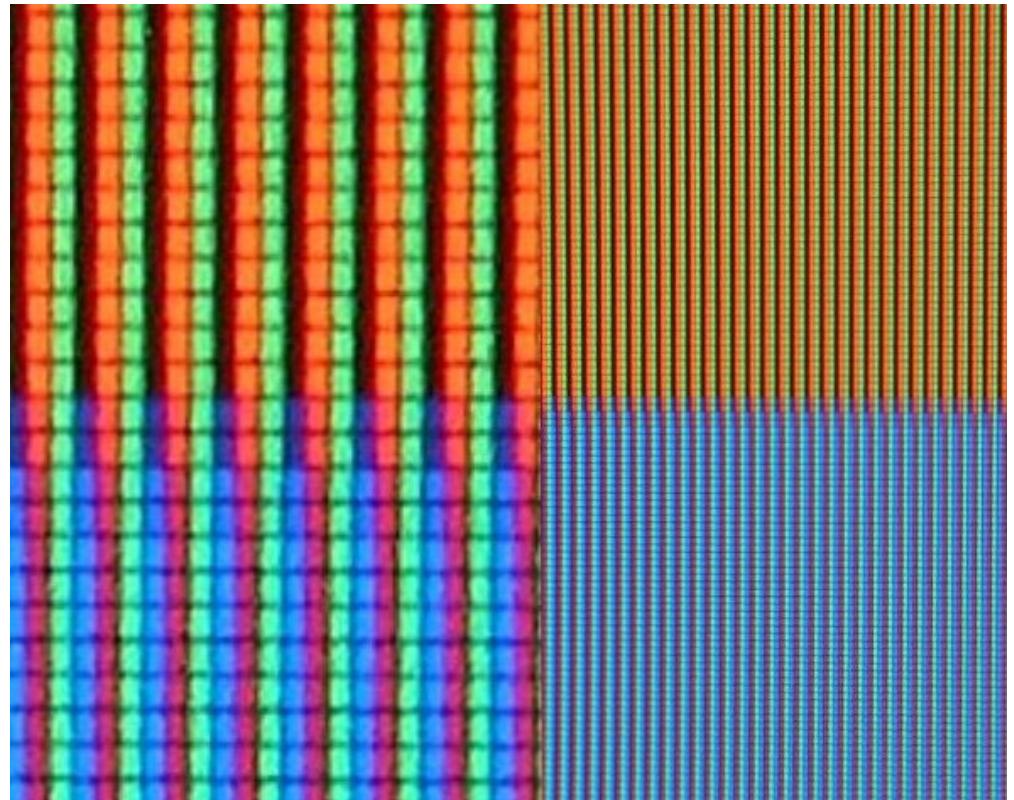


FFABFF

Grafika i wizualizacja

- Modele barw

Model barw RGB w urządzeniach technicznych wykorzystuje się głównie w monitorach i telewizorach.



Źródło: Stan Zurek – [RGB pixels of an LCD television set. On the right - two colours \(orange and blue\), on the left - close up of the pixels](#)

Grafika i wizualizacja

- Modele barw

Model barw RGB – rozbarwienie



R

G

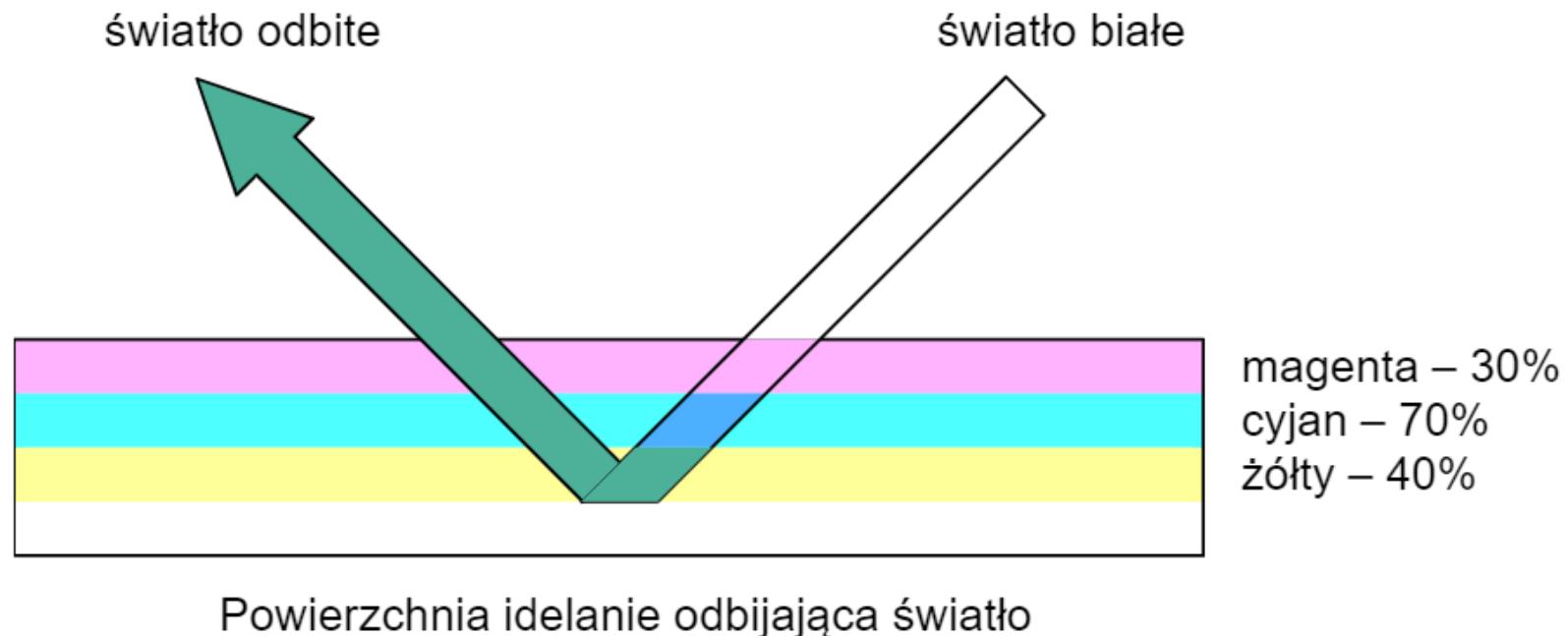
B



Grafika i wizualizacja

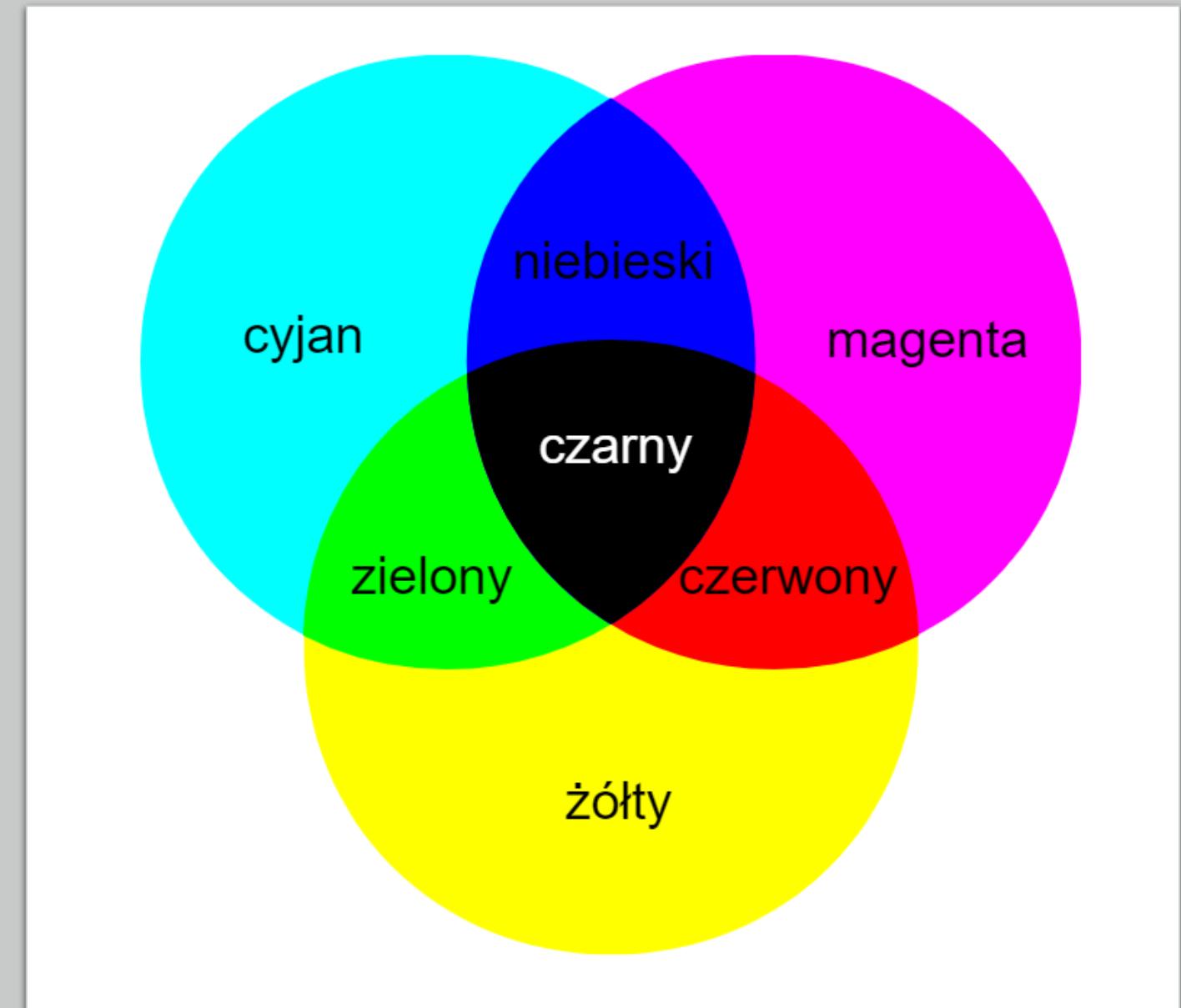
- Modele barw

Model barw CMY – model subtraktynny. Każda barwa powstaje poprzez usunięcie ze światła białego w różnych proporcjach koloru cyjany (C – ang. cyan), magenty (M – ang. magenta) oraz żółtego (Y – ang. yellow)



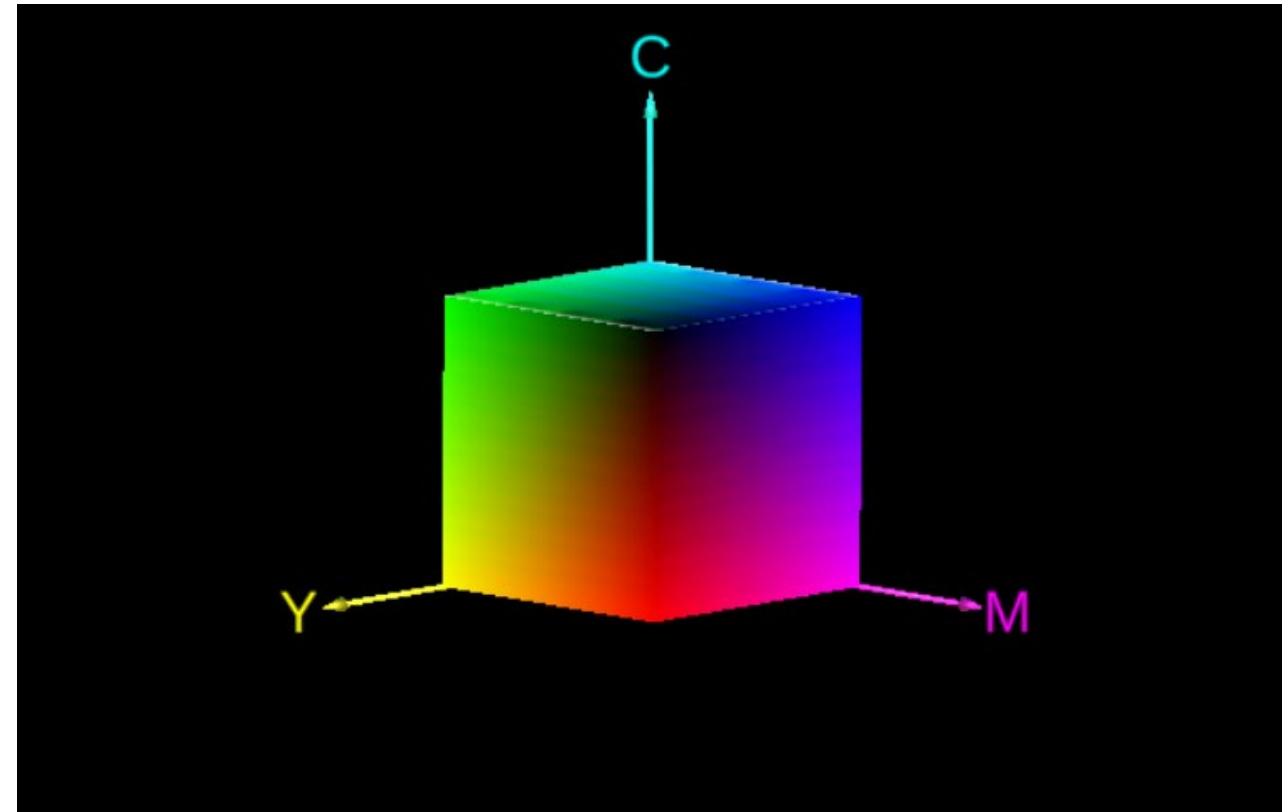
Grafika i wizualizacja - Modele barw

Model barw CMY



Grafika i wizualizacja - Modele barw

Model barw CMY



Grafika i wizualizacja

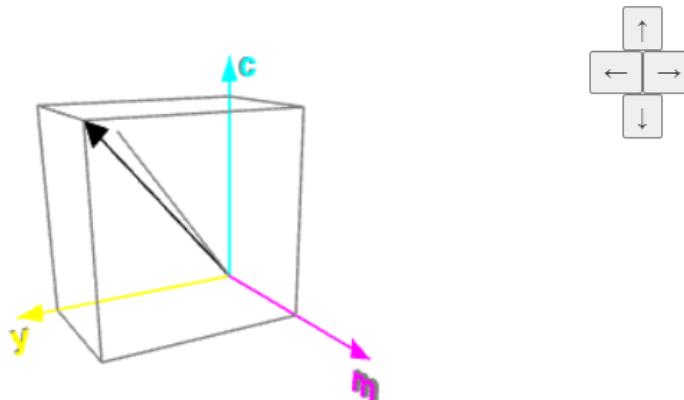
- Modele barw

W modelu barw CMY osiami układu współrzędnych są wektory reprezentujące kolory cyjan, magenta i żółty. Kolor określa się podając wartości składowych cyjanu, magenty i żółtej.

min		max	C =	255
min		max	M =	167
min		max	Y =	255

przestrzeń barw RGB

kolor (wartości w modelu RGB)



Grafika i wizualizacja

- Modele barw

Przejście z modelu RGB do CMY i z CMY do RGB:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

gdzie R, G, B to składowe kolorów modelu RGB, a C, M, Y – składowe kolorów modelu CMY.

Grafika i wizualizacja

- Modele barw

Model barw CMY – rozbarwienie



M

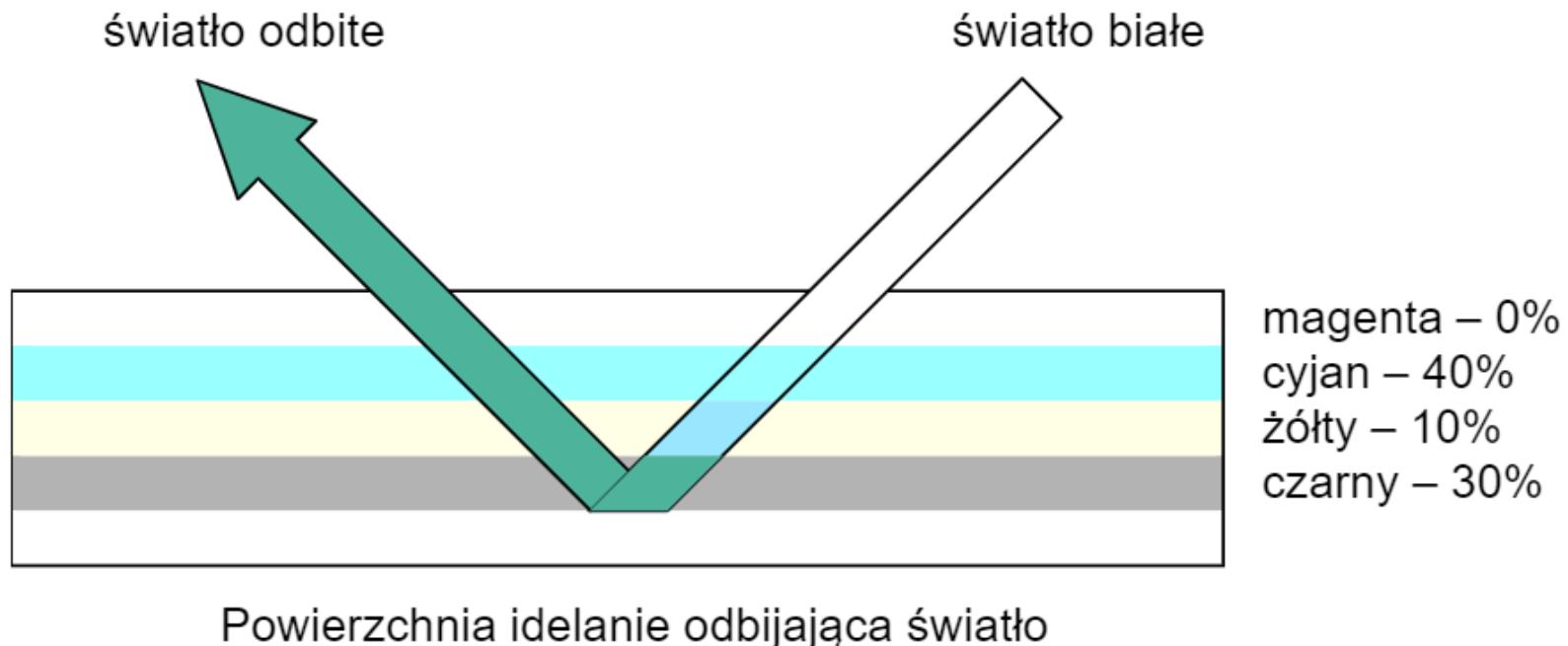
C



Grafika i wizualizacja

- Modele barw

Model barw CMYK – model subtraktywny. Każda barwa powstaje poprzez usunięcie ze światła białego w różnych proporcjach koloru cyjany (C – ang. cyan), magenty (M – ang. magenta), żółtego (Y – ang. yellow), oraz czarnego (K).



Grafika i wizualizacja

- Modele barw

Zależność pomiędzy modelami CMY i CMYK

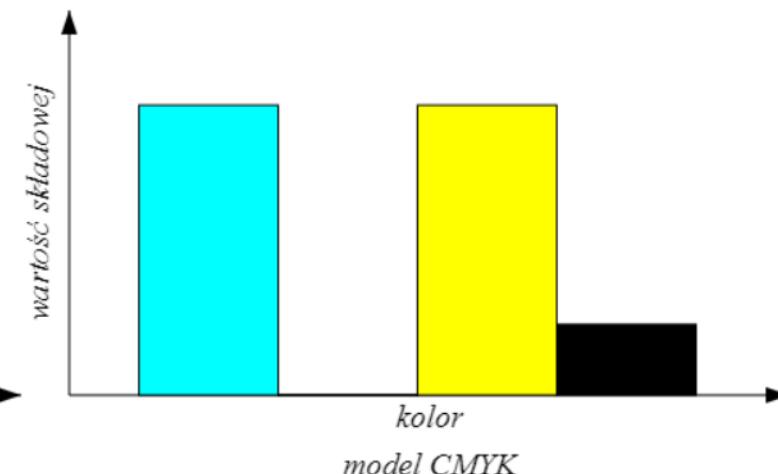
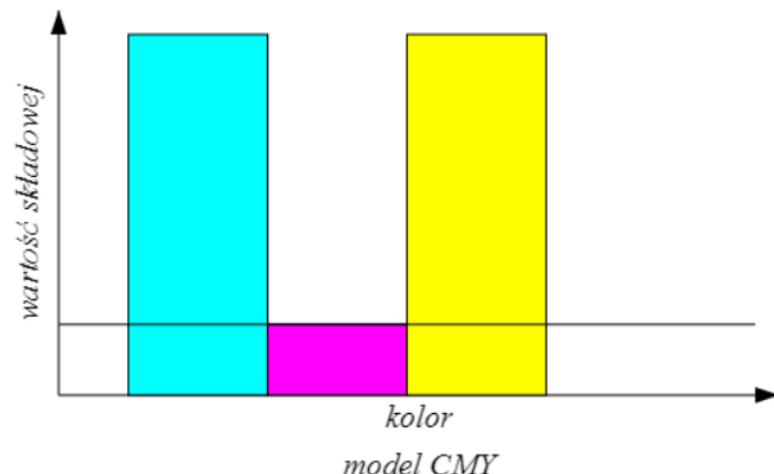
min max C = 255

min max M = 50

min max Y = 255

Kolor:

00CD00



Grafika i wizualizacja

- Modele barw

Przejście z modelu CMY do CMYK:

$$\begin{cases} K' = \min(C, M, Y) \\ C' = C - K' \\ M' = M - K' \\ Y' = Y - K' \end{cases},$$

gdzie C, M, Y to składowe kolorów modelu CMY, a C', M', Y', K' – składowe kolorów modelu CMYK.

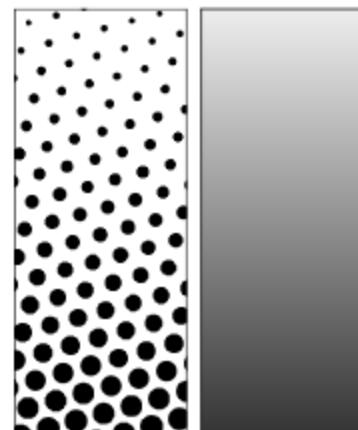
Przejście z modelu CMYK do CMY:

$$\begin{cases} C = C' + K' \\ M = M' + K' \\ Y = Y' + K' \end{cases}.$$

Grafika i wizualizacja

- Modele barw

Model barw CMYK stosowany jest w druku. Na ogół stosuje się farby o maksymalnym natężeniu koloru. Dlatego różne natężenie koloru uzyskuje się tzw. raster, który pozwala na symulowanie różnych natężeń koloru. Odbywa się to przez drukowanie kropek w jednolitym kolorze, w równych odstępach i o różnych rozmiarach lub w tych samych rozmiarach ale w różnych odstępach. Czasami łączy się obie metody.

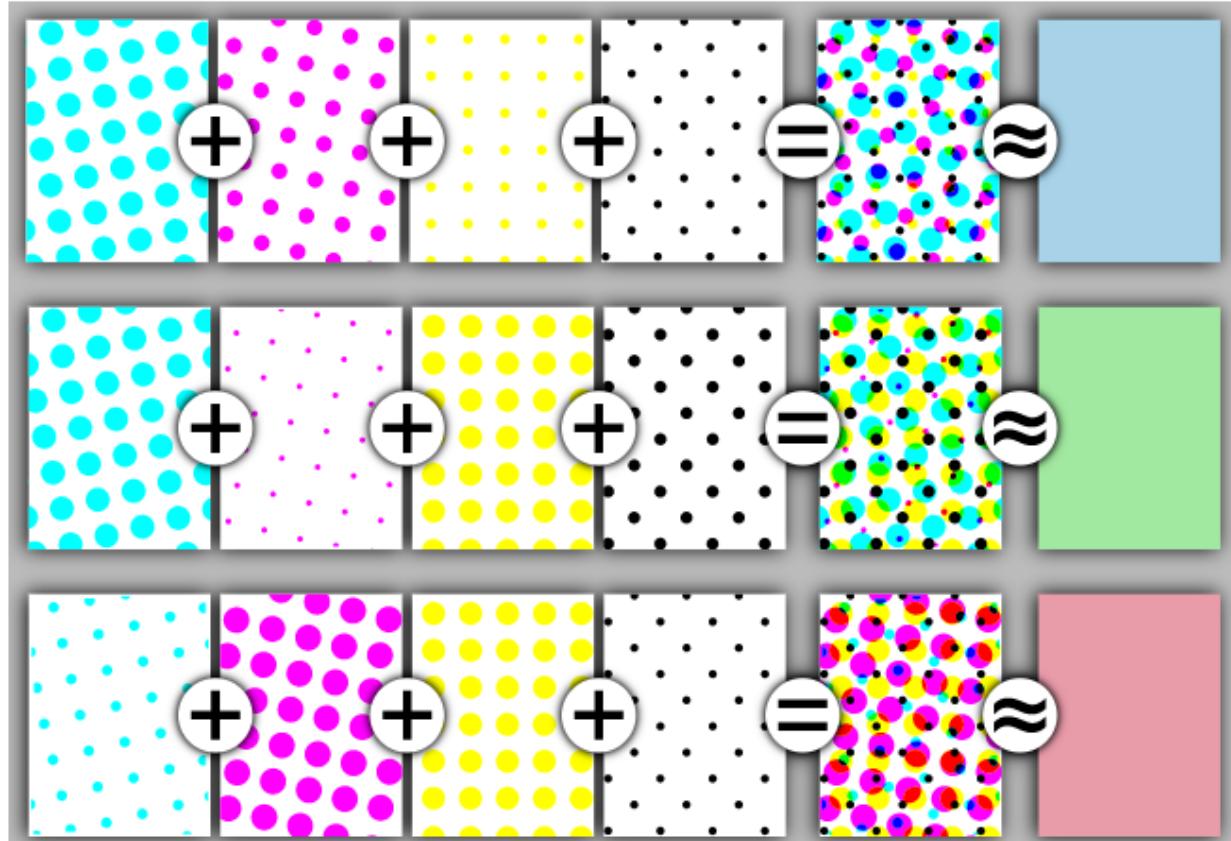


Źródło: Halftoning_introduction.png: Slippensderivative
work: McSush (talk) – [Left: halftone dot pattern with increasing size downwards, Right: how the human eye would see this, when viewed from a sufficient distance.](#)

Grafika i wizualizacja

- Modele barw

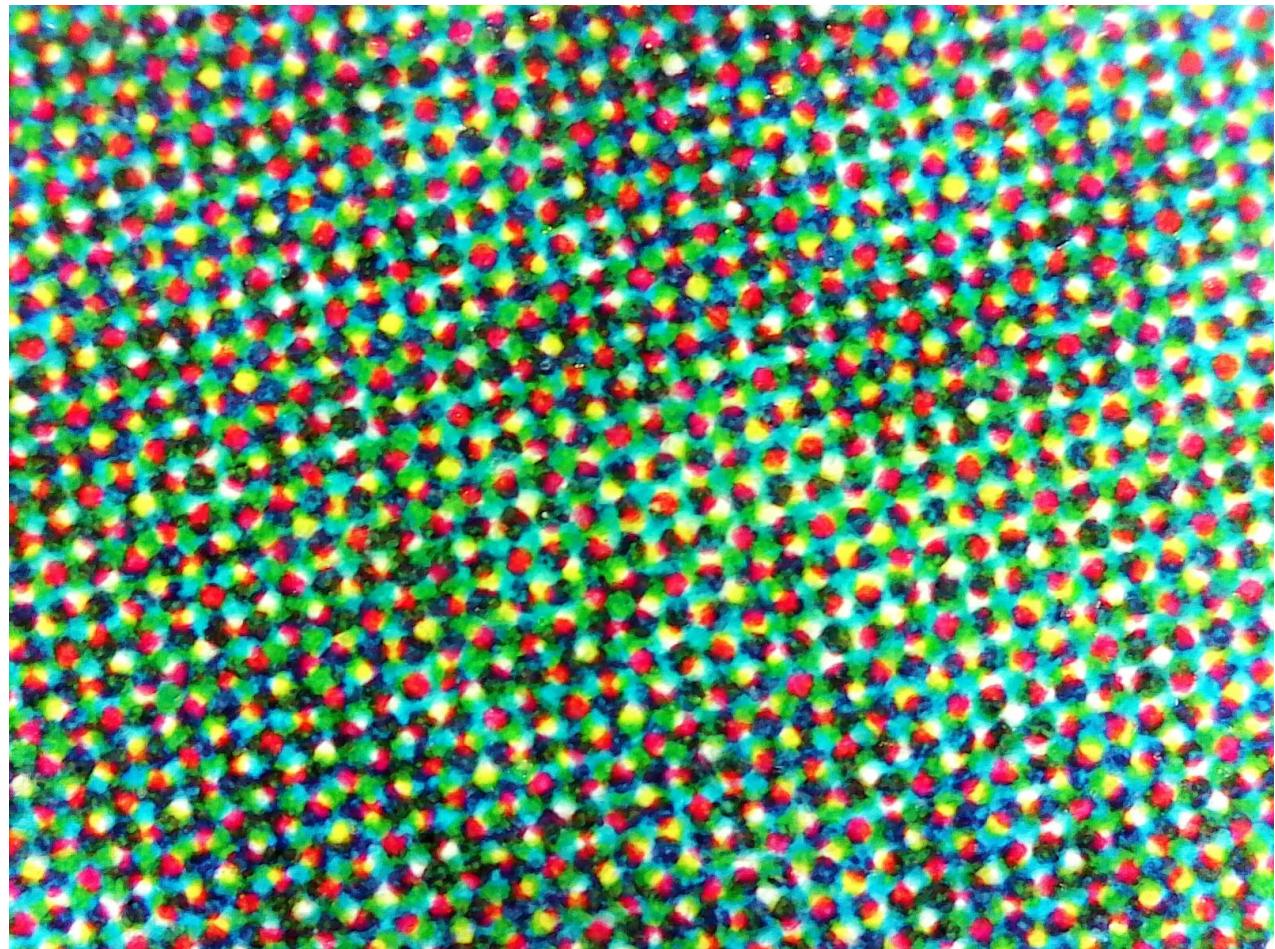
Przykład rastra dla druku wykorzystującego model CMYK.



Źródło: derivative work: Pbroks13 (talk),
Halftoningcolor.png: Slippens – [Three examples of color
halftoning with CMYK separations](#)

Grafika i wizualizacja - Modele barw

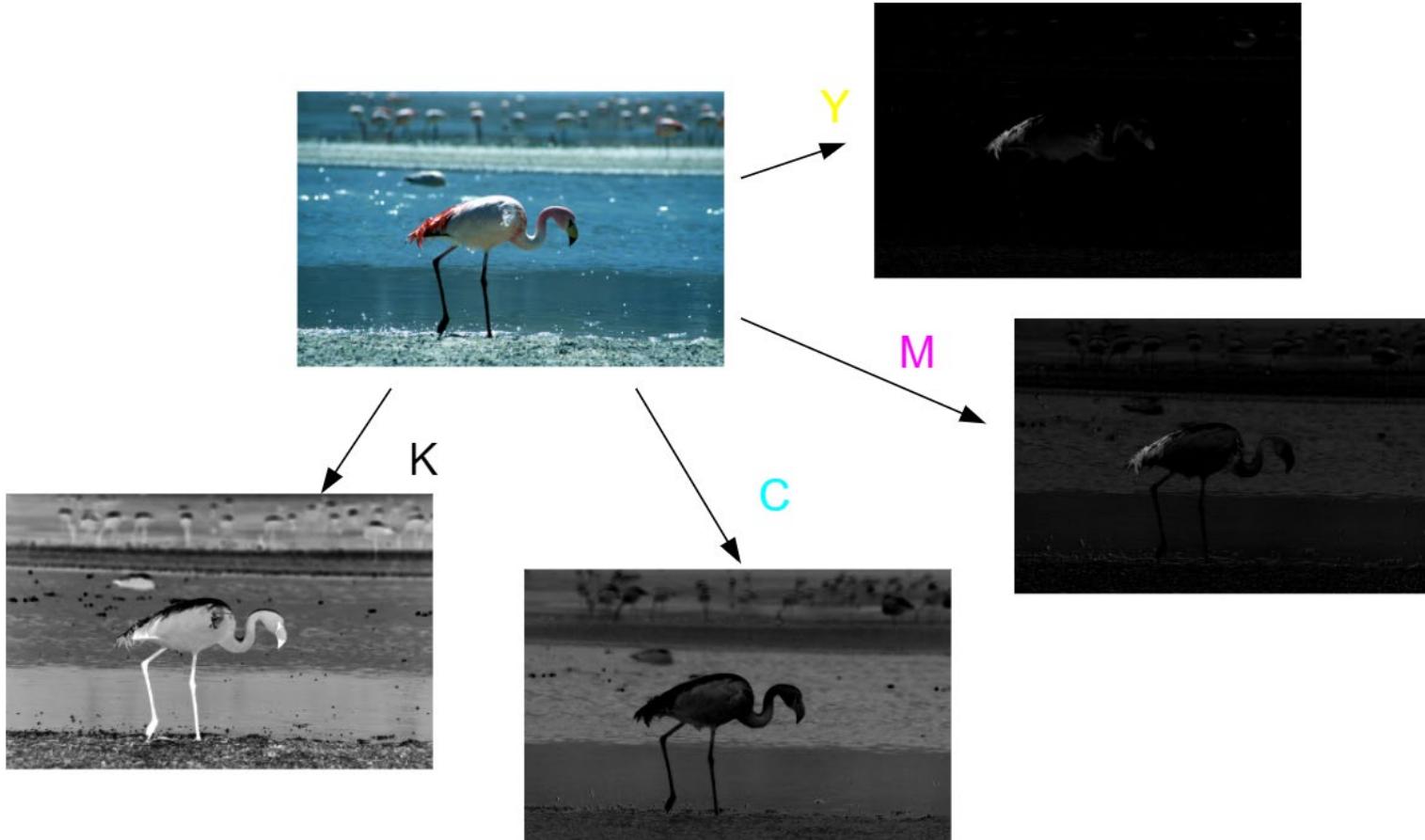
Powiększenie druku
wykorzystującego modelu
CMYK



Grafika i wizualizacja

- Modele barw

Model barw CMYK – rozbarwienie.



Grafika i wizualizacja

- Modele barw

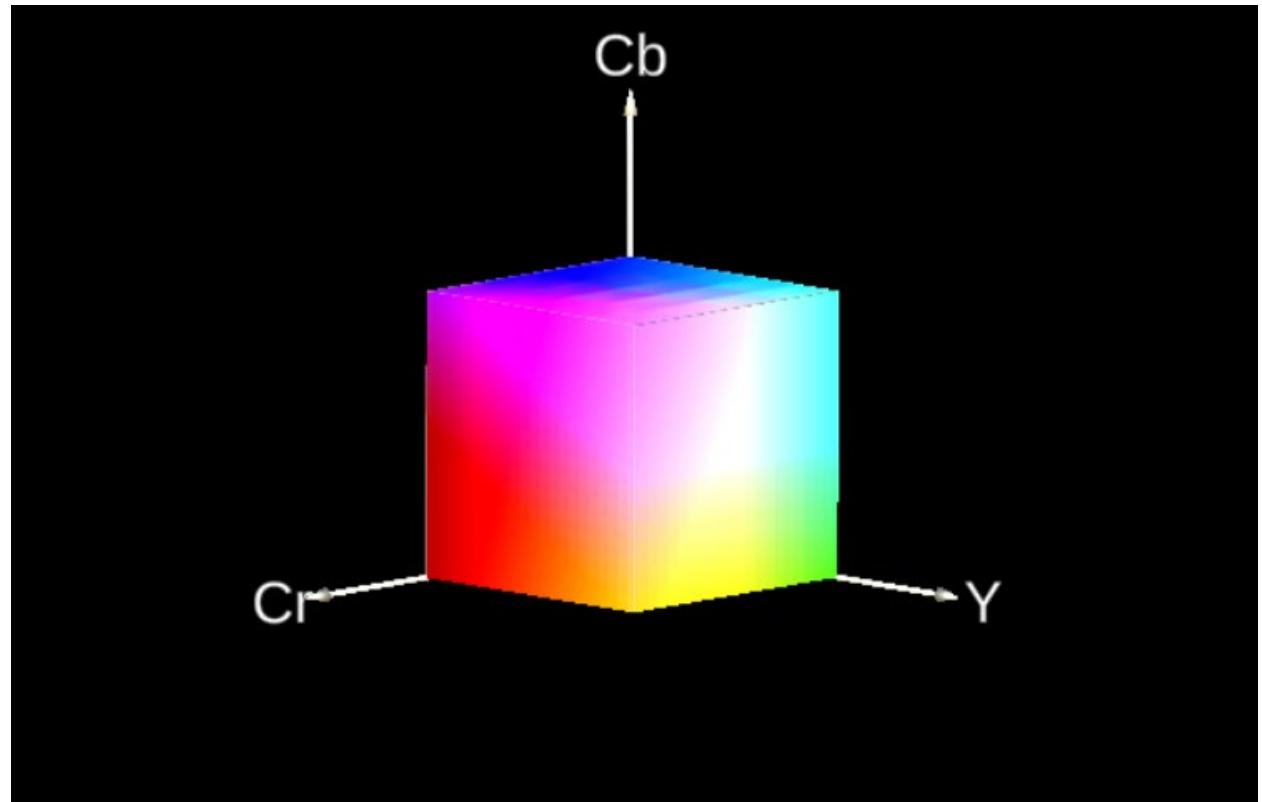
Modele barw YCbCr – rodzina modeli barw stosowana w telewizji cyfrowej oraz kompresji JPEG. Y oznacza składową luminancji, a Cb i Cr są to różnicowe składowe chrominancji. Cb i Cr liczone są na podstawie różnicy składowych B-Y oraz R-Y.

Oko ludzkie jest bardziej wrażliwe na zniekształcenia luminancji niż chrominancji, stąd dzięki tym modelom można przy kodowaniu i kompresji obrazu różnicować stopień utraty informacji dla składowych luminancji oraz chrominancji.

Grafika i wizualizacja

- Modele barw

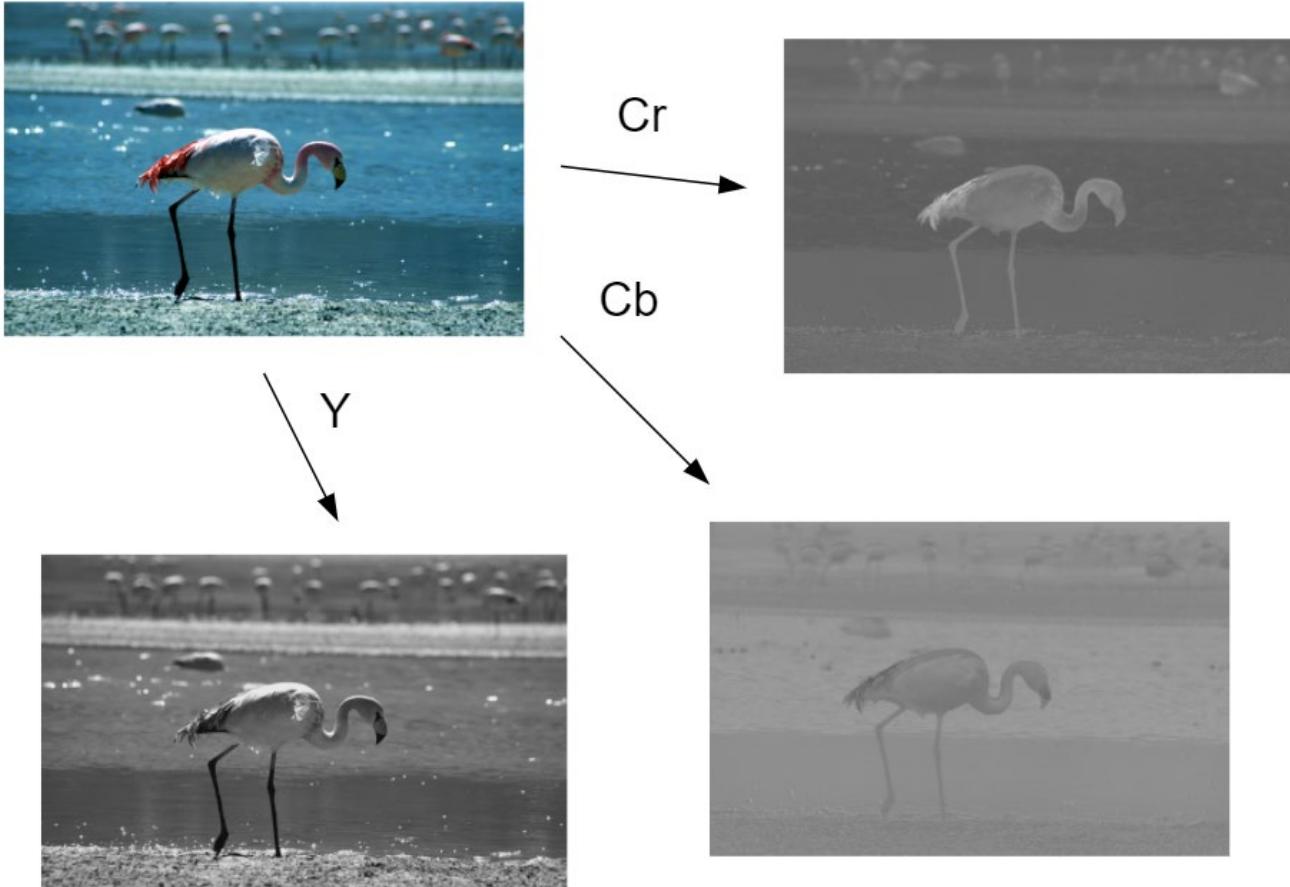
Przykładowy model barw
 $YCbCr$



Grafika i wizualizacja

- Modele barw

Przykładowy model barw YCbCr – rozbarwienie

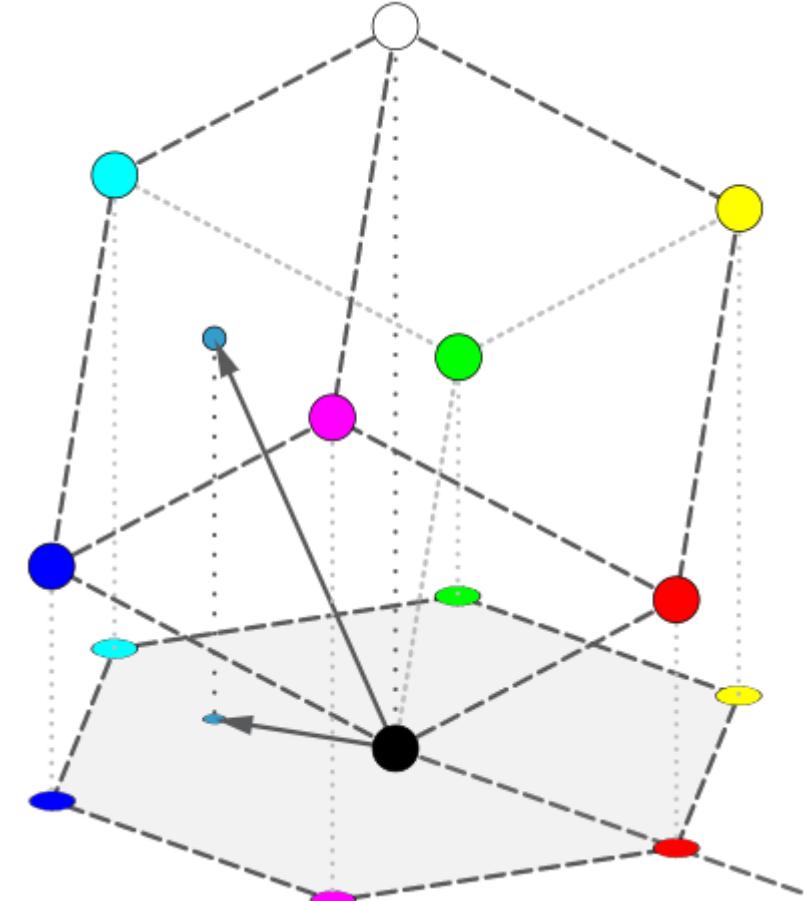


Grafika i wizualizacja

- Modele barw

Modele barw HSL i HSV lepiej odzwierciedlają sposób wyboru barw przez człowieka.

Powstały one z obróconego modelu barw RGB. Powstaje w ten sposób trójwymiarowa figura, której przekrojem jest sześciokąt foremny.

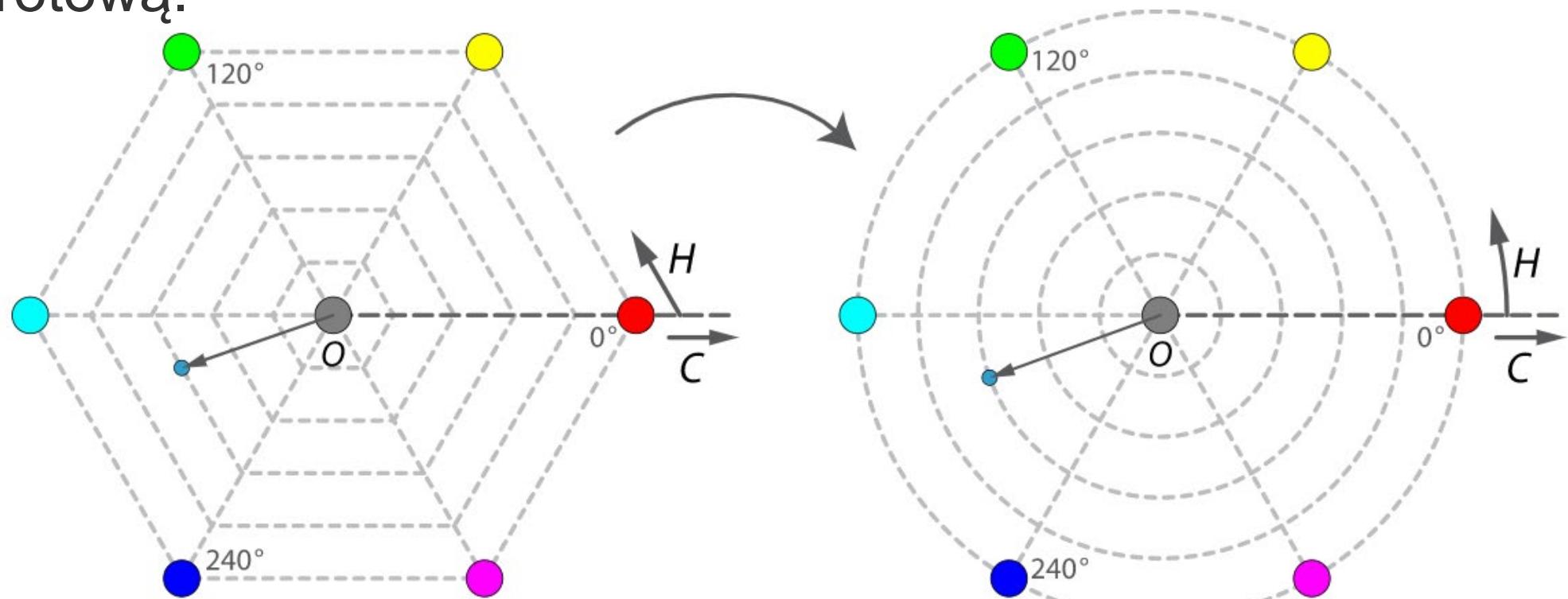


Źródło: Jacob Rus – [For the HSL and HSV article, show how to get hue and "chroma" \[...\], by measuring in the hexagonal](#)

Grafika i wizualizacja

- Modele barw

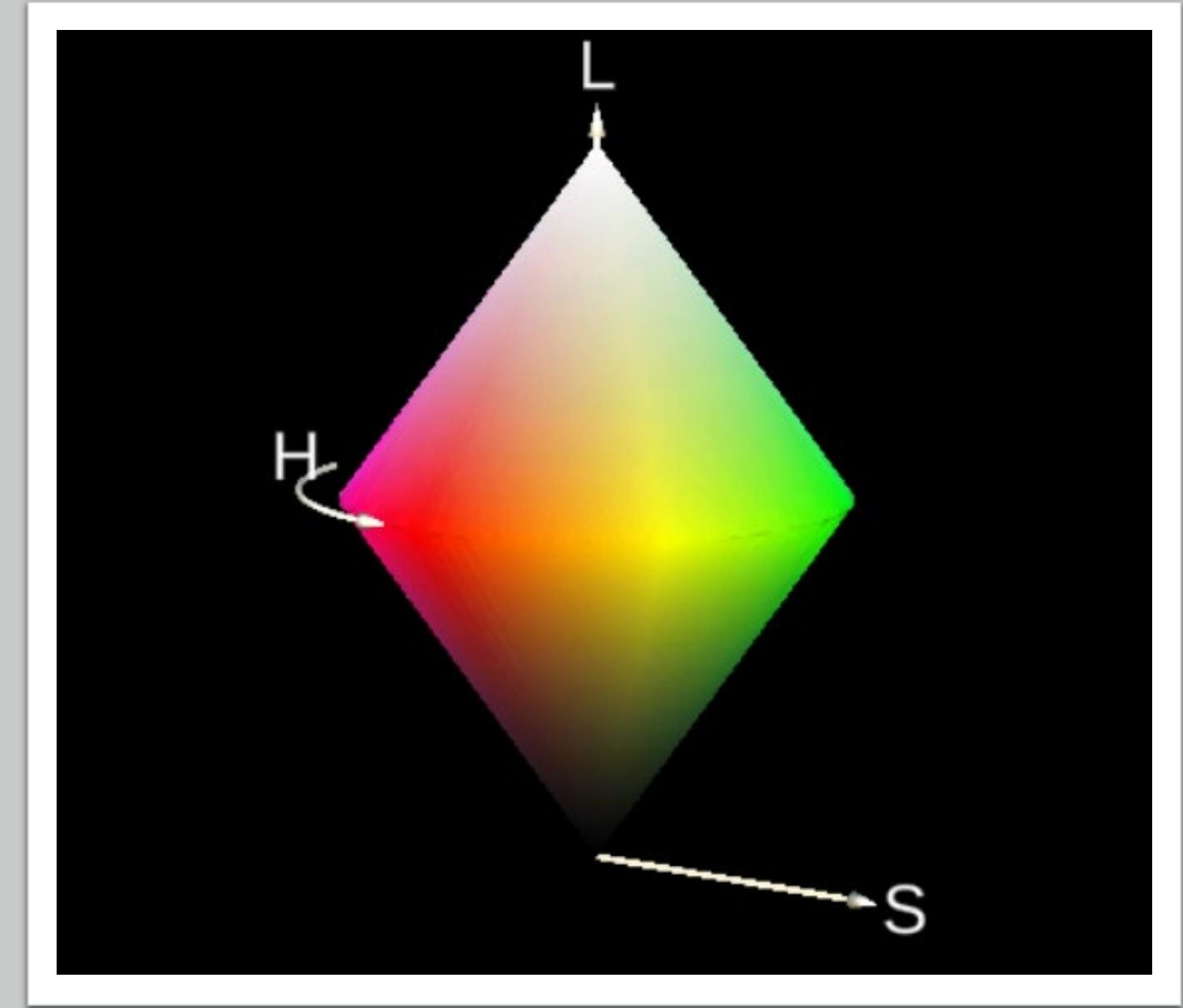
Utworzona trójwymiarowa figura "deformowana" jest w bryłę obrotową.



Źródło: Jacob Rus – [The HSL and HSV representations of RGB color space use a hexagonal norm which has the effect of warping hexagons into circles.](#)

Grafika i wizualizacja - Modele barw

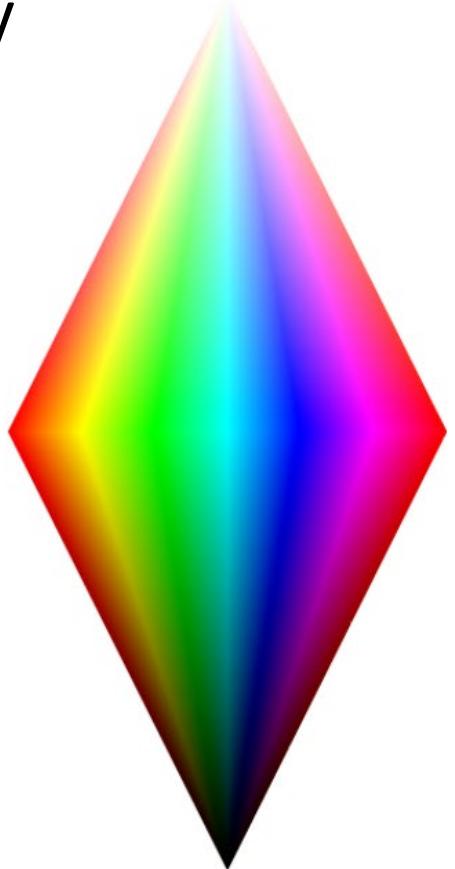
Model barw HSL został stworzony i opatentowany w 1938 przez Georges Valensiego. W modelu tym stosuje się trzy składowe: kolor (H – ang. hue), nasycenie (S – ang. saturation) i jasność (L – ang. lightness).



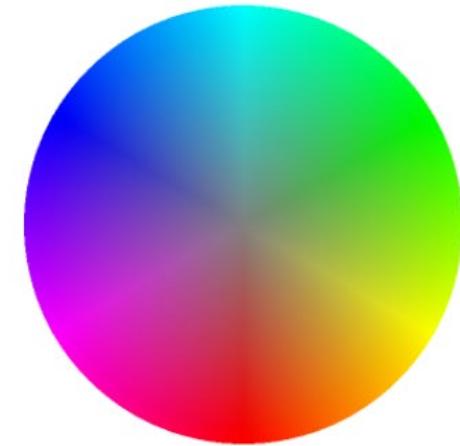
Grafika i wizualizacja

- Modele barw

Model barw HSL – przekroje
przekrój pionowy



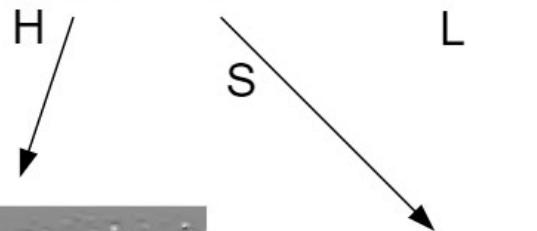
przekrój poziomy



Grafika i wizualizacja

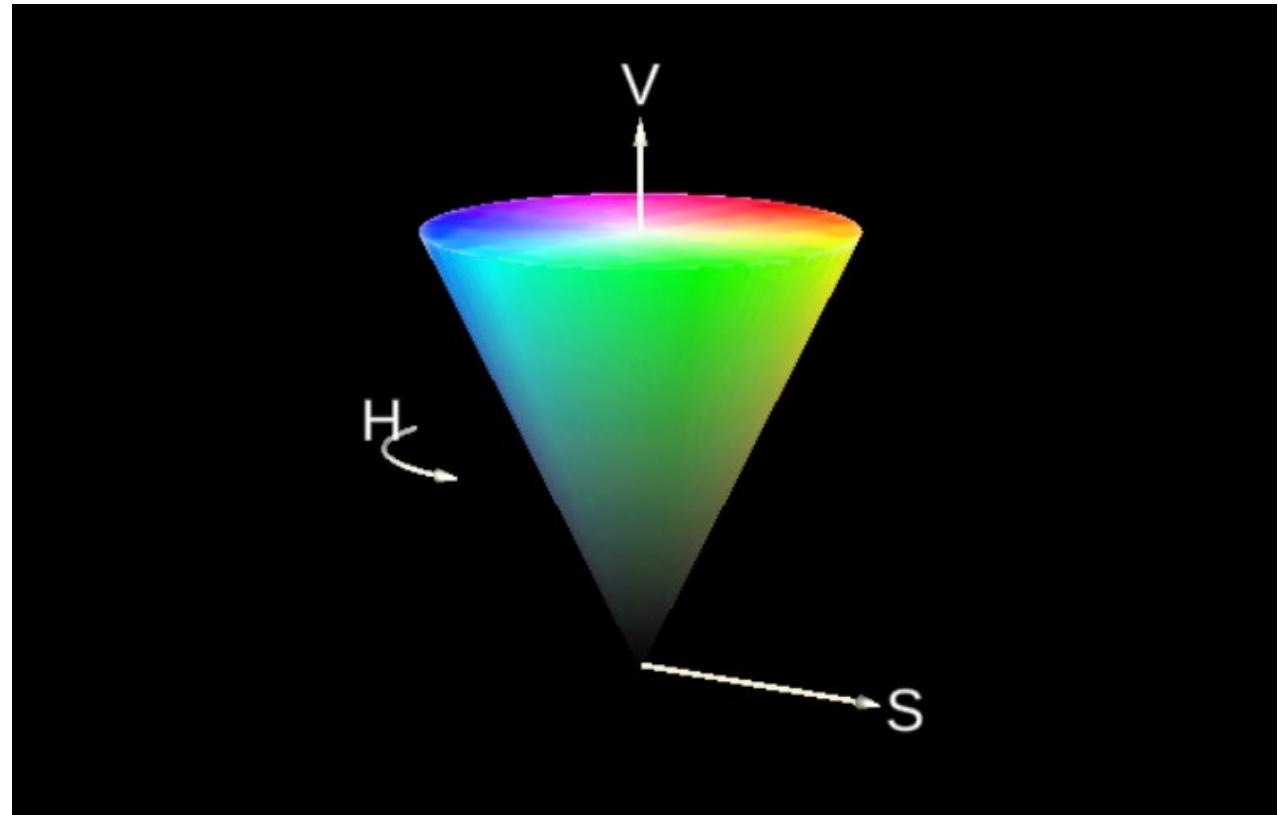
- Modele barw

Model barw HSL – rozbarwienie.



Grafika i wizualizacja - Modele barw

Model barw HSV został opisany w 1978 przez Alvy Ray Smitha. W modelu tym stosuje się trzy składowe: kolor (H – ang. hue), nasycenie (S – ang. saturation) i jasność (V – ang. value).



Grafika i wizualizacja

- Modele barw

Model barw HSL – rozbarwienie.



H

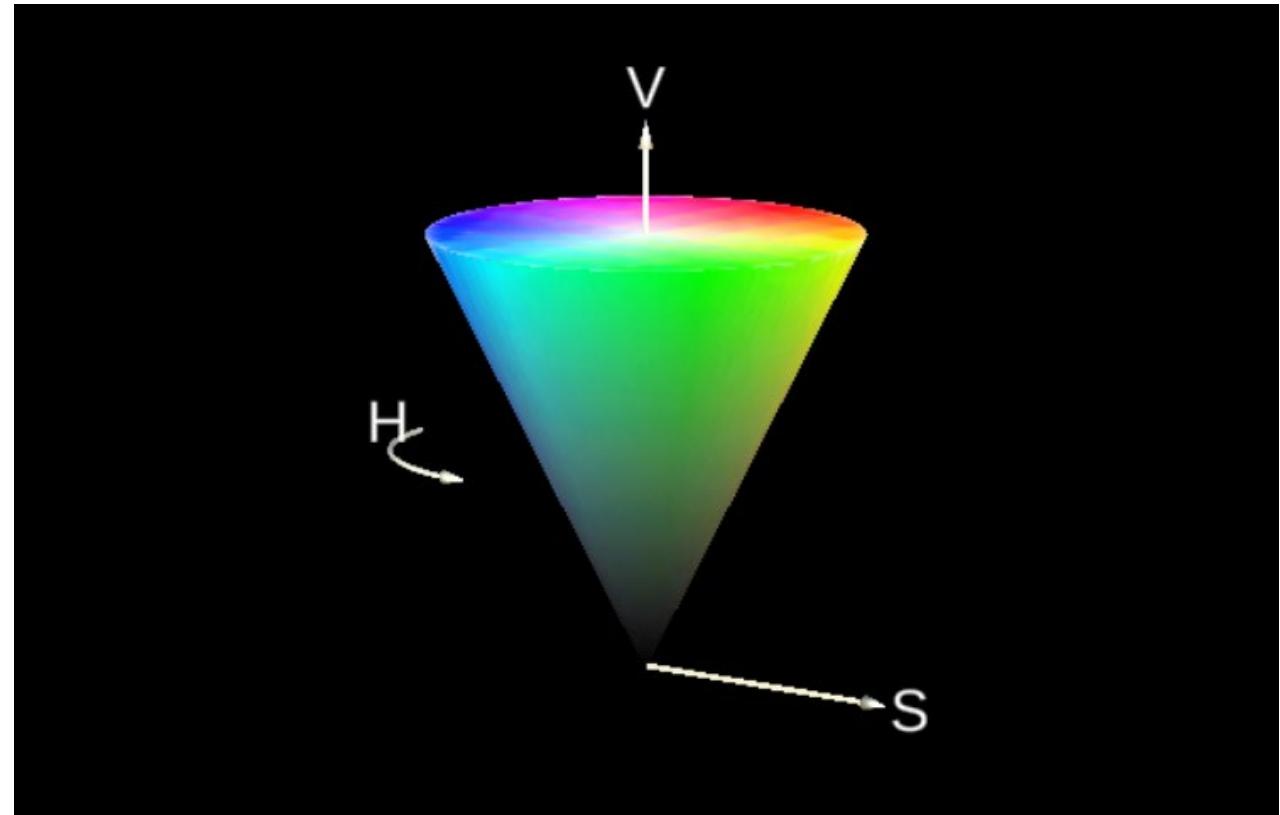


L



Grafika i wizualizacja - Modele barw

Model barw HSV został opisany w 1978 przez Alvy Ray Smitha. W modelu tym stosuje się trzy składowe: kolor (H – ang. hue), nasycenie (S – ang. saturation) i jasność (V – ang. value).

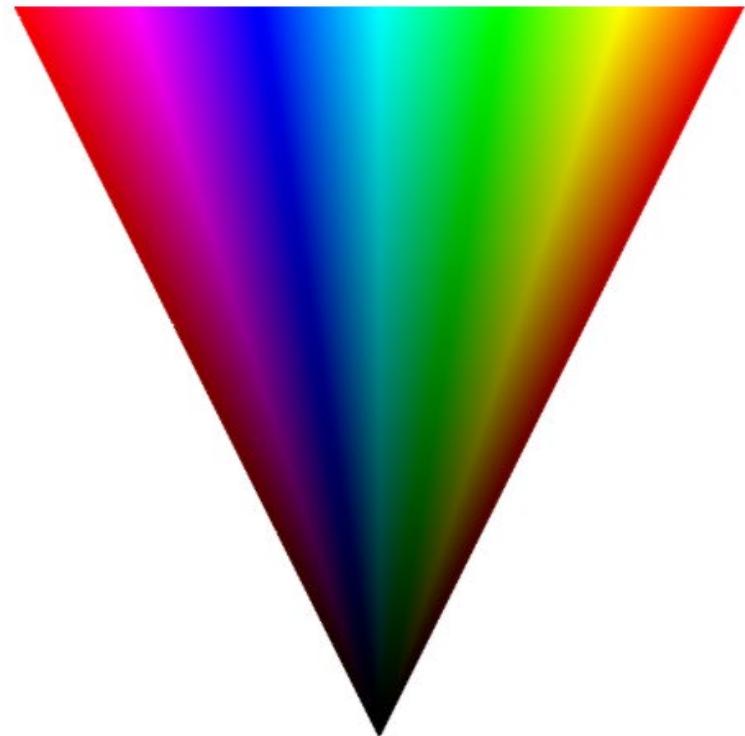


Grafika i wizualizacja

- Modele barw

Model barw HSV

przekrój pionowy



widok z góry



Grafika i wizualizacja

- Modele barw

Model barw HSV – rozbarwienie.



H



V



Grafika i wizualizacja

- Modele barw

Światło białe jest światłem przypominającym światło dzienne. Przyjmuje się, że jest to światło, które powstaje gdy ciało doskonale czarne podgrzeje się do temperatury 6774 K (w przybliżeniu jest to temperatura powierzchni słońca).

Wykorzystując tą definicję CIE (Międzynarodowa Komisja Oświetleniowa) przyjęła szereg wzorców światła białego – iluminantów.

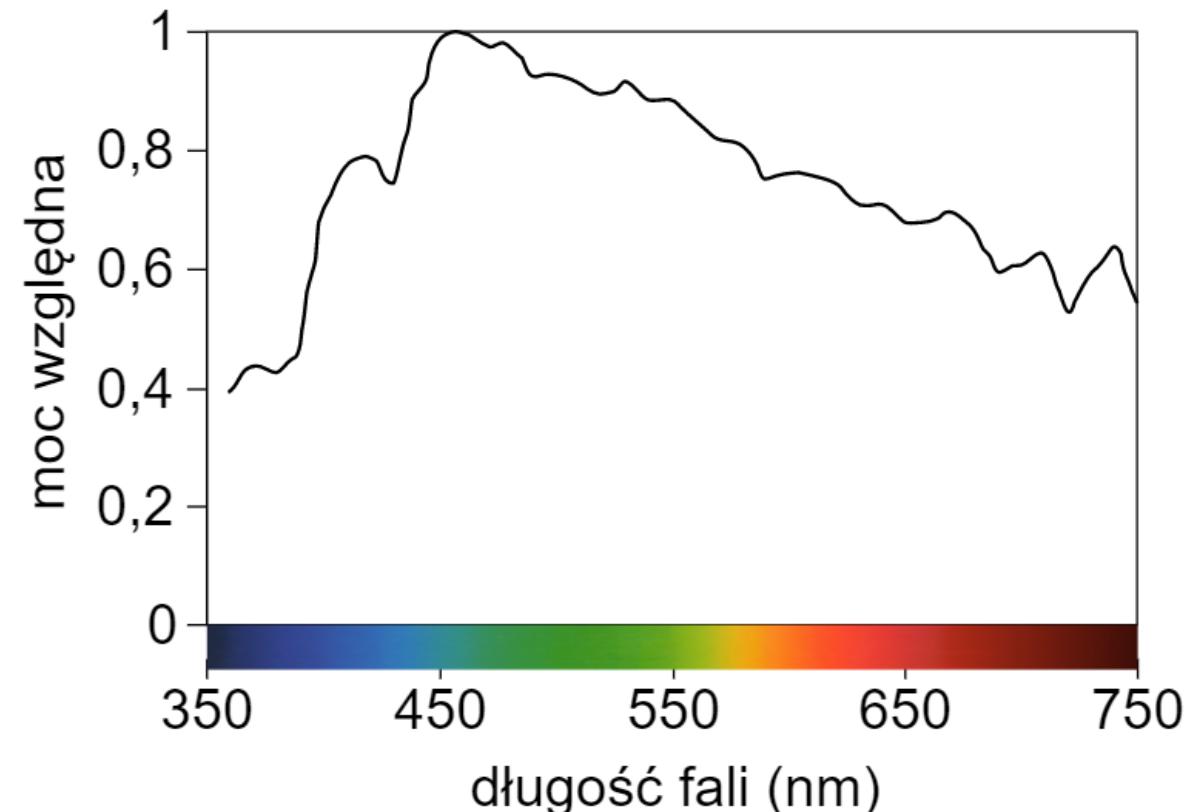
Przykładowe iluminanty:

1. typu D – odzwierciedla naturalne światło dzienne. Wartości po D określają temperaturę do jakiej trzeba podgrzać ciało doskonale czarne. Najbardziej typowa iluminant to D65 (6500 K), inne iluminanty tego typu: D55, D75 itp.;
2. typu A – reprezentuje oświetlenie żarowe;
3. typu E – reprezentuje oświetlenie fluoresencyjne.

Grafika i wizualizacja

- Modele barw

Skład widmowy iluminantu D65.



Źródło: Kevin Houser (Loucetios) – [Relative Spectral Power Distribution for CIE D65](#)

Grafika i wizualizacja

- Modele barw

Problemy z określaniem barwy:

- mała i duża plama o tym samym kolorze mogą mieć inną barwę (indukcja przestrzenna),
- dwie plamy o jednakowej wielkości i kolorze mają barwę zależną od powierzchni na której się znajdują (indukcja przestrzenna),
- sposób odbioru koloru jest zależny od indywidualnych predyspozycji człowieka.

Grafika i wizualizacja

- Modele barw

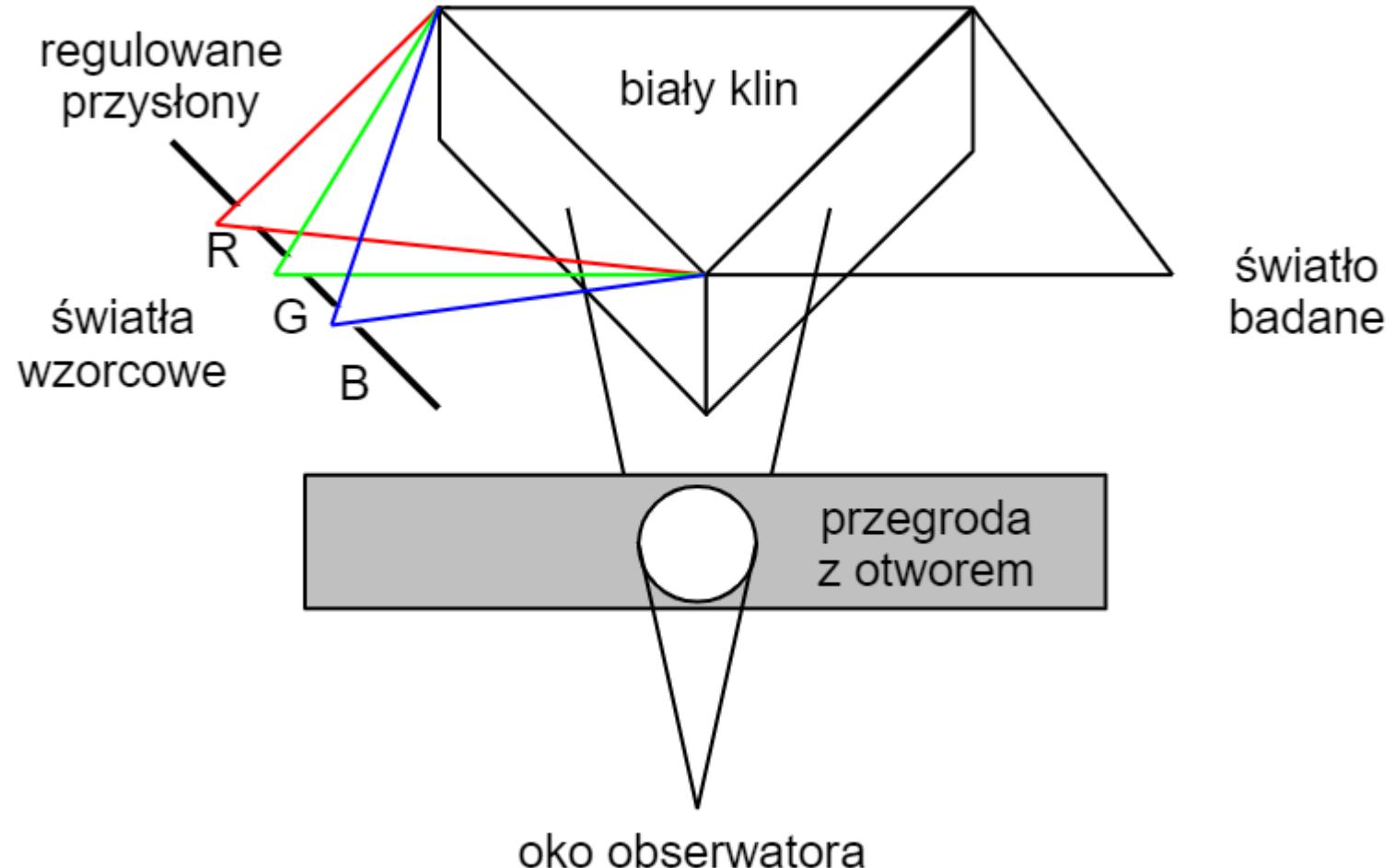
Model barw CIE RGB został stworzony przez CIE w 1931 roku na bazie eksperymentów przeprowadzonych w latach 20-tych przez Williama David-a Wright-a i John-a Guild-a.

Jest to model percepcyjny. Powstał poprzez pomiar wrażeń wzrokowych powstających pod wpływem różnych bodźców świetlnych.

Grafika i wizualizacja

- Modele barw

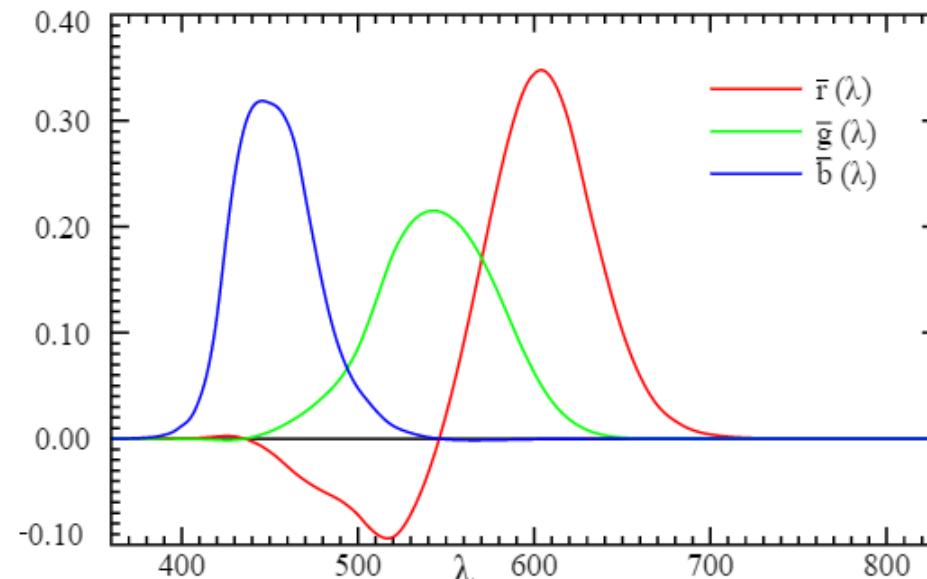
Do badań percepcji kolorów stosuje się odmianę kolorymetru wizualnego stosowanego w chemii.



Grafika i wizualizacja

- Modele barw

Organizacja CIE zdefiniowała standardowego (kolorymetrycznego) obserwatora, który widzi obraz w obrębie dołka środkowego siatkówki (kąt widzenia 2 stopnie). Dla tego obserwatora określono czułość na barwy wyrażając je funkcjami dopasowania kolorów, określającymi ile potrzeba monochromatycznych kolorów testowych do odtworzenia danej barwy.



Źródło: Marco Polo at English Wikipedia – [Normalized RGB functions for monochromatic beams of light of specified \$\lambda\$](#)

Grafika i wizualizacja

- Modele barw

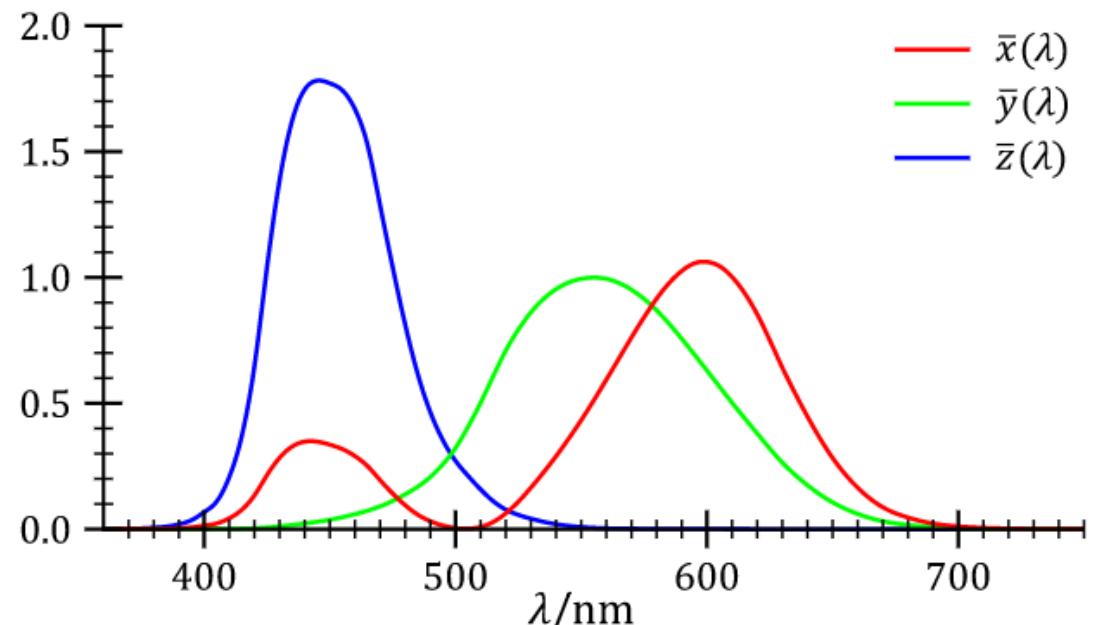
W modelu CIE RGB niektórych barw nie można przedstawić za pomocą dodatnich wartości współrzędnych R, G, B.

Ujemne wartości występują we wszystkich systemach kolorymetrycznych wykorzystujących fizyczne widzialne barwy podstawowe.

Grafika i wizualizacja

- Modele barw

Model barw CIE XYZ osie układu współrzędnych ma tak dobrane, aby otrzymać składowe nieujemne dla wszystkich barw widzianych przez człowieka. Dla osoby o przeciętnym wzroku obejmuje on wszystkie widoczne dla niej wrażenia kolorystyczne.

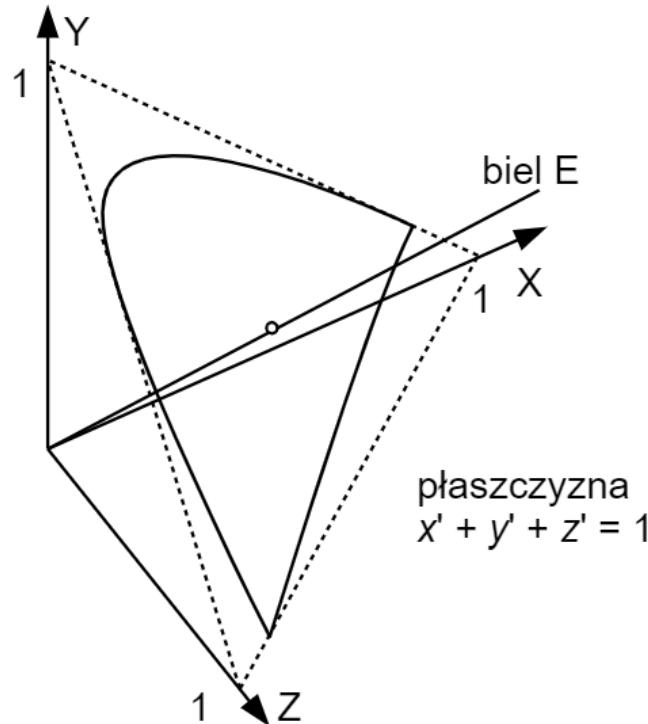


Źródło: User:Acdx – [The CIE 1931 XYZ color matching functions](#).

Grafika i wizualizacja

- Modele barw

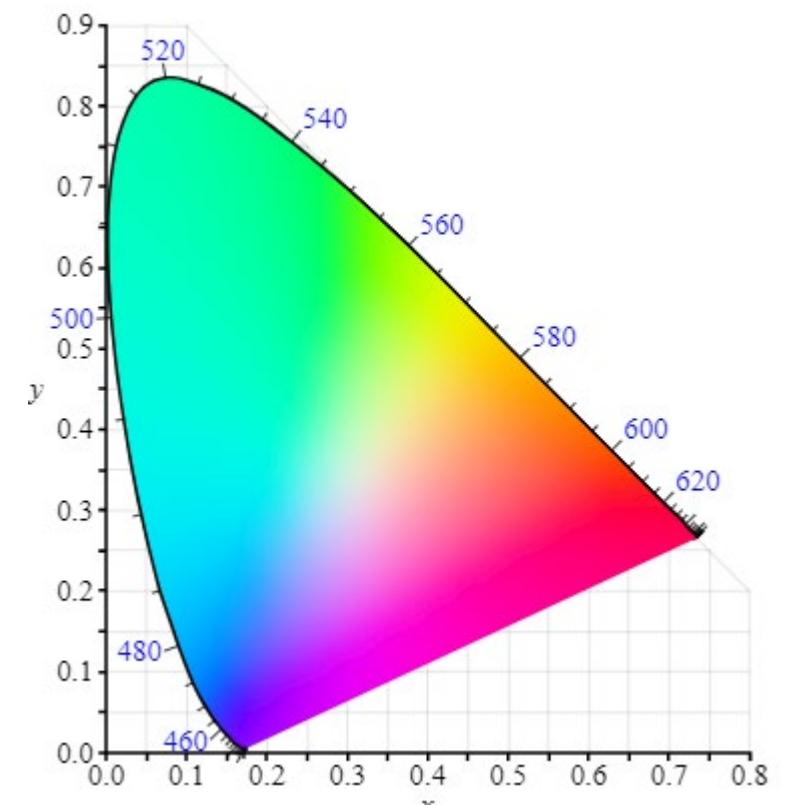
W modelu CIE XYZ barwy widzialne są zawarte w bryle o kształcie przypominającym stożek. W praktyce korzysta się z tzw. wykresu chromatyczności, który tworzy się na przekroju stożka płaszczyzną $x'+y'+z'=1$.



Grafika i wizualizacja

- Modele barw

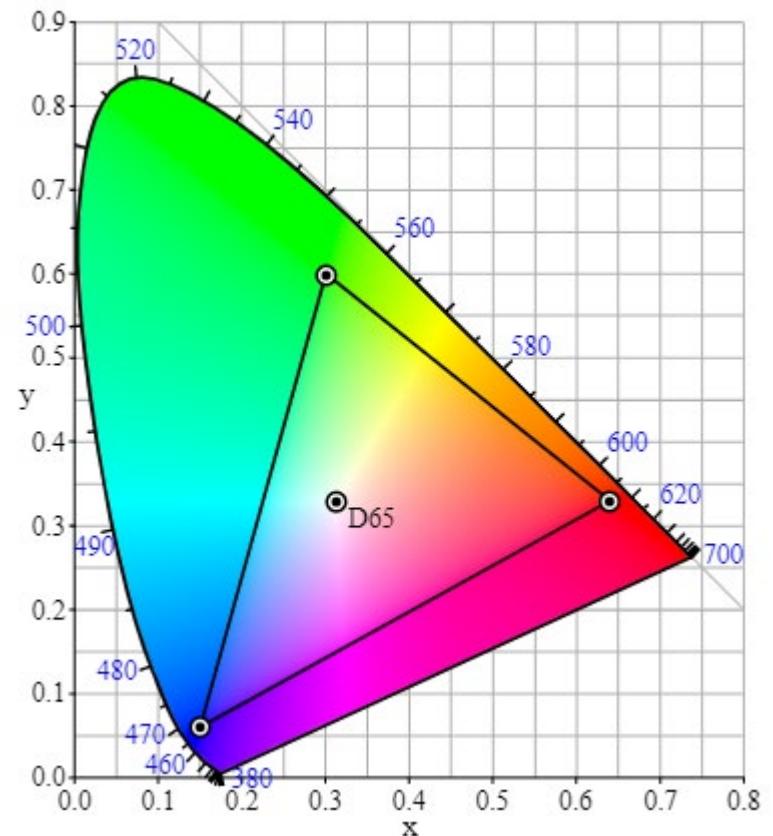
Wnętrze obszaru wraz z brzegiem reprezentuje wszystkie barwy widzialne. Wszystkie barwy o różnej luminancji, ale o tych samych stosunkach współrzędnych reprezentują ten sam punkt. Nie zawiera on barw dla których wrażenie wzrokowe zależy od luminancji (np. brązowej).



Grafika i wizualizacja

- Modele barw

Wykres chromatyczności wykorzystuje się do określania zakresu pracy urządzeń i systemów odzwierciedlających kolor. Przykład wykresu chromatyczności na który nałożono zakres kolorów jaki reprezentuje model barw wykorzystujący 256 wartości do zapisania każdej składowej koloru.



Źródło: Brak informacji o autorze – [CIE 1931 xy chromaticity diagram showing the sRGB gamut and the D65 white point](#)

Grafika i wizualizacja

- Modele barw

System zarządzanie kolorem (ang. Color Management System) jest to kontrolowana konwersja pomiędzy reprezentacjami obrazów dla różnych urządzeń mająca na celu minimalizację różnic w ich wyglądzie. Dzięki prawidłowemu zarządzaniu kolorem dany produkt (na przykład zdjęcie lub film) będzie niewiele się różnił w wyglądzie, przy wizualizacji go na różnych typów monitorów, plakatach, wydrukach itp. System zarządzanie kolorem pozwala do pewnego stopnia kontrolować te różnice.

Grafika i wizualizacja

- Modele barw

System zarządzania kolorem wymaga odpowiednich aplikacji lub modułów systemu operacyjnego oraz urządzeń z którymi mogą one współpracować. Warunkiem współpracy jest określenie dla urządzeń profiliów koloru. Profil koloru są to dane charakteryzujące możliwości reprezentacji koloru przez dane urządzenie lub model barw w którym przechowywane są dane.

Grafika i wizualizacja

- Modele barw

Rodzaje profili kolorów:

- profile urządzeń wyjściowych – charakteryzują możliwości reprodukcji barw przez te urządzenia. Wyróżnia się:
- profile monitorów – charakteryzują możliwości reprodukcji barw przez monitory;
- profile urządzeń drukujących – charakteryzują możliwości reprodukcji barw przez te urządzenia na co składa się rodzaj farby, oraz typ papieru;
- profile urządzeń wejściowych – charakteryzują możliwości rejestracji barw przez te urządzenia;
- profile dokumentów – określają konkretny model barw w jakim dokument został stworzony.

Grafika i wizualizacja

- Modele barw

Profil ICC – profil koloru utworzony według norm wydanych przez International Color Consortium (ICC). Profile ICC są mogą być dostarczane przez producenta sprzętu.

Gdy wierność odtworzenia barw jest istotna profile mogą być korygowane w procesie kalibracji urządzenia.

Gdy nie istnieje profil ICC urządzenia często można go utworzyć wykorzystując specjalne oprogramowanie i sprzęt (kolorymetr).

Przykładem urządzenia opensource umożliwiającego tworzenie profili ICC dla monitora jest ColorHug2 <https://hughski.com/colorhug2.html> .

Grafika i wizualizacja

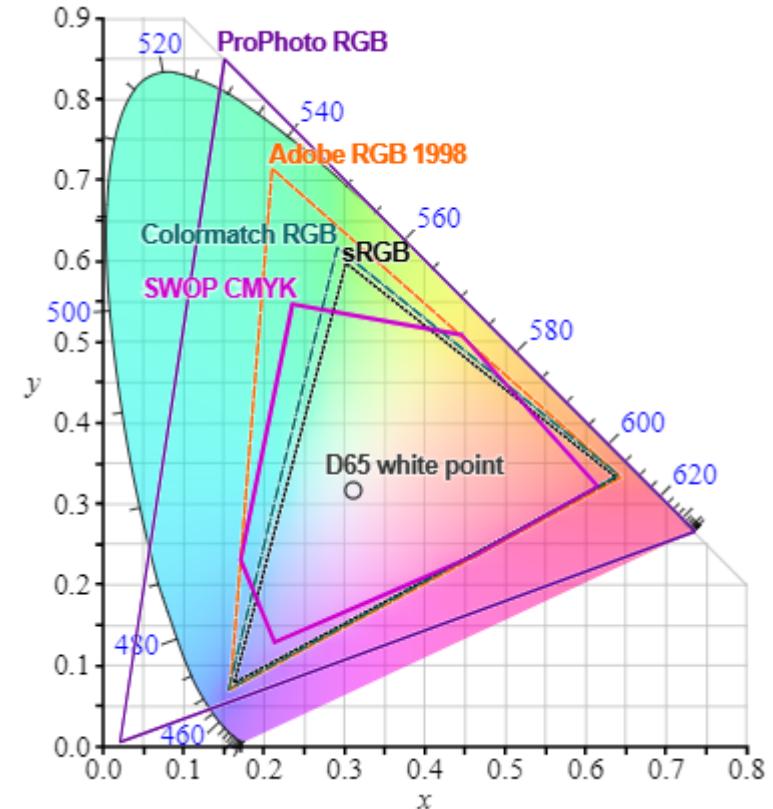
- Modele barw

txt

Grafika i wizualizacja

- Modele barw

Mimo, że mówi się o jednym modelu barw RGB, to w systemach zarządzania kolorem występuje ich wiele odmian. Wiąże się to z różnym przypisaniem rzeczywistych kolorów do współrzędnych barw RGB.



Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Grafika i wizualizacja

Podstawowe algorytmy rasterowe

Edward Półrolniczak, dr inż.,
Katedra Systemów Multimedialnych,
Wydział Informatyki, ZUT w Szczecinie

Grafika i wizualizacja

Zagadnienia

- Wprowadzenie
- Rasteryzacja
- Grafika rastrowa
- Algorytm Bresenhama rysowania linii
- Algorytm Bresenhama rysowania okręgu
- Algorytmy wypełniania
- Inne aspekty



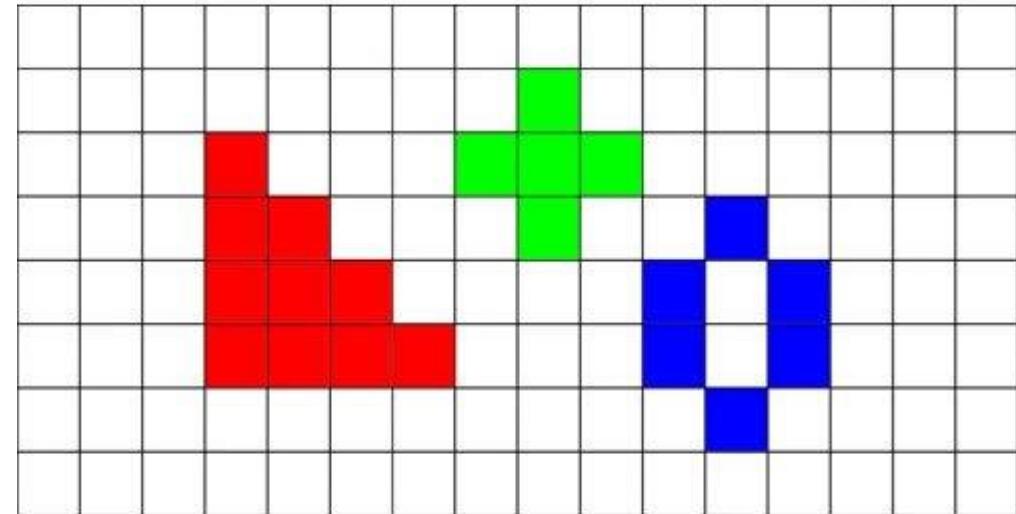
Grafika i wizualizacja

- Wprowadzenie

Algorytmy rastrowe

- Wprowadzenie

Urządzenia wyjścia służące prezentacji grafiki (np. monitory czy drukarki) wykorzystują ograniczony zbiór elementów fizycznych (punktów, pikseli), za pomocą których może być zwizualizowany obraz.



Jest to przestrzeń dyskretna – nie jest to ciągła przestrzeń.

Przed algorytmami graficznymi stawiane jest zatem zadanie jak najwierniejszego odzwierciedlenia obrazu rzeczywistego w przestrzeni dyskretnej monitora lub drukarki.

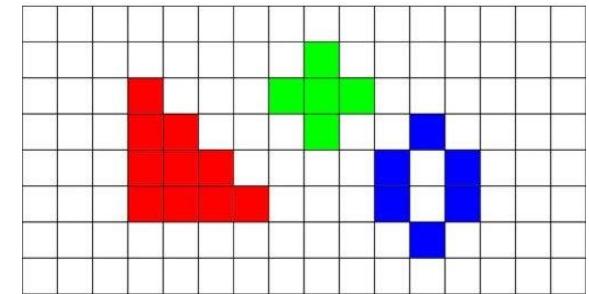
*Rys. Uogólnione wyobrażenie grafiki rastrowej (ekran)

Algorytmy rastrowe

- Rasteryzacja

Rasteryzacja (ang. rasterization or scan-conversion)

Zamiana ciągłej funkcji 2D na funkcję dyskretną
(np. rysowanie okręgu w rastrze na podstawie równania okręgu).



Problem sprowadza się do wyboru pikseli, którym trzeba nadać kolor, aby w efekcie otrzymać wymagany kształt geometryczny.

Problem rasteryzacji pojawił się w momencie upowszechnienia się monitorów rastrowych.

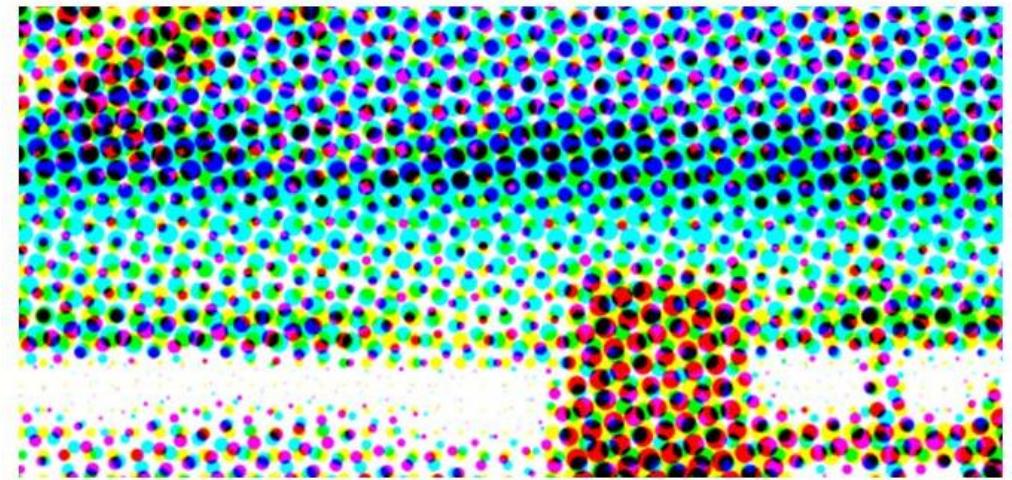
Algorytmy grafiki rastrowej są fundamentem grafiki komputerowej.

*Rys. Uogólnione wyobrażenie grafiki rastrowej (ekran)

Algorytmy rastrowe

- Grafika rastrowa

Grafika rastrowa – prezentacja obrazu za pomocą matrycy punktów w postaci prostokątnej siatki odpowiednio kolorowanych pikseli na monitorze komputera, drukarce lub innym urządzeniu wyjściowym. W systemach komputerowych grafika rastrowa jest przechowywana w sposób skompresowany (stratnie lub bezstratnie) albo nieskompresowany w wielu formatach plików graficznych.



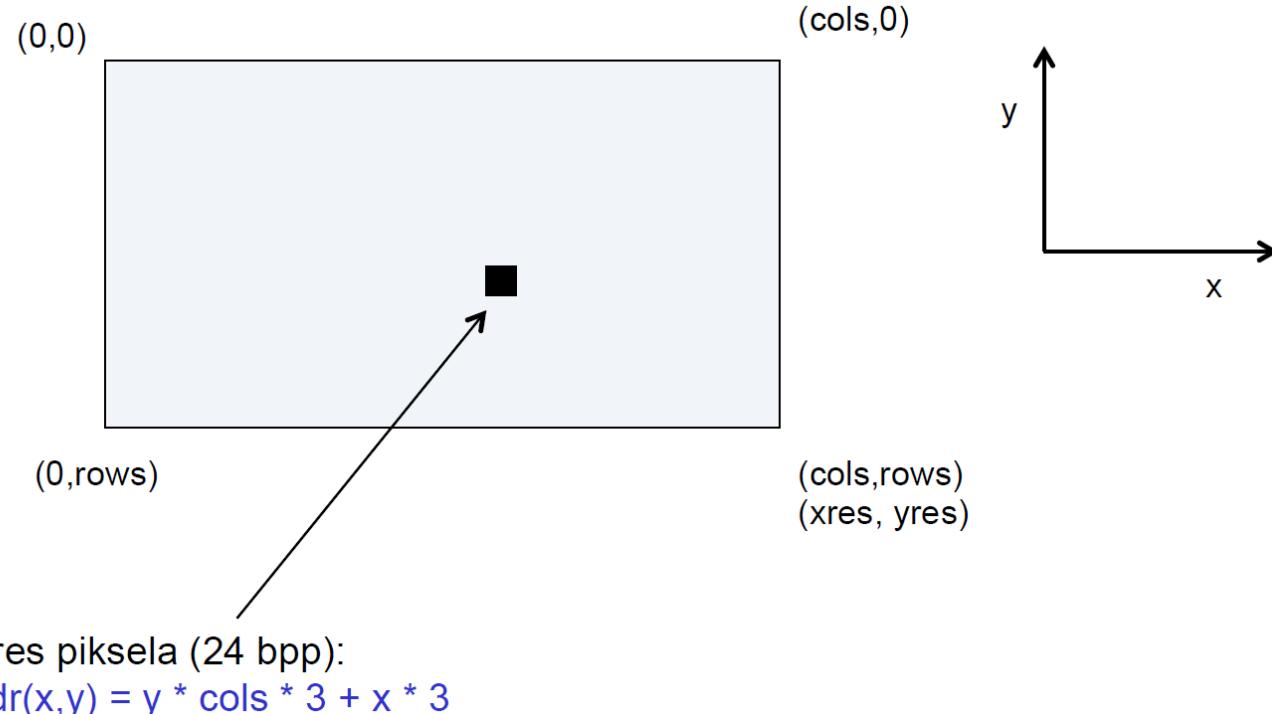
Algorytm rastrowy – algorytm, w wyniku którego uzyskujemy grafikę rastrową.

*Rys. Uogólnione wyobrażenie grafiki rastrowej (druk)

Algorytmy rastrowe

- Rysowanie piksela

Nadanie odpowiedniej komórce pamięci obrazu określonej wartości (koloru).



*Rys. Rysowanie piksela



Grafika i wizualizacja

- Rysowanie odcinka

Algorytmy rastrowe

- Rysowanie odcinka

Jednym z podstawowych zadań (obok narysowania punktu ;-) jest narysowanie odcinka.

Już to zadanie nie jest trywialne.

W celu narysowania odcinka w przestrzeni dyskretnej można skorzystać z matematycznego równania na odcinek...

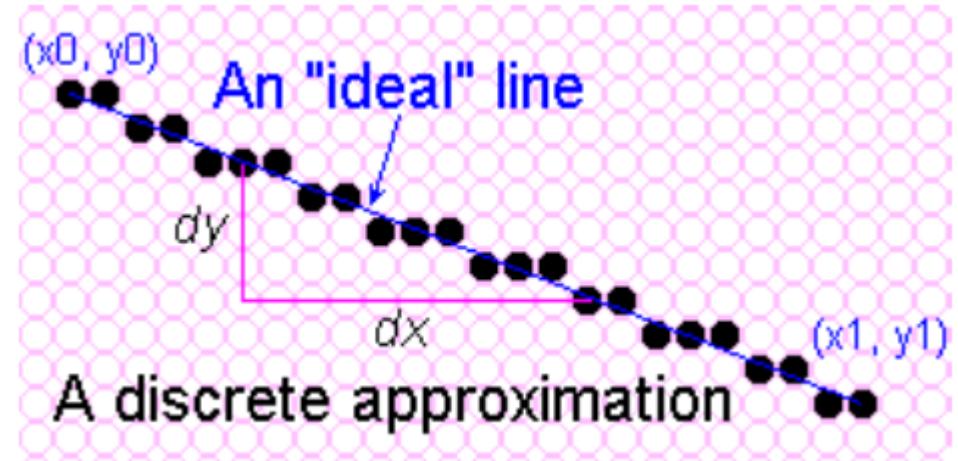
Jak brzmi to równanie?

Algorytmy rastrowe

- Rysowanie odcinka

Zapalanie odpowiednich pikseli pomiędzy pikselami początkowym i końcowym.

- Linia powinna wyglądać w sposób ciągły.
- Musi mieć jednakową grubość i jaskrawość na całej długości.
- Zapalone powinny być piksele jak najbliżej idealnej linii.
- Algorytm rysowania musi być bardzo szybki.



*Rys. <http://www.vrarchitect.net/anu/cg/Line/idealLine.en.html>

Algorytmy rastrowe

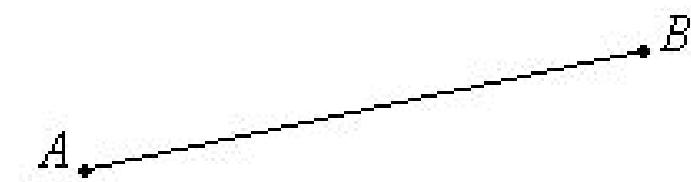
- Rysowanie odcinka

Jednym z podstawowych zadań (obok narysowania punktu ;-) jest narysowanie odcinka.

Już to zadanie nie jest trywialne jeśli wyobrażymy sobie, że rysowanie odbywa się w przestrzeni dyskretnej.

W celu narysowania odcinka w przestrzeni dyskretnej można skorzystać z matematycznego równania na odcinek...

Jak brzmi to równanie?



*Rys. Uogólnione wyobrażenie odcinka w przestrzeni dyskretnej (widać „schodki”)

Algorytmy rastrowe

- Rysowanie odcinka

Matematyczne równanie prostej przechodzącej przez 2 punkty:

*za <http://www.math.edu.pl/rownanie-odcinkowe-prostej>

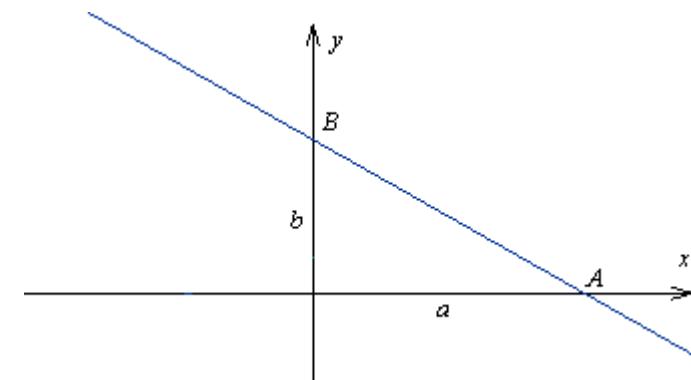
Weźmy pod uwagę prostą przechodzącą przez punkty $A(x_1, y_1)$ i $B(x_2, y_2)$.

Równanie prostej przez dwa punkty możemy zapisać w postaci:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

Gdzie: $(y_2 - y_1) / (x_2 - x_1)$ – to de facto współczynnik kierunkowy prostej (m).

Jak narysować zatem odcinek na ekranie?



*Rys. Prosta odcinkowa

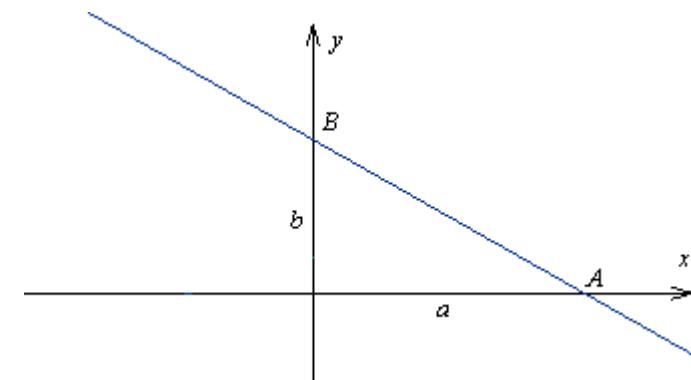
Algorytmy rastrowe

- Rysowanie odcinka

Jak narysować zatem odcinek na ekranie?

Można podstawiać wszystkie dyskretnie wartości x z przedziału $\langle x_1, x_2 \rangle$ wyznaczając w ten sposób

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$



*Rys. Prosta odcinkowa

Algorytmy rastrowe

- Rysowanie odcinka

Jak narysować zatem odcinek na ekranie?

Wykorzystując wcześniej podane wzory można wykreślić linię na ekranie, ale jej wygląd w większości przypadków (kątów nachylenia) nie będzie dla odbiorcy satysfakcjonujący, dlaczego?



Algorytmy rastrowe

- Rysowanie odcinka

Rysowanie odcinka

```
public void lineSimple(int x0, int y0, int x1, int y1,  
Color color) {  
    int pix = color.getRGB();  
    int dx = x1 - x0;  
    int dy = y1 - y0;  
    raster.setPixel(pix, x0, y0);  
    if (dx != 0) {  
        float m = (float) dy / (float) dx;  
        float b = y0 - m*x0;  
        dx = (x1 > x0) ? 1 : -1;  
        while (x0 != x1) {  
            x0 += dx;  
            y0 = Math.round(m*x0 + b);  
            raster.setPixel(pix, x0, y0);  
        }  
    }  
}
```

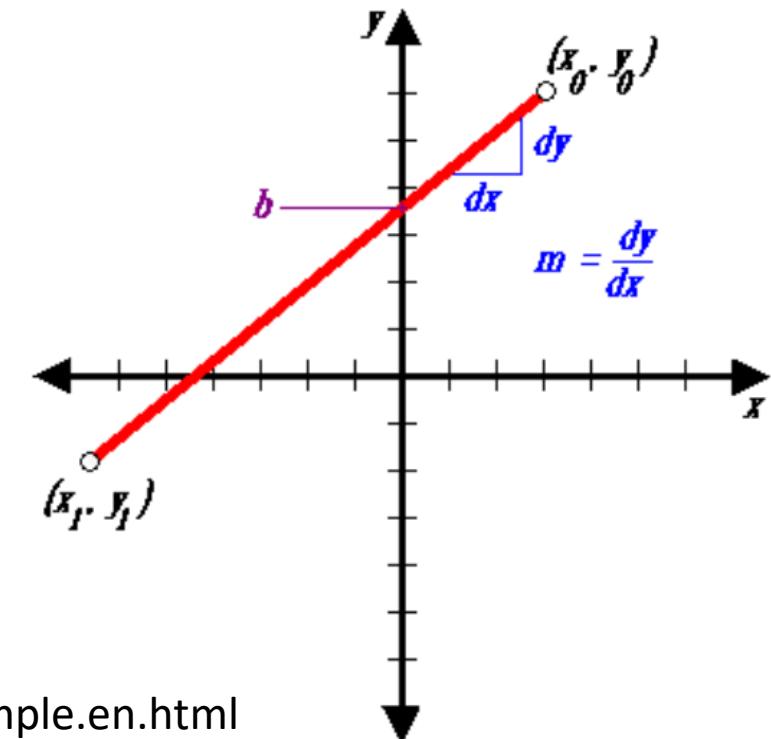
$$y = mx + b$$

$$y_0 = mx_0 + b$$

$$y_1 = mx_1 + b$$

$$m = (y_1 - y_0)/(x_1 - x_0) = dy/dx$$

$$b = y_0 - m x_0$$

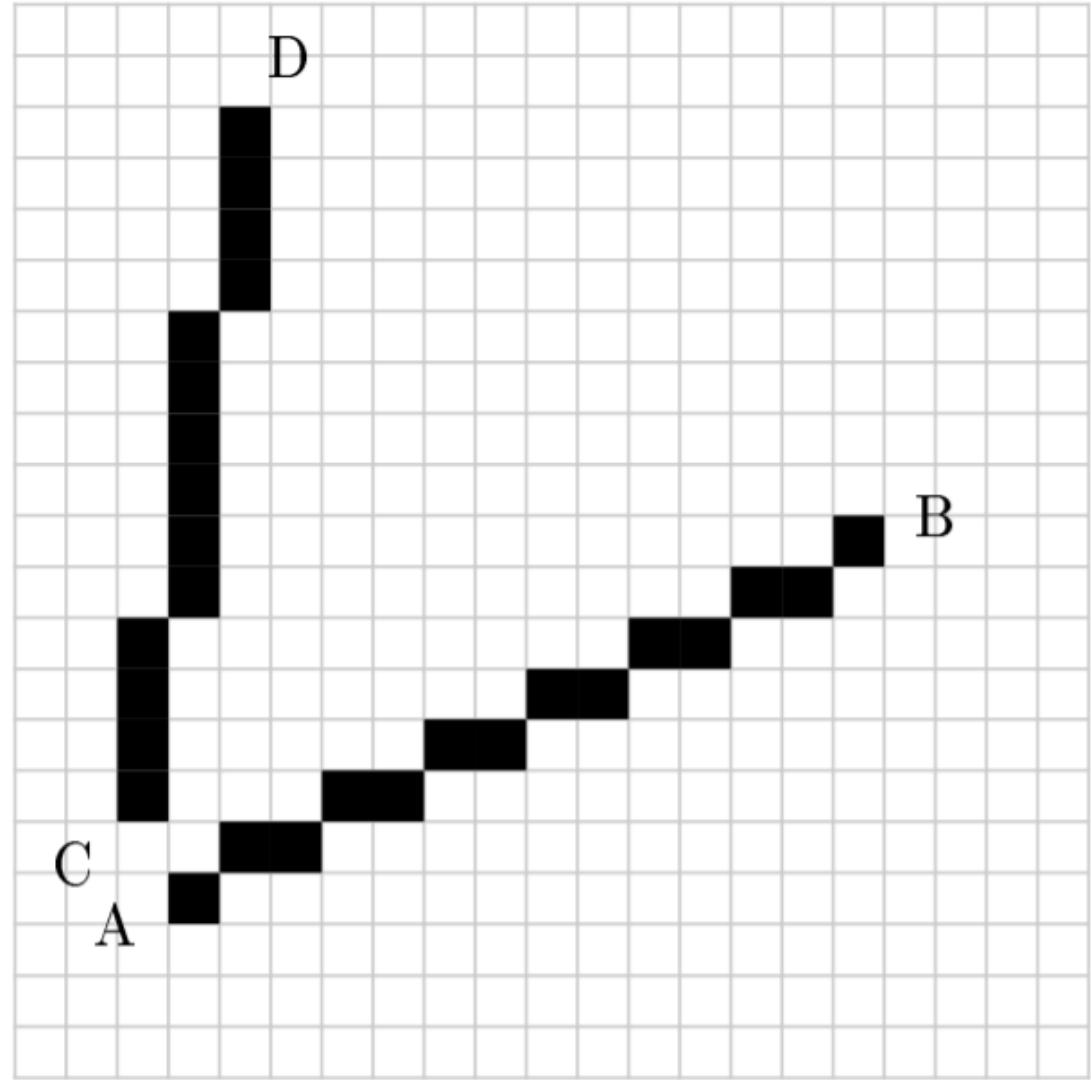


Algorytmy rastrowe - rysowanie odcinka

Algorytm Bresenhama

Jest to jeden z pierwszych
(jeśli nie pierwszy)
znanych i rozpoznawanych
algorytmów rasteryzacji odcinka.
Zaproponował go właśnie
Jack Elton Bresenham

https://en.wikipedia.org/wiki/Jack_Elton_Bresenham



Grafika i wizualizacja

- Algorytm Bresenhama

Algorytm Bresenhama do rysowania linii, opracowany w 1962 roku, jest jego najbardziej znaną innowacją.

Określa, które punkty na dwuwymiarowym rastrze należy narysować, aby utworzyć linię prostą między dwoma danymi punktami i jest powszechnie używany do rysowania linii na ekranie komputera.

Grafika i wizualizacja

- Algorytm Bresenhama

Jest to jeden z najwcześniejszych algorytmów odkrytych w dziedzinie grafiki komputerowej. Algorytm okręgu środkowego ma pewne podobieństwa do jego algorytmu liniowego i jest znany jako algorytm koła Bresenhama.

Grafika i wizualizacja

- Algorytm Bresenhama

W większości praktycznych zadań rysujemy odcinki o współrzędnych rzeczywistych. Czyniąc to na urządzeniu rastrowym, musimy przejść od układu współrzędnych rzeczywistych do układu współrzędnych całkowitych danego urządzenia graficznego.

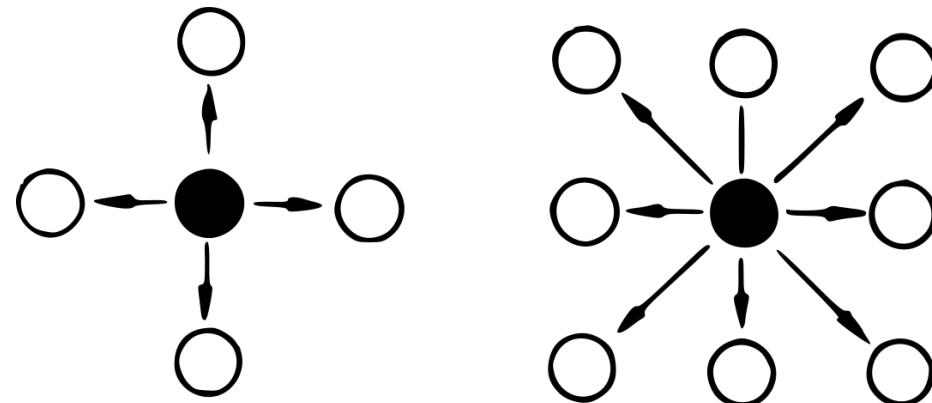
Układ taki nazywamy układem współrzędnych pikselowych.

Grafika i wizualizacja

- Algorytm Bresenhama

Założymy, że końcami rysowanego odcinka są punkty o współrzędnych pikselowych (x_0, y_0) i (x_k, y_k) .

Jako pierwszy piksel przybliżający rysowany odcinek przyjmujemy jeden z krańcowych, np. (x_0, y_0) . Dalej, aby wybrać kolejny piksel, możemy stosować wybór cztero- (lewy) lub ośmiokierunkowy (prawy) (ilustrują je rys., na których czarny piksel to ten wybrany w poprzednim kroku).

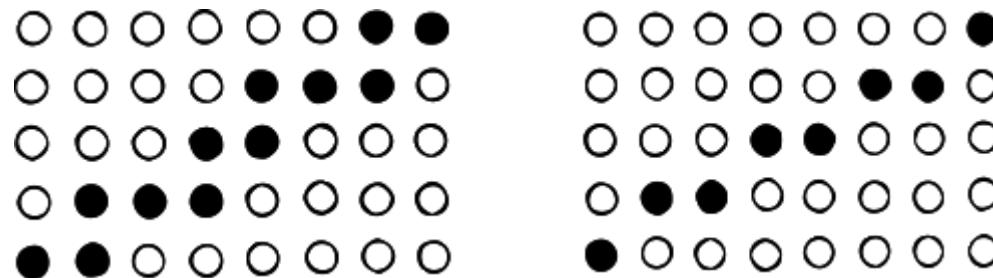


Grafika i wizualizacja

- Algorytm Bresenhama

Z czterech lub ośmiu możliwych wybieramy jako kolejny ten piksel, którego środek leży najbliżej odcinka.

Na rysunku zaznaczono zbiory pikseli przybliżające ten sam odcinek z wyborem cztero- i ośmiokierunkowym (odpowiednio części lewa i prawa rysunku). Różnią się one liczbą elementów!



Grafika i wizualizacja

- Algorytm Bresenhama

Oba rysunki robią wrażenie „schodkowych”. Jeśli dysponujemy urządzeniem ekranowym o wielu kolorach (jasnościach), to ten efekt możemy osłabić, stosując wybór czterokierunkowy i wyświetlając piksele z różnymi odpowiednio dobranymi jasnościami. Podobnie, stosując różne jasności wyświetlanych elementów obrazu, możemy uzyskać „jednolitość” linii poziomych i ukośnych.

Zagadnienia wyrównywania „schodków” rysunku (ang. antialiasing) powinny również zostać wyjaśnione na tym przedmiocie.

Grafika i wizualizacja

- Algorytm Bresenhama

Na ekranie monochromatycznym, tzn. o jednym kolorze (jasności), linie poziome wydają się grubsze (jaśniejsze) niż ukośne, gdyż ta sama ilość światła przy pada na mniejszy obszar. Na takich urządzeniach wizualnie lepsze efekty daje wybór ośmiokierunkowy. Właśnie on jest realizowany w najbardziej popularnym algorytmie rastrowego rysowania odcinków, zaproponowanym w 1962 r. przez Bresenhama.

Grafika i wizualizacja

- Algorytm Bresenhama

Bez zmniejszania ogólności możemy założyć, że:

$$x_0 < x_k,$$

a współczynnik kierunkowy rysowanego odcinka spełnia nierówności

$$0 < dy/dx < 1,$$

gdzie:

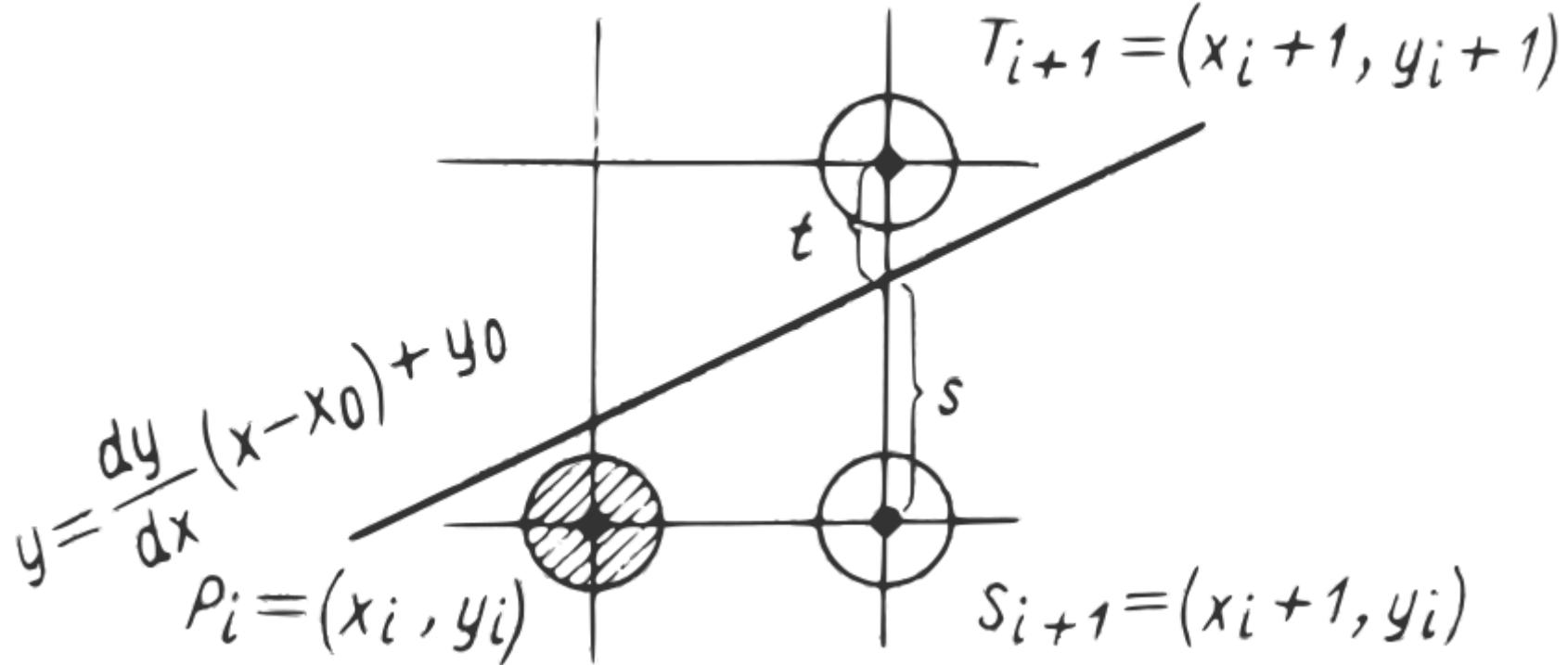
$$dy = y_k - y_0, \ dx = x_k - x_0.$$

W jakim kierunku przebiega ta linia?

Grafika i wizualizacja

- Algorytm Bresenhama

Ilustracja
przyjrzyjmy się oznaczeniom



Grafika i wizualizacja

- Algorytm Bresenhama

Zaczynamy od piksela $P_0 = (x_0, y_0)$.

Ponieważ kąt nachylenia odcinka jest ograniczony do przedziału $<0^\circ, 45^\circ>$, więc po znalezieniu kolejnego $P_i = (x_i, y_i)$ następny piksel wybieramy spośród tylko dwóch:

$S_{i+1} = (x_i + 1, y_i)$ (na prawo) i $T_{i+1} = (x_{i+1}, y_{i+1})$ (prawo-góra).

Wielkości s i t (widoczne na rysunku) są określone równaniami:

$$s = \frac{dy}{dx}(x_i + 1 - x_0) - (y_i - y_0)$$

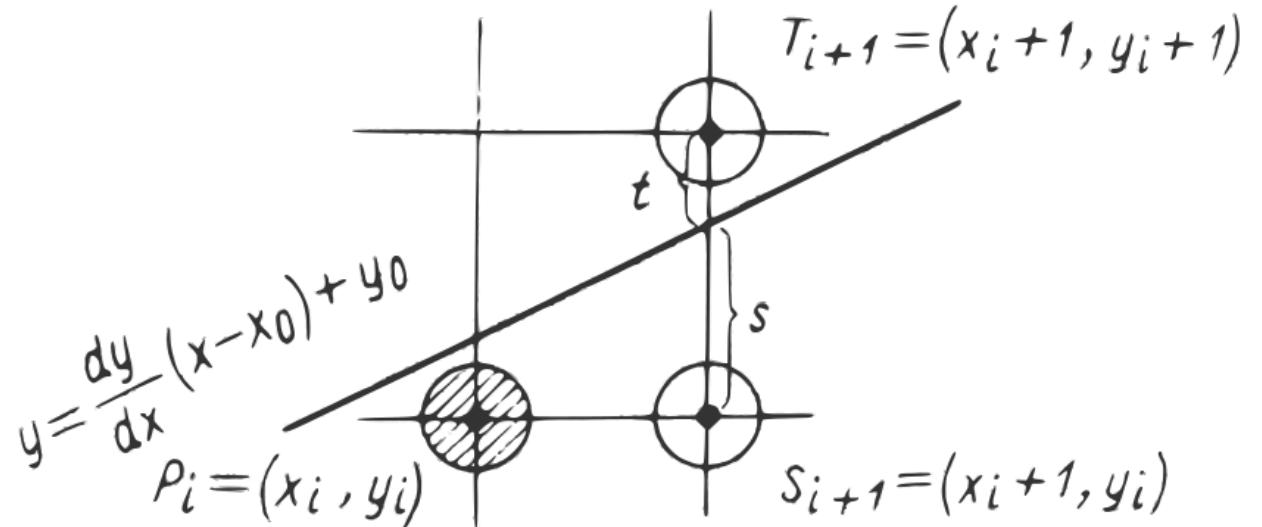
$$t = (y_i + 1 - y_0) - \frac{dy}{dx}(x_i + 1 - x_0)$$

Grafika i wizualizacja

- Algorytm Bresenhama

Odejmując te równania stronami i mnożąc przez dx , dostajemy:

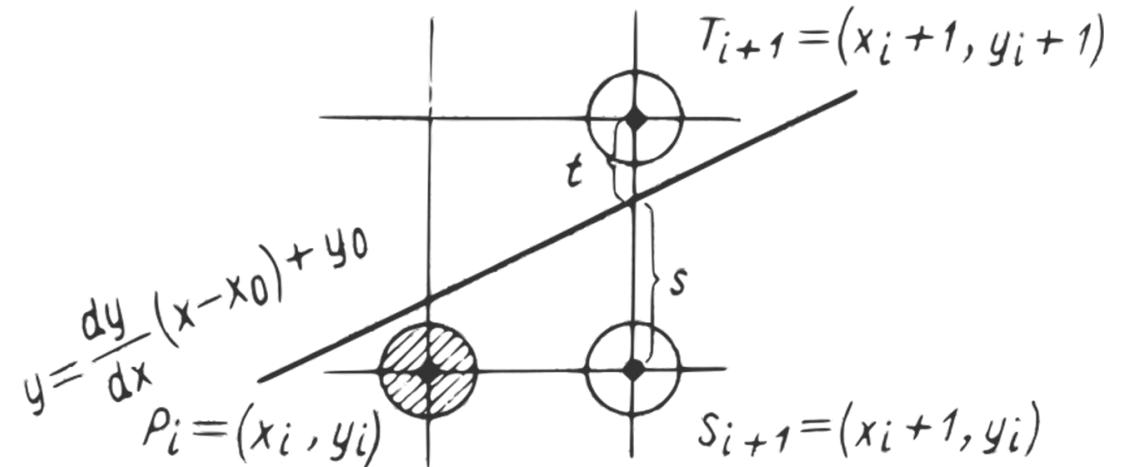
$$d_i = dx(s-t) = 2dy(x_i - x_0) - 2dx(y_i - y_0) + 2dy - dx \quad (1)$$



Grafika i wizualizacja

- Algorytm Bresenhama

Ponieważ $dx > 0$, więc znak d_i określa, która z wielkości s czy t jest większa. Jeśli $d_i > 0$, to $s > t$ i za P_{i+1} przyjmujemy piksel T_{i+1} ; w przeciwnym zaś przypadku, jeżeli $d_i < 0$, wybieramy S_{i+1} . Równość $d_i = 0$ oznacza, że oba piksele S_{i+1} i T_{i+1} leżą w tej samej odległości od odcinka i wtedy możemy arbitralnie decydować na przykład, że $P_{i+1} = T_{i+1}$.



Grafika i wizualizacja

- Algorytm Bresenhama

Dla $i+1$ wzór (1):

$$d_i = dx(s-t) = 2dy(x_i - x_0) - 2dx(y_i - y_0) + 2dy - dx \quad (1)$$

ma postać:

$$d_{i+1} = 2dy(x_{i+1} - x_0) - 2dx(y_{i+1} - y_0) + 2dy - dx \quad (2)$$

odejmując od niego stronami równanie (1) otrzymujemy:

$$d_{i+1} - d_i = 2dy(x_{i+1} - x_i) - 2dx(y_{i+1} - y_i) \quad (3) \quad \text{czyli}$$

$$d_{i+1} = d_i + 2dy - 2dx(y_{i+1} - y_i), \text{ gdy } x_{i+1} - x_i = 1$$

Grafika i wizualizacja

- Algorytm Bresenhama

Jeśli $d_i \geq 0$ (wybieramy wtedy $P_{i+1} = T_{i+1}$) to:

$y_{i+1} = y_i + 1$ i zależność rekurencyjna (3) upraszcza się do postaci:

$$d_{i+1} = d_i + 2(dy - dx)$$

a jeżeli $d_i < 0$ (przyjmujemy wówczas $P_{i+1} = S_{i+1}$), to $y_{i+1} = y_i$ co daje:

$$d_{i+1} = d_i + 2dy$$

Grafika i wizualizacja

- Algorytm Bresenhama

Dla $i=0$ ze wzoru (1):

$$d_i = dx(s-t) = 2dy(x_i - x_0) - 2dx(y_i - y_0) + 2dy - dx \quad (1)$$

otrzymujemy:

$$d_0 = 2dy - dx$$

Przy przyjętych założeniach algorytm Bresenhama dla wyznaczania lokalizacji pikseli rysowanego odcinka $(x_0, y_0) \rightarrow (x_k, y_k)$ będzie następujący: →

Grafika i wizualizacja

- Algorytm Bresenhama

Dla $i=0$ ze wzoru (1):

$$d_i = dx(s-t) = 2dy(x_i - x_0) - 2dx(y_i - y_0) + 2dy - dx \quad (1)$$

otrzymujemy:

$$d_0 = 2dy - dx$$

Przy przyjętych założeniach algorytm Bresenhama dla wyznaczania lokalizacji pikseli rysowanego odcinka $(x_0, y_0) \rightarrow (x_k, y_k)$ będzie następujący: →

Grafika i wizualizacja

- Algorytm Bresenhama

...następujący: →

start:

$$P_0 = (x_0, y_0); \quad dx = x_k - x_0; \quad dy = y_k - y_0;$$
$$dp = 2 * dy; \quad dd = 2 * (dy - dx); \quad d_0 = 2 * dy - dx;$$

dla $i = 0, 1, \dots, x_k - x_0 - 1$

$$x_{i+1} = x_i + 1;$$

jesli $d_i \geq 0$, to $\{d_{i+1} = d_i + dd; \quad y_{i+1} = y_i + 1\}$

w przeciwnym razie $\{d_{i+1} = d_i + dp; \quad y_{i+1} = y_i\}$

$$P_{i+1} = (x_{i+1}, y_{i+1})$$

Grafika i wizualizacja

- Algorytm Bresenhama

Pełny algorytm Bresenhama dla $x_k > x_0$, $y_k > y_0$, $0 < m < 1$ wyglądałby tak:

```
procedure Bresline (x0, xk, y0, yk, )
begin
dx := xk-x0; dy := yk-y0;
d0 := 2*dy-dx; dd := 2*(dy-dx);
x := x0; y := y0;
xend := xk;
set_pixel(x,y);
while (x<xend) do begin
x := x+1;
if (d<0) d := d+d0;
else begin
d := d+dd;
y := y+1;
end
set_pixel(x,y);
end
end
```

PASCAL ;-)

start:

$$P_0 = (x_0, y_0); \quad dx = x_k - x_0; \quad dy = y_k - y_0;$$
$$dp = 2 * dy; \quad dd = 2 * (dy - dx); \quad d_0 = 2 * dy - dx;$$

dla $i = 0, 1, \dots, x_k - x_0 - 1$

$$x_{i+1} = x_i + 1;$$

jeśli $d_i \geq 0$, **to** $\{d_{i+1} = d_i + dd; \quad y_{i+1} = y_i + 1\}$

w przeciwnym razie $\{d_{i+1} = d_i + dp; \quad y_{i+1} = y_i\}$

$$P_{i+1} = (x_{i+1}, y_{i+1})$$

Grafika i wizualizacja

- Algorytm Bresenhama

Podkreślmy, że algorytm wymaga działań tylko na liczbach całkowitych.

Ograniczmy się nadal do odcinków o współczynniku kierunkowym $dy/dx \in (0, 1]$.

Startując od (x_0, y_0) , w algorytmie Bresenhama wybieramy kolejne piksele i wykonujemy na siatce rastrowej ruchy poziome (wybór S_i) oraz diagonalne (wybór T_i).

Niech S oznacza ruch poziomy, T diagonalny, a $(S^2T)^3$ niech będzie skrótownym zapisem ciągu SST SST SST.

Rysując np. odcinek $(0,0) – (131, 16)$, wykonamy następujące ruchy:

$S^4T(S^7T)^4S^8T(S^7T)^5S^8T(S^7T)^4S^4$

Grafika i wizualizacja

- Algorytm Bresenhama

Wyznaczenie powtarzających się grup ruchów pozwala na przyspieszenie (szczególnie w przypadku długich linii) rysowania odcinków i kompresję danych potrzebnych do ich zapamiętania oraz może ułatwić skalowanie rysunków i obcinanie ich do zadanych obszarów.



Grafika i wizualizacja

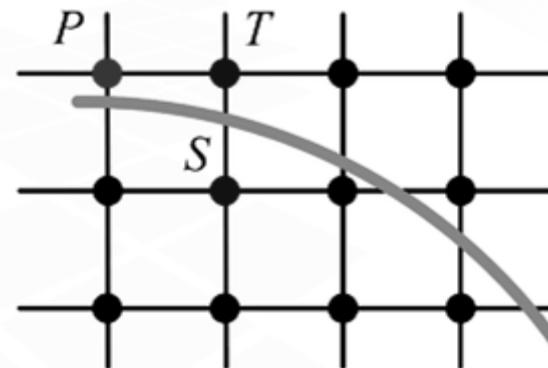
- Rysowanie okręgu

Grafika i wizualizacja

- Rysowanie okręgu

Na podobnej zasadzie jak rysowanie odcinka, Bresenham opracował algorytm rysowania łuku okręgu. Oczywiście zmieniają się w tym przypadku warunki wyboru pikseli, ale algorytm ten również wykorzystuje tylko operacje na liczbach całkowitych.

Wychodząc z piksela P wybieramy piksel następny spośród S i T



Grafika i wizualizacja

- Rysowanie okręgu

O ile przy rysowaniu odcinków wartość aspektu a urządzenia graficznego była nieistotna (aspekt a definiuje się jako stosunek odległości środków pikseli sąsiednich w poziomie do odległości środków pikseli sąsiednich w pionie), o tyle w przypadku okręgu musimy go uwzględnić.

Inaczej dla $a <> 1$ zamiast okręgu otrzymalibyśmy elipę.

Dalej będziemy zakładać, że a jest liczbą wymierną,

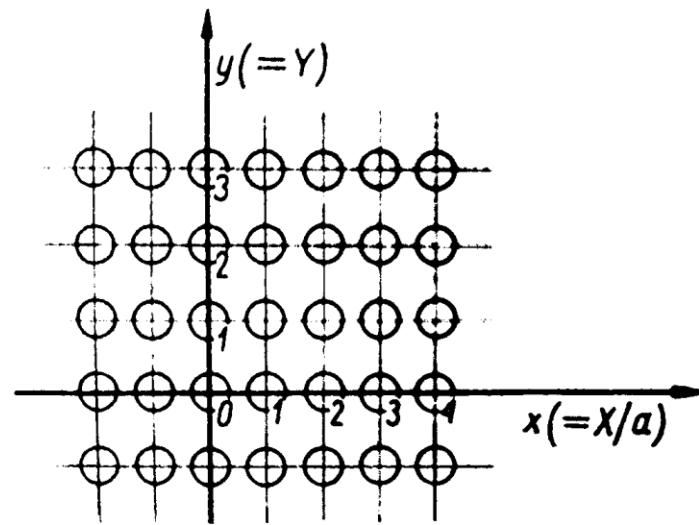
$a = p/q$ (p, q – liczby naturalne).

Grafika i wizualizacja

- Rysowanie okręgu

W ogólnym przypadku piksele urządzenia rastrowego tworzą siatkę prostokątną. Opiszmy ją dwoma układami układem zmiennych pikselowych Oxy i układem zmiennych rzeczywistych OXY - takimi, że

$$X = xa, \quad Y = y.$$



Grafika i wizualizacja

- Rysowanie okręgu

Zakładamy, że promień R okręgu, który chcemy narysować, jest liczbą naturalną, a jego środek leży w początku układu (wykonanie ewentualnego przesunięcia nie powinno sprawić trudności).

Zadanie polega na wybraniu pikseli przybliżających krzywą

$$X^2 + Y^2 - R^2 = 0 \quad (\text{czyli opisywanej równaniem okręgu})$$

czyli

$$(x_a)^2 + y^2 - R^2 = 0 \quad (\text{biorąc pod uwagę aspekt})$$

czyli

$$f(x, y) = p^2 x^2 + q^2 y^2 - q^2 R^2 = 0 \quad (4) \quad (\text{gdyż założyliśmy, że } a \text{ jest liczbą wymierną, } a = p/q \text{ (} p, q \text{ - liczby naturalne).})$$

Grafika i wizualizacja

- Rysowanie okręgu

Tak jak poprzednio przy rysowaniu odcinka zastosowany jest tu wybór ośmiokierunkowy.

Ze względu na symetrię można ograniczyć się do pierwszej ćwiartki okręgu (a przy $a = 1$ wystarczy rozważyć tylko 1/8 okręgu).

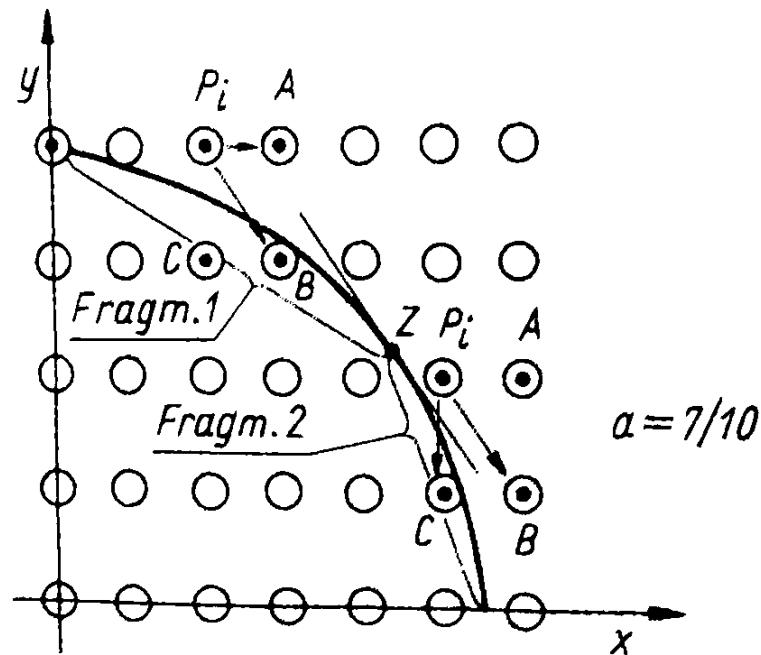
Zaczynamy od piksela $P_0 = (0, R)$, a następnie wyznaczamy zgodnie z kierunkiem obrotu wskazówek zegara.

Łatwo zauważyć, że przy takiej kolejności w pierwszej ćwiartce po znalezieniu piksela P_i , wybór następnego P_{i+1} ogranicza się do jednego z trzech pikseli oznaczonych na rys. (kolejny slajd) literami A, B i C.

Grafika i wizualizacja

- Rysowanie okręgu

Rysunek. Podział ćwiartki okręgu na fragmenty w algorytmie Bresenhama.



Grafika i wizualizacja

- Rysowanie okręgu

Punkt Z , w którym współczynnik kierunkowy wektora stycznego

$$\frac{dy}{dx} = -fx/fy = -2p^2x/2q^2y$$

jest równy -1 , dzieli ćwiartkę okręgu na dwa fragmenty (rys.).

W pierwszym fragmencie (tam, gdzie $p^2x < q^2y$), zwiększając kolejno wartość x o jeden wybieramy P_{i+1} spośród pikseli A i B , a w drugim fragmencie zmniejszamy kolejno y o jeden i wybieramy P_{i+1} spośród B i C .

Grafika i wizualizacja

- Rysowanie okręgu

Aby rozstrzygnąć, środek którego z dwóch pikseli leży bliżej przybliżanego okręgu, moglibyśmy obliczyć wartości $f(x,y)$ dla odpowiednich współrzędnych x i y .

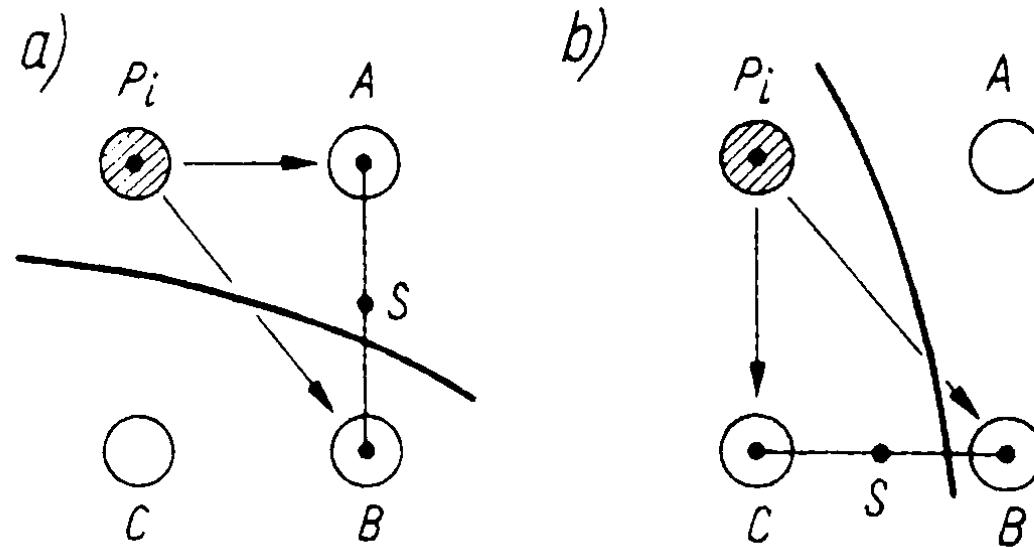
W wypadku krzywych zadanych bardziej skomplikowanym równaniem niż (4) znalezienie odległości środka piksela od krzywej może być zadaniem znacznie trudniejszym.

Wygodniej jest wtedy stosować inne kryterium wyboru. O wyborze będzie decydować wartość f w punkcie środkowym S między alternatywnymi pikselami.

Grafika i wizualizacja

- Rysowanie okręgu

*Rysunek. Wybór pikseli w algorytmie Bresenhama rysowania okręgu:
a) we fragmencie 1, b) we fragmencie 2)*



Grafika i wizualizacja

- Rysowanie okręgu

Ostateczny algorytm w formie pseudokodu:

start:

$$x = 0; \quad y = R;$$

$$pp = p * p; \quad pp4 = 4 * pp; \quad pp8 = 8 * pp;$$

$$qq = q * q; \quad qq4 = 4 * qq; \quad qq8 = 8 * qq;$$

$$fx = 0; \quad fy = qq8 * R; \quad fs = pp4 - qq4 * R + qq;$$

dopóki $fx < fy$

$$P = (x, y); \quad x = x + 1; \quad fx = fx + pp8;$$

jeśli $fs \leq 0$, **to** $fs = fs + fx + pp4$

w przeciwnym razie $\{y = y - 1; \quad fy = fy - qq8;$

$$fs = fs + fx + pp4 - fy\}$$

$$fs = fs - (fx + fy)/2 + 3 * (pp - qq)$$

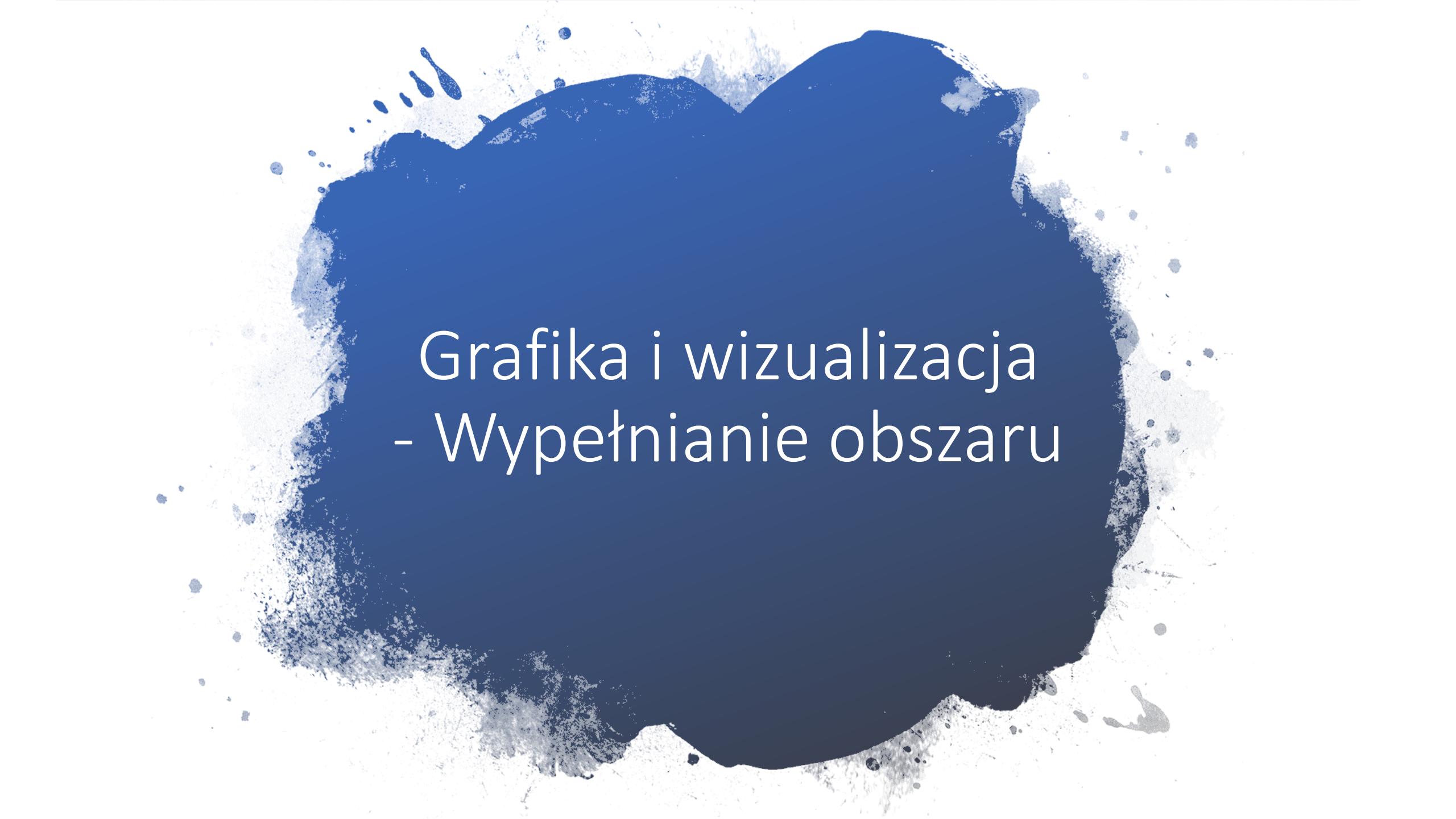
dopóki $y \geq 0$

$$P = (x, y); \quad y = y - 1; \quad fy = fy - qq8;$$

jeśli $fs \leq 0$, **to** $\{x = x + 1; \quad fx = fx + pp8;$

$$fs = fs + fx - fy + qq4\}$$

w przeciwnym razie $fs = fs - fy + qq4$



Grafika i wizualizacja

- Wypełnianie obszaru

Grafika i wizualizacja

- Wypełnianie obszaru

Problemem pojawiającym się w wielu praktycznych zadaniach jest wypełnianie obszaru zadanym kolorem lub wzorcem.

W systemach interakcyjnych użytkownik często rysuje brzeg obszaru, wskazuje jego wnętrze i wybiera kolor (wzorzec), jakim to wnętrze ma być wypełnione.

W tym przypadku brzeg obszaru jest pewnym zbiorem pikseli o ustalonej własności, np. danym kolorze.

Grafika i wizualizacja

- Wypełnianie obszaru

Wypełnianie obszaru jest drugim po rysowaniu odcinka lub łuku, najczęściej występującym problemem związanym z prymitywami.

Zadanie dla szczególnych przypadków (np. dla prostokąta) jest zadaniem trywialnym. Natomiast w ogólnym przypadku algorytm powinien pracować poprawnie dla dowolnego wielokąta (także wklęsłego) lub dla wielokątów z „dziurami”.

Grafika i wizualizacja

- Wypełnianie obszaru

W wielu innych zadaniach wypełniany obszar to wielobok, a ogólniej jest zadany analitycznie (we współrzędnych rzeczywistych, a nie pikselowych) przez krzywą zamkniętą składającą się z odcinków, łuków itp.

Zadanie wypełniania obszaru może więc być różnie formułowane.
Zacznijmy od sprecyzowania pierwszego - dyskretnego przypadku.

Grafika i wizualizacja

- Wypełnianie obszaru

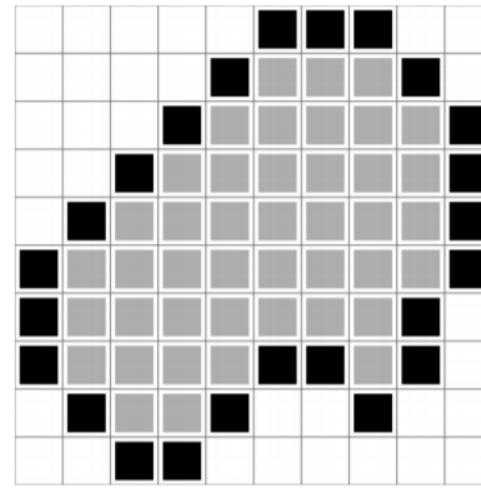
Zbiór pikseli nazywamy spójnym, gdy od dowolnego piksela z tego zbioru możemy przejść do każdego innego przez piksele sąsiednie.

Dokładniej, jeśli za piksele sąsiednie z danym uważamy te leżące bezpośrednio pod, nad, na lewo i na prawo (tak jak na rys.), to zbiór jest czterospójny; jeśli relację sąsiedztwa rozszerzamy na piksele leżące obok wzdłuż przekątnych (rys.), to zbiór nazywamy ośmiospójnym.

Grafika i wizualizacja

- Wypełnianie obszaru

Wypełnianie przez sianie (spójność)



Jeśli wybierzemy jeden punkt, który jest wewnątrz wypełnianego obszaru, to punkt sąsiedni będzie również punktem wewnątrz albo będzie punktem brzegowym.

Wypełnianie przez spójność zakłada znajomość punktu startowego (tzw. „ziarna”) wewnątrz obszaru. Punkt ten jest wypełniany, a następnie startując z niego wypełniamy punkty sąsiednie (jeśli oczywiście istnieją – jeśli nie są już wypełnione, ani nie są punktami granicznymi obszaru). Jednocześnie punkty sąsiednie stają się wyjściowymi dla wypełniania w następnym kroku. Procedura ta jest powtarzana dopóki można wskazać punkty wyjściowe (niewypełnione) wewnątrz obszaru.

Grafika i wizualizacja

- Wypełnianie obszaru

Warto zwrócić tutaj uwagę na problem sąsiedztwa i konieczność dostosowania do niego kształtu brzegu.

Jak powinien wyglądać rozkład pikseli brzegu dla obu przypadków aby stanowił on figurę zamkniętą z punktu widzenia możliwości ruchu?

Grafika i wizualizacja

- Wypełnianie obszaru

Algorytm wypełniania przez sianie (spójność) dla siatki czterospójnej.

Przyjęto: k_b – kolor brzegu, k_w – kolor wypełnienia

```
procedure wypełnij (x, y)
begin
set_pixel(x, y, c_f);
if (barwa(x-1, y) <>k_b i k_w) wypełnij (x-1, y);
if (barwa(x+1, y) <>k_b i k_w) wypełnij (x+1, y);
if (barwa(x, y-1) <>k_b i k_w) wypełnij (x, y-1);
if (barwa(x, y+1) <>k_b i k_w) wypełnij (x, y+1);
end
```

Grafika i wizualizacja

- Wypełnianie obszaru

Algorytm wypełniania przez sianie (spójność) dla siatki czterospójnej.

Proszę zauważyć, że przykładowy kod jest rekurencyjny. Jakie to rozwiązanie ma wady a jakie zalety?

Grafika i wizualizacja

- Wypełnianie obszaru

Algorytm wypełniania przez sianie (spójność) dla siatki czterospójnej.

Proszę zauważyć, że przykładowy kod jest rekurencyjny. Jakie to rozwiązanie ma wady a jakie zalety?

Grafika i wizualizacja

- Wypełnianie obszaru

Algorytm wypełniania przez sianie (spójność) dla siatki czterospójnej.

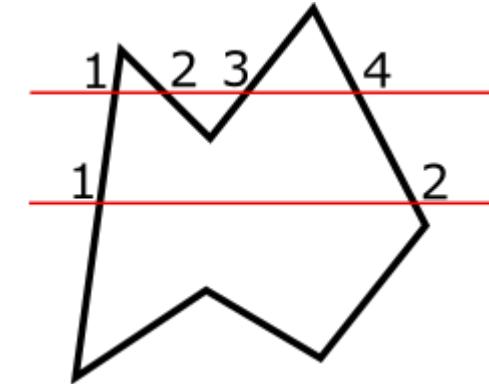
Proszę zauważyć, że przykładowy kod jest rekurencyjny. Jakie to rozwiązanie ma wady a jakie zalety?

Implementacja tego algorytmu jest prosta i krótka, jednak jego praktyczna realizacja może być kosztowna obliczeniowo. Przy dużym obszarze głębokość rekurencji staje się znaczna i może dochodzić do przepełnienia stosu. Dodatkowo algorytm jest rozrzutny – bada kolor danego piksela czasem nawet kilka razy.

Grafika i wizualizacja

- Wypełnianie obszaru

Wypełnianie przez kontrolę parzystości

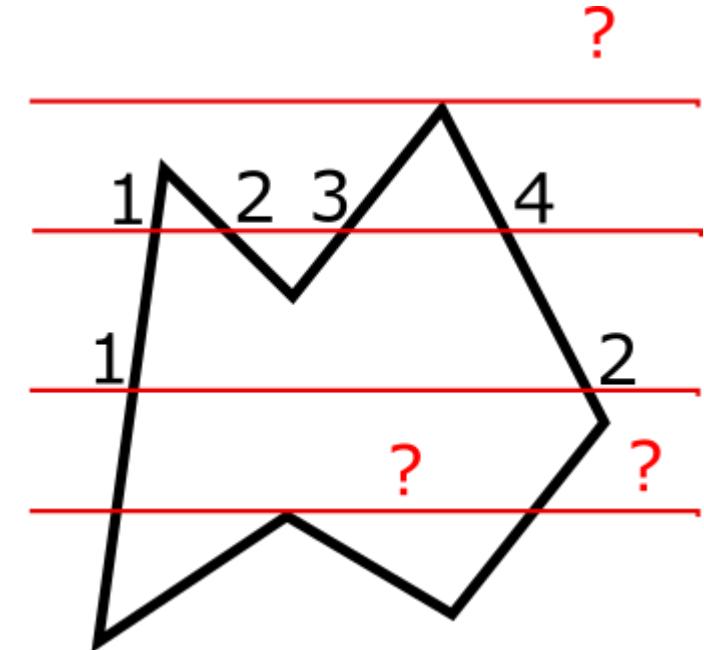


Wypełnianie przez kontrolę parzystości wykorzystuje pewną właściwość przecięcia brzegu linią prostą. Jeśli punkty przecięcia ponumerujemy kolejnymi liczbami naturalnymi zgodnie z orientacją prostej (na rysunku od lewej do prawej dla prostej poziomej) i jeśli będziemy poruszać się po prostej zgodnie z jej orientacją to każde nieparzyste przecięcie będzie „wejściem” do wnętrza obszaru, natomiast każde parzyste będzie „wyjściem” na zewnątrz. Zatem, aby wypełnić obszar należy go przeciąć prostymi odpowiadającymi kolejnym rzędem pikseli, a następnie wypełnić odcinkami pomiędzy każdym nieparzystym przecięciem, a najbliższym parzystym.

Grafika i wizualizacja

- Wypełnianie obszaru

Wypełnianie przez kontrolę parzystości - problemy



Należy zmodyfikować algorytm dla przypadków szczególnych, gdy punkt przecięcia jest jednocześnie lokalnym ekstremum brzegu, co zaburza prostą regułę parzystości. Lokalizacja lokalnego ekstremum możliwa jest na podstawie położenia końców przecinanych odcinków. Jeśli są po tej samej stronie prostej wypełniającej – przecinany punkt jest lokalnym ekstremum i nie należy jego liczyć przy analizie parzystości

Grafika i wizualizacja

- Wypełnianie obszaru

Przeglądanie liniami poziomymi (Scanline interpolation)

Grafika i wizualizacja

- Wypełnianie obszaru

Wejście: lista krawędzi wieloboku $\{ (x_i, y_i), (x_{i+1}, y_{i+1}) \}$,
 $i = 0, \dots, n$, $x_n = x_0$, $y_n = y_0$

Wynik: wypełnione wnętrze wieloboku

uporządkuj wierzchołki w krawędziach, aby $y_i < y_{i+1}$, usuń krawędzie poziome, uporządkuj
krawędzie w kolejności rosnących y_i

TAK $\leftarrow \emptyset$ (Tabela Aktywnych Krawędzi)

$y \leftarrow y_i$ pierwszej krawędzi

repeat

TAK \leftarrow TAK $\cup \{ \text{krawędzie, których pierwszy koniec jest na linii } y \}$

Opracowanie poziomu y

$y++;$

TAK \leftarrow TAK $\setminus \{ \text{krawędzie, których drugi koniec jest na linii } y \}$

until TAK = \emptyset

Grafika i wizualizacja

- Wypełnianie obszaru

Opracowanie poziomu y

Wejście: TAK $\langle\rangle\emptyset$ (zawiera parzystą ilość elementów)

Wynik: wypełniony poziom y

for all krawędzi z TAK do

Oblicz współrzędną x punktu przecięcia z linią poziomu y

end for

Posortuj TAK w kolejności rosnących współrzędnych x punktów przecięcia

for all kolejnych par krawędzi z TKA do

rysuj odcinek poziomy na linii y, między ich punktami przecięcia z linią y;

end for



Grafika i wizualizacja

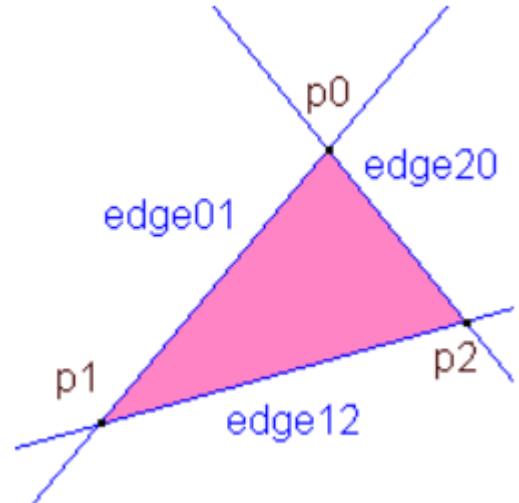
- Inne aspekty grafiki rastrowej

Grafika i wizualizacja

- Trójkąt

Kształt geometryczny **najpowszechniej** wykorzystywany w grafice komputerowej:

- bardzo prosta reprezentacja (3 wierzchołki i 3 brzegi),
- trójkąt zawsze jest wypukły (ang. convex),
- wierzchołki trójkąta zawsze leżą na jednej płaszczyźnie,
- za pomocą siatki trójkątów można aproksymować dowolny kształt 3D.



Grafika i wizualizacja

- Trójkąt

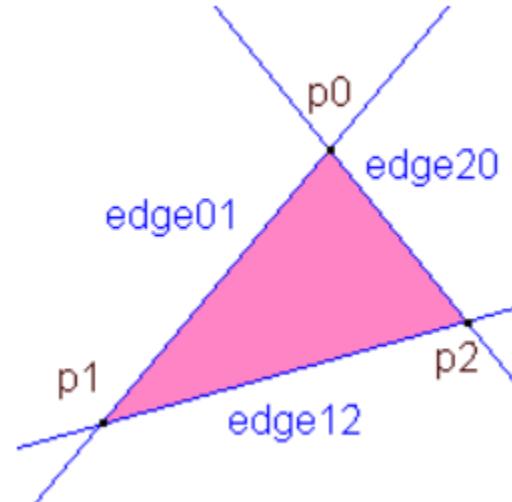
Trójkąty są określane za pomocą 3 punktów lub 3 krawędzi.

Trójkąt możemy zdefiniować za pomocą trzech punktów, na przykład:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3)$$

lub możemy zdefiniować trójkąt między jego trzema krawędziami, na przykład:

$$A_1 x + B_1 y + C_1 = 0, \quad A_2 x + B_2 y + C_2 = 0, \quad A_3 x + B_3 y + C_3 = 0$$



W rezultacie trójkąty są matematycznie bardzo proste. Matematyka związana ze skanowaniem trójkątów konwertujących obejmuje tylko proste równania liniowe.

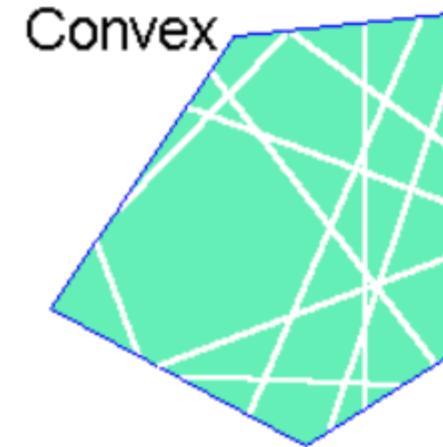
Grafika i wizualizacja

- Trójkąt

Trójkąty są zawsze wypukłymi wielokątami

Co to znaczy być wypukłym?

Obiekt jest wypukły wtedy i tylko wtedy, gdy jakikolwiek odcinek linii łączący dwa punkty na jego granicy jest w całości zawarty w obiekcie lub w jednej z jego granic.



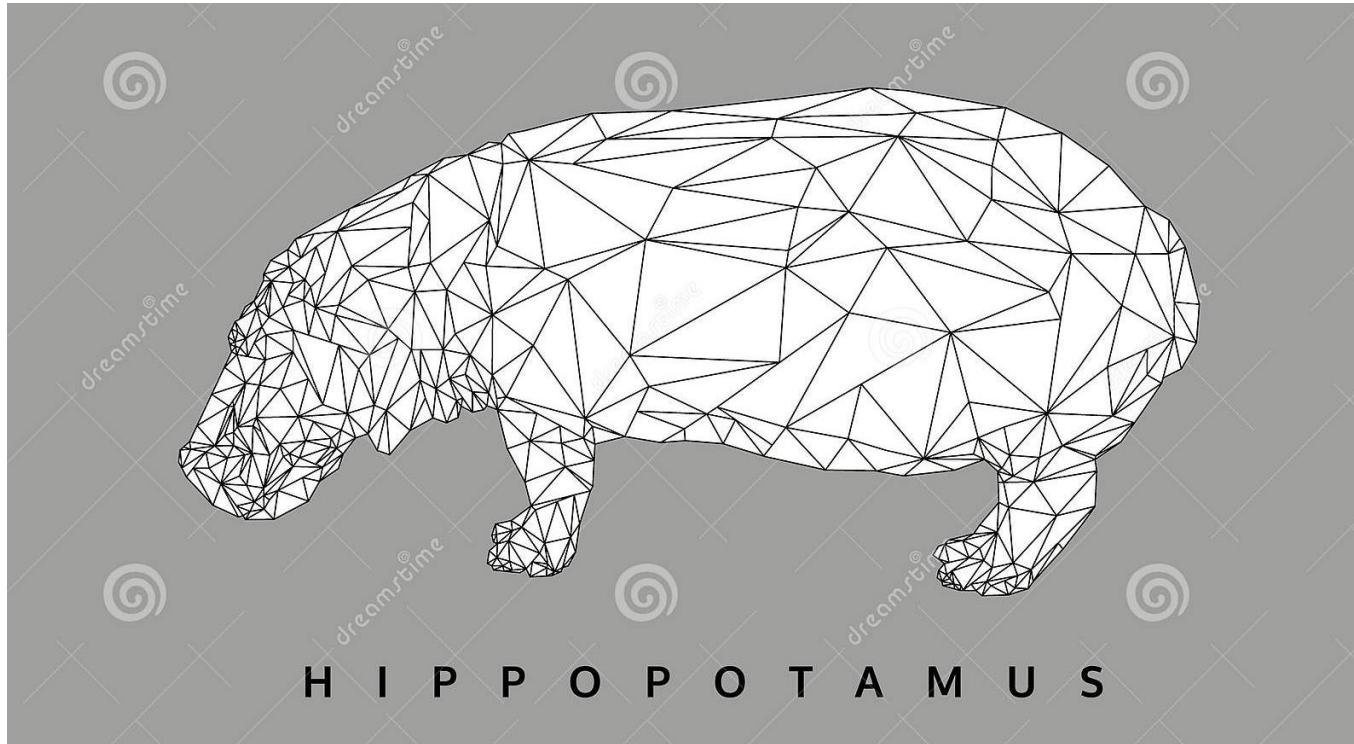
Dlaczego wypukłość jest ważna?

Ponieważ bez względu na to, jak trójkąt jest zorientowany na ekranie, dana linia skanowania będzie zawierała tylko jeden segment lub rozpiętość tego trójkąta.

Grafika i wizualizacja

- ...

Nawet hipopotamy są z trójkątów



Grafika i wizualizacja

- txt

Formaty zapisu grafiki rastrowej

ppm

Portable PixMap, zapis bez kompresji.

xpm

X-Window PixMap, zapis znakowy bez kompresji, z paletą barw dowolnego rozmiaru.
Grafikę w formacie xpm można tworzyć nawet zwykłym edytorem znakowym.

bmp

Bitmap, zapis bez kompresji lub z prymitywną kompresją.

pcx

Paintbrush, zapis z prymitywną kompresją.

gif

Graphics Interchange Format, zapis z kompresją, z paletą do 256 barw.

Grafika i wizualizacja

- txt

Formaty zapisu grafiki rastrowej

jpeg

Joint Photo Expert Group, zapis z kompresją stratną, z bezpośrednim kodowaniem barw pikseli (bez palety).

j2

Joint Photo Expert Group 2000, zapis z kompresją stratną albo bezstratną, z bezpośrednim kodowaniem barw pikseli.

png

Portable Network Graphics, zapis z kompresją.

tif

Tag Image File Format, otwarty format roboczy grafiki rastrowej dla urządzeń (np. skanerów) i edytorów graficznych.

djvu

Deja Vu, format zaawansowanej kompresji stratnej do przechowywania zeskanowanych wydawnictw, opracowany przez AT&T.

Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Grafika i wizualizacja

Wypełnianie obszarów

Edward Półrolniczak, dr inż.,

Katedra Systemów Multimedialnych,

Wydział Informatyki, ZUT w Szczecinie

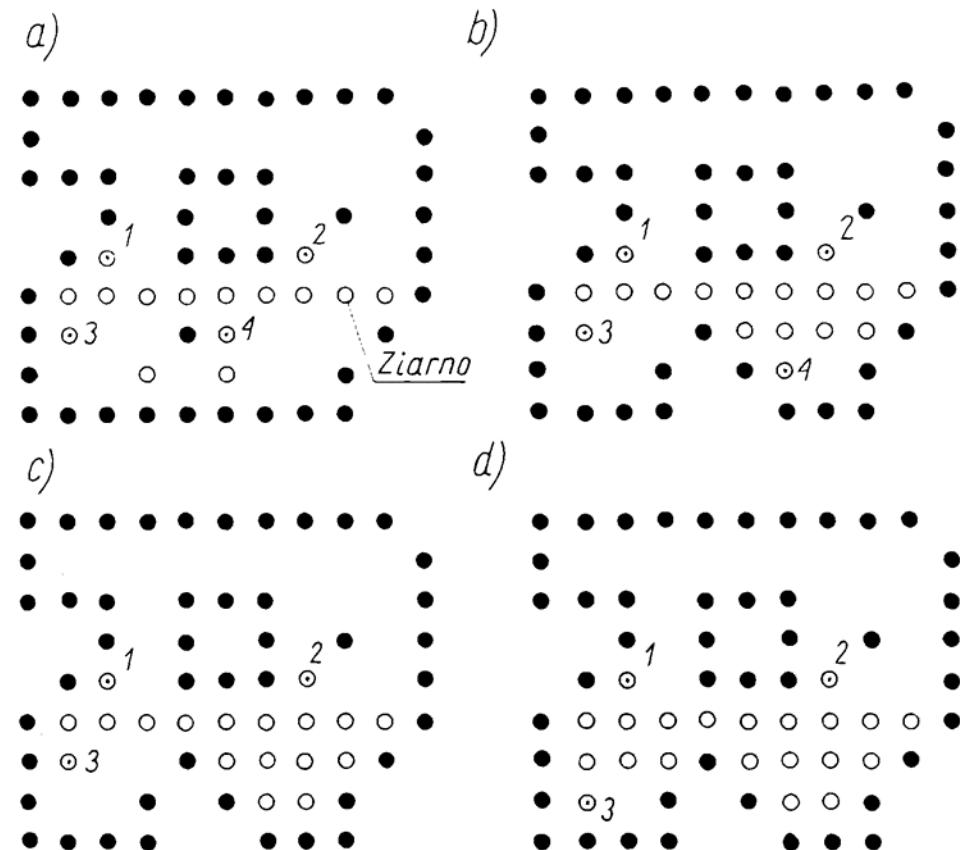
Grafika i wizualizacja

Zagadnienia

- Wypełnianie wieloboku zadanego analitycznie
- Problemy związane z wypełnianiem
- Rysowanie pogrubionych elementów pierwotnych

Schemat postępowania dla algorytmu Smitha

Po wypełnieniu pierwszego segmentu zawierającego ziarno przeglądamy (od lewej do prawej) linie leżące bezpośrednio nad oraz pod linią poprzednią i wyznaczamy początkowe (czyli leżące najbardziej na lewo) piksele segmentów sąsiednich, nie wypełnionych jeszcze nowym kolorem.

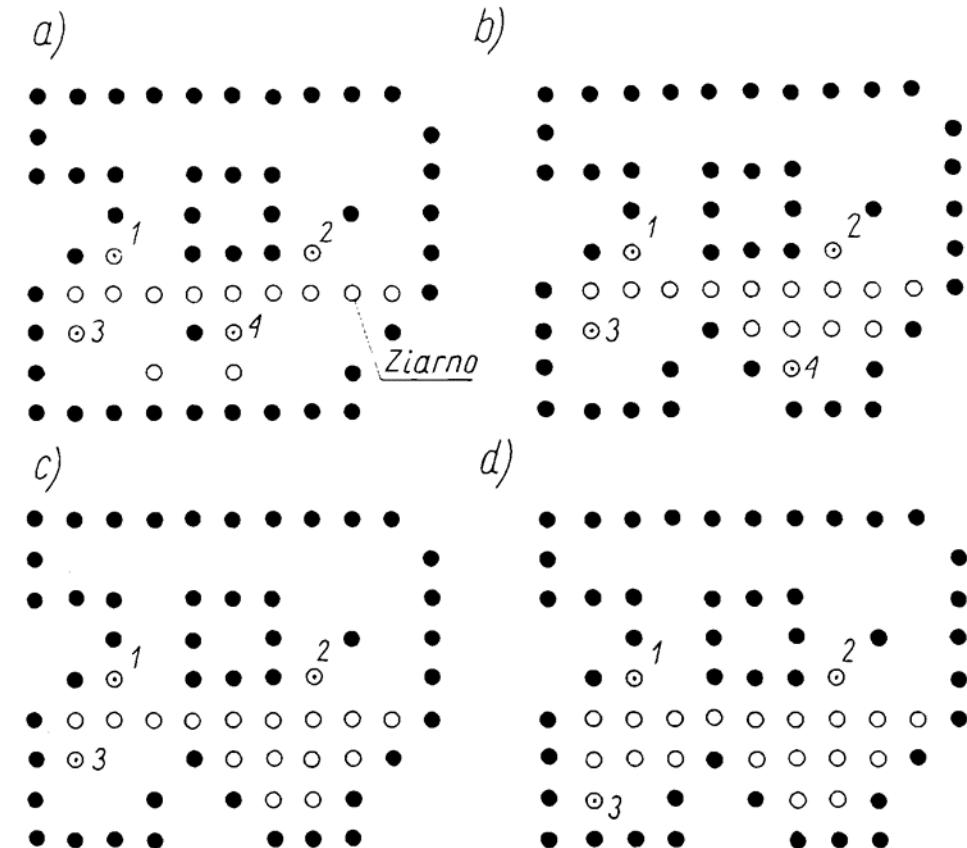


a) Zaznaczone ziarno i wypełniona linia zawierająca ziarno; 1, 2, 3 i 4 to nowe ziarna. b) linia zawierająca ziarno 4 została wypełniona, znalezione nowe ziarno nr 4 itd..

Schemat postępowania dla algorytmu Smitha

Współrzędne tych pikseli i długości segmentów odkładamy na stos.

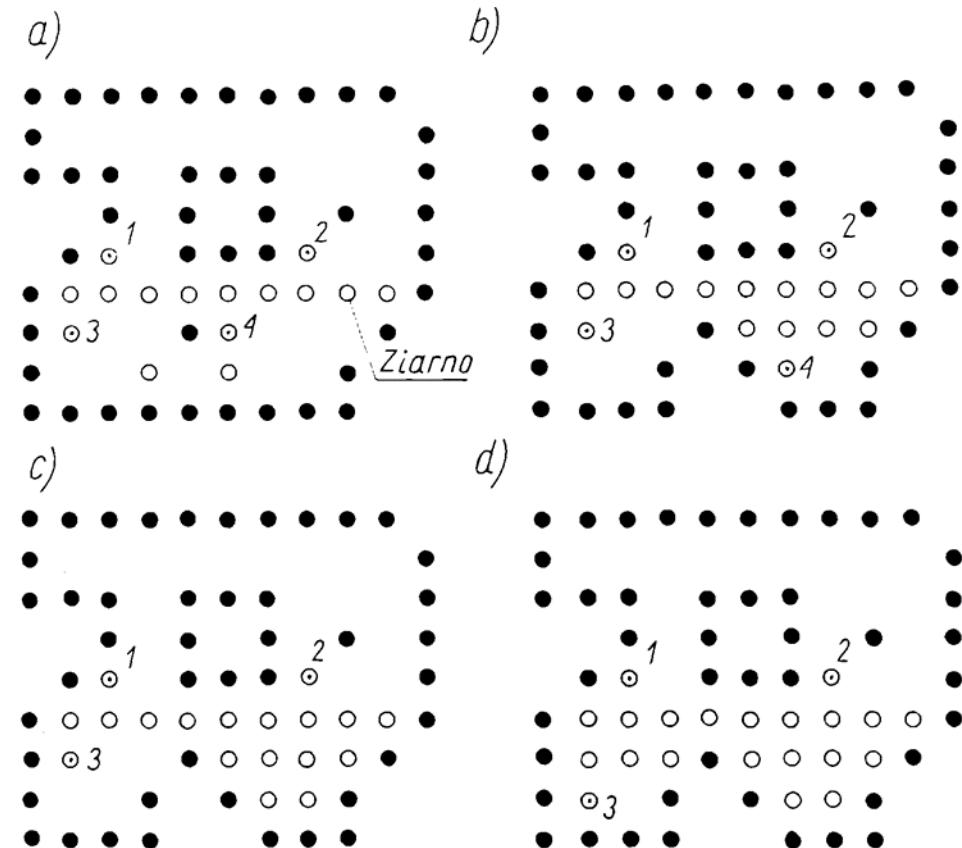
Piksel, którego adres jest na wierzchu stosu, bierzemy następnie jako nowe ziarno.



Schemat postępowania dla algorytmu Smitha

Rysunek ilustruje początkowe kroki takiego algorytmu...

Samodzielne dokonczenie postępowania prowadzącego do wypełnienia całego przykładowego obszaru jest intuicyjne.



Grafika i wizualizacja

Wypełnianie wieloboku zadanego analitycznie

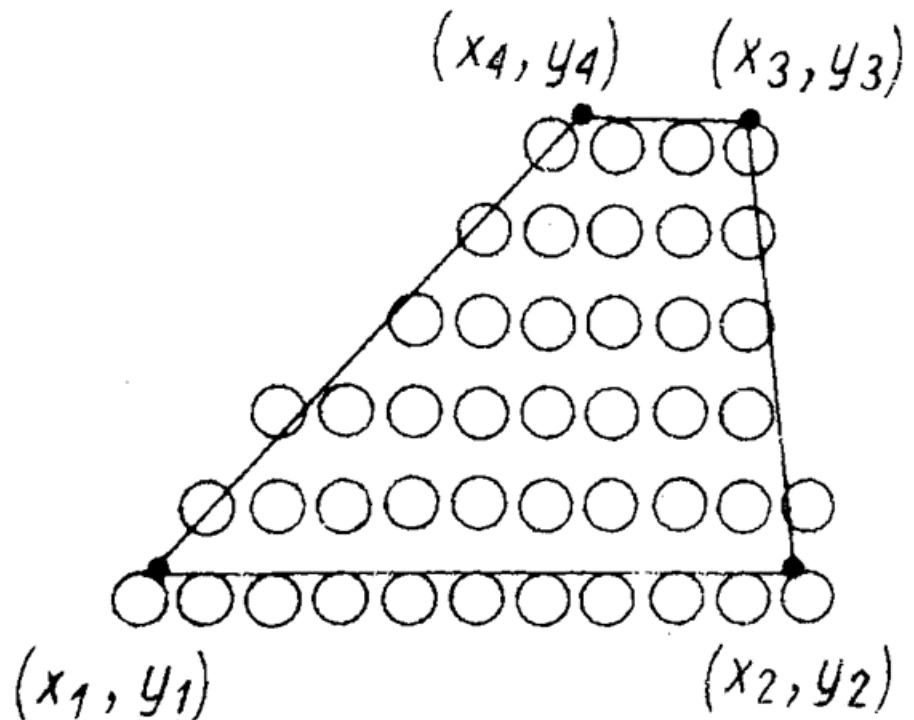
Obszar jest określony analitycznie jako wielokąt

Niech płaszczyzna rysunku będzie opisana współrzędnymi pikselowymi.

Dla uproszczenia zadania wypełniania na początek przyjęty zostanie szczególny przykład wielokąta - trapez o podstawach równoległych do osi x.

Obszar jest określony analitycznie jako wielokąt

Na ogół wierzchołki trapezu rzeczywistego (analogowego) mają współrzędne nie całkowite, lecz rzeczywiste, czyli nie pokrywające się ze środkami pikseli, tak jak na rys.:



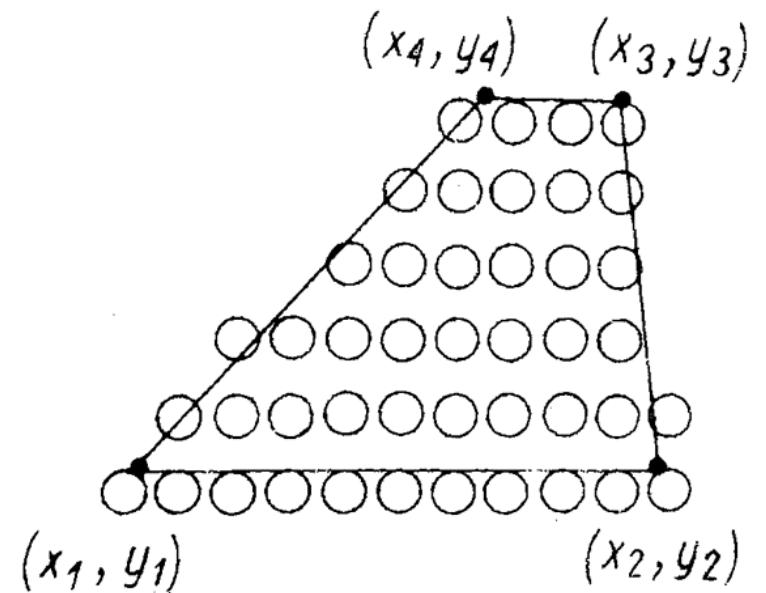
Niech $y_{\min} = \text{round}(y_1)$, a $y_{\max} = \text{round}(y_3)$, gdzie $\text{round}(r)$ jest zaokrągleniem liczby rzeczywistej r do najbliższej liczby całkowitej.

Z założenia $y_1 = y_2$, a $y_3 = y_4$

//trapez ma 2 równoległe do osi x boki

Dalej niech $c_l = (x_4 - x_1)/(y_4 - y_1)$ i $c_p = (x_3 - x_2)/(y_3 - y_2)$.
// c_l – lewy bok, c_p – prawy bok

Przy tych oznaczeniach algorytm wypełniania trapezu może przebiegać następująco...



Algorytm wypełniania trapezu

dla $y = y_{min}, y_{min} + 1, \dots, y_{max}$

- wyznacz współrzędne x_l i x_p przecięcia linii poziomej y z prostymi
 $x = x_1 + (y - y_1)c_l$ i $x = x_2 + (y - y_2)c_p$

- wypełnij nowym kolorem (wzorcem) piksele leżące na tej linii od $round(x_l)$ do $round(x_p)$.

Pamiętając, że: $c_l = (x_4 - x_1)/(y_4 - y_1)$ i $c_p = (x_3 - x_2)/(y_3 - y_2)$

oraz $y = y_{min}, y_{min} + 1, \dots, y_{max}$ będzie jednakowe dla x_l i x_p
to

$x_l = x_1 + (y - y_1)c_l$ i $x_p = x_2 + (y - y_2)c_p$

...dalej zauważmy, że dla $y = y_1 = y_2$ jest $x_l = x_1, x_p = x_2$ oraz że między przecięciami x_l i x_p obu prostych z linią poziomą y a przecięciami x'_l i x'_p , z kolejną linią poziomą $y+1$ zachodzą zależności

$$x'_l = x_l + c_l, \quad x'_p = x_p + c_p$$

Jeśli wykorzystamy to spostrzeżenie, to algorytm uprości się do postaci:

podstaw $x_l = x_1; x_p = x_2$; dla $y = y_{min}, y_{min}+1, \dots, y_{max}$

- wypełnij nowym kolorem (wzorcem) piksele leżące na linii y od $round(x_l)$ do $round(x_p)$;
- zmień $x_l = x_l + c_l; \quad x_p = x_p + c_p$;

Podział wielokąta na trapezy

Zadanie wypełniania dowolnego wielokąta można sprowadzić do omówionego wyżej przypadku szczególnego.

Podział wielokąta na trapezy

Istnieją algorytmy rozkładu wielokąta na sumę trapezów (np. Brassela i Fegeasa) .

Rozkład na trapezy może być przydatny:

- do wypełniania obszaru danym kolorem (wzorcem),
- przy działaniach na wielokątach i przy rysowaniu obiektów trójwymiarowych z uwzględnieniem oświetlenia,
- do zaoszczędzenia obliczeń, gdy wielokrotnie wypełniamy obszary tego samego kształtu, ale różnej wielkości, na przykład przy drukowaniu mniejszych i większych czcionek ustalonego kroju.

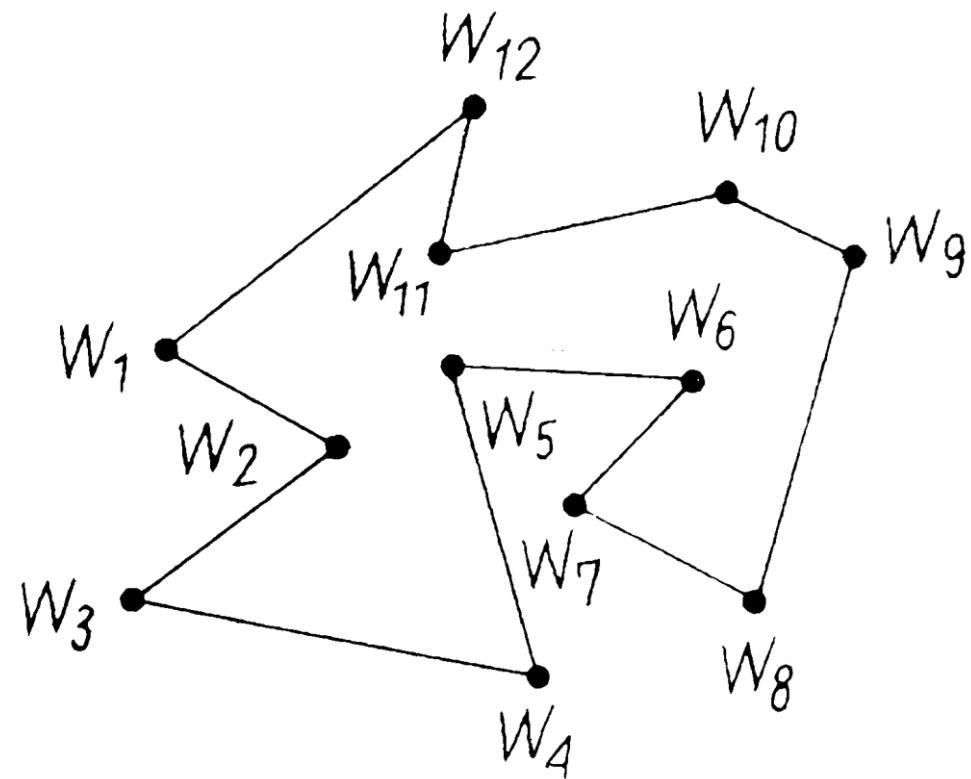
Podział wielokąta na trapezy

Algorytm może naśladować wypełnianie zbiornika wodą, przy czym dopływ wody jest umieszczony w najniższym punkcie zbiornika - obszaru.

Kluczem jest podział wielokąta na trapezy, potem można implementować, dla każdego z trapezów, przymocowany wcześniej algorytm.

Podział wielokąta na trapezy

Figura wejściowa

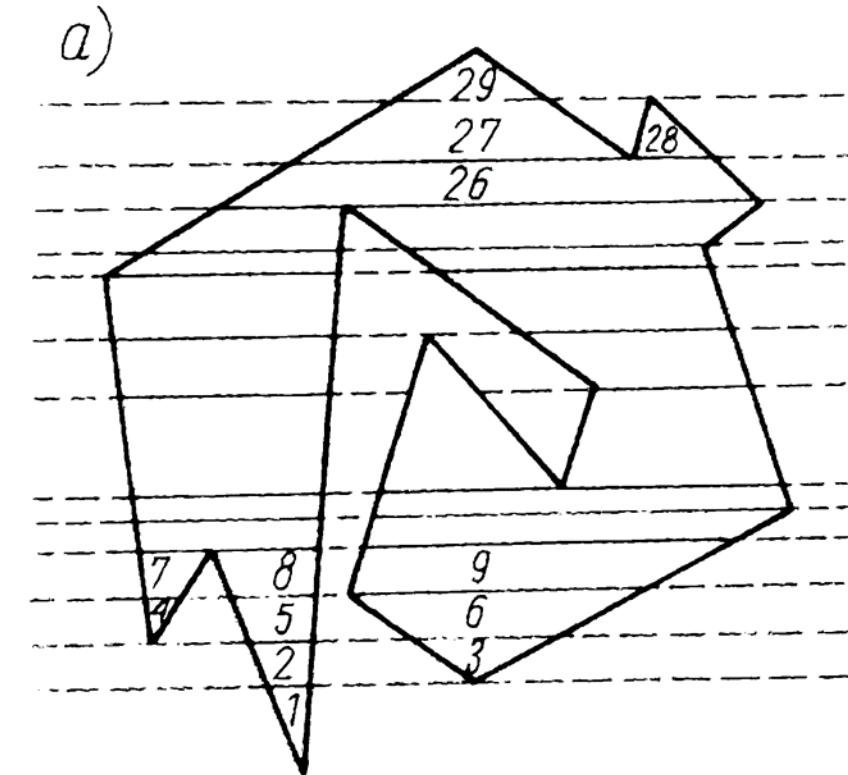


Podział wielokąta na trapezy

Narzucającym się sposobem podziału wielokąta na trapezy jest poprowadzenie linii poziomych przez wszystkie wierzchołki.

Odcinki kolejnych linii zawarte wewnętrz
wielokąta są podstawami trapezów, a fragmenty
odpowiednich krawędzi - ich bokami (rys. a).

W szczególnych przypadkach trapez może
degenerować się do trójkąta.



Podział wielokąta na trapezy

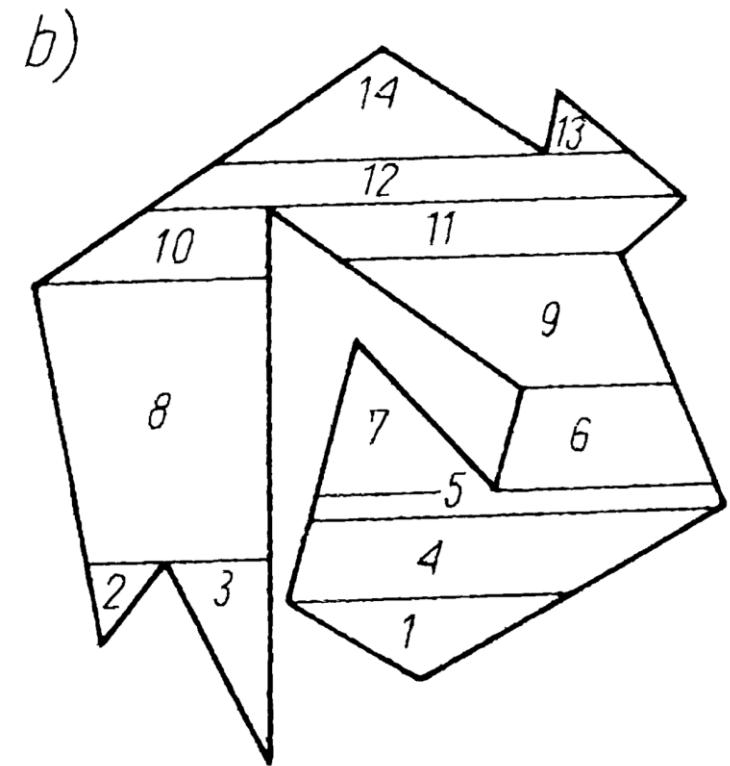
Przedstawiona wyżej pierwsza wersja algorytmu może dzielić wielokąt na sumę dużej liczby trapezów.

Przy n wierzchołkach, tych trapezów może być nawet rzędu n^2 . Tego samego rzędu będzie też liczba wykonywanych działań.

Koszt algorytmu i liczbę trapezów można zmniejszyć (rys. b), jeśli będziemy dzielili wielokąt nie wszystkimi leżącymi wewnątrz niego odcinkami linii poziomych, lecz tylko tymi odcinkami, które zawierają wierzchołki.

Podział wielokąta na trapezy

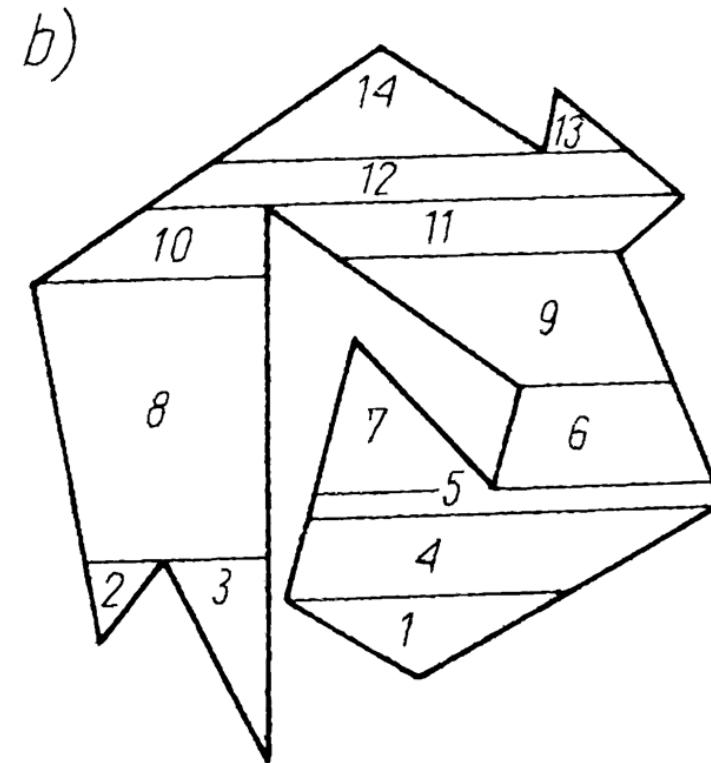
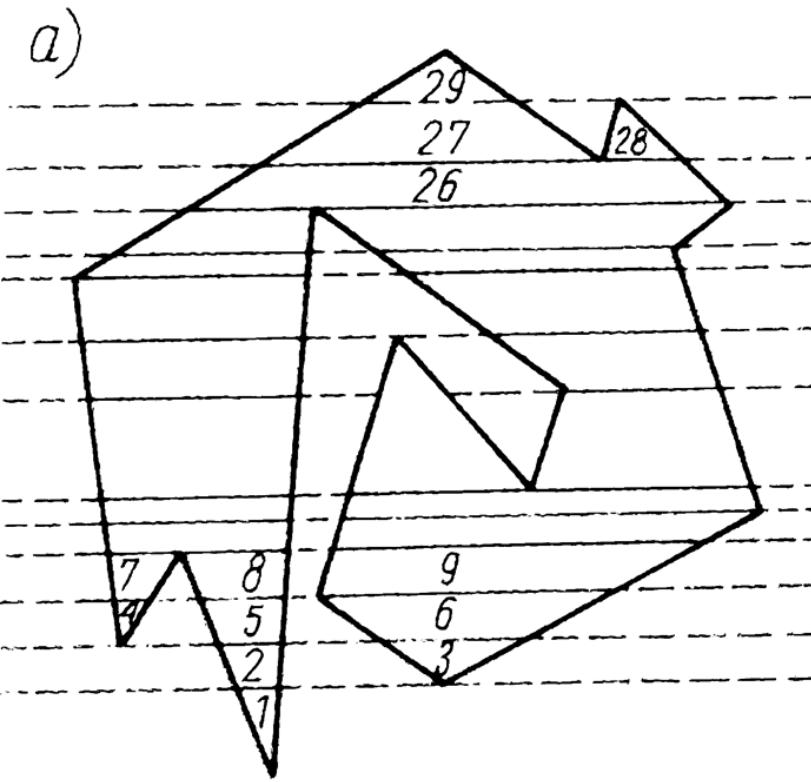
Koszt algorytmu i liczbę trapezów można zmniejszyć (rys. b), jeśli będziemy dzielili wielokąt nie wszystkimi leżącymi wewnątrz niego odcinkami linii poziomych, lecz tylko tymi odcinkami, które zawierają wierzchołki.



Podział wielokąta na trapezy

Porównanie podziałów:

- a) Linie przechodzące przez wierzchołki
- b) Wybrane fragmenty linii z tylko tymi odcinkami, które zawierają wierzchołki.



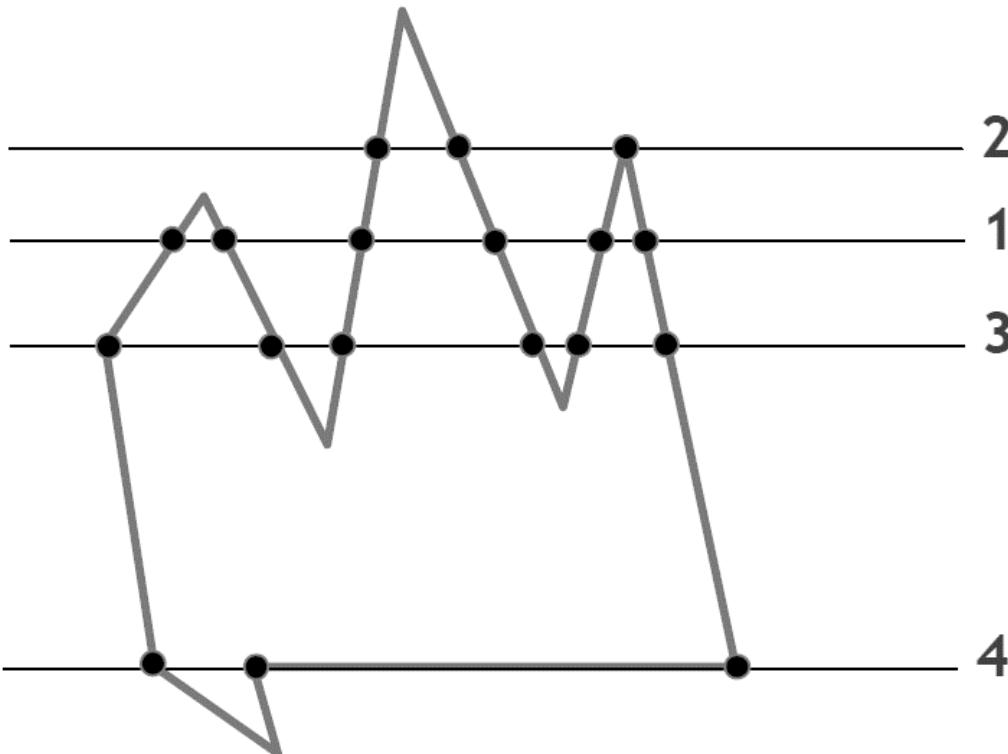
Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Zasada parzystości:

Prosta, która nie przechodzi przez wierzchołek przecina wielobok parzystą ilość razy.

położenie 1 – 6 przecięć
położenie 2 – 3 przecięcia
położenie 3 – 6 przecięć
położenie 4 – ? przecięć



Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Dodatkowe uwagi:

- dla $0 \leq a \leq 1$ równanie prostej ma postać:

$$y = ax + b, \text{ (na rysunku } b=0)$$

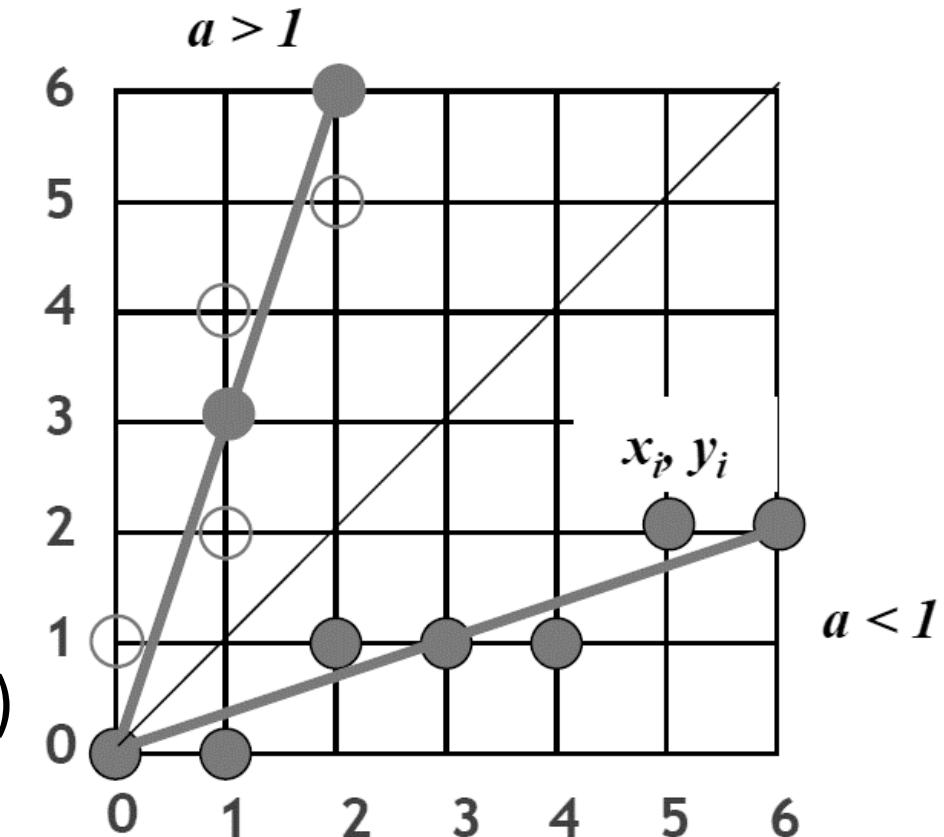
- dla $a > 0$ analizujemy równanie:

$$y = 1/a)x + b.$$

Jeśli $b=0$ to rysowane są punkty x_i, y_i

Jeśli $b > 0$ to rysowane są punkty $x_i, y_i + \text{round}(b)$

//algorytm DDA – Digital Differential Analyzer

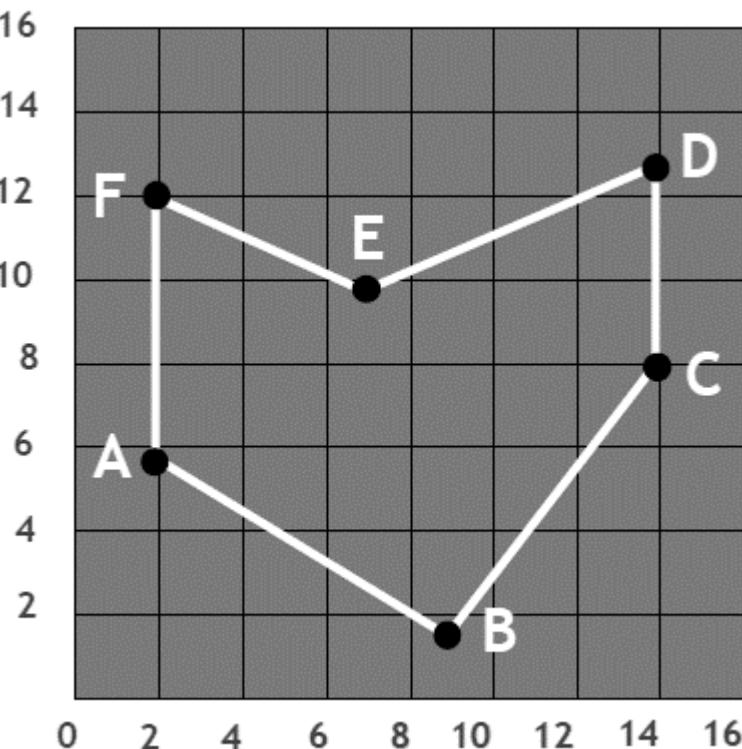


Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Wersja, gdy dany jest wielobok (bez krawędzi poziomych) opisany jako zbiór kolejnych krawędzi.

KRAWĘDŹ	y _{min}	y _{max}	x _{min}	x _{max}
AB	1	5	9	2
BC	1	7	9	14
CD	7	12	14	14
DE	9	12	7	14
EF	9	11	7	2
FA	5	11	2	2



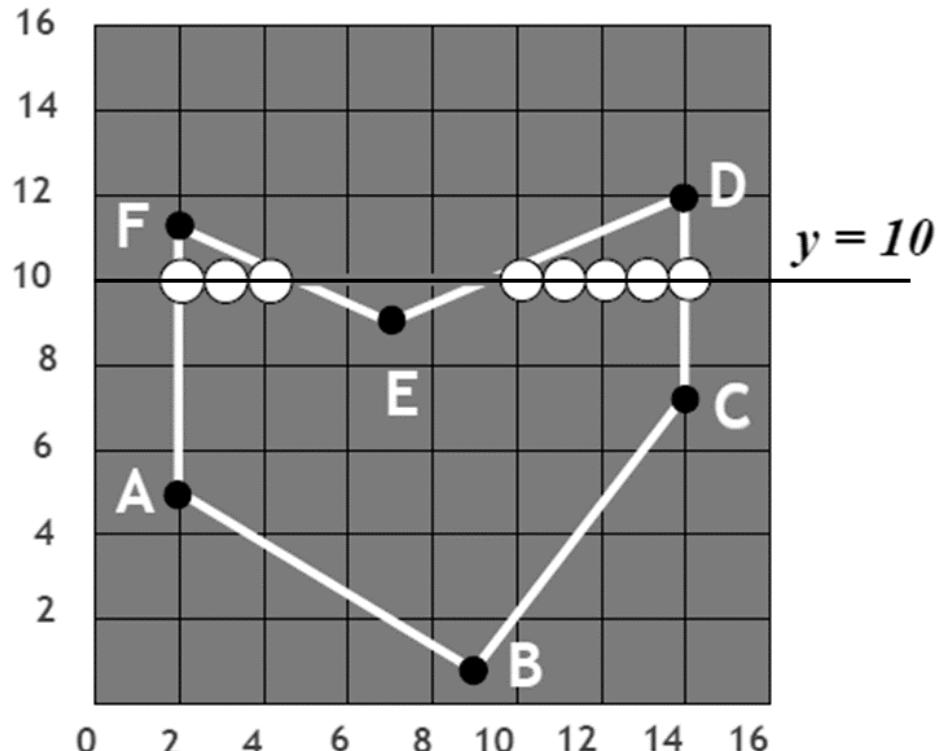
Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Podstawowe elementy działania algorytmu:

- przesuwać pozioma linię skanującą od dołu do góry,
- dla danego położenia linii należy wypełniać piksele pomiędzy poszczególnymi parami krawędzi .

W ogólności, dla linii skanującej w położeniu $y = 10$ należy wypełnić piksele (narysować poziome odcinki) pomiędzy krawędziami AF i FE oraz DE i CD.



Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Opis algorytmu:

Krok 0: Utworzyć globalną tablicę krawędzi (ET).

Krok 1: Ustawić y na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi (ET), czyli y dla pierwszej niepustej grupy krawędzi

Krok 2: Wyzerować aktywną tablicę krawędzi (AT).

Krok 3: Powtarzać tak długo, dopóki tablica globalna (ET) i tablica aktywna (AT) nie będą puste.

Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

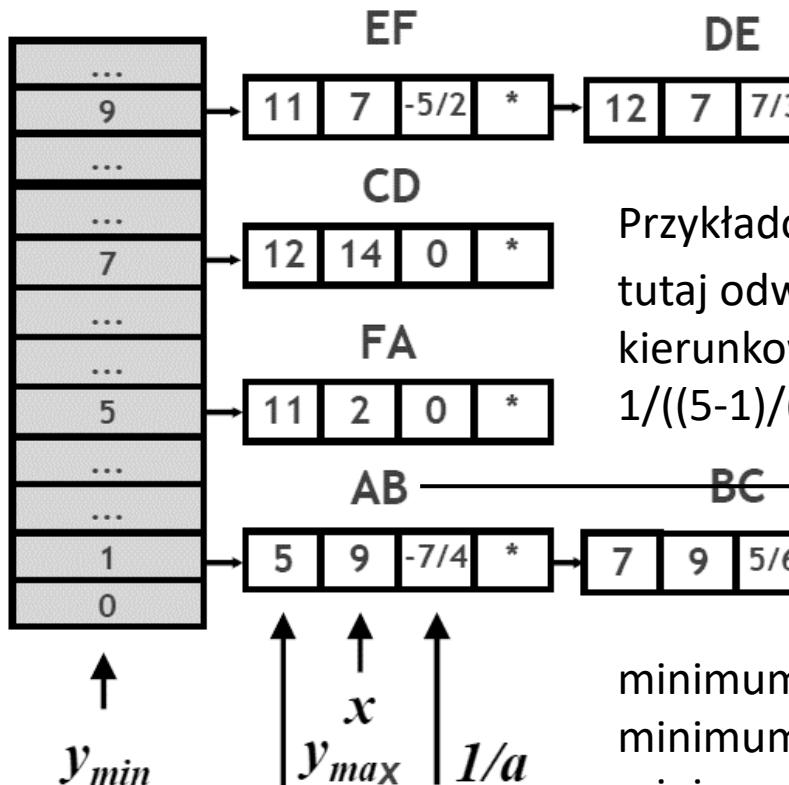
- Przenieść z grupy y tablicy globalnej (ET) do tablicy aktywnej (AT) te krawędzie, dla których $y_{\min} = y$ i posortować je ze względu na x.
- Wypełnić piksele w linii y, wykorzystując pary x z tablicy aktywnej (AT).
- Usunąć z tablicy aktywnej (AT) te krawędzie, dla których $y = y_{\max}$.
- Zwiększyć y o 1 (następna linia).
- Dla każdej pary krawędzi, która nie jest pionowa wyliczyć i wstawić do tablicy aktywnej (AT) nowe wartości x.



Grafika i wizualizacja

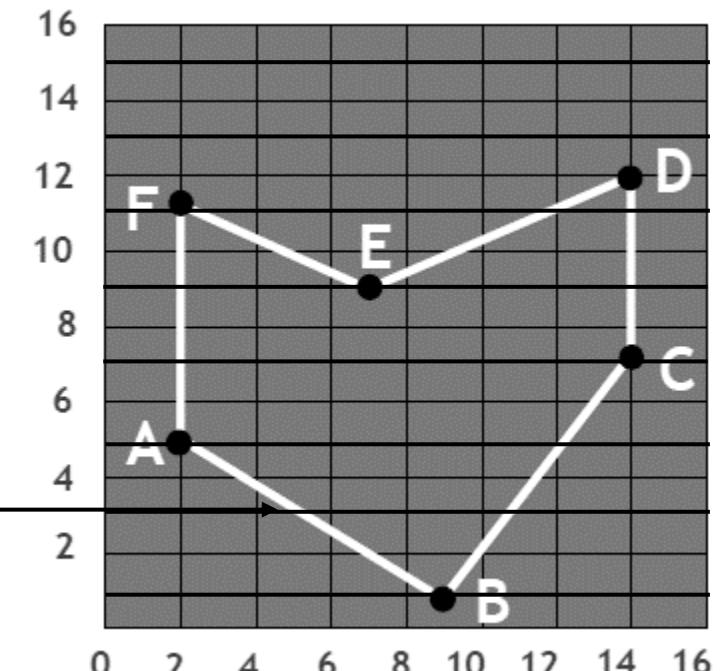
- Algorytm przeglądania linii (scan-line)

Krok 0: Tworzenie globalnej tablicy krawędzi (ET)



Przykładowe $-7/4$ wynika z a - $a = \frac{y_B - y_A}{x_B - x_A}$
tutaj odwróconego współczynnika
kierunkowego prostej AB
 $1/((5-1)/(2-9))=-7/4$

minimum y_{min} - przypisanie do grupy
minimum x_{min} - porządkowanie w grupie
minimum y_{max} - porządkowanie w grupie
minimum x_{max} - porządkowanie w grupie



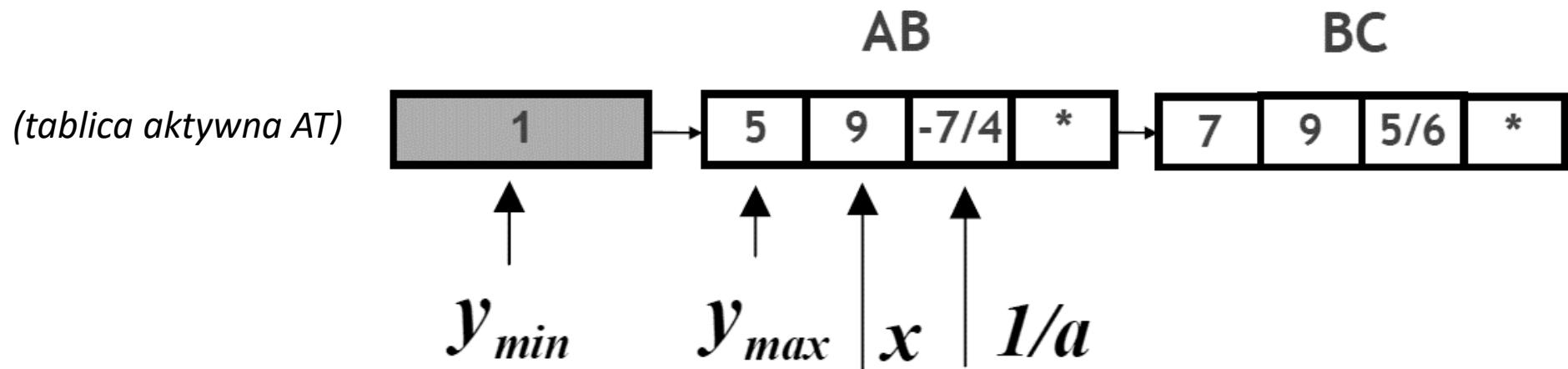
Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Krok 1: Ustawiamy y na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi. ($t u y = 1$)

Krok 2: Zerujemy aktywną tablicę krawędzi (AT).

Krok 3: Przenosimy z grupy y tablicy globalnej (ET) do tablicy aktywnej (AT) te krawędzie, dla których $y_{min} = y$ i sortujemy krawędzie ze względu na x .

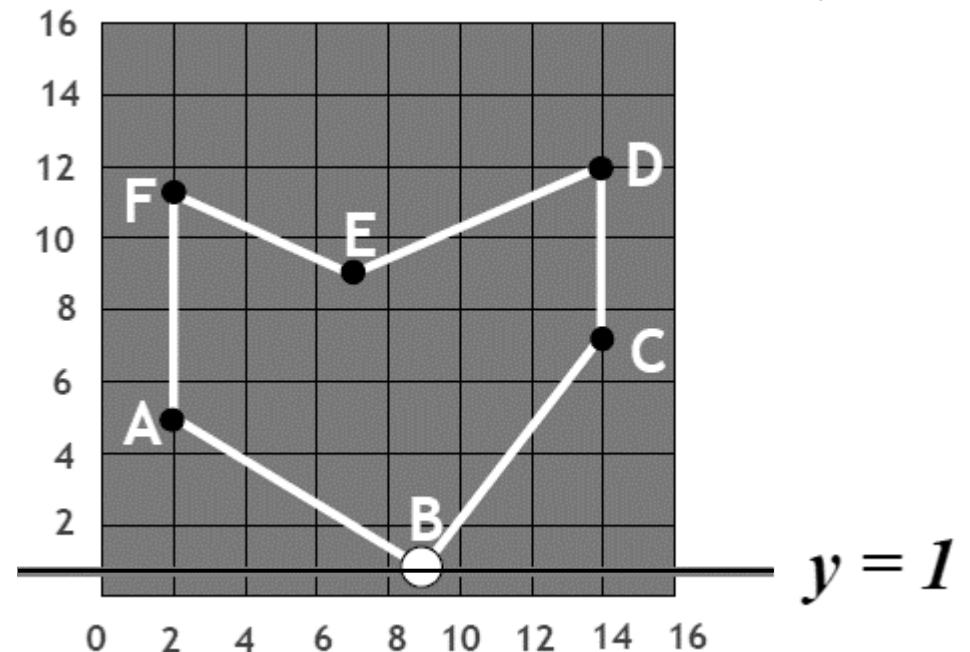


Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Wypełniamy piksele w linii y , wykorzystując pary x z tablicy aktywnej (AT) i zasadę parzystości.

- Usuwamy z tablicy aktywnej (AT) te krawędzie, dla których $y = y_{\max}$ (ew. brak takich krawędzi)
- Zwiększamy y o 1 ($y = 2$, przejście linię wyżej).



Grafika i wizualizacja

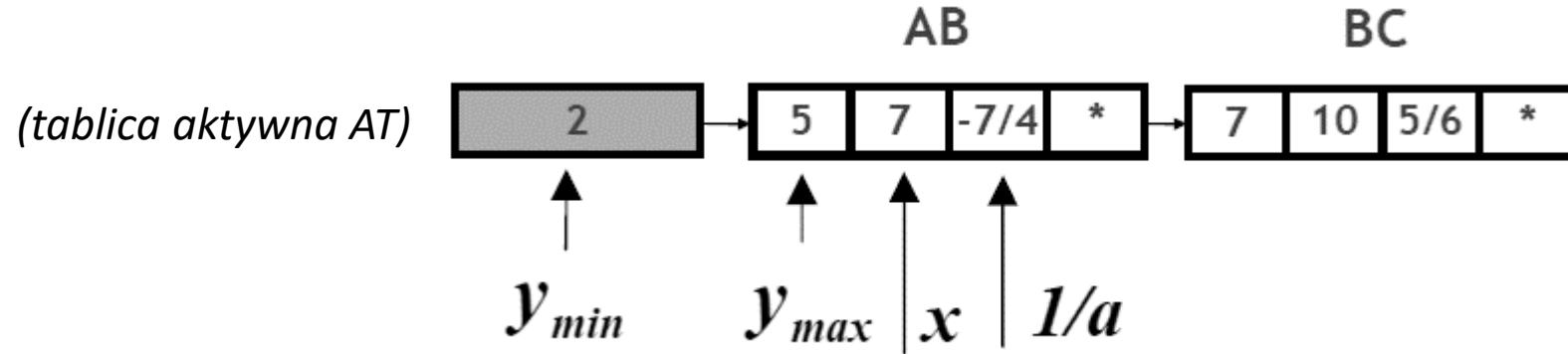
- Algorytm przeglądania linii (scan-line)

Dla każdej pary krawędzi, która nie jest pionowa wyliczamy i wstawiamy do tablicy aktywnej (AT) nowe wartości x.

Sposób obliczania nowych wartości x:

a) $x(y) = x(y - 1) + 1/a$ (algorytm DDA)

b) algorytm Bresenhama

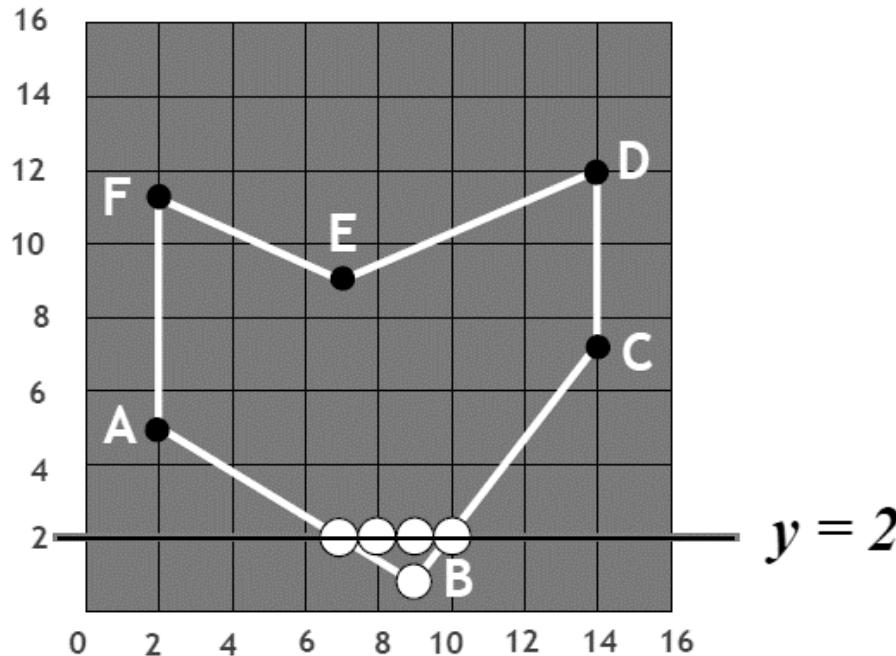


Koniec pierwszego przebiegu dla kroku 3

Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

W kolejnym przebiegu kroku 3 uzyskamy rysunek :



Postępujemy analogicznie aż do osiągnięcia $y = 5$.

Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Dla $y = 5$ rysunek wygląda następująco:

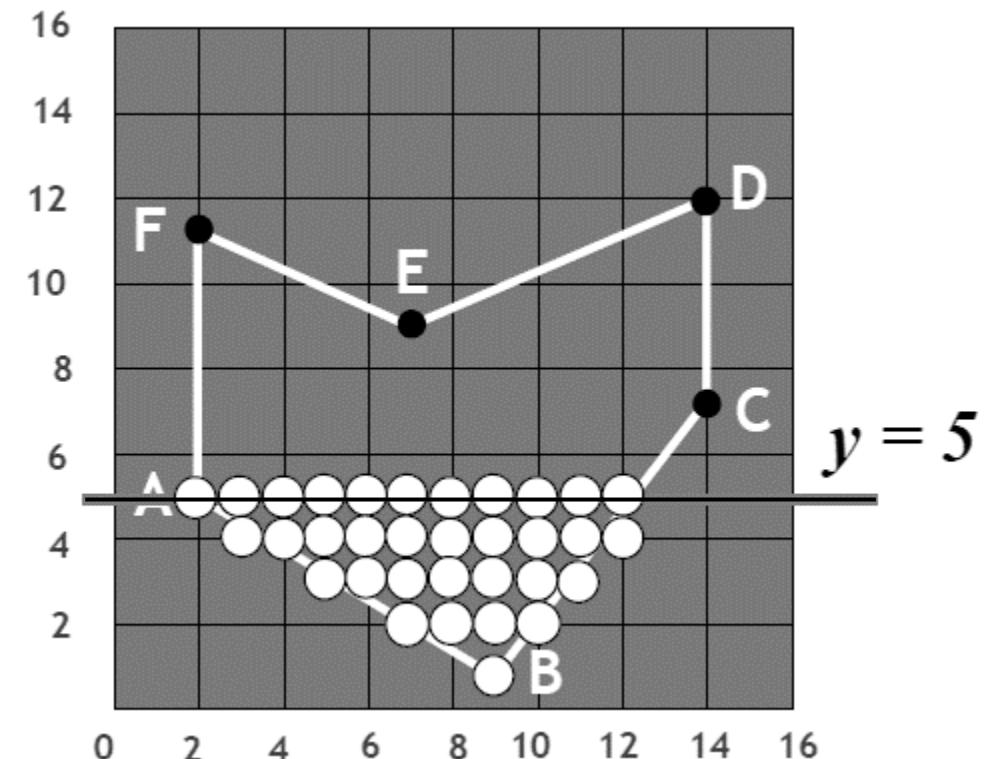
Dla krawędzi AB zachodzi $y = y_{\max}$ (koniec krawędzi).

Usuwamy AB z tablicy aktywnej (AT).

Przenosimy do tablicy aktywnej
z tablicy globalnej (ET) krawędź FA,
dla której $y_{\min} = y$.

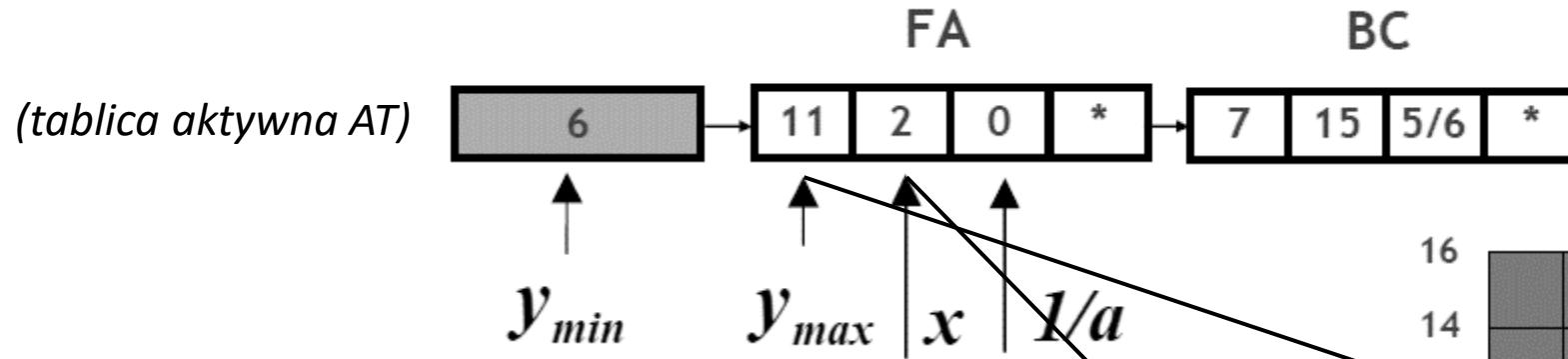
Porządkujemy tablicę (AT) ze
względu na x.

Tablica aktywna przybiera postać:



Grafika i wizualizacja

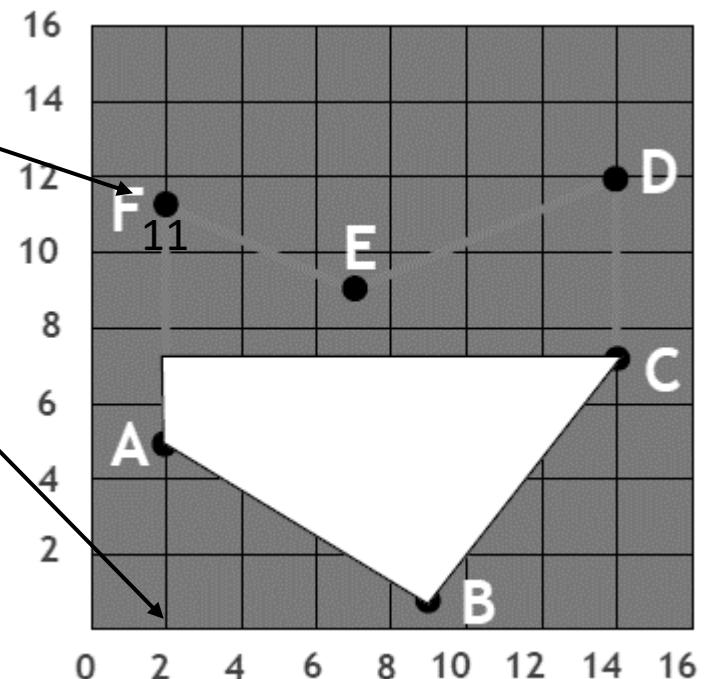
- Algorytm przeglądania linii (scan-line)



Kontynuujemy wypełnianie aż osiągniemy koniec krawędzi występującej w tablicy aktywnej (AT), lub początek krawędzi z tablicy globalnej (ET).

W rozważanym przypadku dla $y = 7$, osiągamy koniec krawędzi BC.

Uzyskany efekt wypełnienia pokazuje rysunek.

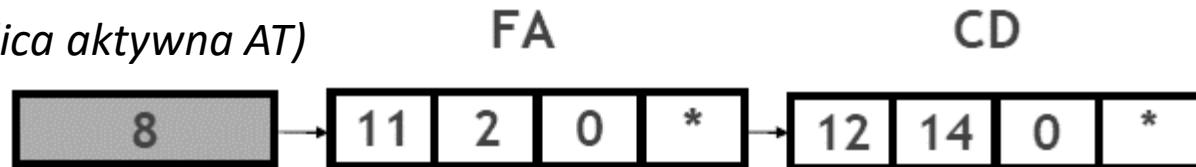


Grafika i wizualizacja

- txt

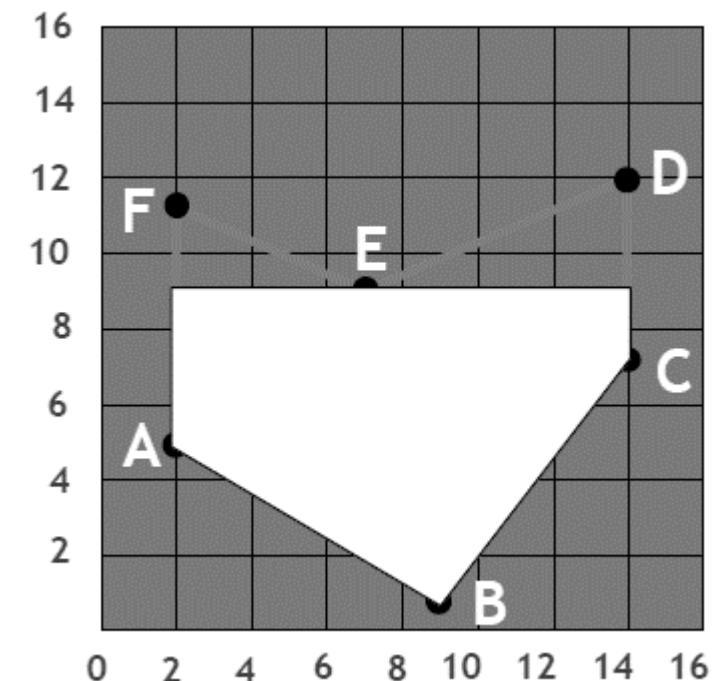
Po usunięciu z tablicy aktywnej (AT) krawędzi BC, zwiększeniu y i przeniesieniu z tablicy globalnej (ET) krawędzi CD tablica aktywna przyjmuje postać:

(tablica aktywna AT)



Wypełniając dalej, osiągamy wartość $y = 9$.

Rysunek wygląda teraz tak:

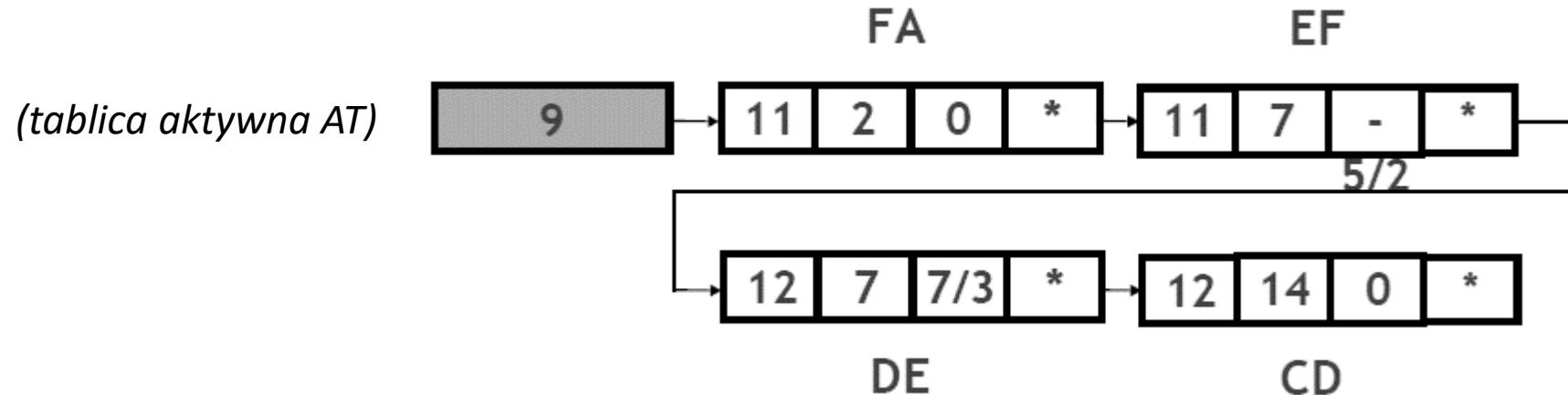


Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)

Stwierdzamy, że dla $y = 9$ para krawędzi EF i DE spełnia warunek $y_{\min} = y$

W związku z tym przenosimy krawędzie EF i DE do tablicy aktywnej (AT), która po uporządkowaniu elementów ze względu na x wygląda tak:

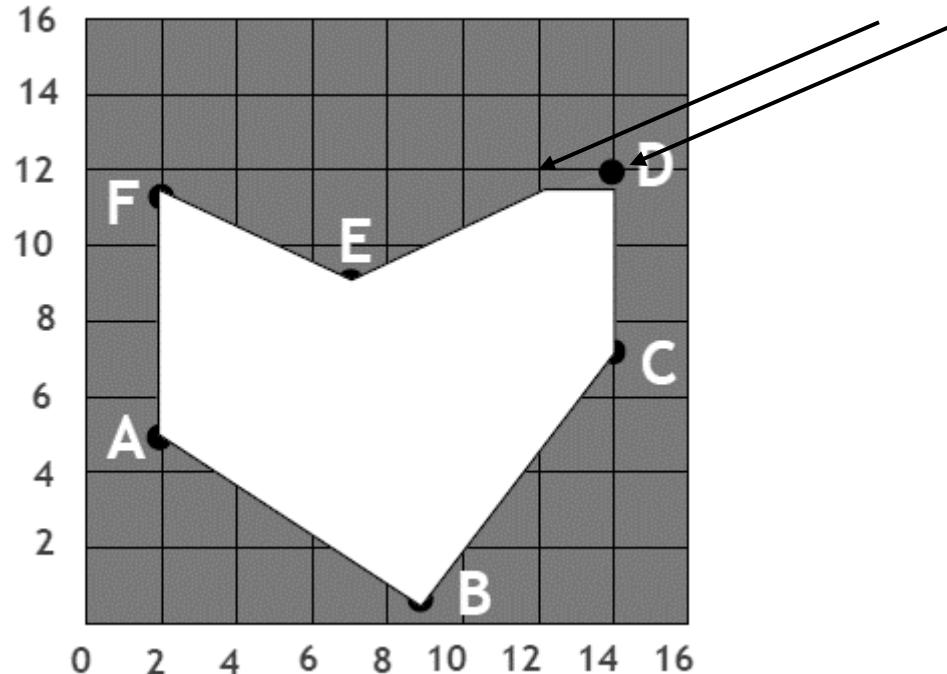


Tablica globalna (ET) staje się pusta.

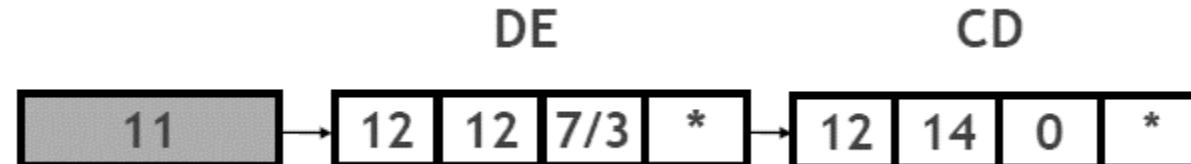
Kontynuujemy wypełnianie między parami krawędzi z tablicy (AT), aż do osiągnięcia $y = 11$. Uzyskujemy efekt jak na kolejnym rysunku:

Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line)



Usuwamy z tablicy (AT) parę krawędzi FA, EF.



Grafika i wizualizacja

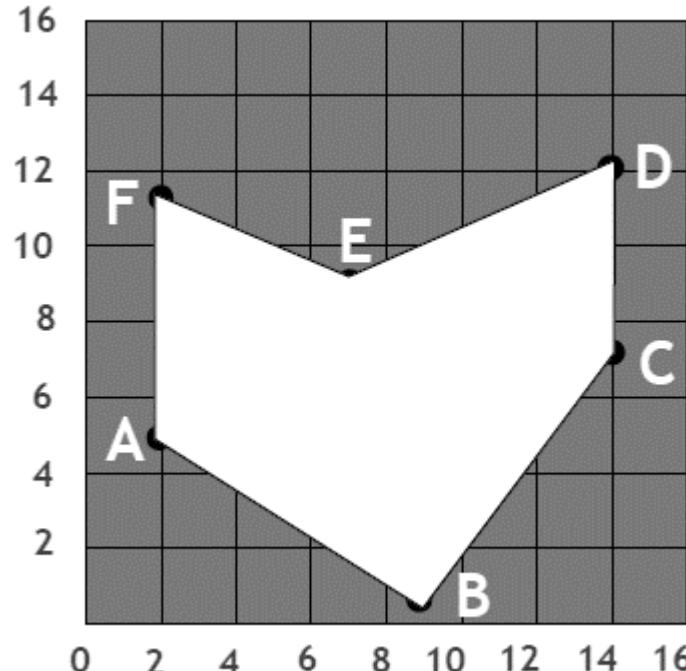
- Algorytm przeglądania linii (scan-line)

Wypełniamy następną linię i osiągamy końce krawędzi DE i CD.

Usuwamy krawędzie z tablicy AT.

Tablica (AT) jest już pusta, czyli kończymy wypełnianie.

Ostateczny efekt pokazuje rysunek:



Grafika i wizualizacja

- Algorytm przeglądania linii (scan-line), problem

Problem krawędzi poziomych:

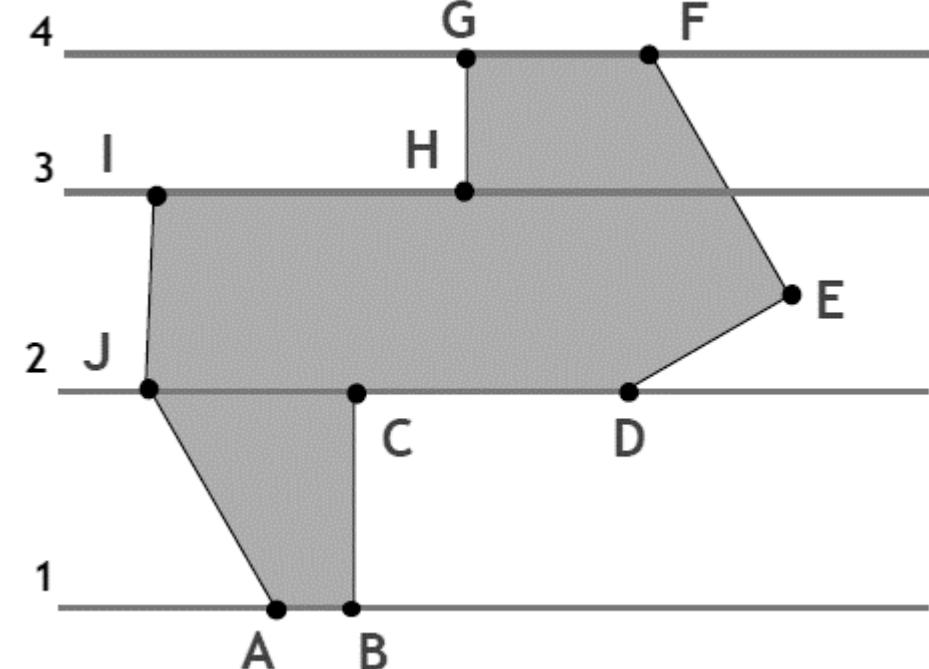
Na początku, przyjęto założenie, że wypełniany wielobok nie ma krawędzi poziomych. Jak pozbyć się tego założenia?

Rozwiązanie:

Pomijanie krawędzi poziomych w tablicy (AT).

Przykłady:

- 1 - w tablicy (AT) znajdzie się AJ i BC,
- 2 - w tablicy (AT) znajdzie się IJ i DE,
- 3 - w tablicy (AT) znajdzie się IJ i EF,
- 4 - w tablicy (AT) znajdzie się GH i FE.



Grafika i wizualizacja

Wypełnianie kształtu zadanego mapą pikseli

Grafika i wizualizacja

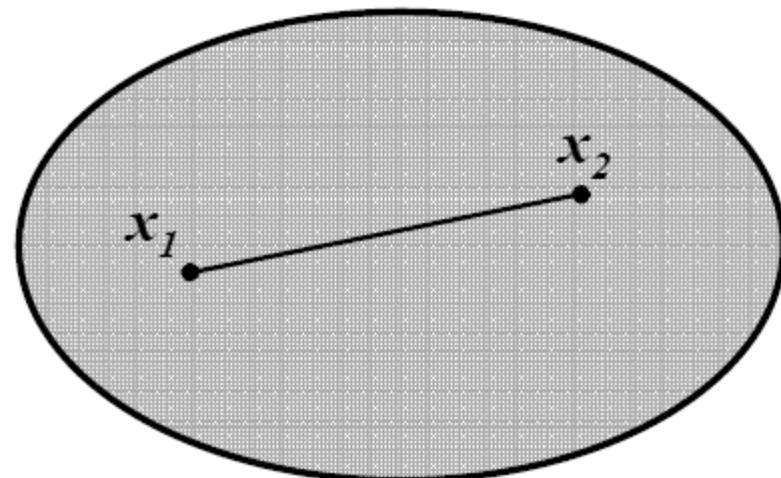
- Kształt wypukły, założenie

Założenie:

Dany jest kontur w postaci zbioru różnych od tła pikseli i punkt leżący wewnątrz konturu.

Kontur jest wypukły:

Dla dowolnej pary punktów x_1, x_2 leżących wewnątrz konturu odcinek, którego końcami są te punkty, leży w całości wewnątrz konturu.

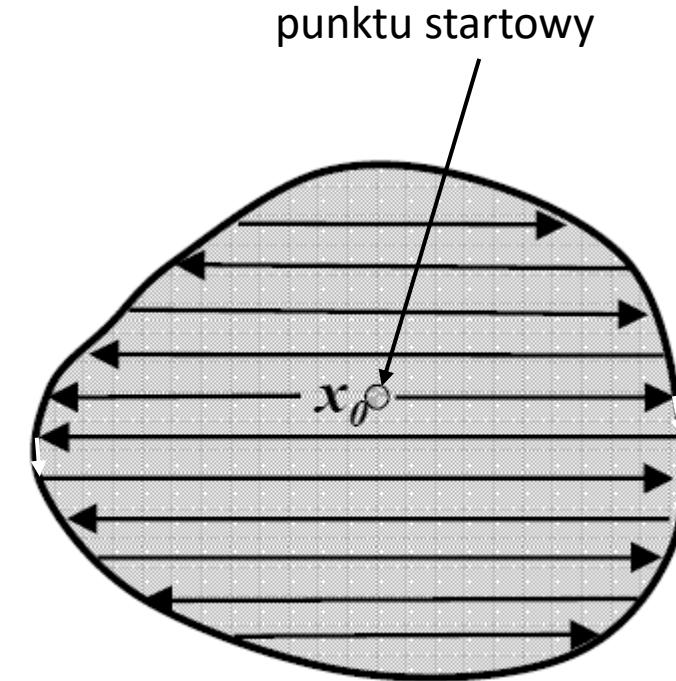


Grafika i wizualizacja

- Kształt wypukły, algorytm prosty

Prosty algorytm wypełniania konturu wypukłego:

- wewnątrz kształtu wypukłego obierany jest punkt startowy x_0 ,
- wypełniane są linie poczynając od punktu startowego, aż do prawej granicy konturu,
- znajdowany jest „niższy” punkt konturu i wypełniane są linie do lewej granicy konturu,
- proces powtarzany jest tak długo, dopóki można znaleźć punkt „niższy”,
- w przeciwnym razie algorytm wraca do punktu startowego i analogiczny proces realizowany jest poruszając się „w górę”,
- proces powtarzany jest tak długo, dopóki można znaleźć punkt „wyższy”, jeśli nie – koniec algorytmu.



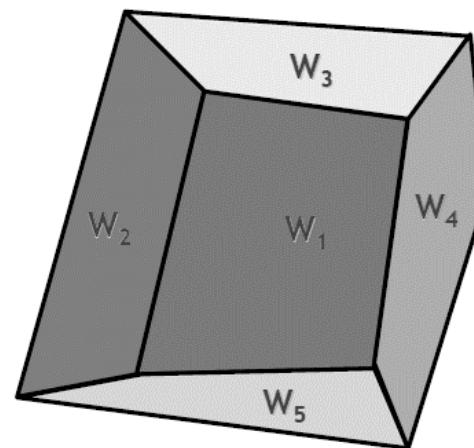
Grafika i wizualizacja

Problemy wypełniania

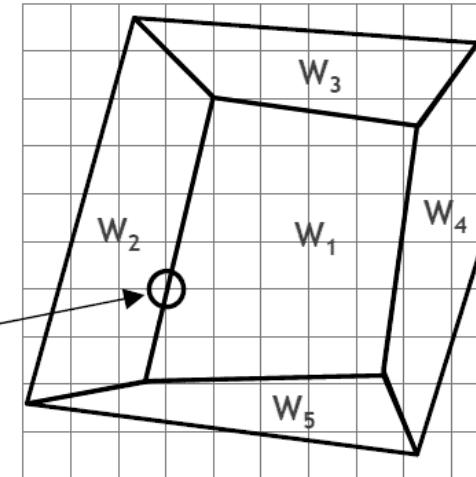
Grafika i wizualizacja

- Problem brzegu wieloboku

Trzeba narysować kilka wypełnionych wieloboków o wspólnych krawędziach. Każdy z wieloboków jest wypełniony innym kolorem.



piksel



Jak rysować obrazy krawędzi wieloboków ?

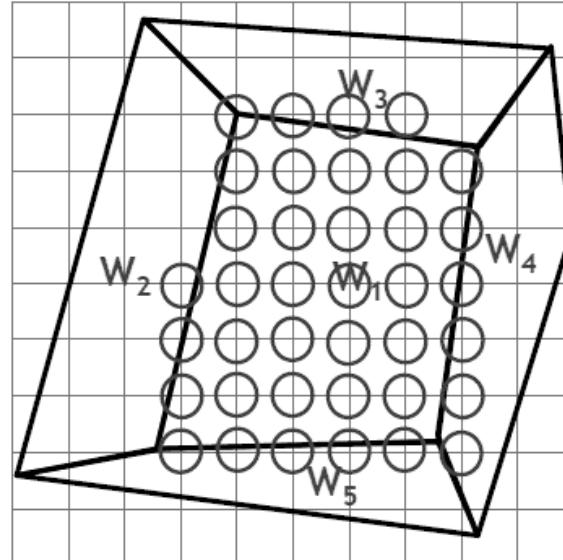
Grafika i wizualizacja

- txt

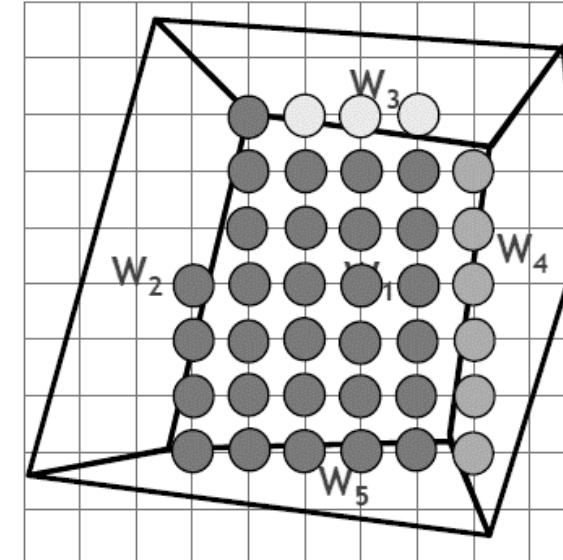
Często stosowane rozwiązanie:

- wypełniać piksele leżące wewnątrz wieloboku, ale nie na brzegu,
- wypełniać piksele należące do lewej krawędzi,
- wypełniać piksele należące do dolnej krawędzi.

Piksele kandydackie



Piksele wypełnione



Grafika i wizualizacja

- Problem wieloboków „wąskich”

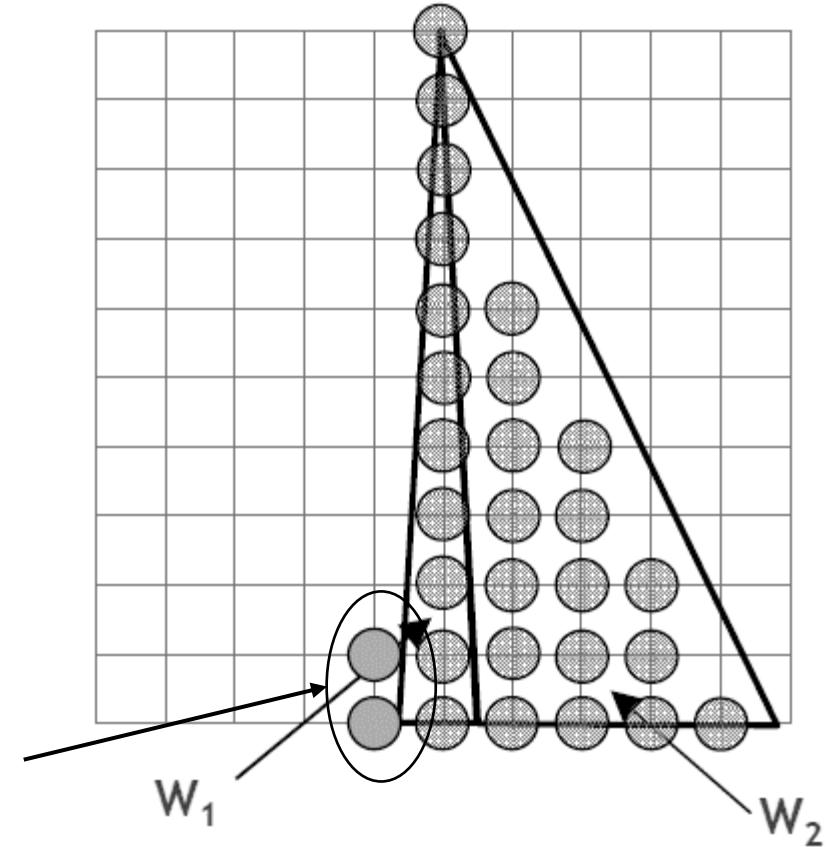
Wykorzystano poprzednio opisaną konwencję rysowania.

W wyniku jej zastosowania obraz wieloboku W_1 składa się tylko z dwóch punktów.

Niestety brak zadawalającego rozwiązania przy tym sposobie rysowania.

Należy zastosować wypełnianie wielotonowe.

Dwa punkty tworzące W_1 po wypełnieniu.



Grafika i wizualizacja

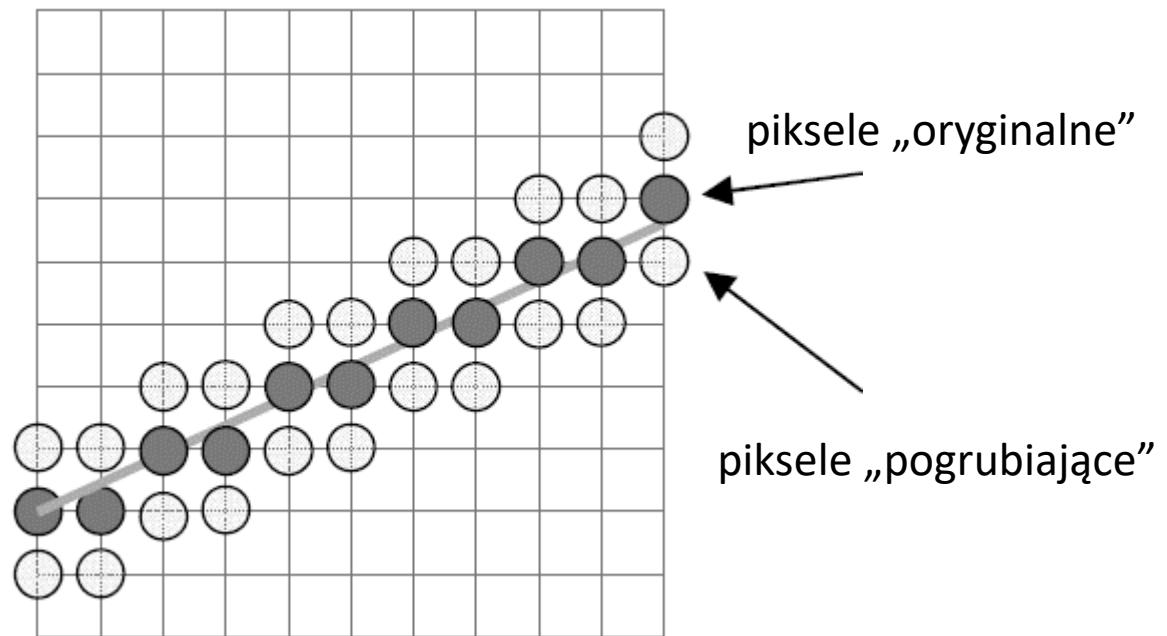
Rysowanie pogrubionych elementów pierwotnych

Grafika i wizualizacja

- Metoda powielania pikseli

Dla każdego piksela rysunku podstawowego, rysowane są dodatkowe piksele „pogrubiające”:

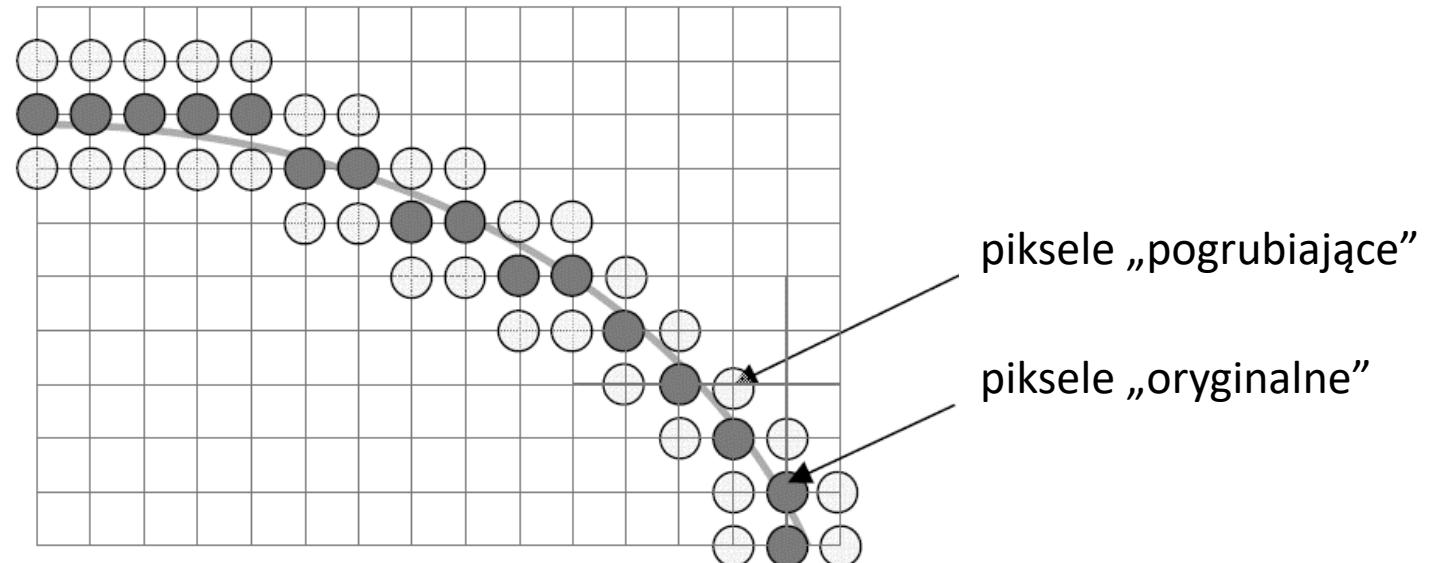
- gdy $|a| < 1$, w kolumnie,
- w przypadku przeciwnym, w wierszu.



Grafika i wizualizacja

- Metoda powielania pikseli

Zastosowanie powyższej metody dla krzywej:



Zaleta - prosty algorytm

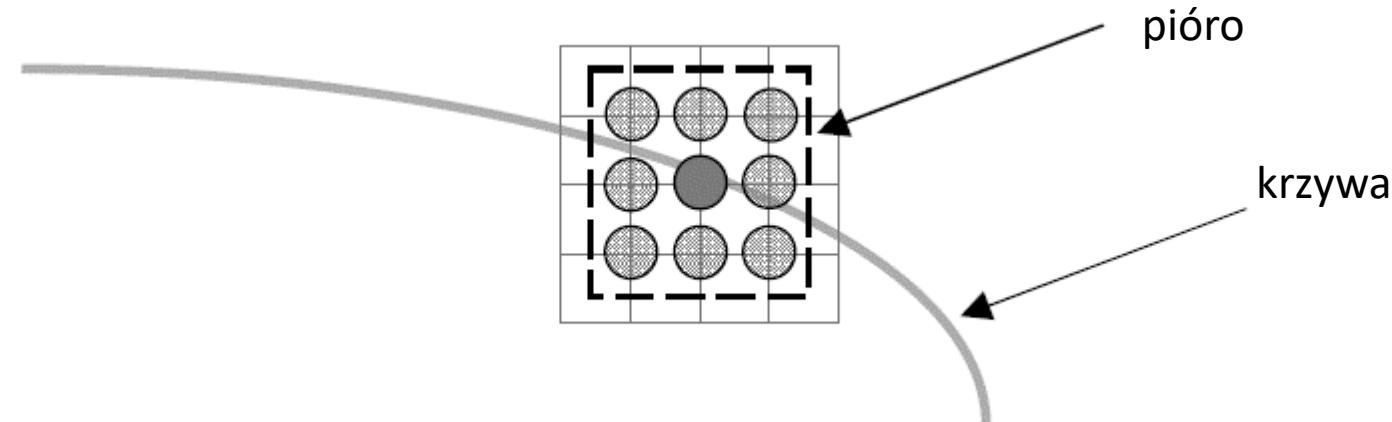
Wada - grubość rysunku dla różnych fragmentów krzywej jest różna

Grafika i wizualizacja

- Metoda prostokątnego pióra

Zasady:

- „ślad” pióra porusza się wzdłuż krzywej,
- piksel rysunku podstawowego znajduje się w centrum śladu.



Zaleta - lepsze niż w poprzedniej metodzie odwzorowanie kształtu

Wada - występuje powtórne wypełnianie pikseli

Grafika i wizualizacja

Kwantyzacja obrazu

Grafika i wizualizacja

Kwantyzacja obrazu

Przydatna definicja:

Kwantyzacja - nazwa grupy przekształceń sygnałów (np. dźwięku, obrazu, wartości odczytanej z czujnika), która zmniejsza precyzję sygnałów, aby mogły one zostać przetworzone przez jakieś urządzenie. Przykładem kwantyzacji jest konwersja analogowo-cyfrowa, która sygnał analogowy (liczbę rzeczywistą) przekształca w ograniczony zbiór liczb całkowitych, dzięki czemu możliwa jest dalsze przetwarzanie tego sygnału (obrazu, dźwięku) przy pomocy komputera.

Grafika i wizualizacja

Kwantyzacja obrazu

Przydatna definicja:

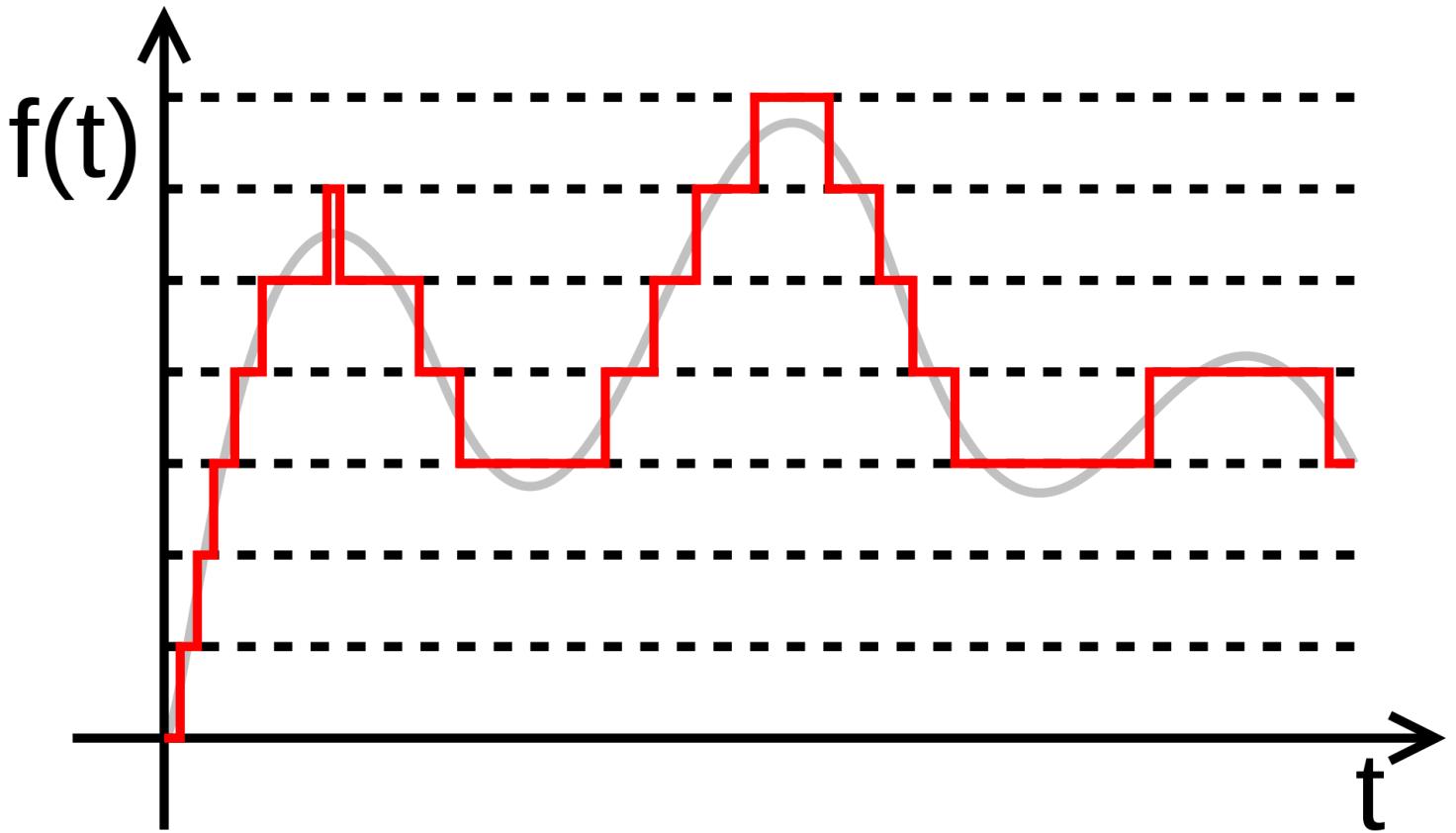
Rozdzielcość bitowa – liczbę bitów na jakich zapisana jest jedna próbka danych. Określona liczba bitów determinuje dokładność przybliżenia zapisywanej próbki (im więcej bitów tym dokładniej).

Próbki wartości (dla każdej ze składowych) są często zapisywane jako 8 bitowa liczba całkowita bez znaku, czyli zawierająca wartości pikseli o wartościach całkowitych z zakresu $<0,255>$. Oczywiście stosowane są również podejścia 16, 32 lub 64 –bitowe lub float o wartościach z przedziału $<0,1>$. Zapis próbek w formie zmiennoprzecinkowej ma zastosowanie przetwarzaniu obrazów.

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzacja
sygnału
analogowego



Grafika i wizualizacja

Kwantyzacja obrazu

Rodzaje kwantyzacji:

- skalarna,
- wektorowa.

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzacja skalarna:

Kwantyzacja skalarna to taka, w której kwantowane są niezależnie pojedyncze wartości.

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzacja skalarna może być:

- równomierna (ang. uniform),
- nierównomierna (ang. nonuniform).

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzacja skalarna równomierna a nierównomierna:

Przy kwantyzacji równomiernej różnica pomiędzy sąsiednimi poziomami decyzyjnymi jest jednakowa, przy kwantyzacji nierównomiernej tak nie jest.

Kwantyzacja nierównomierna jest stosowana tam gdzie rozkład prawdopodobieństwa wartości analogowych nie jest jednostajny. Gdy w jakimś przedziale wartości pojawiają się częściej niż w innych, jest mu przypisywane odpowiednio więcej poziomów kwantyzacji (patrz format mp3), dzięki czemu błąd kwantyzacji maleje.

Istnieje możliwość wyznaczenia poziomów decyzyjnych i reprezentacji, tak aby błąd średniokwadratowy był dla danego rozkładu minimalny i w praktyce stosuje się do tego iteracyjny algorytm Maxa-Lloyda.

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzacja wektorowa:

W przypadku kwantyzacji wektorowej odpowiednikami przedziałów kwantyzacji są bryły w przestrzeni wielowymiarowej zwane komórkami, przy czym liczba wymiarów zależy od liczby wymiarów kwantowanych wektorów (liczba jednocześnie kwantowanych wartości).

Odpowiednikami poziomów decyzyjnych są ściany pomiędzy komórkami (płaszczyzny lub hiperplaszczyzny), a odpowiednikami poziomów reprezentacji są centroidy komórek.

Grafika i wizualizacja

Kwantyzacja obrazu

Kwantyzację można również stosować do sygnału, który już był wcześniej skwantowany (np. zmiana rozdzielczości bitowej zapisanego już wcześniej obrazu). Taki zabieg zazwyczaj zmniejsza wymaganą liczbę bitów kosztem zmniejszenia dokładności reprezentacji sygnału i jest częstym elementem technik kompresji danych.

Grafika i wizualizacja

Kwantyzacja obrazu

Szum kwantyzacji:

Szum kwantyzacji powstaje m.in. w przetwornikach analogowo-cyfrowych w trakcie dyskretyzacji sygnału audio lub obrazu. Podczas konwersji ciągłego sygnału analogowego na postać cyfrową, każda wartość jest przybliżana do najbliższej wartości dyskretnej.

Grafika i wizualizacja

Kwantyzacja obrazu

Szum kwantyzacji:

Powstający w ten sposób błąd określany jest mianem szumu kwantyzacji. Szum ten można zmniejszyć poprzez zwiększenie liczby bitów opisujących każdą próbkę. Zwiększenie liczby bitów o jeden powoduje dwukrotne zwiększenie liczby poziomów kwantyzacji i w rezultacie zmniejszenie szumu kwantyzacji o:

$$20 \log(2) \approx 6,02 \text{ dB}$$

Grafika i wizualizacja

Kwantyzacja obrazu

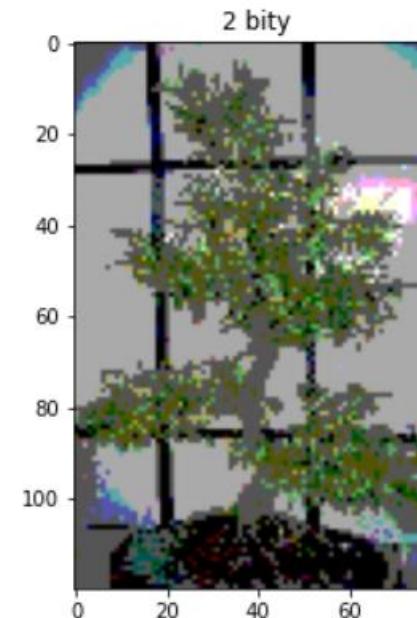
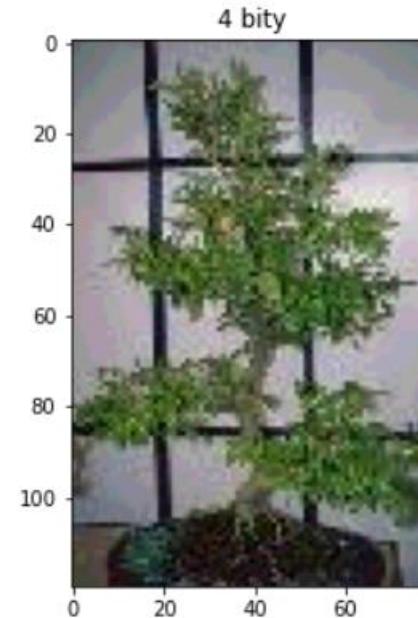
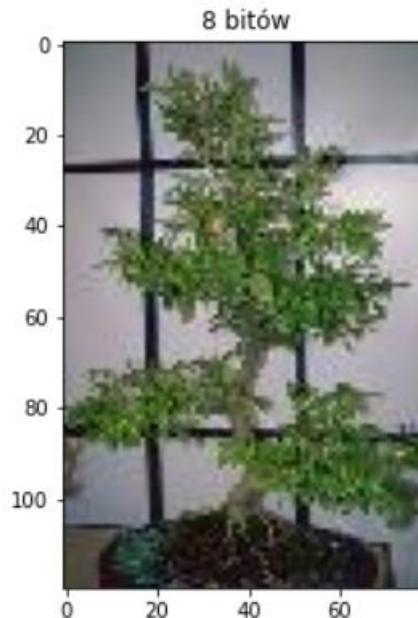
Przykład:

W sporze pomiędzy firmami Philips (która to optowała za 14-bitowym kwantyzerem dla standardu Compact Disc) a Sony (która preferowała kwantyzer 16-bitowy), wygrała ta druga, dzięki czemu obecny standard CD-Audio oferuje dynamikę (parametr SNR) na poziomie 96 dB. Wynika to z faktu, że w standardzie zastosowano kwantyzację równomierną (PCM), dla której każdy dodatkowy bit kwantyzera (podwojenie liczby poziomów kwantyzacji) podnosi wartość parametru SNR dla spróbkowanego sygnału o wartość 6 dB ($6 \text{ dB/bit} * 16 \text{ bit} = 96 \text{ dB}$). Gdyby zwyciężyła propozycja firmy Philips, dzisiejsze płyty standardu Audio-CD oferowałyby możliwość nagrania tematu muzycznego o ok. 14% dłuższym czasie trwania, jednakże kosztem niższej dynamiki (84 dB w miejsce 96 dB) sygnału.

Grafika i wizualizacja

Kwantyzacja obrazu

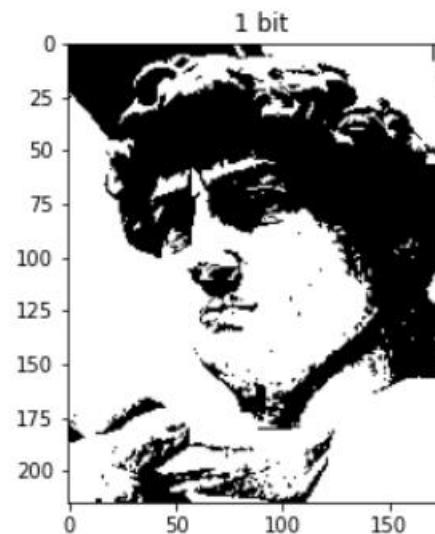
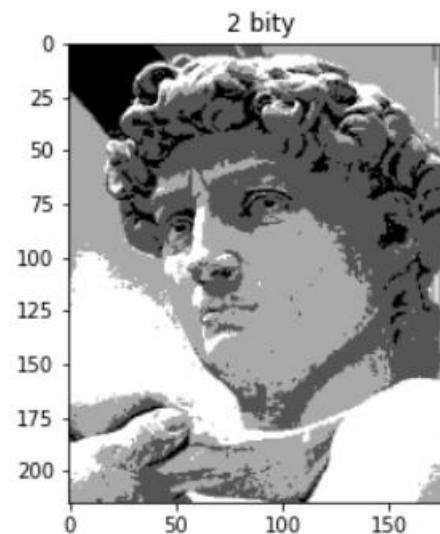
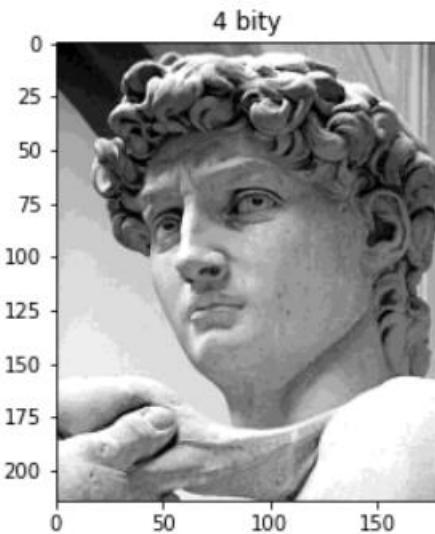
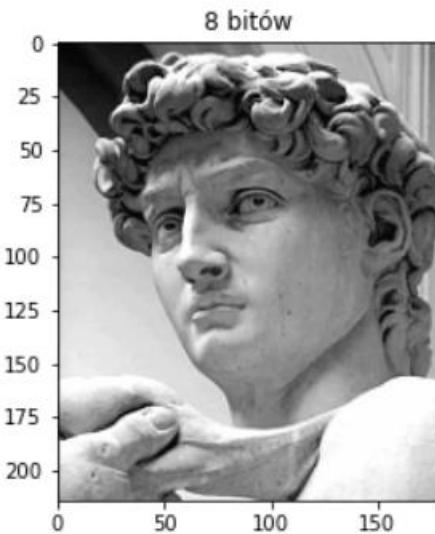
Kwantyzacja obrazu w dół (zmnieszenie rozdzielczości bitowej) prowadzi zawsze do redukcji zapisywanej informacji co prowadzi to do pogorszenia jakości obrazu:



Grafika i wizualizacja

Kwantyzacja obrazu

Obraz w odcieniach szarości:



Grafika i wizualizacja

Kwantyzacja obrazu

W celu zniwelowania błędów kwantyzacji stosuje się dithering.

Grafika i wizualizacja

Kwantyzacja obrazu

W trybie kolorowym dithering realizowany jest jako próba uzyskania wrażenia danej barwy poprzez kompozycję kilku kolorów z dostępnej palety, gdy kolor oryginalny nie może zostać bezpośrednio wyświetlony.

W trybie czarno-białym realizowana jest podobna symulacja jak w trybie kolorowym, lecz z zastosowaniem do odcieni szarości.

Grafika i wizualizacja

Kwantyzacja obrazu

Istnieje przynajmniej kilka podejść do realizacji ditheringu:

- dithering przez zastosowanie metody losowej,
- dithering zorganizowany,
- dithering metodą Floyda–Steinberga.

Grafika i wizualizacja

Kwantyzacja obrazu

Metoda losowa:

Metoda nadająca się głównie do obrazów binarnych. Jest mało dokładna, ale posiada niską złożoność czasową i obliczeniową.

Polega na generowaniu losowych wartości dla każdego ze sprawdzanych pikseli (można od razu wygenerować całą ich macierz o rozmiarze naszego obrazu) i sprawdzaniu czy wartość piksela w skali odcieni szarości jest większa niż losowej wartości i w zależności od wyniku oznaczania naszego nowego piksela jako czarny lub biały.

Grafika i wizualizacja

Kwantyzacja obrazu

Dithering zorganizowany:

Dithering zorganizowany wprowadza uporządkowane przesunięcia kolorów wykorzystując do tego mapami progowania (Threshold map), zwane również macierze indeksowanymi lub macierzami Bayera.

Macierze te można w dowolny sposób obracać i odbijać lustrzenie bez zmniejszania skuteczności algorytmu. Obok zaprezentowano mapy dla dwóch najmniejszych rozmiarów.

$$M_1 = \frac{1}{4} \times \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

$$M_2 = \frac{1}{16} \times \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

Grafika i wizualizacja

Kwantyzacja obrazu

Mapa progowania służy do zmiany źródłowych właściwości koloru. Dużo łatwiej pracować operując na wartościach przesunięcia. Dlatego mapa zostaje przekształcona do przestrzeni, które ułatwią dalsze obliczenia. Wykorzystywany jest do tego poniższy wzór:

$$M_{1-pre} = \frac{1}{4} \times (\begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 1) - 0.5 = \begin{bmatrix} -0.25 & 0.25 \\ 0.5 & 0 \end{bmatrix}$$

Grafika i wizualizacja

Kwantyzacja obrazu

*colorFit - funkcji kwantyzująca, przypisująca dowolnej barwie obrazu źródłowego - barwę z palety docelowej (reorganizująca paletę barw)

Dalej wykorzystywana jest funkcja kwantyfikująca, która dopasowuje wartości nowych pikseli do palety kolorów (nazywaną dalej colorFit). W przypadku obrazu w skali odcienni szarości funkcja ta ma proste zadanie - wystarczy, że dopasuje ona wartość do odpowiedniego przedziału lub najbliższej wartości z palety.

Gdy posiadamy już taką funkcję sama funkcja wyliczania wartości dla nowych pikseli wygląda tak:

$$C_n = \text{colorFit}(c + r \times (M_{\text{pre}}(a \mod (2 * n)), b \mod (2 * n))))$$

Grafika i wizualizacja

Kwantyzacja obrazu

Współczynnik r określa skalowanie koloru pomiędzy oryginalnymi wartościami pikseli a nową paletą.

Jeżeli obie palety są w tym samym zakresie pikseli parametr będzie wynosił 1.

W przeciwnym wypadku będzie wymagał on wyliczenia na podstawie ich wzajemnego stosunku zakresu wartości.

Grafika i wizualizacja

Kwantyzacja obrazu

Znając najmniejszą macierz M_1 można generować większe macierze z założeniem, że każdy kolejny poziom będzie dwukrotnie większy niż poprzedni.

Służy do tego poniższy wzór:

$$M_{2n} = \frac{1}{(2N)^2} \times \begin{bmatrix} 4 \times M_n & 4 \times M_n + 2 \\ 4 \times M_n + 3 & 4 \times M_N + 1 \end{bmatrix}$$

Grafika i wizualizacja

Kwantyzacja obrazu

Dithering metodą Floyd–Steinberga:

Podstawą algorytmu Floyda-Steinberga jest rozpraszanie błędów wprowadzonej redukcji. Przy wykorzystaniu funkcji kwantyzującej (colorFit), przypisującej dowolnej barwie – barwę z palety docelowej. Metoda polega na stopniowym przekazywaniu błędu kwantyzacji do kolejnych pikseli zgodnie z wskazanymi wagami. Przykład z wagami dla otoczenia piksela został zaprezentowany poniżej.

$$\begin{bmatrix} & * & \frac{7}{16} & \dots \\ \dots & \frac{3}{16} & \frac{5}{16} & \frac{1}{16} & \dots \end{bmatrix}$$

Grafika i wizualizacja

Kwantyzacja obrazu

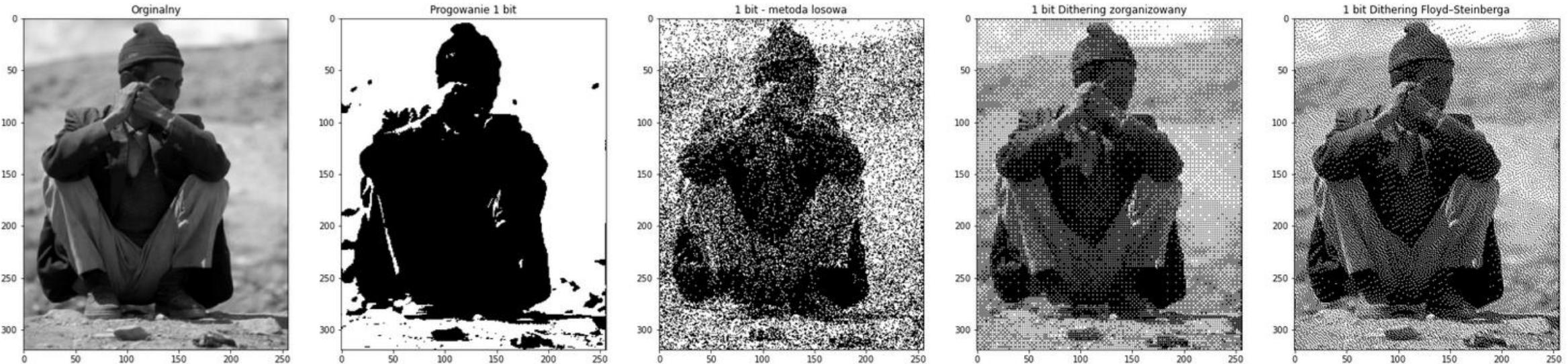
Sam algorytm wyliczania wartości piksela (x,y) wraz z procedurą rozpraszania błędu kwantyzacji został zaprezentowany poniżej (należy jednak pamiętać, że nie wszystkich miejscach mamy dostęp do wszystkich pikseli, aby przekazać błąd dalej - przy krawędziach obrazu).

```
oldpixel = img[x][y]
newpixel = colorFit(oldpixel)
img[x][y] = newpixel
quant_error = oldpixel - newpixel
img[x + 1][y    ] = img[x + 1][y    ] + quant_error × 7 / 16
img[x - 1][y + 1] = img[x - 1][y + 1] + quant_error × 3 / 16
img[x    ][y + 1] = img[x    ][y + 1] + quant_error × 5 / 16
img[x + 1][y + 1] = img[x + 1][y + 1] + quant_error × 1 / 16
```

Grafika i wizualizacja

Kwantyzacja obrazu

Przykłady:



Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Geometria w przestrzeni 2D i 3D

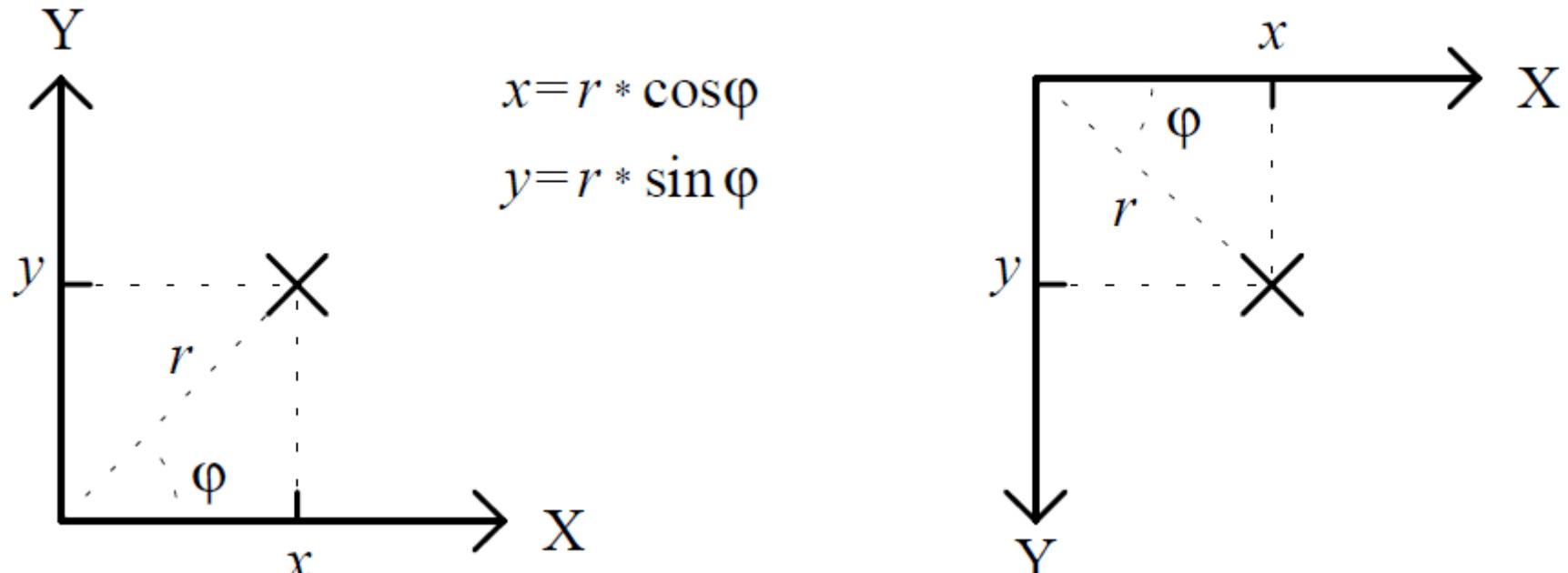
Przekształcenia obiektów

Reprezentacja i przekształcenia w przestrzeni 2D

Wprowadzenie

Reprezentacja i przekształcenia w przestrzeni 2D

Do opisu obiektów dwuwymiarowych będziemy najczęściej używali układu kartezjańskiego x,y. Alternatywą dla układu kartezjańskiego mogą być współrzędne biegunkowe r, φ.



Rys. Prawoskrętny i lewoskrętny dwuwymiarowy układ współrzędnych (pokazana zależność współrzędnych kartezjańskich i biegunkowych)

Reprezentacja i przekształcenia w przestrzeni 2D

Każdy punkt w przestrzeni (zarówno 2D jak i więcej-wymiarowej) można poddać przekształceniu (transformacji) zmieniającemu wartości jego współrzędnych.

Transformacje elementarne:

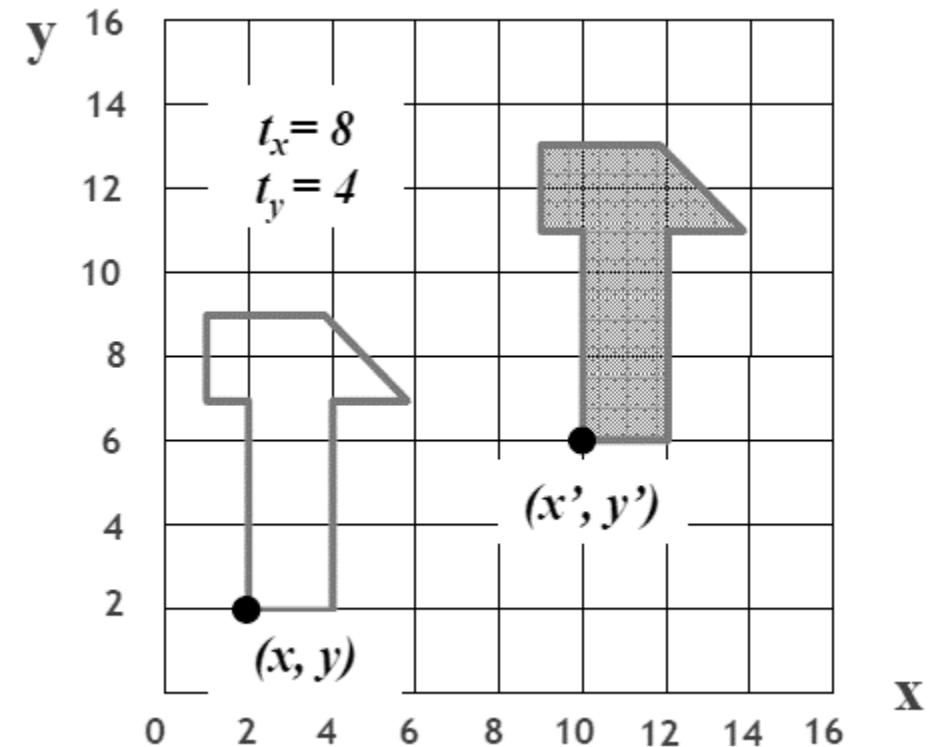
- przesunięcie,
- zmiana skali,
- obrót wokół środka układu współrzędnych.

Reprezentacja i przekształcenia w przestrzeni 2D

Przesunięcie (translacja)

$$x' = x + t_x$$

$$y' = y + t_y$$



Reprezentacja i przekształcenia w przestrzeni 2D

Podstawowe przekształcenia w przestrzeni dwuwymiarowej:

- oznaczenia: p – punkt wejściowy, p' – punkt przekształcony

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- przesunięcie (translacja o wektor b)

$$\mathbf{p}' = \mathbf{I} * \mathbf{p} + \mathbf{b}$$

gdzie: $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

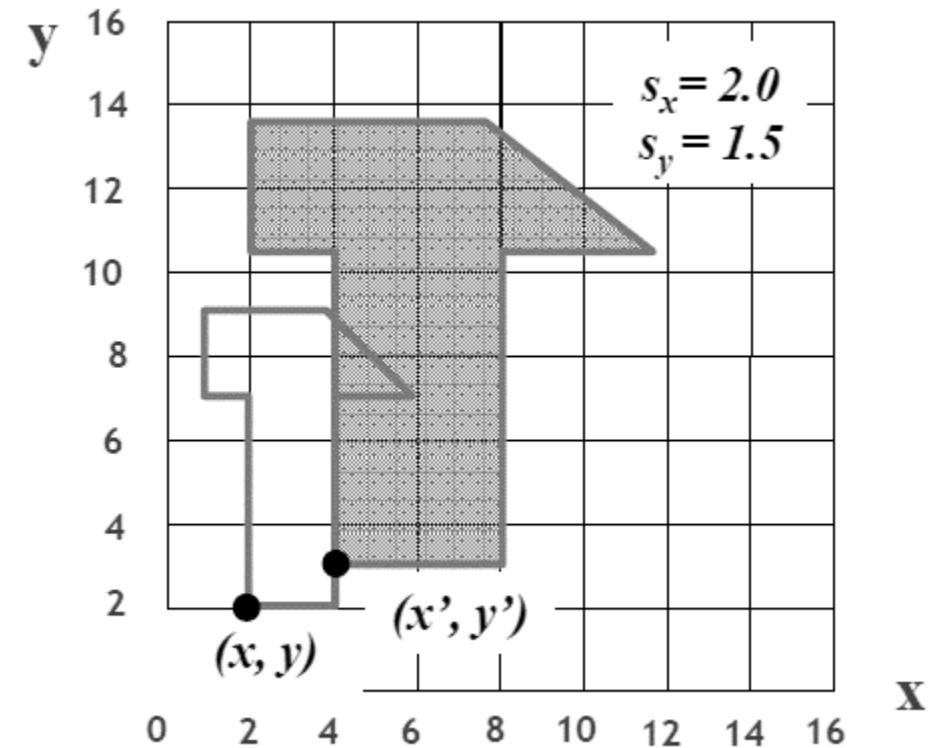
$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} =$$
$$= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 2D

Skalowanie

$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

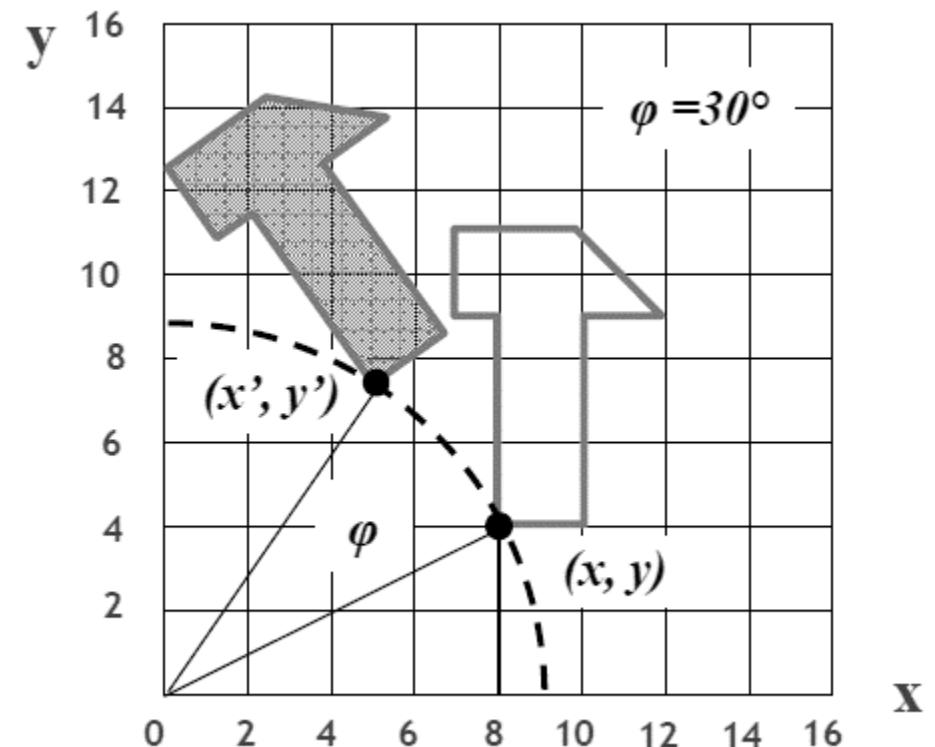


Reprezentacja i przekształcenia w przestrzeni 2D

Obrót wokół środka układu współrzędnych

$$x' = x \cos \phi - y \sin \phi$$

$$y' = y \cos \phi + x \sin \phi$$



Reprezentacja i przekształcenia w przestrzeni 2D

Obrót wokół środka układu współrzędnych

$$x' = r \cos(\alpha + \phi) = r \cos \alpha \cos \phi - r \sin \alpha \sin \phi$$

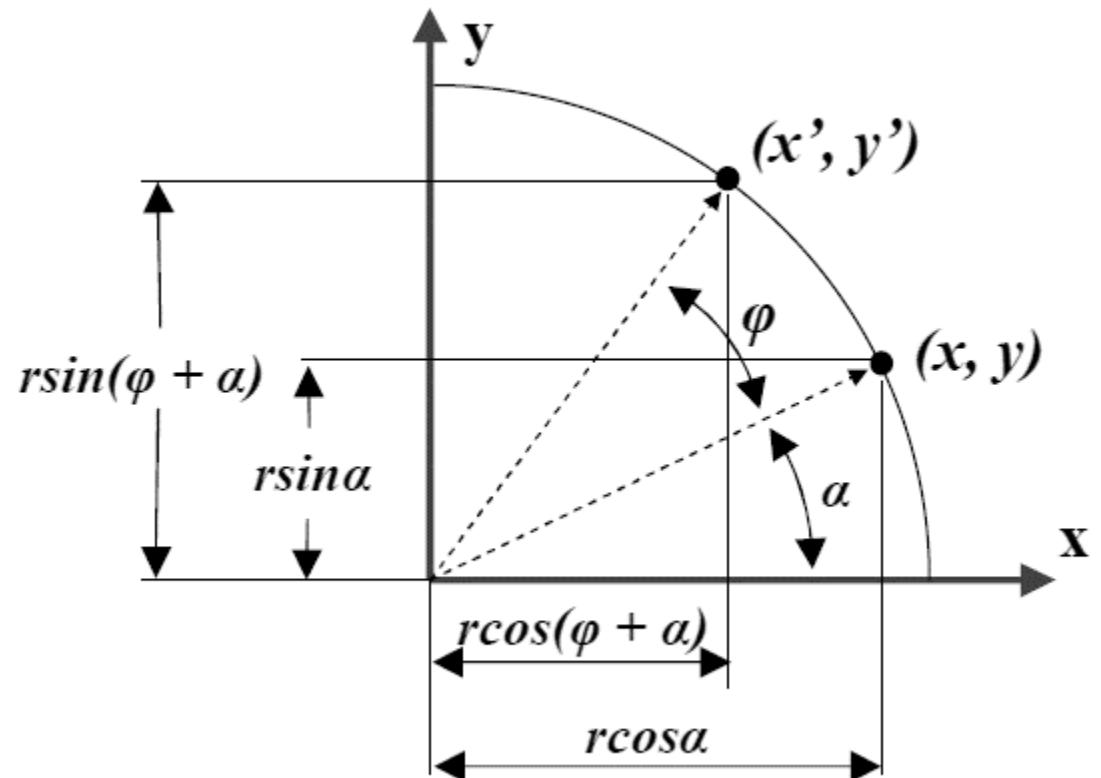
$$y' = r \sin(\alpha + \phi) = r \sin \alpha \cos \phi - r \cos \alpha \sin \phi$$

$$x = r \cos \alpha, \quad y = r \sin \alpha$$



$$x' = x \cos \phi - y \sin \phi$$

$$y' = y \cos \phi + x \sin \phi$$



Reprezentacja i przekształcenia w przestrzeni 2D

Współrzędne jednorodne – sposób reprezentacji punktów n-wymiarowej przestrzeni rzutowej za pomocą układu $n+1$ współrzędnych.

Pojęcie to opiera się na konstrukcji przestrzeni rzutowej, w której n-wymiarową przestrzeń euklidesową E^n uzupełnia się o kierunki zwane punktami w nieskończoności lub punktami niewłaściwymi.

Reprezentacja i przekształcenia w przestrzeni 3D

...

Jeśli punkt właściwy ma współrzędne kartezjańskie (x_1, x_2, \dots, x_n) , to jego współrzędne jednorodne mają postać $(x_1, x_2, \dots, x_n, 1)$.

Np. punkt na płaszczyźnie o współrzędnych kartezjańskich (x, y) ma zarazem współrzędne jednorodne postaci $(x, y, 1)$.

Podobnie punkt w przestrzeni trójwymiarowej o współrzędnych kartezjańskich (x, y, z) ma współrzędne jednorodne $(x, y, z, 1)$.

Reprezentacja i przekształcenia w przestrzeni 3D

...

Punkt właściwy o współrzędnych jednorodnych $(x_1, x_2, \dots, x_n, z)$ i $z > 0$ ma współrzędne kartezjańskie postaci $\left(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}\right)$

Jeśli natomiast $z=0$, to jest to punkt w nieskończoności i nie istnieją dla niego współrzędne kartezjańskie - żaden punkt nie może mieć współrzędnych jednorodnych $(0, 0, \dots, 0, 0)$.

Reprezentacja i przekształcenia w przestrzeni 3D

...

Dwa układy $(x_1, x_2, \dots, x_{n+1})$ i $(y_1, y_2, \dots, y_{n+1})$ są współrzędnymi jednorodnymi tego samego punktu, gdy jeden z tych układów jest wielokrotnością drugiego tj. $x_i = k \cdot y_i$ dla pewnego $k \in \mathbb{R}, k \neq 0$.

Inaczej mówiąc, każdy punkt (właściwy lub niewłaściwy) można reprezentować na nieskończoność wiele sposobów we współrzędnych jednorodnych i wszystkie te reprezentacje są do siebie proporcjonalne.

Reprezentacja i przekształcenia w przestrzeni 3D

...

Współrzędne jednorodne zostały wprowadzone do geometrii w 1827 przez Augusta Möbiusa w pracy *Der barycentrische Calcul*.

W 1946 E. Maxwell użył ich do rozwiązywania problemów związanych z rzutowaniem. Są narzędziem do stosowania metod analitycznych w przestrzeniach rzutowych.

Ze względu na kilka zalet znalazły też zastosowanie w grafice komputerowej. W roku 1965 L. Roberts zauważył, że współrzędne jednorodne znakomicie nadają się do macierzowego opisu przekształceń w przestrzeniach n-wymiarowych.

Reprezentacja i przekształcenia w przestrzeni 2D

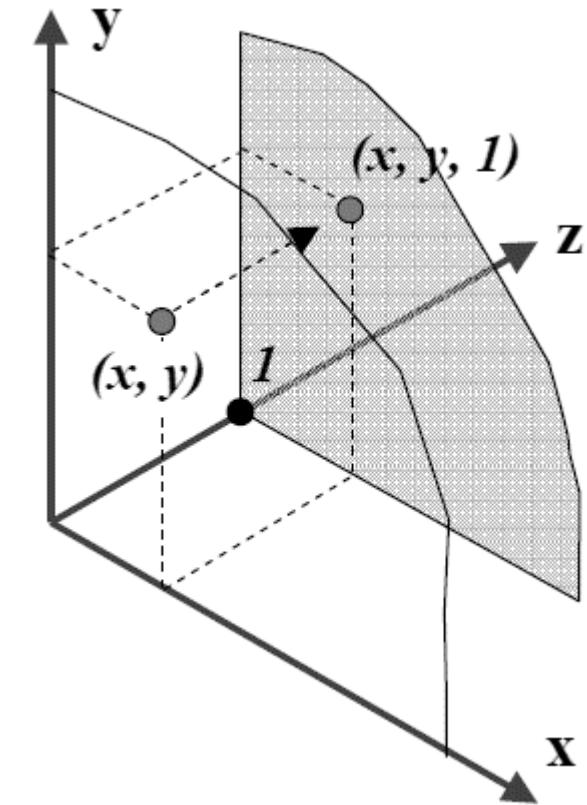
Współrzędne jednorodne w grafice komputerowej

Współrzędne przed transformacją

$$(x, y) \rightarrow (x, y, 1) \rightarrow [x \quad y \quad 1]$$

(i analogicznie) współrzędne po transformacji

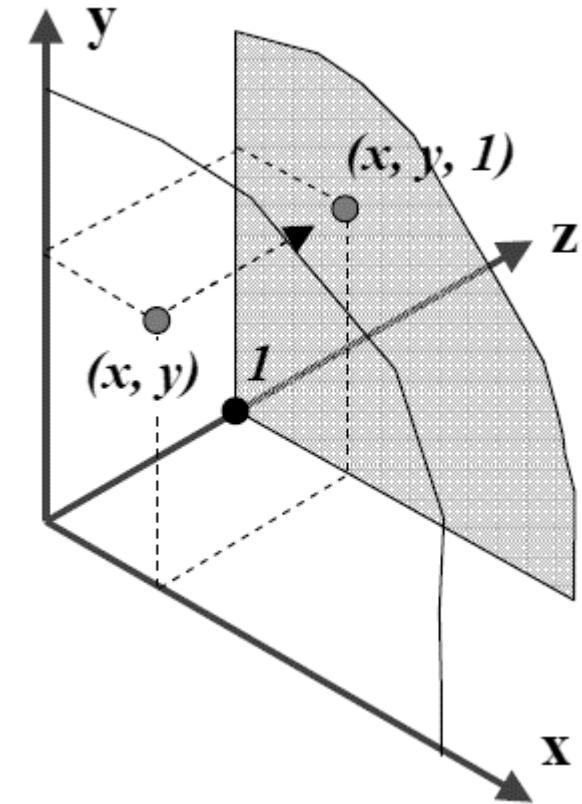
$$(x', y') \rightarrow (x', y', 1) \rightarrow [x' \quad y' \quad 1]$$



Reprezentacja i przekształcenia w przestrzeni 2D

Związek pomiędzy określonymi wyżej przekształceniami można zapisać w postaci ogólnej:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$



Reprezentacja i przekształcenia w przestrzeni 2D

Translację (przesunięcie):

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y\end{aligned}$$

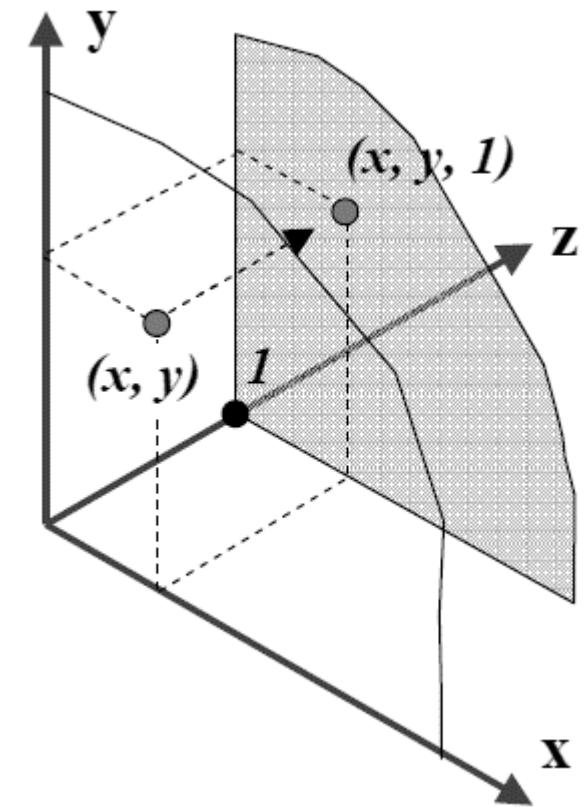
zapisane zostanie następująco:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

$$\begin{aligned}x' &= x \cdot 1 + y \cdot 0 + 1 \cdot t_x \\y' &= x \cdot 0 + y \cdot 1 + 1 \cdot t_y\end{aligned}$$

(czyli to co chcieliśmy otrzymać)

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y\end{aligned}$$



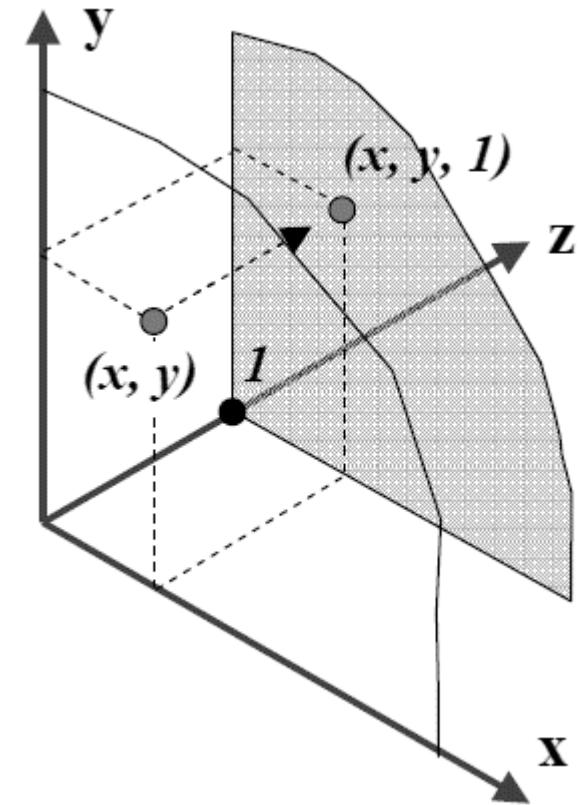
Reprezentacja i przekształcenia w przestrzeni 2D

Translację (przesunięcie):

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y\end{aligned}$$

opisuje więc macierz:

$$T(t_x, t_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$



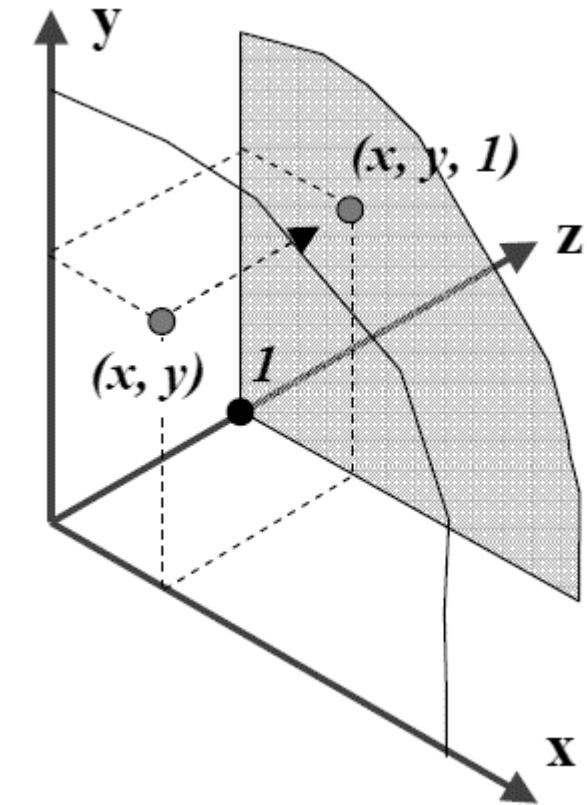
Reprezentacja i przekształcenia w przestrzeni 2D

Skalowanie:

$$\begin{aligned}x' &= x \cdot s_x \\y' &= y \cdot s_y\end{aligned}$$

zapisane zostanie następująco:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



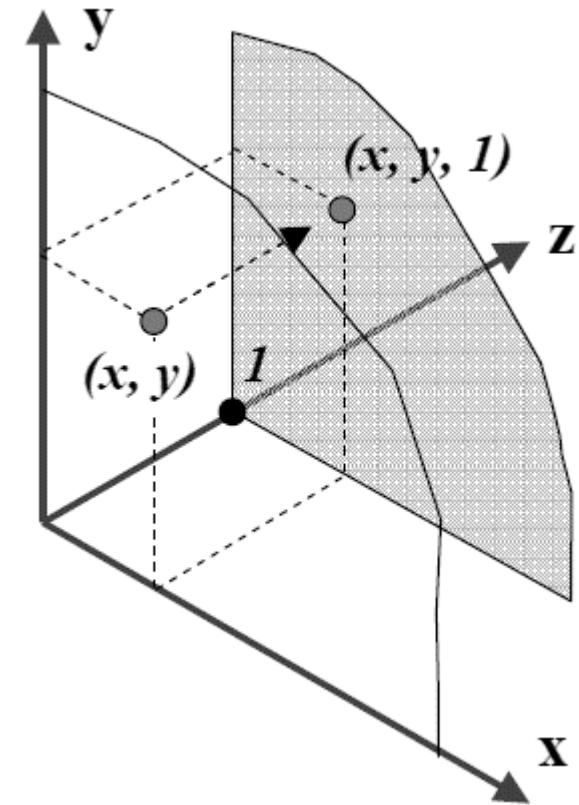
Reprezentacja i przekształcenia w przestrzeni 2D

Skalowanie:

$$\begin{aligned}x' &= x \cdot s_x \\y' &= y \cdot s_y\end{aligned}$$

opisuje więc macierz:

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



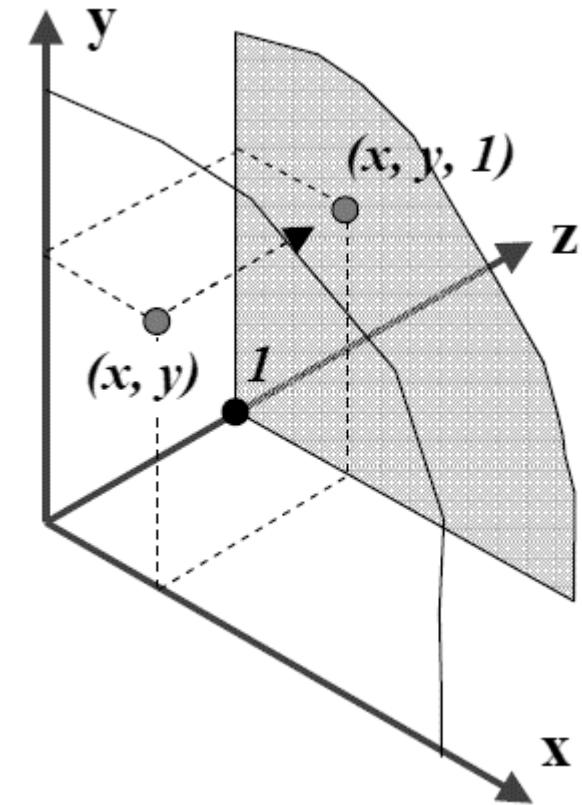
Reprezentacja i przekształcenia w przestrzeni 2D

Obrót wokół środka układu współrzędnych:

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi \\y' &= y \cos \varphi + x \sin \varphi\end{aligned}$$

zapisany zostanie następująco:

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



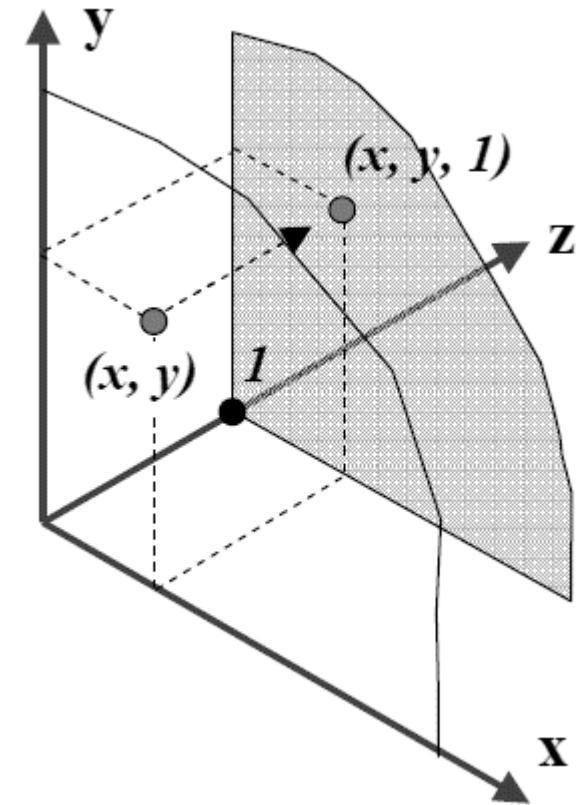
Reprezentacja i przekształcenia w przestrzeni 2D

Obrót wokół środka układu współrzędnych:

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi \\y' &= y \cos \varphi + x \sin \varphi\end{aligned}$$

opisuje więc macierz:

$$R(\varphi) = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Reprezentacja i przekształcenia w przestrzeni 2D

Podsumowując:

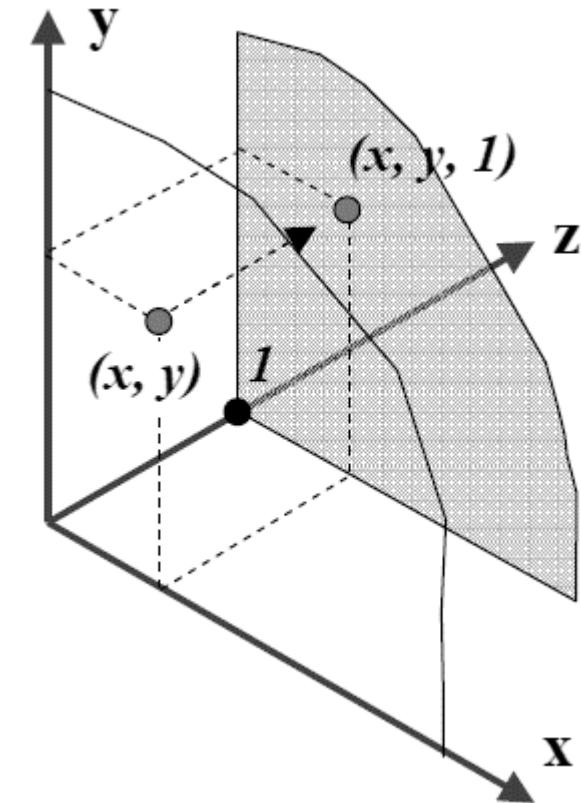
Po wprowadzeniu współrzędnych jednorodnych wszystkie trzy transformacje elementarne opisane zostały w ten sam sposób.

Współrzędne punktu po wykonaniu transformacji można wyznaczyć mnożąc wektor opisujący współrzędne punktu przed transformacją przez odpowiednią macierz M .

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \cdot M$$

gdzie $M: T, S$ lub O .

Transformacje te można składać →



Reprezentacja i przekształcenia w przestrzeni 2D

Składanie przekształceń:

Macierz będąca iloczynem dowolnej liczby macierzy reprezentujących różne przekształcenia zawiera złożenie tych przekształceń. Dzięki temu zamiast osobno wykonywać kolejne przekształcenia,

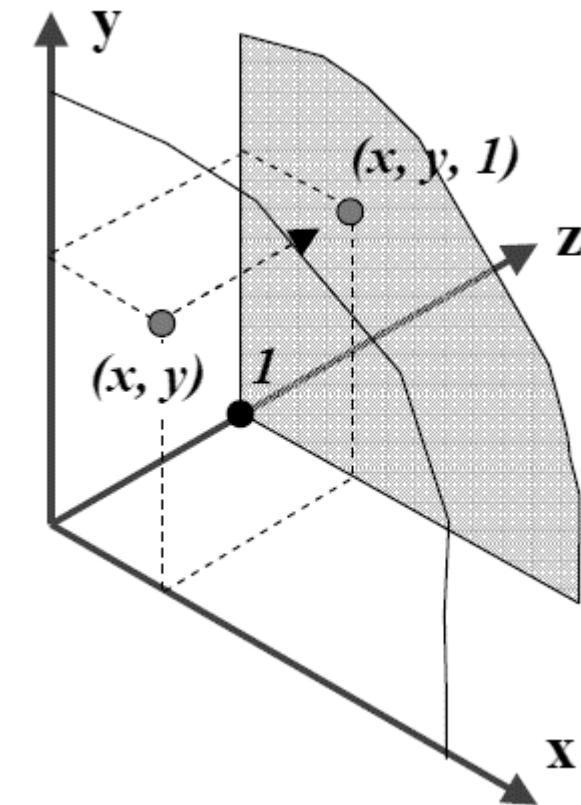
np. $x' = x \cdot M_1, x'' = x' \cdot M_2, \dots, x^{(n)} = x^{(n-1)} \cdot M_n$

Można najpierw wykonać mnożenie odpowiednich macierzy przekształceń,

np. $M = M_1 \cdot M_2 \cdot \dots \cdot M_n$, a dalej używać macierzy wynikowej $x' = x \cdot M$.

Czyli zamiast wykonać n mnożeń punktu przez macierze, to samo uzyskuje się jednym mnożeniem punktu przez macierz, dzięki uprzedniemu wykonaniu n-1 mnożeń macierzy.

Ma to szczególne znaczenie w momencie, gdy dane przekształcenie jest wielokrotnie powtarzane.

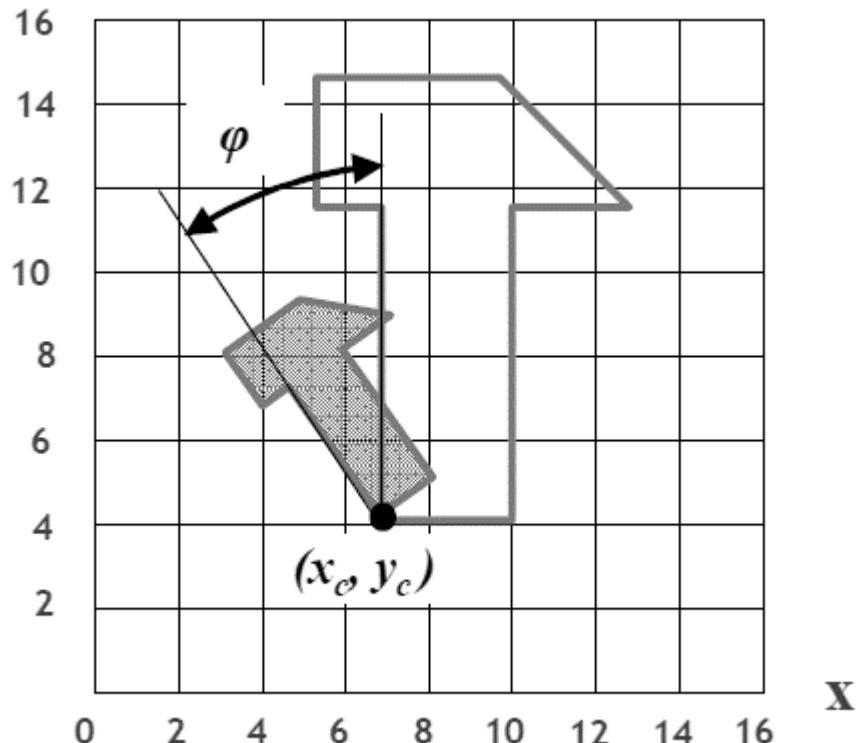


Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Przykład:

Obrócić obiekt wokół punktu (x_c, y_c) o kąt φ , pomniejszając go dwukrotnie.

Jak opisać określoną wyżej transformację przy pomocy wzoru?



Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

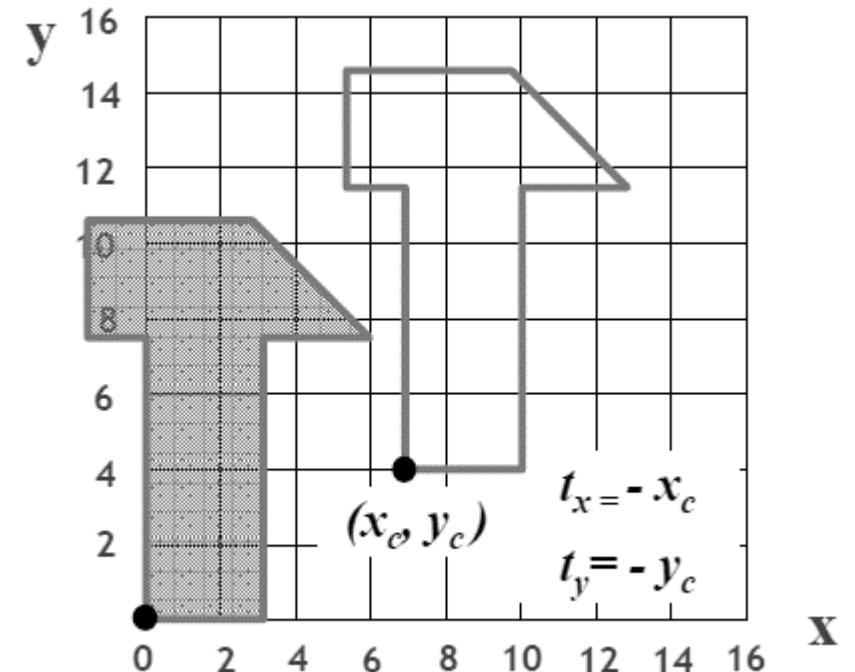
Krok 1:

Przesunięcie obiektu tak, aby punkt obrotu (x_c, y_c) znalazł się w punkcie $(0, 0)$, czyli odjęcie od współrzędnych (x_c, y_c) wartości (x_c, y_c) . Odjęcie tych wartości od wszystkich punktów obiektu spowoduje przesunięcie całego obiektu o wektor $[-x_c, -y_c]$.

$$p = [x \ y \ 1]$$

$$p' = [x' \ y' \ 1]$$

$$p' = p \cdot T(-x_c, -y_c)$$

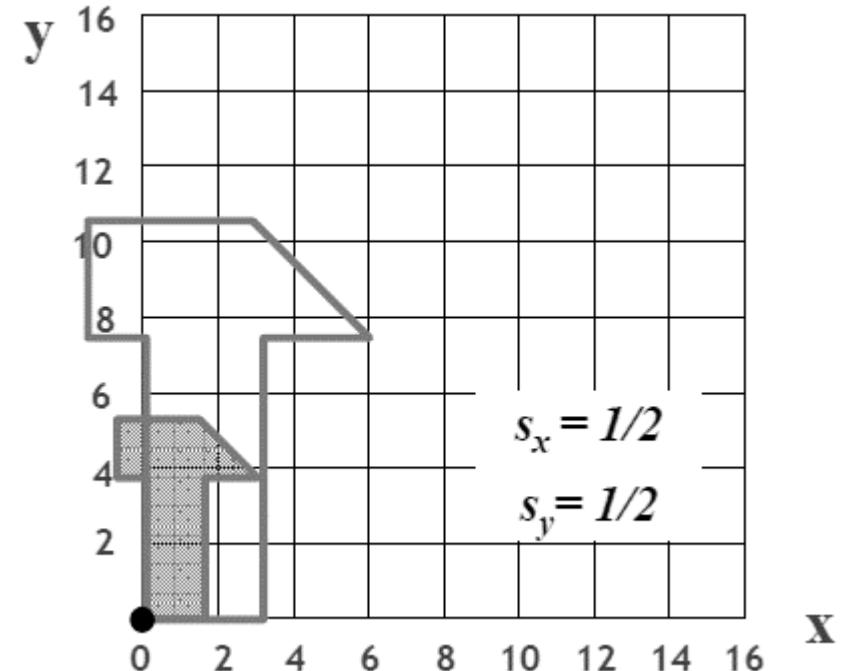


Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Krok 2:

Przeskalowanie obiektu z parametrami skalowania $s_x=1/2$, $s_y=1/2$

$$p' = p \cdot T(-x_c, -y_c) \cdot S(s_x, s_y)$$

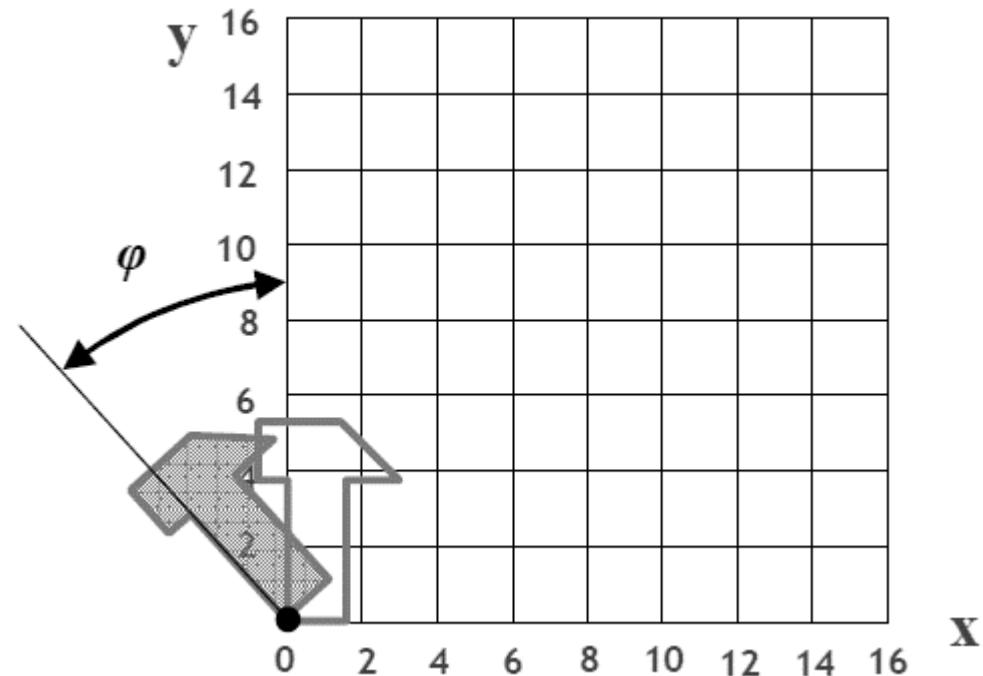


Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Krok 3:

Obrócenie obiektu wokół środka układu
współrzędnych o kąt φ .

$$p' = p \cdot T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\varphi)$$



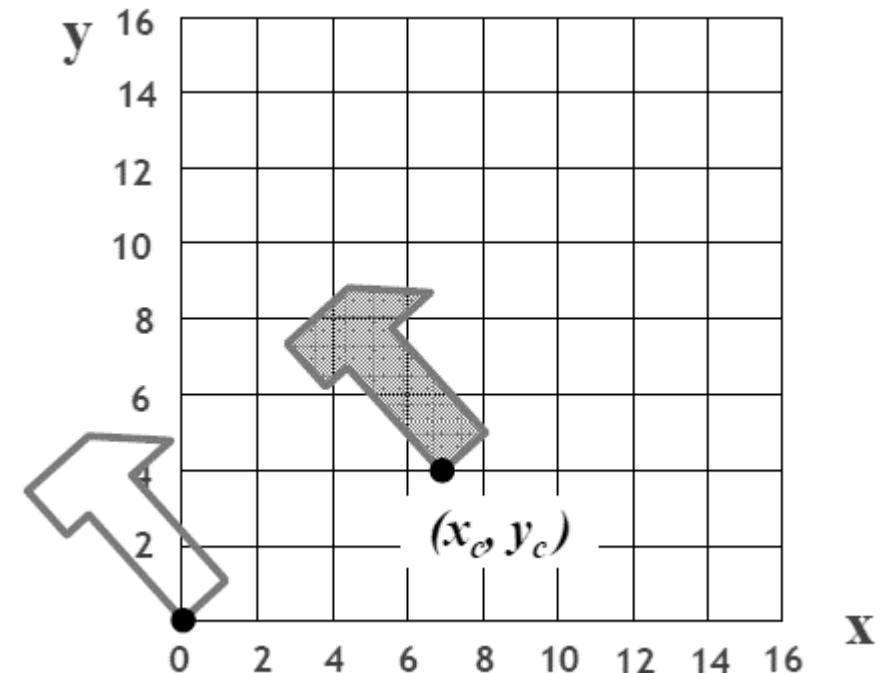
Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Krok 4:

Przesunięcie obiektu tak, aby punkt (0, 0)
znalazł się w punkcie (x_c , y_c) .

$$p' = p \cdot T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\varphi) \cdot T(x_c, y_c)$$

Poszczególne transformacje i ich kolejność
zostały określone – wiedząc to wszystko można
wyznaczyć jedną macierz transformacji.



Reprezentacja i przekształcenia w przestrzeni 2D i 3D

Przypomnienie - Mnożenie macierzy A i B:

$$\begin{bmatrix} -1 & -2 & 3 \\ 0 & 2 & -1 \\ -1 & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 5 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 3 \end{bmatrix}$$

Każda komórka wynikowej macierzy zawiera wynik sumowania iloczynów odpowiadającego jej wiersza macierzy A i kolumny macierzy B.

Przykładowo komórka (3,3) wynika z wymnożenia wiersza 3go z A i kolumny 3ciej z B.

$$\begin{bmatrix} -1 & -2 & 3 \\ 0 & 2 & -1 \\ \boxed{-1} & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 5 & \boxed{1} \\ 2 & 1 & 2 \\ 3 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & -1 & 4 \\ 1 & 0 & 1 \\ 5 & -2 & \boxed{} \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 2D i 3D

Przypomnienie - Mnożenie macierzy A i B możemy zapisać tak: $A \cdot B$ albo krócej tak: AB .

Własności mnożenia macierzy:

- mnożenie macierzy AB jest możliwe, tylko wtedy gdy:
liczba kolumn A=liczba wierszy B
- mnożenie macierzy jest łączne: $A(BC) = (AB)C$
- mnożenie macierzy jest rozdzielne względem dodawania:
 $A(B+C) = AB+AC$ oraz $(A+B)C = AC+BC$
- mnożenie macierzy nie jest przemienne: $AB \neq BA$

Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

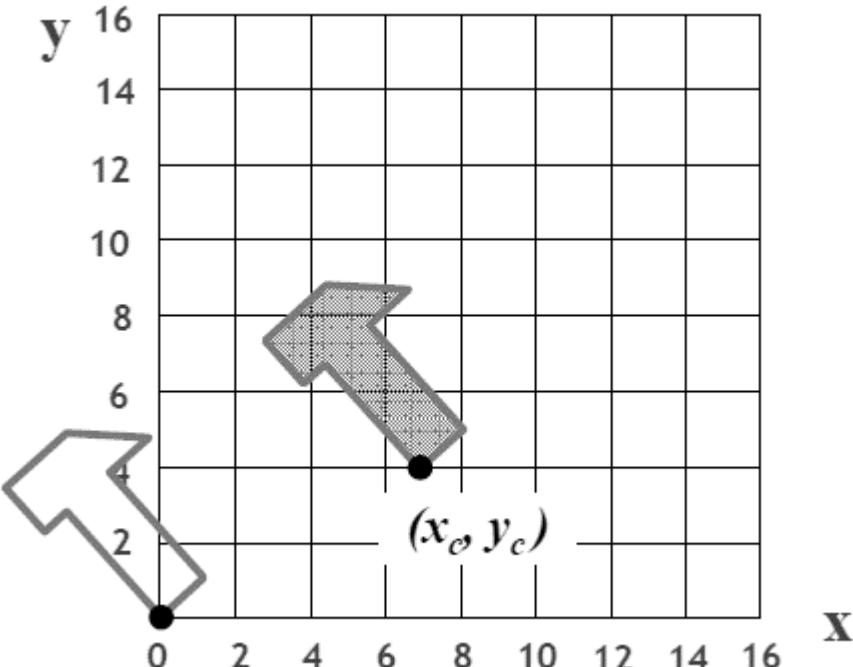
Macierz transformacji dla opisywanego przykładu będzie wynosić:

$$M = T(-x_c, -y_c) \cdot S(s_x, s_y) \cdot R(\varphi) \cdot T(x_c, y_c)$$

czyli:

$$M = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

, m' oznaczają wartości wyznaczone dla poszczególnych komórek macierzy w przypadku gdy znane są wartości x_c, y_c, s_x, s_y i φ



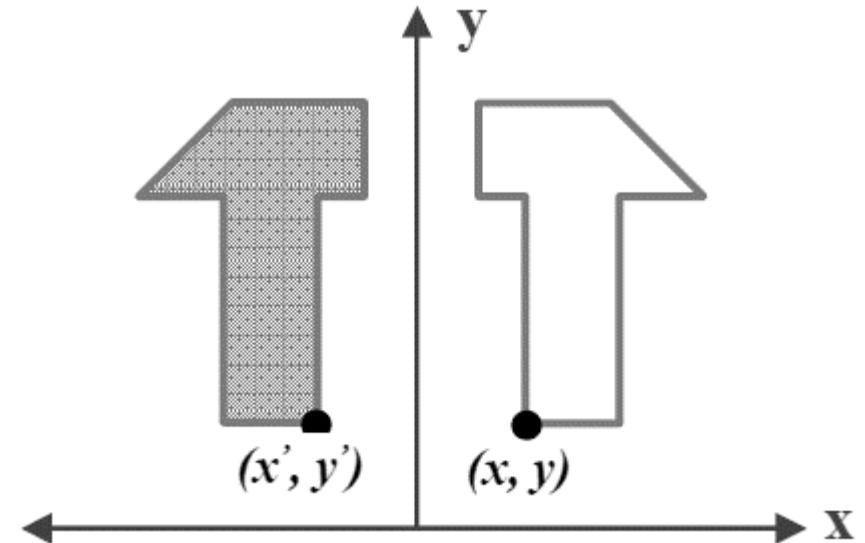
$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T(t_x, t_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \quad R(\varphi) = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Nie wszystkie transformacje daje się sprowadzić do złożenia trzech transformacji elementarnych (ale daje się je zapisać macierzowo)

Przykład 1 – odbicie lustrzane:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

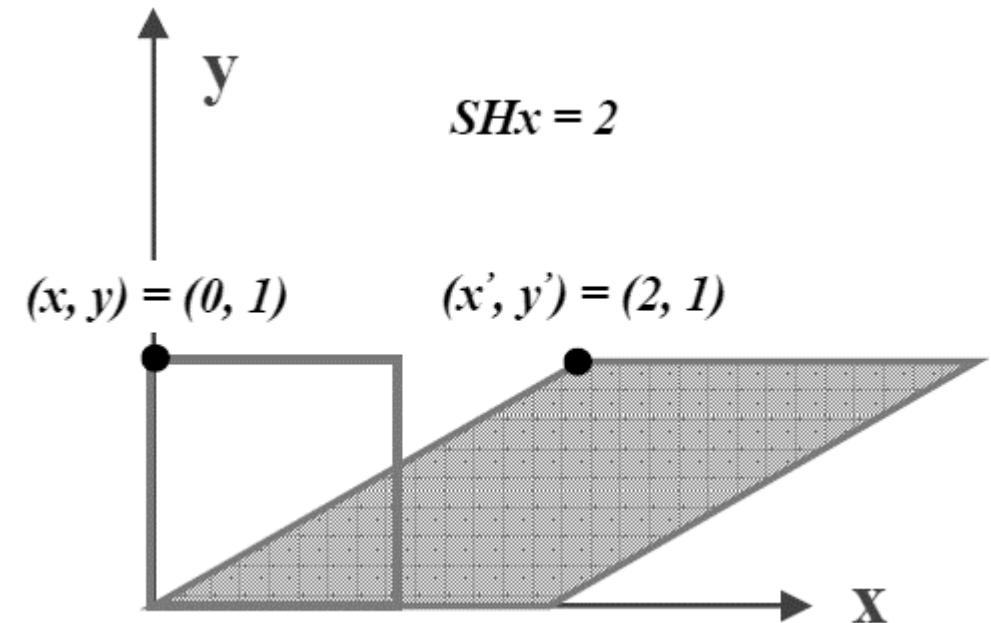


Reprezentacja i przekształcenia w przestrzeni 2D – składanie transformacji

Nie wszystkie transformacje daje się sprowadzić do złożenia trzech transformacji elementarnych.

Przykład 2 – ścinanie:

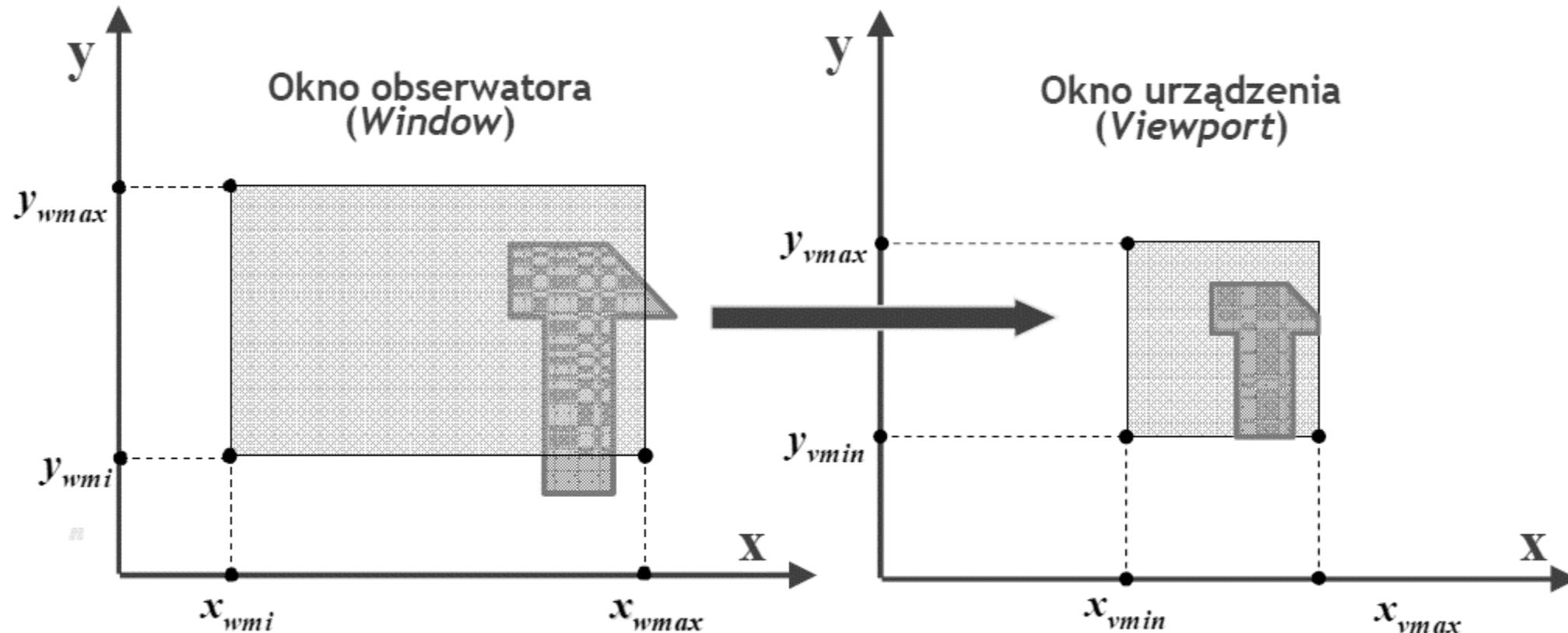
$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ SHx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Reprezentacja i przekształcenia w przestrzeni 2D – wizualizacja w oknie

Wprowadzenie

Reprezentacja i przekształcenia w przestrzeni 2D – wizualizacja w oknie



Układ współrzędnych
obserwatora
(World Coordinates)

Układ współrzędnych
urządzenia
(Device Coordinates)

Reprezentacja i przekształcenia w przestrzeni 2D – wizualizacja w oknie

Algorytm wizualizacji 2-D:

1. Zdefiniować obiekt w układzie współrzędnych obserwatora .
2. W układzie współrzędnych obserwatora określić okno obserwatora
3. W układzie współrzędnych urządzenia określić okno urządzenia.
4. Zmodyfikować opis obiektu usuwając te elementy, które znajdują się poza oknem obserwatora (obcinanie).
5. Przetransformować opis obiektu z wnętrza okna obserwatora do wnętrza okna urządzenia, stosując transformacje:

Reprezentacja i przekształcenia w przestrzeni 2D – wizualizacja w oknie

$$p_v = p_w \cdot T(-x_{w\min}, -y_{w\min}) S(s_x, s_y) T(x_{v\min}, y_{v\min})$$

przy czym:

$$s_x = \frac{x_{v\max} - x_{v\min}}{x_{w\max} - x_{w\min}}$$

$$s_y = \frac{y_{v\max} - y_{v\min}}{y_{w\max} - y_{w\min}}$$

a następnie

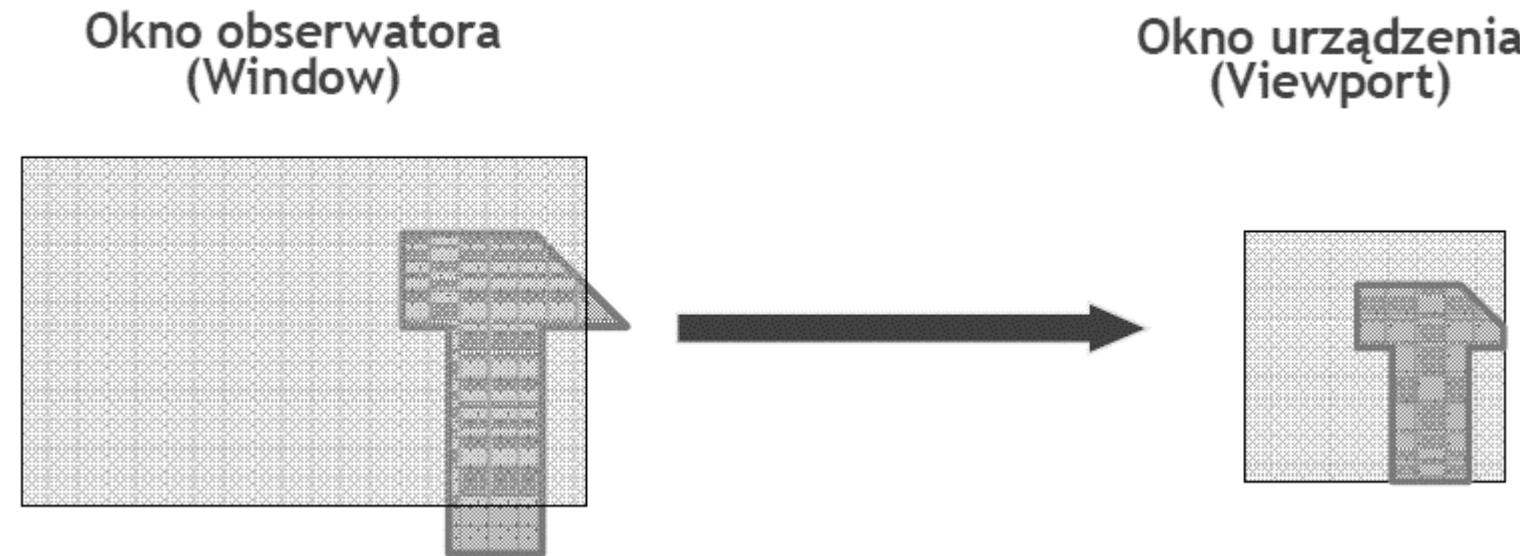
6. Narysować obiekt na ekranie.

Reprezentacja i przekształcenia w przestrzeni 2D – wizualizacja w oknie

Już w momencie określenia okna obserwatora mamy do czynienia z obcinaniem (ang. clipping).

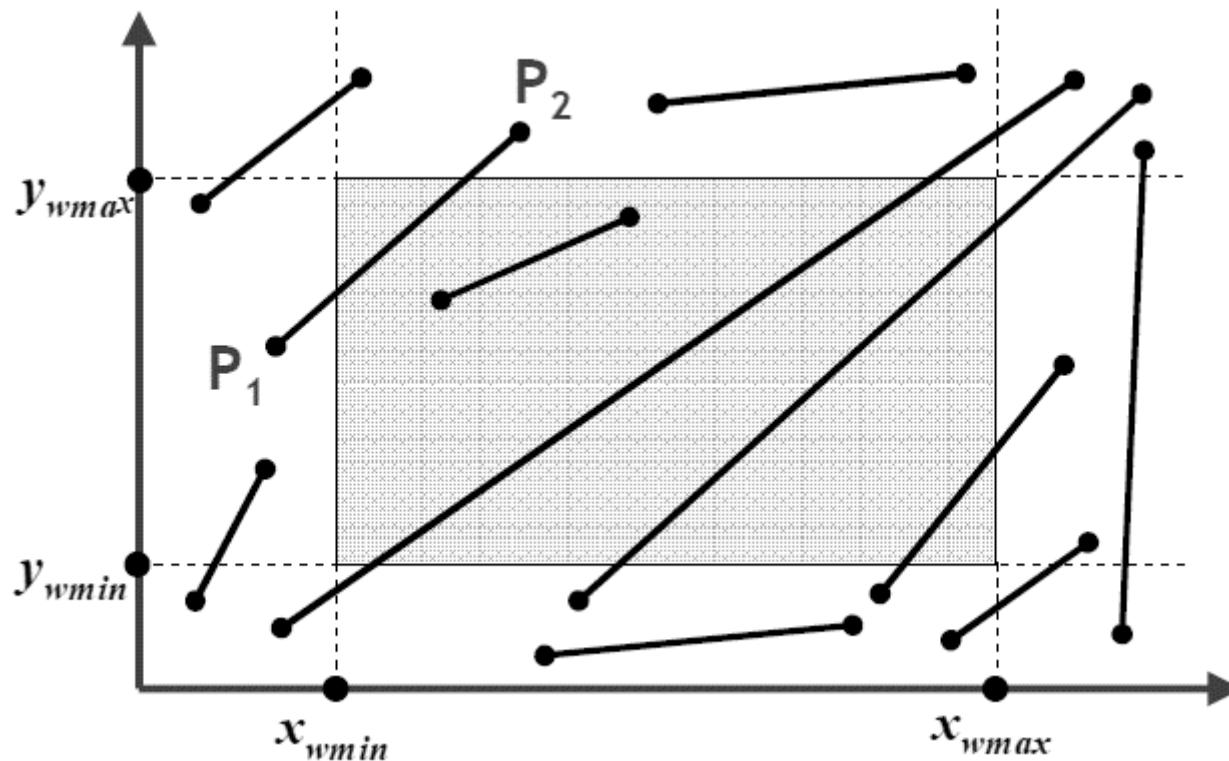
Punk 4. mówił „Zmodyfikować opis obiektu usuwając te elementy, które znajdują się poza oknem obserwatora (obcinanie).”

Służy temu algorytm Cohen'a i Sutherlanda.



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Przykładowy układ odcinków w przestrzeni obserwatora:



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

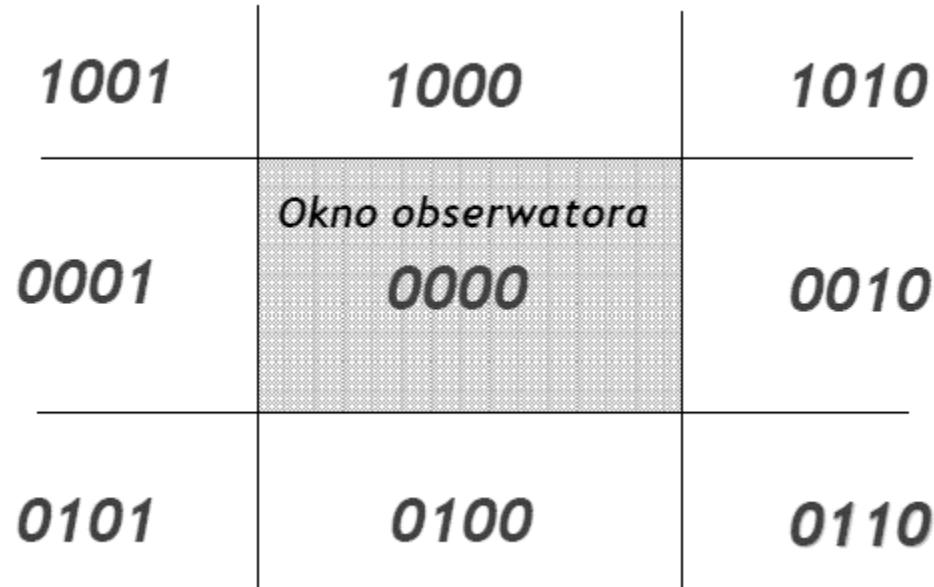
Kodowanie obszarów w przestrzeni obserwatora:

bit1 = 1 - na lewo od okna obserwatora,

bit2 = 1 - na prawo od okna obserwatora,

bit3 = 1 - w dół od okna obserwatora,

bit4 = 1 - w góre od okna obserwatora.



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Krok 1

Dla każdego punktu końcowego odcinka obliczyć różnicę

$$\alpha_1 = x - x_{w \min}$$

współrzędnych punktu końcowego i granic okna obserwatora:

$$\alpha_2 = x_{w \max} - x$$

$$\alpha_3 = y - y_{w \min}$$

$$\alpha_4 = y_{w \max} - y$$

Krok 2

Zakodować wszystkie punkty końcowe odcinków według reguły:

jeżeli $\alpha_i > 0$ to $\text{bit}_i = 1$

jeżeli $\alpha_i \leq 0$ to $\text{bit}_i = 0$

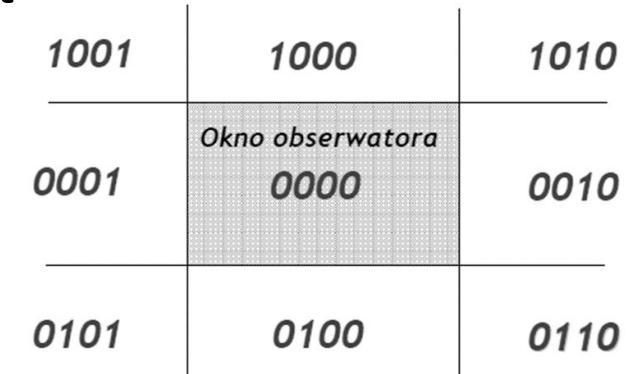
Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Krok 3

Sprawdzić kody par punktów końcowych dla wszystkich odcinków.

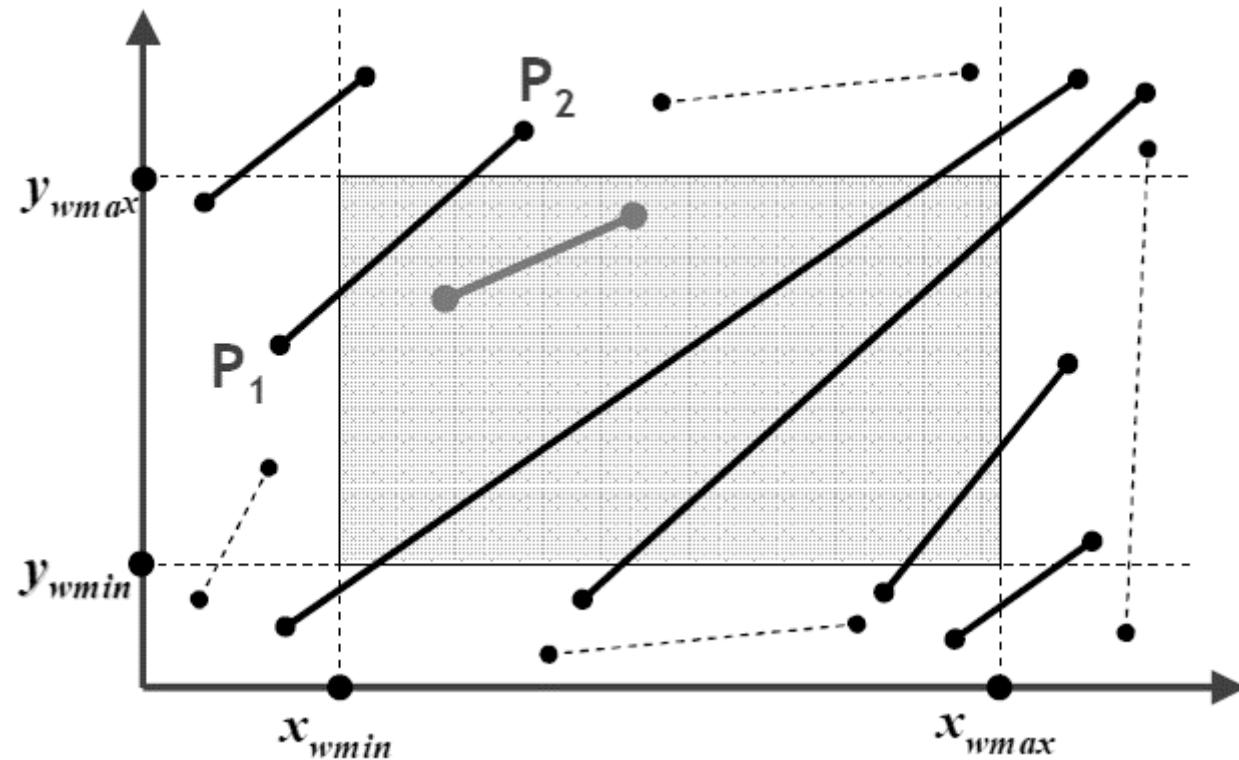
Jeżeli:

1. kod $P_1 = P_2 = 0000$ - odcinek leży całkowicie wewnątrz okna obserwatora.
2. którykolwiek biti dla $P_1 =$ biti dla $P_2 = 1$ - odcinek leży całkowicie na zewnątrz okna obserwatora.
 - pozostawić odcinki leżące wewnątrz okna,
 - usunąć odcinki leżące na zewnątrz okna,
 - jeśli wyczerpano w ten sposób wszystkie odcinki zakończyć algorytm,
 - w przeciwnym przypadku wykonać krok 4.



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Efekt działania algorytmu po wykonaniu kroku 3 (usunięcie odcinków poza oknem):



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Krok 4

Dla pozostałych odcinków, których punkty końcowe leżą na lewo, lub na prawo od granicy okna obliczyć nowe współrzędne tych punktów według wzorów:

$x = x_{w\min}$ dla punktów leżących na lewo od okna,

$x = x_{w\max}$ dla punktów leżących na prawo okna,

$y = y_k + m(x - x_k)$

gdzie

$$y = y_k + m(x - x_k)$$

x, y - nowe współrzędne punktu końcowego,

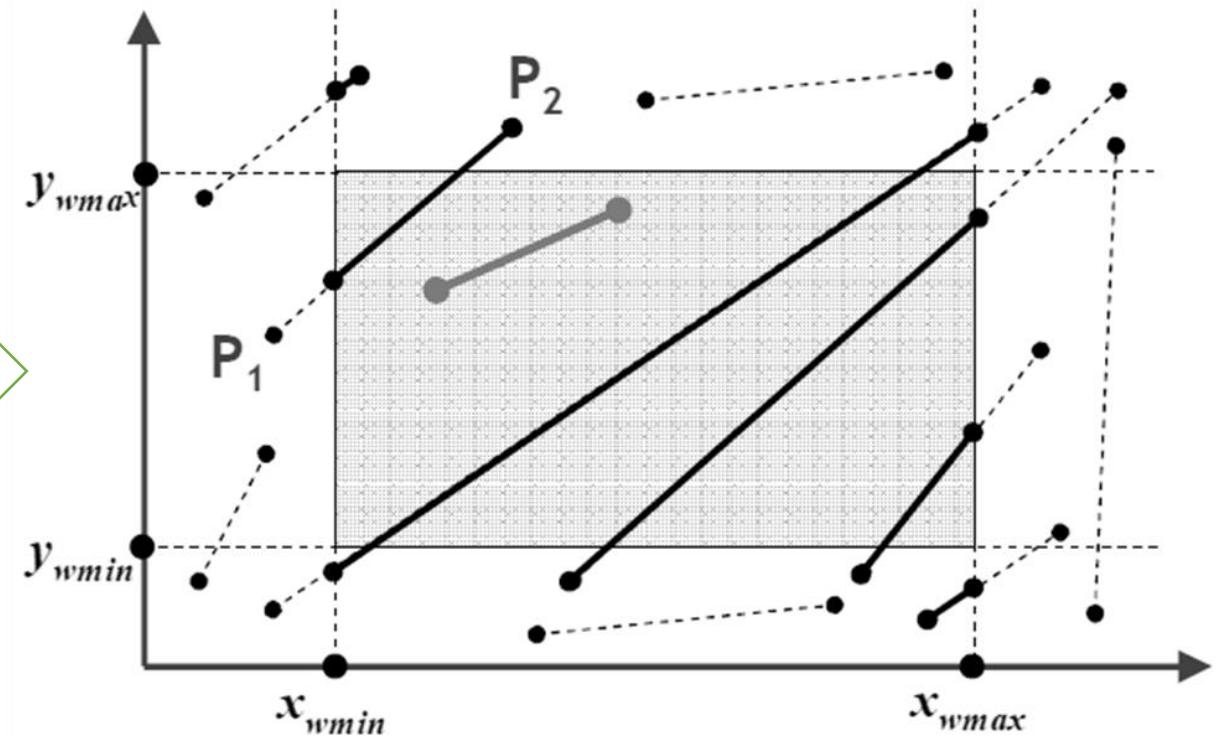
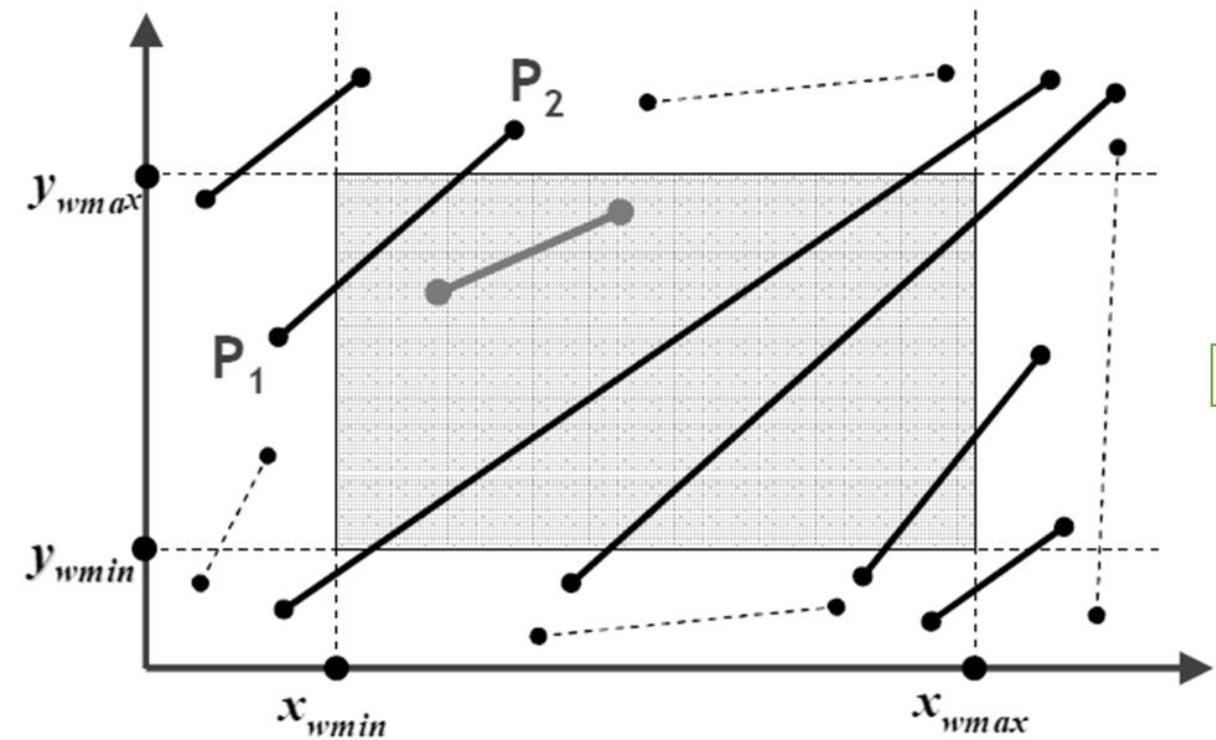
x_k, y_k - poprzednie współrzędne punktu.

Zakodować nowe punkty końcowe według reguły opisanej w kroku 2.

Powtórzyć krok 3.

Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Efekt po wykonaniu pierwszego przebiegu kroku 3:



Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Dla pozostałych odcinków, których punkty końcowe leżą pod, lub ponad granicami okna obliczyć nowe współrzędne tych punktów według wzorów:

$y = y_{w\min}$ dla punktów leżących poniżej okna,

$y = y_{w\max}$ dla punktów leżących powyżej okna,

$$x = x_k + (y - y_k)/m$$

gdzie:

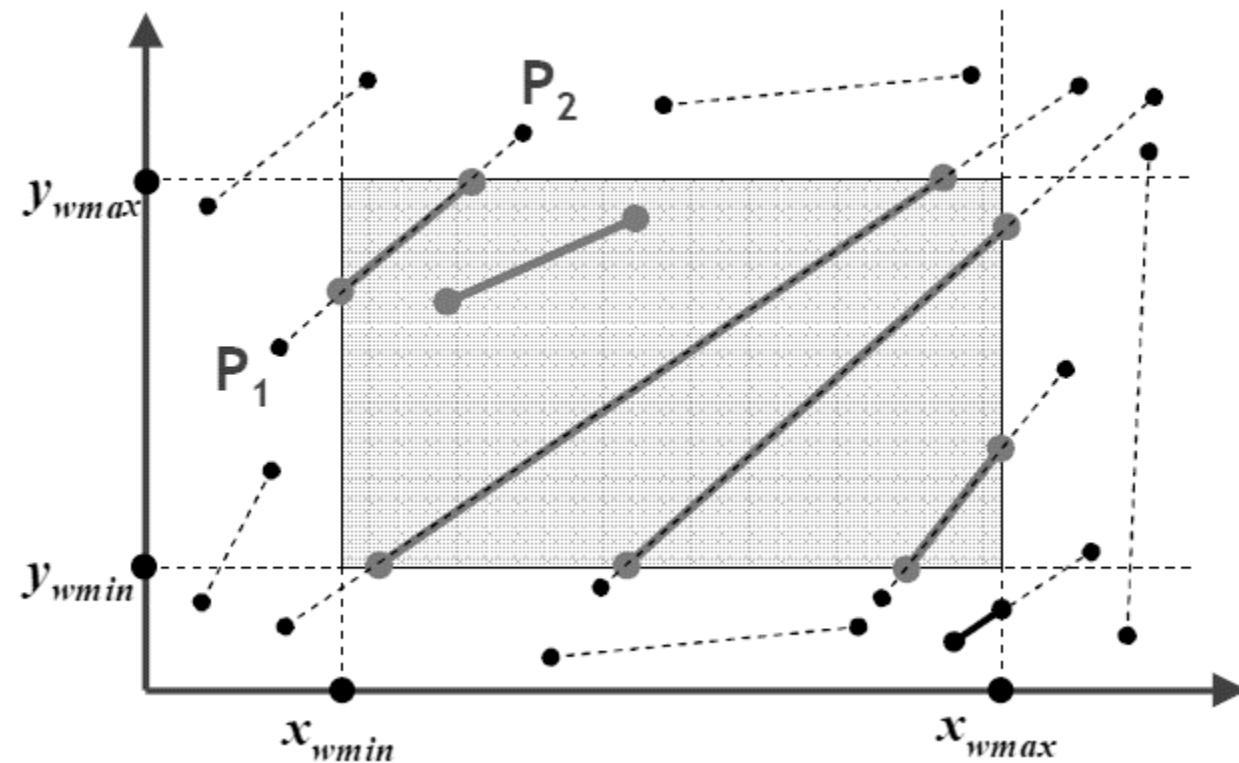
x, y - nowe współrzędne punktu końcowego,

x_k, y_k - poprzednie współrzędne punktu.

Zakodować nowe punkty końcowe według reguły opisanej w kroku 2.
Powtórzyć krok 3.

Reprezentacja i przekształcenia w przestrzeni 2D – obcinanie/algorytm Cohena i Sutherlanda

Końcowy efekt działania algorytmu obcinania:



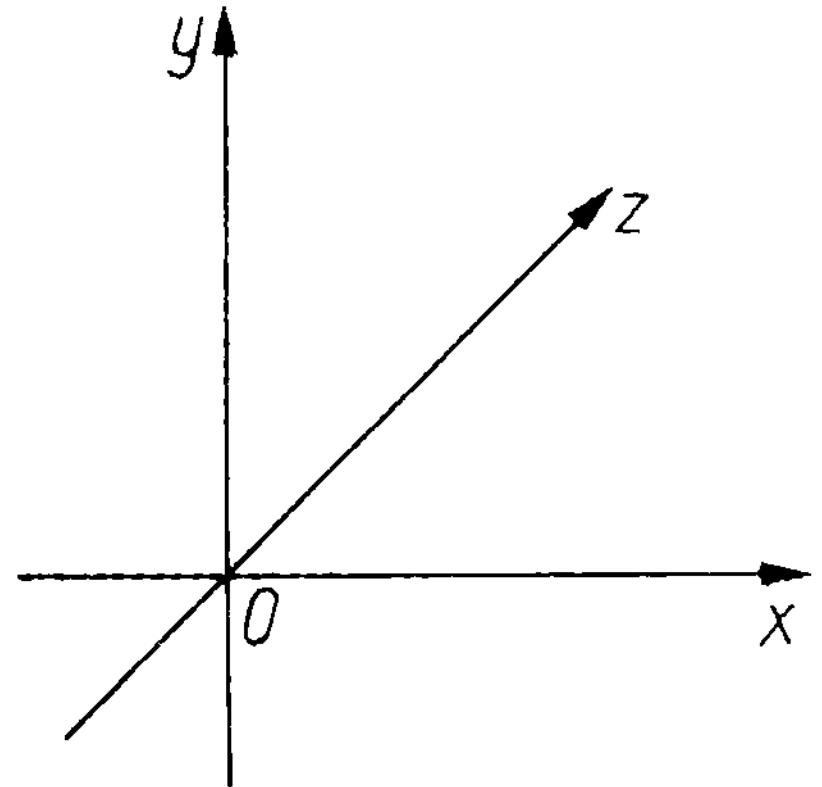
Reprezentacja i przekształcenia w przestrzeni 3D

Wprowadzenie

Reprezentacja i przekształcenia w przestrzeni 3D – lewoskrętny układ współrzędnych

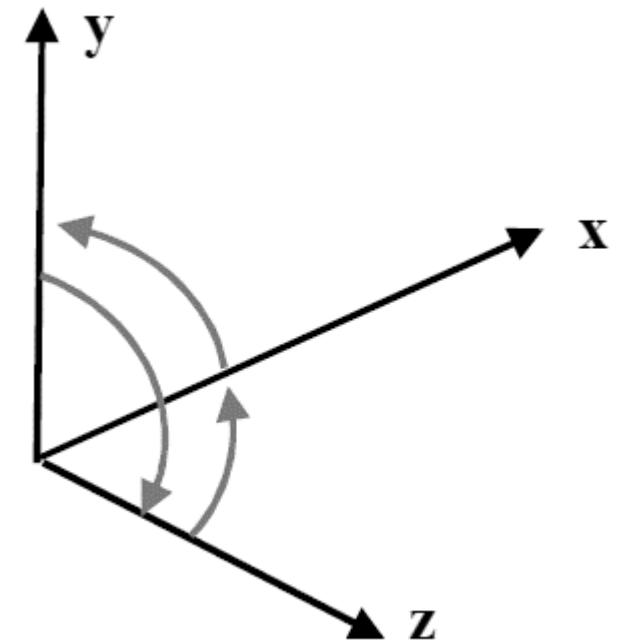
Do opisu obiektów trójwymiarowych często używamy układu kartezjańskiego lewoskrętnego (na rys.).

Wygodnie jest patrzeć na ten układ w ten sposób, że osz x przebiega poziomo od lewej do prawej, osz y pionowo od dołu do góry, a osz z prostopadle w głąb kierunku patrzenia.



Reprezentacja i przekształcenia w przestrzeni 3D – prawoskrętny układ współrzędnych

W przypadku prawoskrętnego układu współrzędnych - jeśli patrzymy z dodatniego kierunku osi w stronę środka układu współrzędnych, to obrót o 90° w kierunku przeciwnym do ruchu wskazówek zegara, przekształci jedną dodatnią oś w drugą.



Reprezentacja i przekształcenia w przestrzeni 3D

Transformacje elementarne:

- przesunięcie,
- zmiana skali,
- obroty wokół poszczególnych osi układu.

Współrzędne jednorodne: $(x \ y \ z) \rightarrow (x \ y \ z \ 1)$

$$(x' \ y' \ z') \rightarrow (x' \ y' \ z' \ 1)$$

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 3D

Przesunięcie:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

Skalowanie:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 3D

Obroty wokół osi:

Jeśli patrzymy z dodatniego kierunku osi w stronę środka układu współrzędnych, za obrót o kąt dodatni uważa się obrót w kierunku przeciwnym do ruchu wskazówek zegara.

Obrót wokół osi x:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 3D

Obrót wokół osi y:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obrót wokół osi z:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

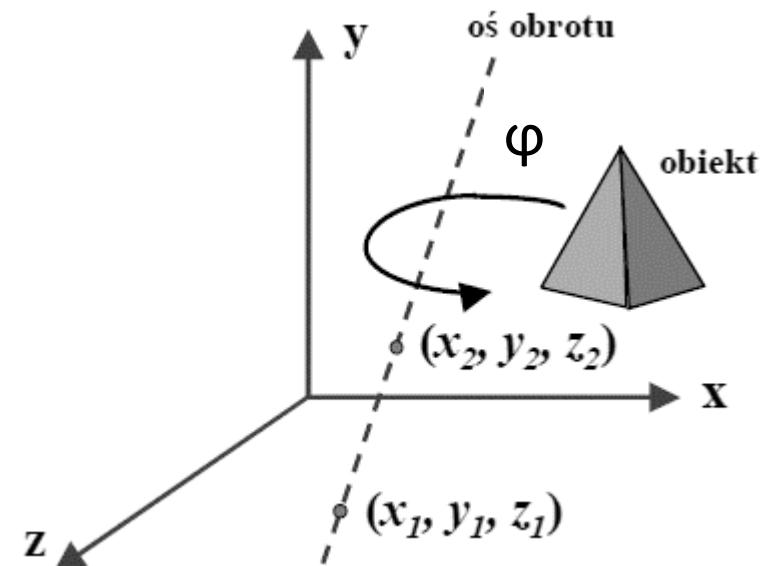
Reprezentacja i przekształcenia w przestrzeni 3D

Obrót punktu (obiektu) wokół dowolnie zadanej osi.

Dane:

- opis osi obrotu $(x_1, y_1, z_1), (x_2, y_2, z_2)$,
- opis obiektu (np. siatka wieloboków),
- kąt obrotu: φ

Zadanie: należy wyznaczyć macierz transformacji realizującej obrót obiektu o kąt φ .



Reprezentacja i przekształcenia w przestrzeni 3D

Reprezentacja osi obrotu:

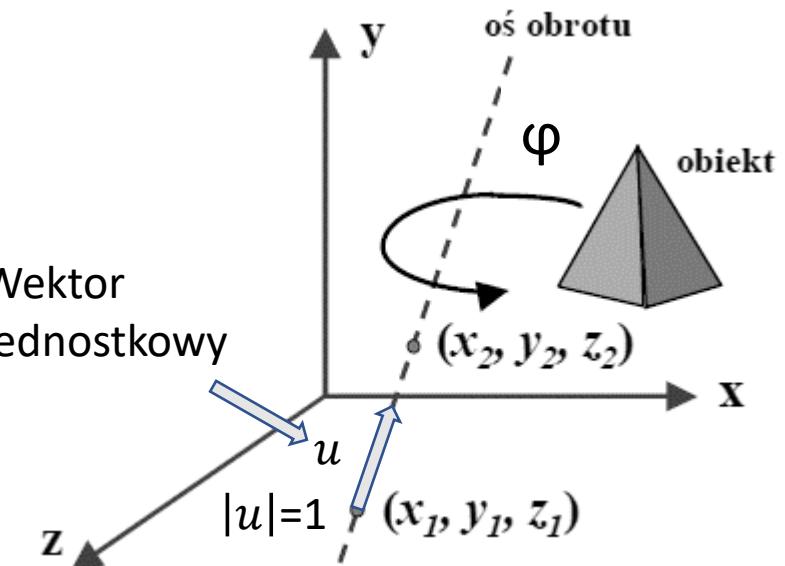
Oś obrotu reprezentowana będzie przez wektor $u = [a \ b \ c]$ zaczepiony w punkcie (x_1, y_1, z_1) , przy czym

$$a = \frac{x_2 - x_1}{|v|} \quad b = \frac{y_2 - y_1}{|v|} \quad c = \frac{z_2 - z_1}{|v|}$$

gdzie

$$|v| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

//dzięki dzieleniu przez długość wektor jest jednostkowy



Reprezentacja i przekształcenia w przestrzeni 3D

Kroki procedury obrotu obiektu wokół osi:

1. Przesunięcie osi i obiektu, tak aby oś obrotu przechodziła przez środek układu współrzędnych.

2. Obrócenie osi i obiektu, tak aby oś obrotu stała się współliniowa z jedną z osi układu współrzędnych
(np. z osią z).

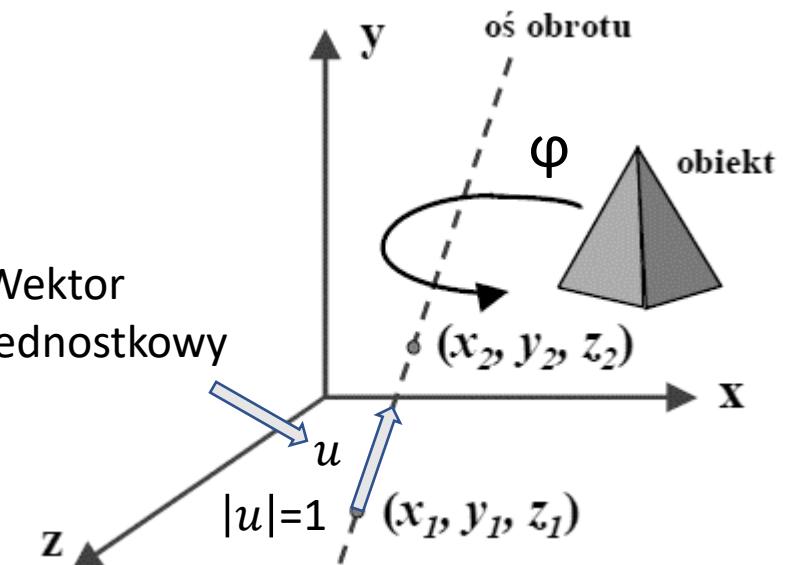
- „położenie”* osi na płaszczyźnie (x-z),
- obrót wokół osi y

3. Obrót obiektu wokół osi z o kąt φ .

4. Transformacja odwrotna do wykonanej w kroku 2.

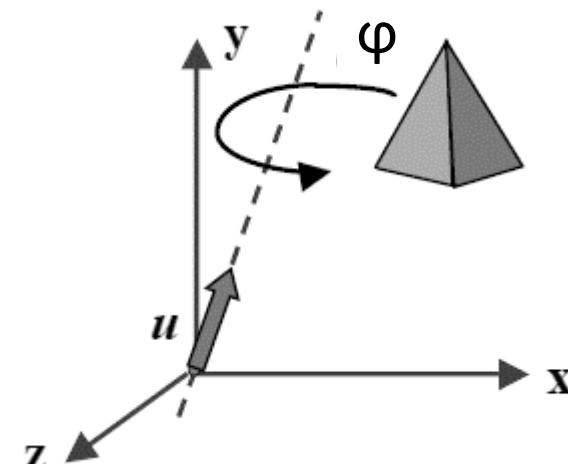
5. Transformacja odwrotna do wykonanej w kroku 1.

* „położenie” – należy oś położyć na płaszczyźnie



Reprezentacja i przekształcenia w przestrzeni 3D

Krok 1: Przesunięcie osi i obiektu, tak aby oś przechodziła przez środek układu współrzędnych.



Wykonane przesunięcie opisuje macierz:

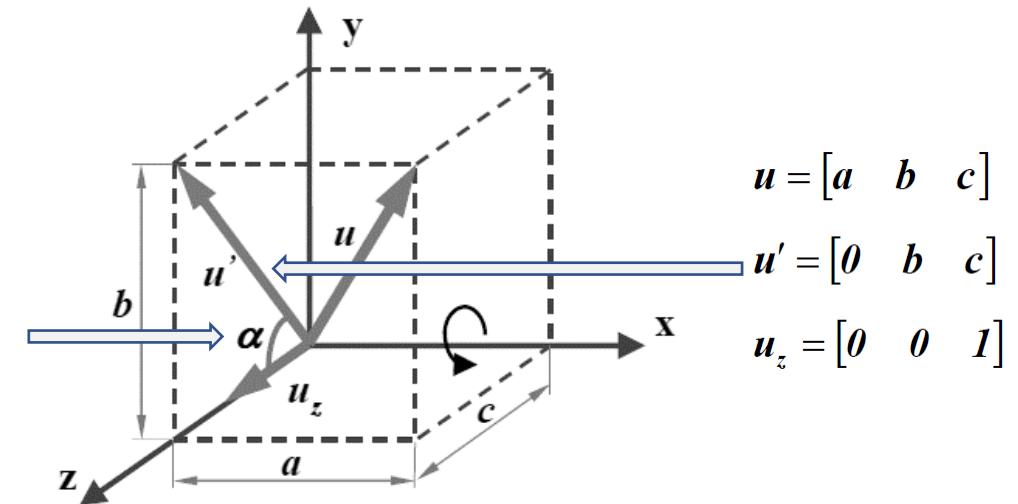
$$T(-x_1, -y_1, -z_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_1 & -y_1 & -z_1 & 1 \end{bmatrix}$$

Reprezentacja i przekształcenia w przestrzeni 3D

Krok 2: Obrócenie osi i obiektu, tak aby oś stała się współliniowa z osią z.

Położenie osi obrotu na płaszczyźnie (x-z):

Położenie osi obrotu na płaszczyźnie (x-z) jest równoważne obrotowi wektora u' o kąt α .



Reprezentacja i przekształcenia w przestrzeni 3D

Kąt obrotu α lub nawet $\sin \alpha$ i $\cos \alpha$ można wyznaczyć łatwo korzystając z iloczynu skalarnego i wektorowego:

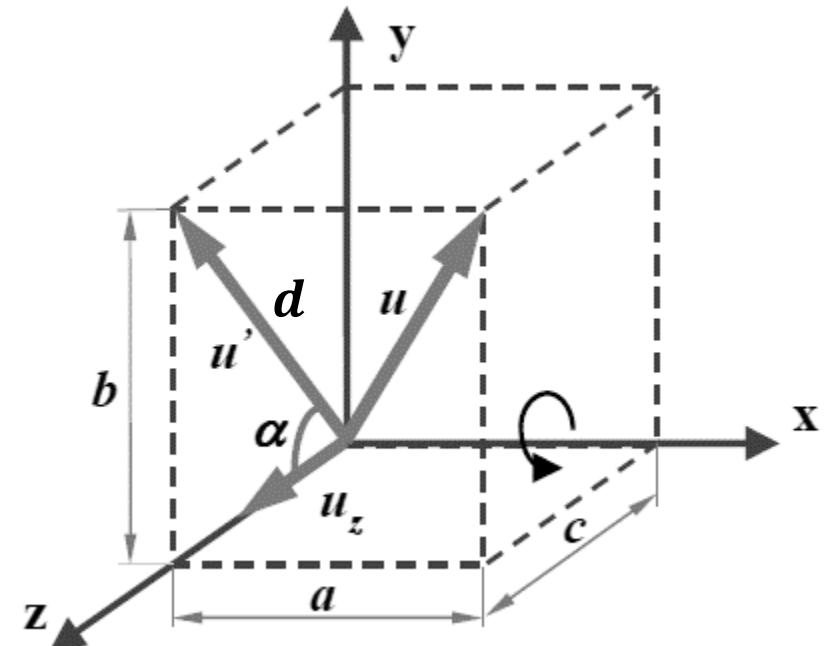
//iloczyn skalarny

$$\mathbf{u}' \cdot \mathbf{u}_z = 0 \cdot 0 + b \cdot 0 + c \cdot 1 = c$$

$$\mathbf{u}' \cdot \mathbf{u}_z = |\mathbf{u}'| \cdot |\mathbf{u}_z| \cdot \cos \alpha$$

$$|\mathbf{u}'| = \sqrt{b^2 + c^2} = d \quad |\mathbf{u}_z| = 1$$

$$c = d \cdot \cos \alpha \quad \xrightarrow{\text{blue arrow}} \quad \cos \alpha = \frac{c}{d}$$



Reprezentacja i przekształcenia w przestrzeni 3D

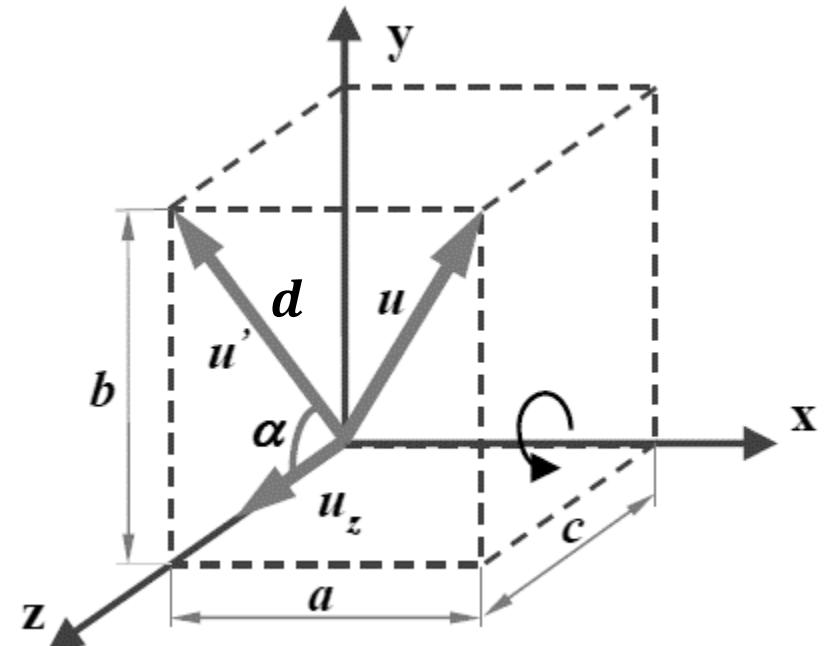
Wartości $\sin \alpha$ i $\cos \alpha$ można wyznaczyć korzystając z iloczynu skalarnego i wektorowego:

//iloczyn wektorowy

$$u' \times u_z = \det \begin{bmatrix} u_x & u_y & u_z \\ 0 & b & c \\ 0 & 0 & 1 \end{bmatrix} = u_x \cdot b$$

$$u' \times u_z = u_x \cdot |u'| \cdot |u_z| \cdot \sin \alpha = u_x \cdot d \cdot \sin \alpha$$

$$\implies \sin \alpha = \frac{b}{d}$$



Reprezentacja i przekształcenia w przestrzeni 3D

Położenie osi obrotu na płaszczyźnie (x-z) można wykonać przez obrót wokół osi x o kąt α .

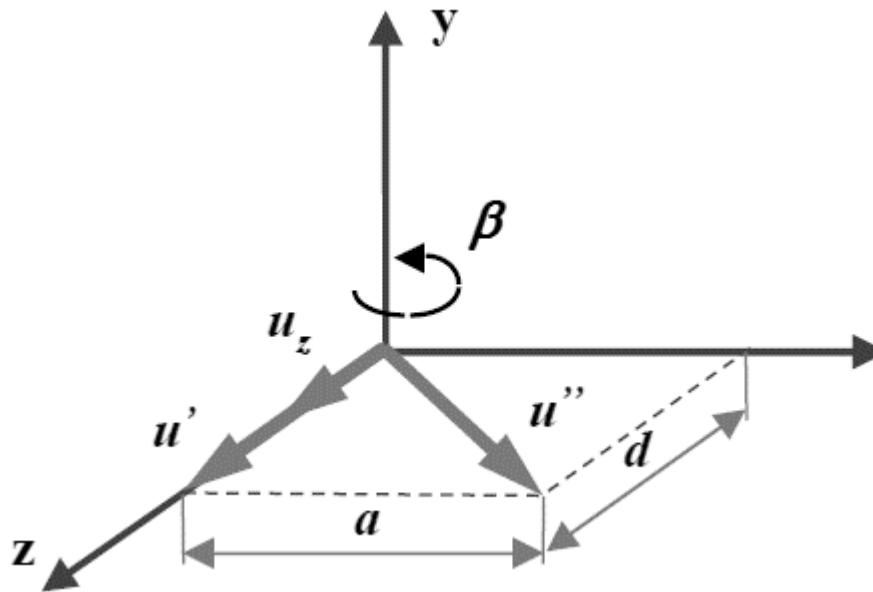
Obrót ten opisuje macierz:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & b/d & 0 \\ 0 & -b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Korzystne jest, że w miejsce funkcji trygonometrycznych w macierzy udało się wprowadzić łatwiejsze do obliczenia ilorazy.

Reprezentacja i przekształcenia w przestrzeni 3D

Położenie osi obrotu na osi z.



$$u'' = [a \ 0 \ d]$$

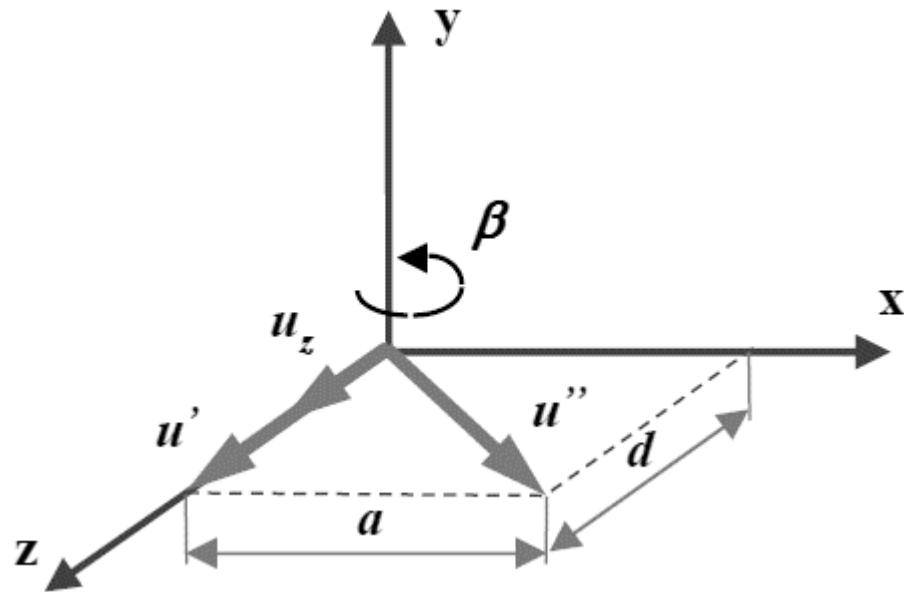
$$u' = [0 \ 0 \ d]$$

$$u_z = [0 \ 0 \ 1]$$

Położenie osi obrotu na osi z jest równoważne obrotowi wektora u'' o kąt β .

Reprezentacja i przekształcenia w przestrzeni 3D

Iloczyn skalarny



$$u'' \cdot u_z = a \cdot 0 + 0 \cdot 0 + d \cdot 1 = d$$

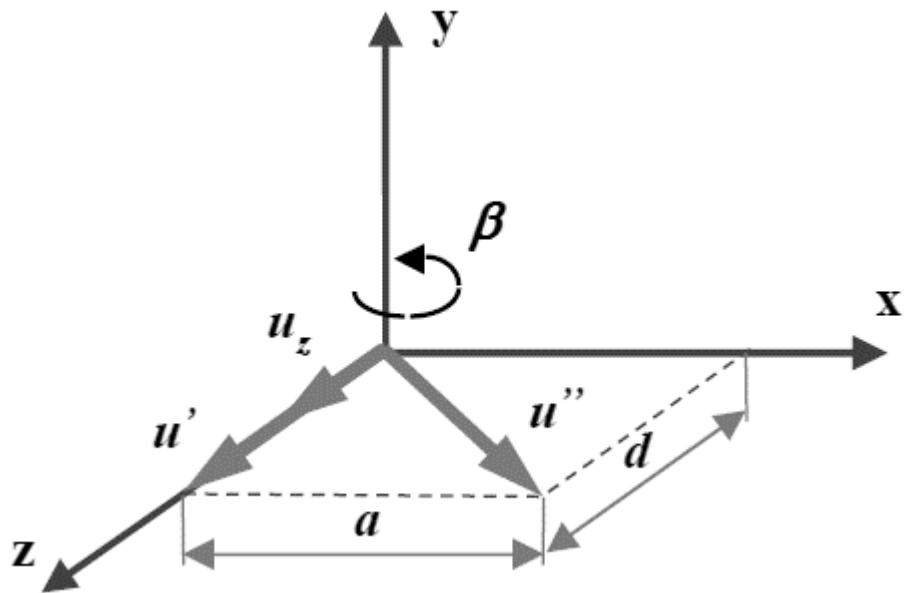
$$u'' \cdot u_z = |u''| \cdot |u_z| \cdot \cos \beta$$

$$|u''| = \sqrt{a^2 + d^2} = 1 \quad |u_z| = 1$$

$$\cos \alpha = d$$

Reprezentacja i przekształcenia w przestrzeni 3D

Iloczyn wektorowy



$$u'' \times u_z = \det \begin{bmatrix} u_x & u_y & u_z \\ a & 0 & d \\ 0 & 0 & 1 \end{bmatrix} = u_y \cdot (-a)$$

$$u'' \times u_z = u_y |u''| \cdot |u_z| \cdot \sin \beta = u_y \cdot \sin \beta$$

$$\sin \beta = -a$$

Reprezentacja i przekształcenia w przestrzeni 3D

Położenie osi obrotu na osi z można przeprowadzić przez obrót wokół osi y o kąt β .

Obrót ten opisuje macierz:

$$R_y(\beta) = \begin{bmatrix} d & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Podobnie jak poprzednio w miejsce funkcji trygonometrycznych w macierzy udało się wprowadzić łatwiejsze do obliczenia elementy.

Reprezentacja i przekształcenia w przestrzeni 3D

Krok 3: Obrócenie obiektu wokół osi z o kąt θ .

Obrót ten opisuje macierz:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Krok 4: Transformacja odwrotna do wykonanej w kroku 2

Macierz transformacji: $R_y^{-1}(\beta) \cdot R_x^{-1}(\alpha)$

Reprezentacja i przekształcenia w przestrzeni 3D

Krok 5: Transformacja odwrotna do wykonanej w kroku 1

Macierz transformacji: $\mathbf{T}^{-1}(-x_I, -y_I, -z_I)$

Ostatecznie, macierz realizująca transformację obrotu obiektu wokół zadanej osi o kat θ przyjmuje postać*:

$$\begin{aligned}\mathbf{R}(\boldsymbol{\Theta}) = & \mathbf{T}(-x_I, -y_I, -z_I) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_z(\theta) \cdot \\ & \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{T}^{-1}(-x_I, -y_I, -z_I)\end{aligned}$$

*złożenie 7 macierzy transformacji

Reprezentacja i przekształcenia w przestrzeni 2D i 3D

Dziękuję za uwagę

Modelowanie obiektów

Edward Półrolniczak, dr inż.,
Katedra Systemów Multimedialnych,
Wydział Informatyki, ZUT w Szczecinie

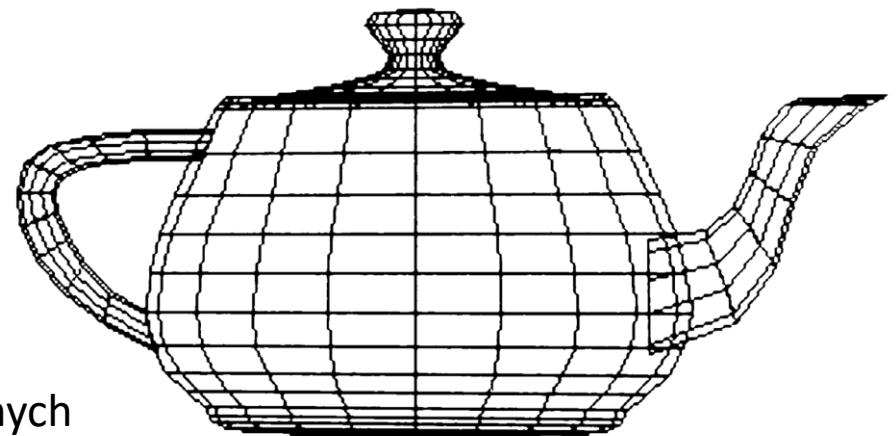
Zagadnienia

- Wprowadzenie
- Modele szkieletowe
- Równanie powierzchni w postaci uwikłanej
- Powierzchnie opisane parametrycznie
- Krzywe parametryczne w 3-D (krzywa Hermite'a)

Klasyczny czajniczek (obiekt zbudowany z krzywych) - najbardziej znana ikona grafiki komputerowej. W 1975 r. opracowany przez Martina Newella.

Był używany przez dziesiątki badaczy jako struktura do demonstrowania najnowszych metod modelowania realistycznych powierzchni i tekstur.

Modelowanie eleganckiego czajniczka wymagało określenia jego kształtu jako zbioru elementów gładkiej powierzchni, znanych jako płaty bikubiczne.



Gładkie krzywe i powierzchnie są przydatne w wielu zastosowaniach grafiki komputerowej.

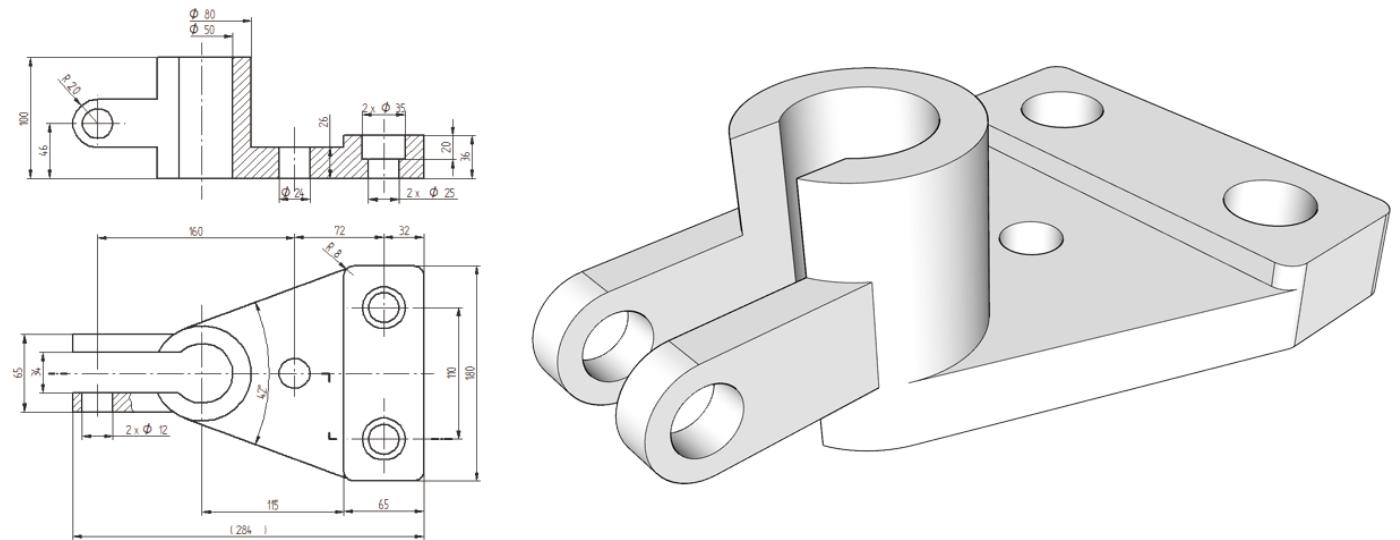
Wiele obiektów świata rzeczywistego to z natury obiekty gładkie a wiele programów grafiki komputerowej jest związanych z modelowaniem świata rzeczywistego.

Zastosowanie gładkich powierzchni:

- w projektowaniu wspomaganym komputerowo (CAD),
- w dobrej jakości fontach,
- wykresach,
- rysunkach artystycznych,
- w animacji (ścieżka kamery albo obiektu jest prawie zawsze gładka),
- operacje na przestrzeni barw (często gładka musi być ścieżka przechodząca przez przestrzeń barw).

Konieczność reprezentowania krzywych i powierzchni powstaje w przypadkach:

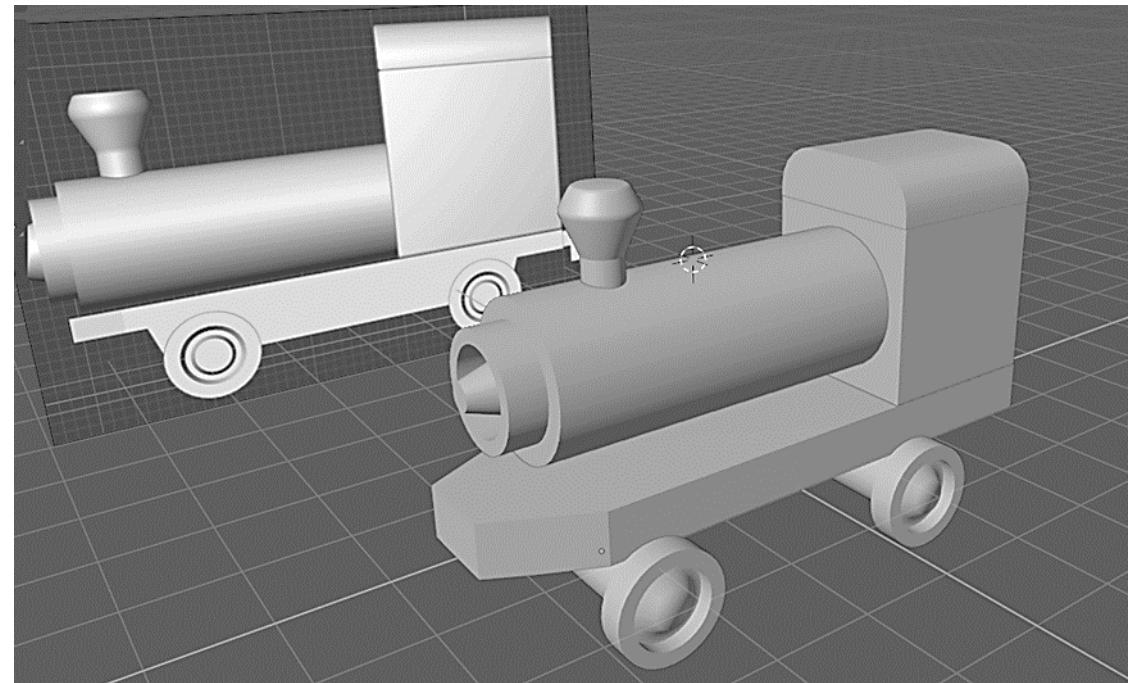
- w modelowaniu istniejących obiektów (odwzorowywanie, np. samochód, twarz albo góra),
- w modelowaniu obiektu nie istniejącego (projektowanie).



https://copio.pl/wp-content/uploads/2020/05/rysunek_model.png

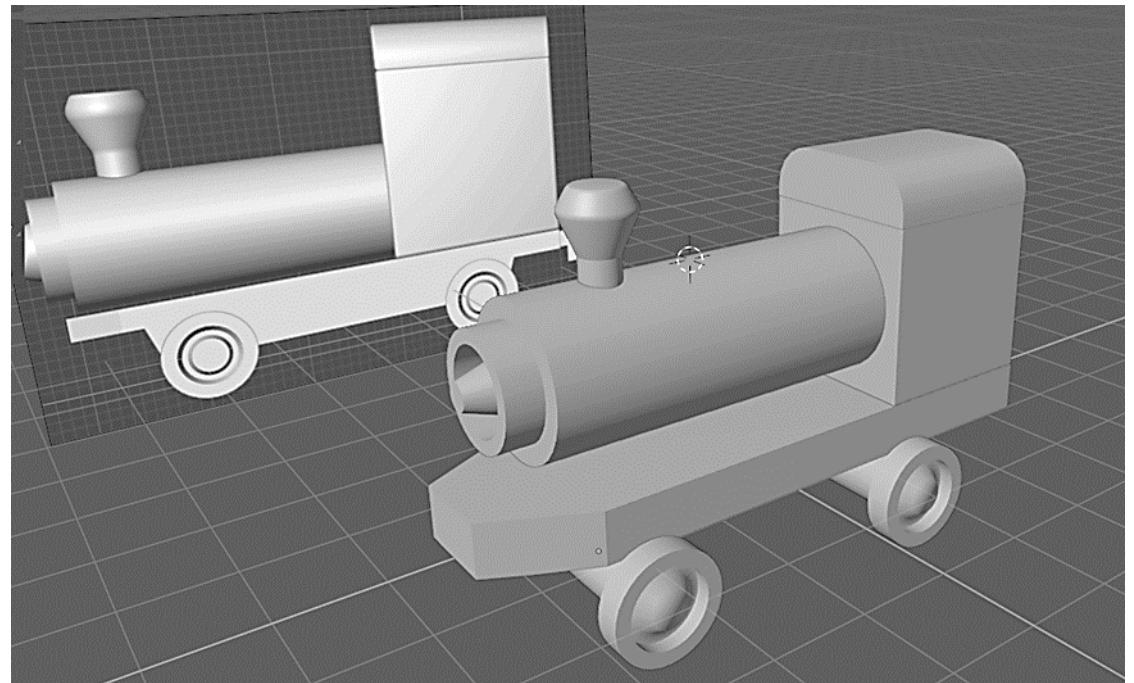
W pierwszym przypadku może nie istnieć matematyczny opis obiektu – musimy ten opis niejako stworzyć. *Możemy użyć jako modelu współrzędnych nieskończoność wielu punktów obiektu w świecie rzeczywistym, ale to podejście nie jest możliwe dla komputera o skończonej pamięci.

Ale...

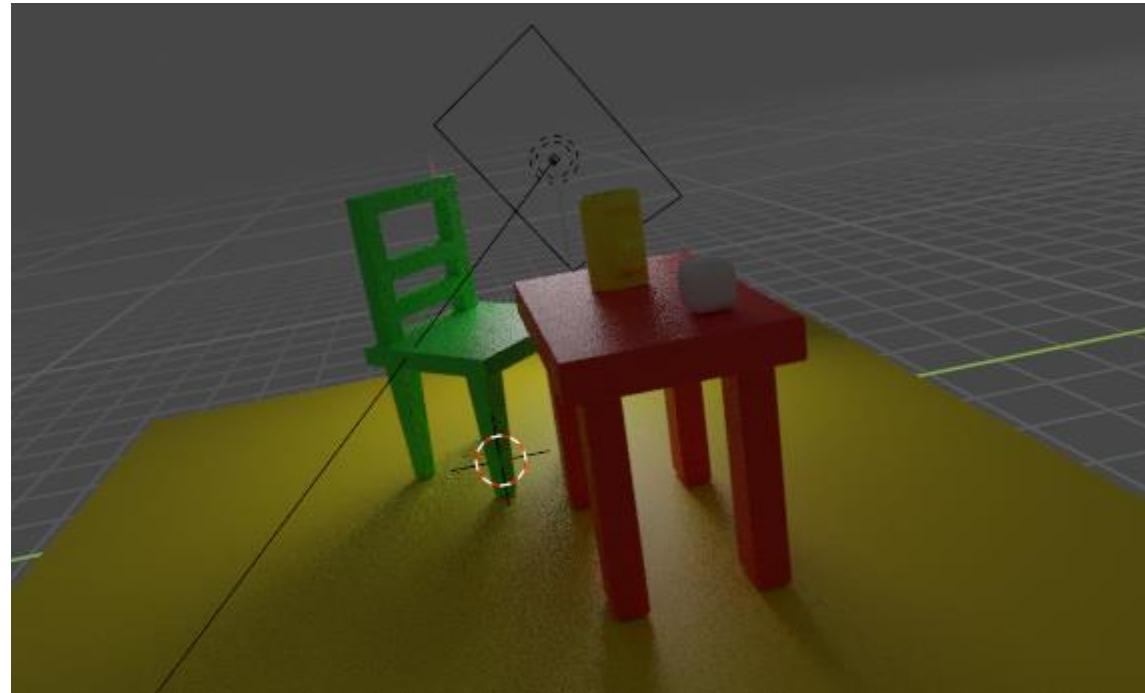


Ale...

Znacznie częściej aproksymujemy obiekt za pomocą kawałków płaszczyzn, powierzchni kuli i innych kształtów, które jest łatwo opisać matematycznie, i dobieramy składowe modelu tak, aby punkty naszego modelu były blisko odpowiednich punktów obiektu.



W drugim przypadku, gdy nie istnieje obiekt, który modelujemy, użytkownik tworzy obiekt w procesie modelowania. *Dlatego obiekt przybliża dokładnie swoją reprezentację, ponieważ stanowi jedynie jej urzeczywistnienie.



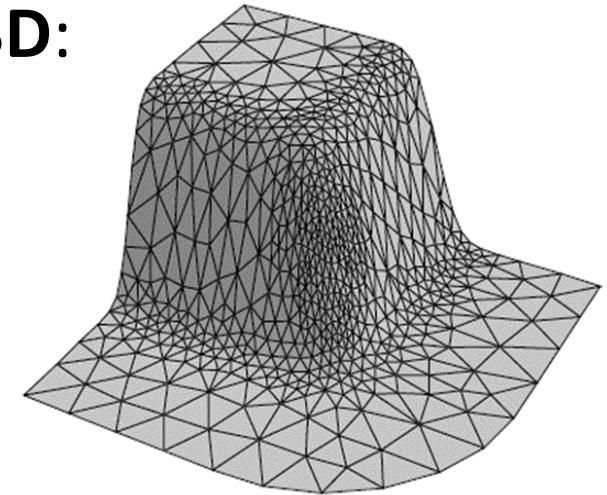
Wtedy...

W celu utworzenia obiektu użytkownik może tworzyć obiekt interakcyjnie (patrz Blender), opisać go matematycznie (np. Matlab) albo dać przybliżony opis do realizacji przez jakiś program.

Komputerowa reprezentacja może być później używana do generowania fizycznej realizacji abstrakcyjnie zaprojektowanego obiektu (obecnie chociażby druk 3D, CAD).

Trzy najpopularniejsze **reprezentacje powierzchni 3D**:

- siatki wielokątów (modele szkieletowe),
- powierzchnie parametryczne,
- powierzchnie drugiego stopnia.



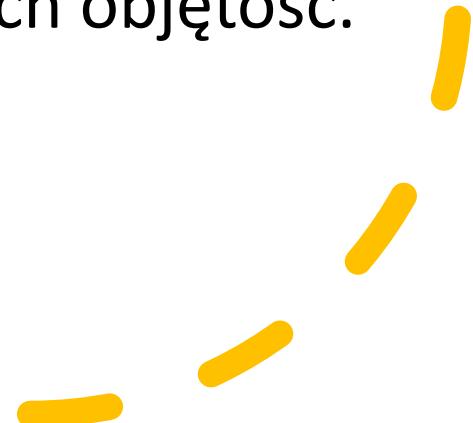
Interesujące są również krzywe parametryczne również dla tego, że powierzchnie parametryczne są uogólnieniem krzywych.

Krzywe parametryczne mogą również być podstawą kształtu powierzchni.

Modelowanie brył

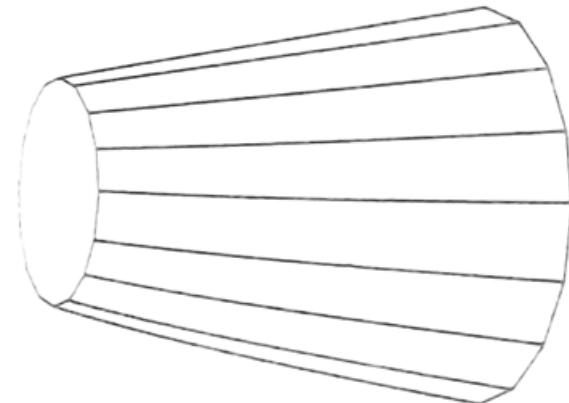
Modelowanie brył jest reprezentacją objętości całkowicie otoczonych przez powierzchnie takie, jak powierzchnia sześcianu, samolotu albo budynku.

Omawiane reprezentacje powierzchni mogą być wykorzystane w modelowaniu brył do określania powierzchni otaczających objętość.



Siatka wielokątów jest to zbiór połączonych płaskich powierzchni ograniczonych przez łamane zamknięte.

Taka reprezentacja stanowi jednak tylko przybliżenie.



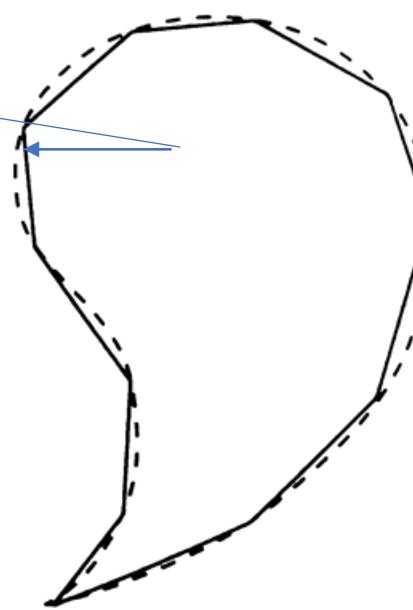
Siatki wielokątów mogą stanowić naturalną reprezentację otwartych skrzynek, pomieszczeń i otoczenia budynków, podobnie jak objętości ograniczonych przez płaskie powierzchnie.

Siatki wielokątów mogą być użyte, chociaż nie tak łatwo, do reprezentowania obiektów o powierzchniach krzywoliniowych (rys.).

Na rysunku obok pokazano przekrój kształtu krzywoliniowego i siatki wielokątowej reprezentującej ten kształt.

Możemy zmniejszać błędy aproksymacji wykorzystując większą liczbę wielokątów do utworzenia dokładniejszej liniowej aproksymacji, jednak takie podejście zwiększa wymagania co do pamięci i czas wykonywania algorytmów przetwarzania tej reprezentacji.

Jeżeli obraz zostanie powiększony, to proste krawędzie znowu staną się widoczne.



Wielomianowe krzywe parametryczne definiują punkty na krzywej 3D za pomocą trzech wielomianów z parametrem t, dla x, y i z.

Współczynniki wielomianów są tak dobierane, żeby krzywa przebiegała wzdłuż pożądanej ścieżki.

*Chociaż można wykorzystać różne stopnie wielomianów, najpopularniejszy przypadek to wielomiany trzeciego stopnia (zawierają trzecie potęgi parametru).

W odniesieniu do takich krzywych często jest używane określenie krzywa trzeciego stopnia.

Parametryczne wielomianowe płyty powierzchni z dwiema zmiennymi określają współrzędne punktów na powierzchni krzywoliniowej za pomocą trzech wielomianów dwóch zmiennych, po jednym dla x, y i z.
Brzegi płyt są parametrycznymi krzywymi wielomianowymi.

W celu otrzymania aproksymacji powierzchni krzywoliniowej o zadanej dokładności potrzebna jest znacznie mniejsza liczba płytów powierzchni wielomianowych dwóch zmiennych niż płytów wielokątowych (zresztą płaskich).

*Algorytmy potrzebne w przypadku korzystania z wielomianów z dwiema zmiennymi są jednak bardziej złożone niż w przypadku korzystania z wielokątów.

Podobnie jak w przypadku krzywych można korzystać z wielomianów różnych stopni, niemniej najpopularniejsze są wielomiany trzeciego stopnia.

Powierzchnie takie (3 stopnia) są nazywane powierzchniami bikubicznymi.

Z kolei powierzchnie drugiego stopnia są to powierzchnie zdefiniowane bezpośrednio równaniami $f(x, y, z) = 0$, przy czym f jest wielomianem drugiego stopnia zmiennych x, y i z .

Powierzchnie drugiego stopnia są wygodną reprezentacją kuli, elipsoidy i walca.

Reprezentacje te mogą być wykorzystane w systemach do reprezentowania nie tylko powierzchni, ale również ograniczonych (pełnych) objętości.

Modele szkieletowe 3D

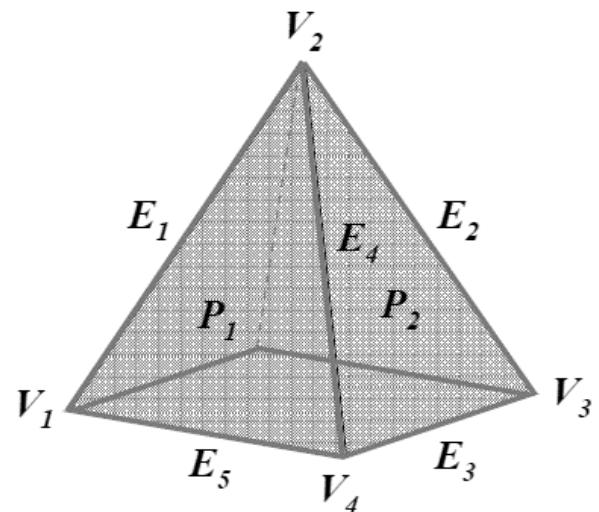
Ogólny model szkieletowy

Model szkieletowy - zbiór wierzchołków, krawędzi i ścian (wieloboków) połączonych tak, że każda krawędź jest wspólna przynajmniej dla dwóch ścian.

V_1, V_2, \dots - *wierzchołki (vertex)*

E_1, E_2, \dots - *krawędzie (edge)*

P_1, P_2, \dots - *ściany (poligon surface)*



*patrz np. Blender

Stawiane w grafice pytania:

- Jak zbudować model szkieletowy ?
- Jak efektywnie zapisać zbudowany model ?

Metody budowy modeli szkieletowych:

- opis modelowanej powierzchni w postaci równania matematycznego i jego wizualizacja,
- interakcyjne sposoby tworzenia modelu wykorzystujące realny obiekt w postaci bryły, urządzenie do wyznaczania współrzędnych punktu w przestrzeni (skaner 3-D) i odpowiednie algorytmy, łączące uzyskane w wyniku skanowania punkty krawędziami.

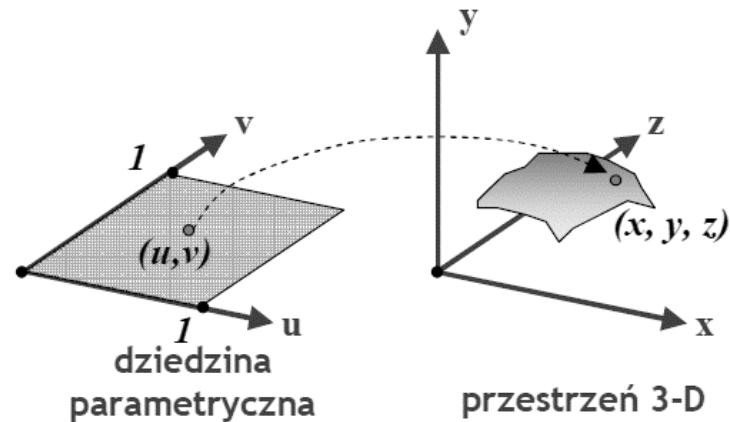
Przetwarzanie opisu zadanego w postaci równania

Modele najczęściej przetwarzane są w równania powierzchni zapisane w postaci parametrycznej.

$$x = f_x(u, v)$$

$$y = f_y(u, v) \quad 0 \leq u, v \leq 1$$

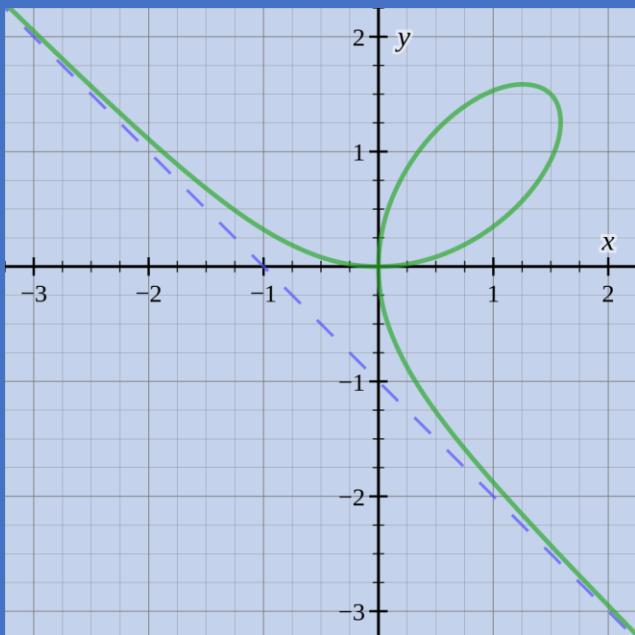
$$z = f_z(u, v)$$



Budowa modelu szkieletowego powierzchni polega na:

- podziale dziedziny parametrycznej na wieloboki,
- obliczeniu dla wierzchołków wieloboków punktów w 3-D,
- połączeniu wyliczonych punktów w wieloboki w 3-D.

Liść Kartezjusza



Przykład krzywej parametrycznej (dla wyobrażenia wykorzystania parametru do wyznaczania kolejnych wartości (x,y)).

Tych wyznaczonych tu punktów jest t (gdzie zbiór t może być ograniczony).

Liść Kartezjusza – płaska krzywa geometryczna trzeciego stopnia opisana równaniem:

$$x^3 + y^3 = 3axy$$

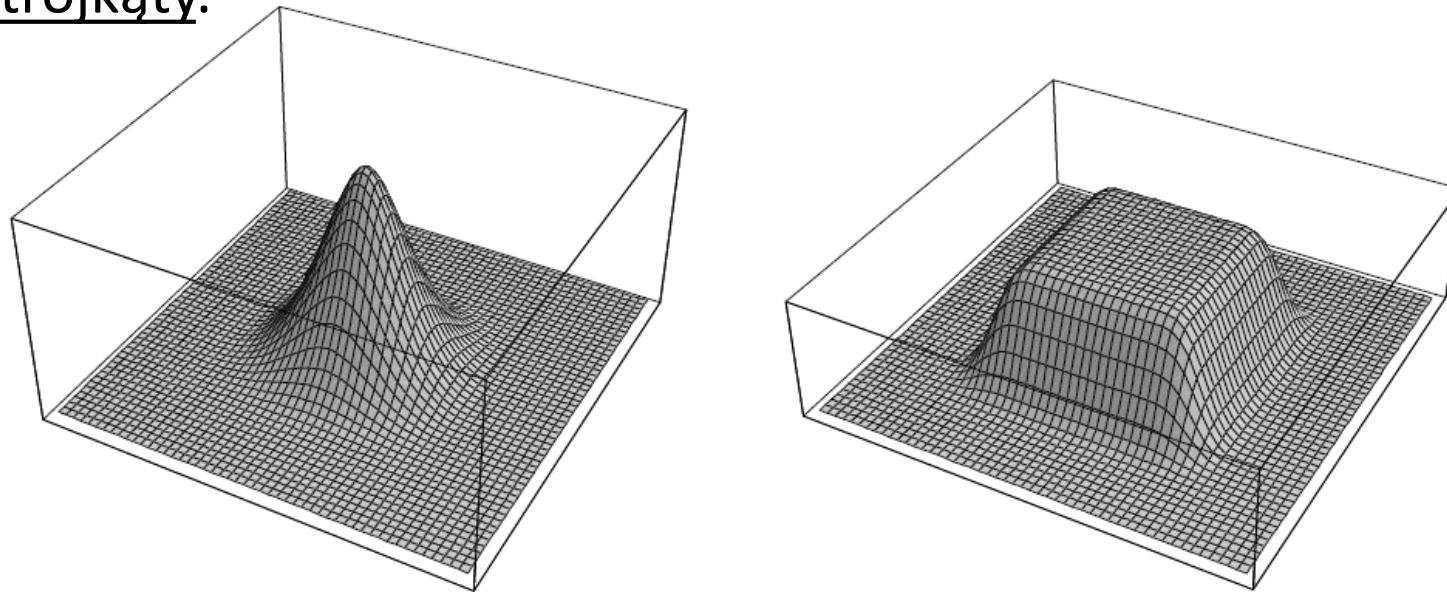
gdzie $a > 0$

Krzywą tę można również opisać równaniami parametrycznymi:

$$x = 3at / 1+t^3, \quad y = 3at^2 / 1+t^3$$

Jednorodny podział dziedziny parametrycznej

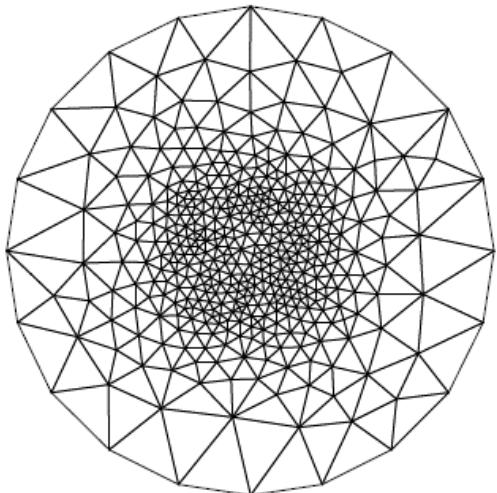
Dziedzina parametryczna jest dzielona na takie same wieloboki np. prostokąty, kwadraty lub trójkąty.



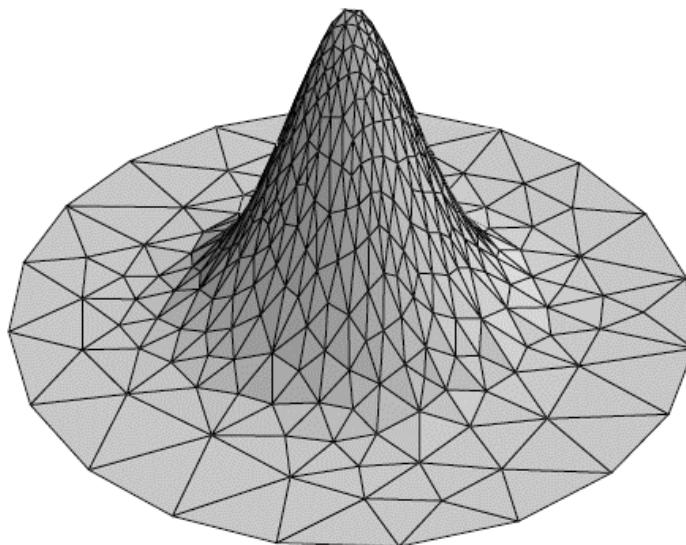
Przykład uzyskany przy podziale dziedziny parametrycznej na równe kwadraty.

Niejednorodny podział dziedziny parametrycznej (triangulacja niejednorodna)

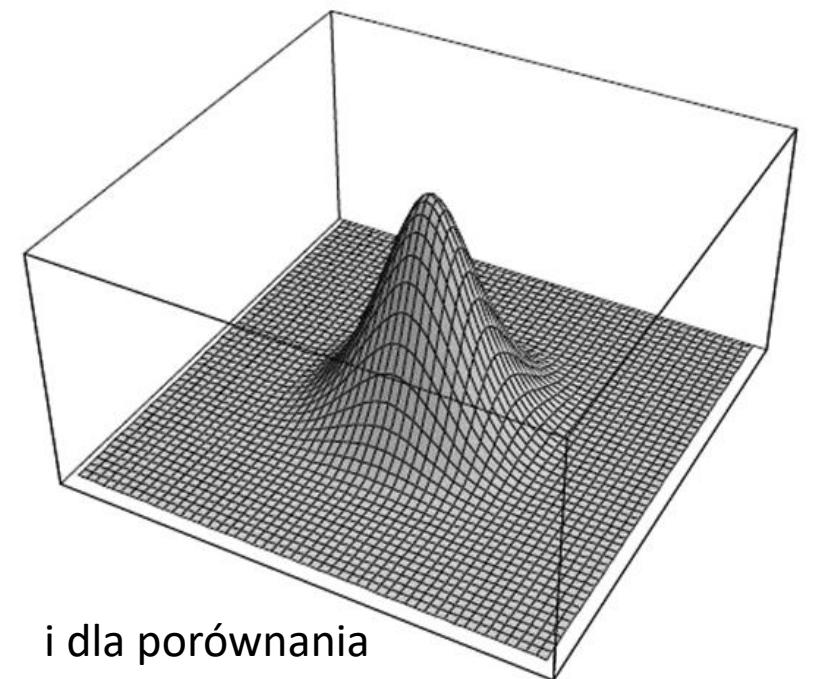
Dziedzina parametryczna jest dzielona według algorytmu na trójkąty o różnej wielkości.



podział dziedziny parametrycznej

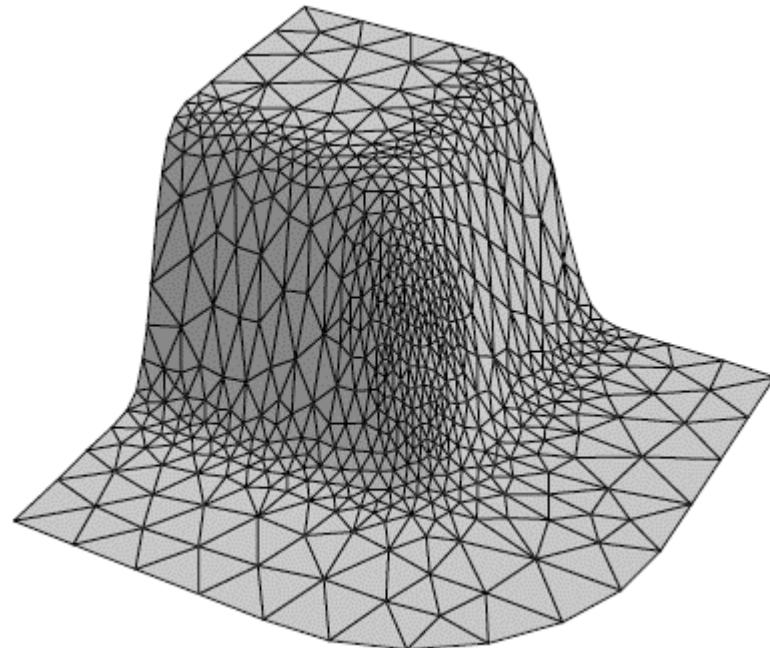
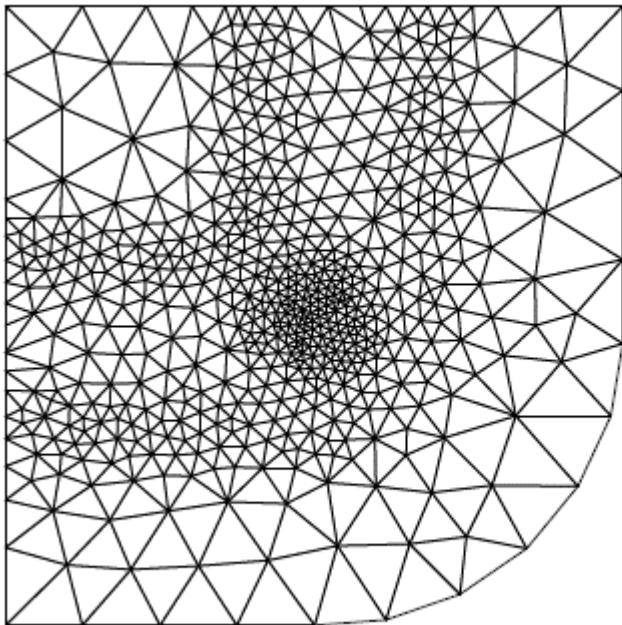


model szkieletowy powierzchni



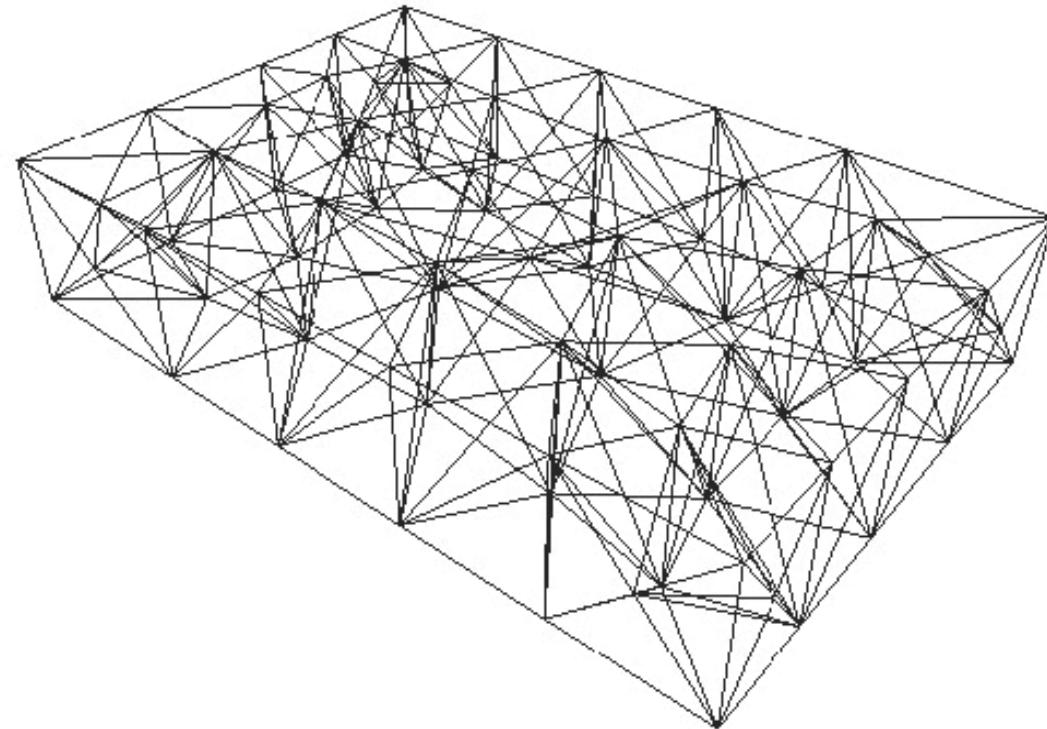
i dla porównania
podział jednorodny

Triangulacja niejednorodna pozwala na uzyskanie modelu złożonego ze znacznie mniejszej liczby trójkątów niż podział jednorodny, przy zachowaniu tej samej dokładności*.



*modele wykonał dr Jarosław Sugier

„Triangulacja” 3-D – bryła zapisywana jest jako sieć czworościanów (figura 3D, której boki są trójkątami)



Metoda elementów skończonych (Finite Element Analysis FEA)

Skanowanie powierzchni istniejącego obiektu

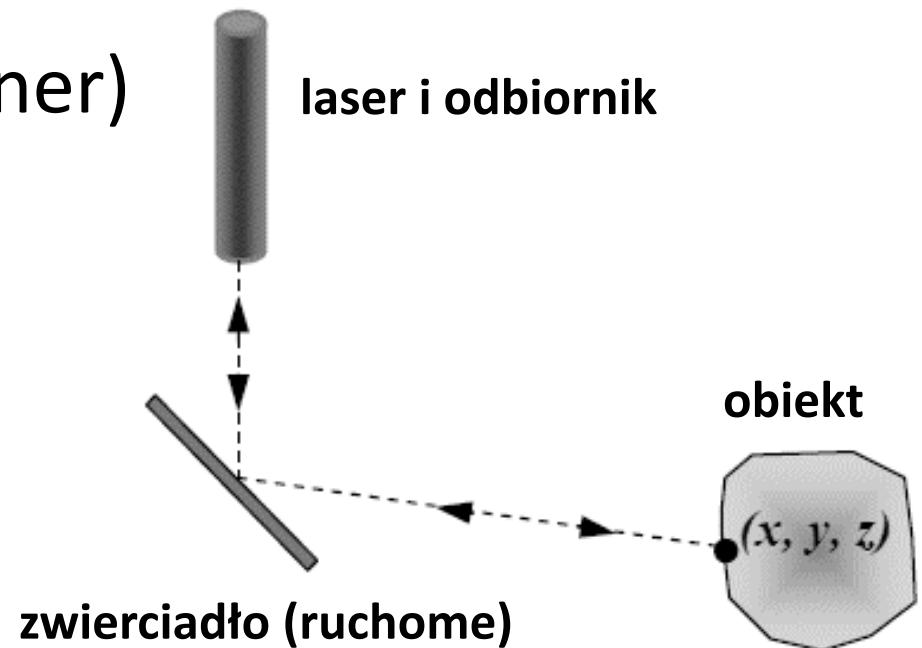
Skanery laserowe:

a) skanery z dalmierzem (ranging scanner)

Przez laser wysyłany jest impuls świetlny.

Mierzy się czas od wysłania impulsu do zarejestrowania sygnału odbitego od obiektu w odbiorniku.

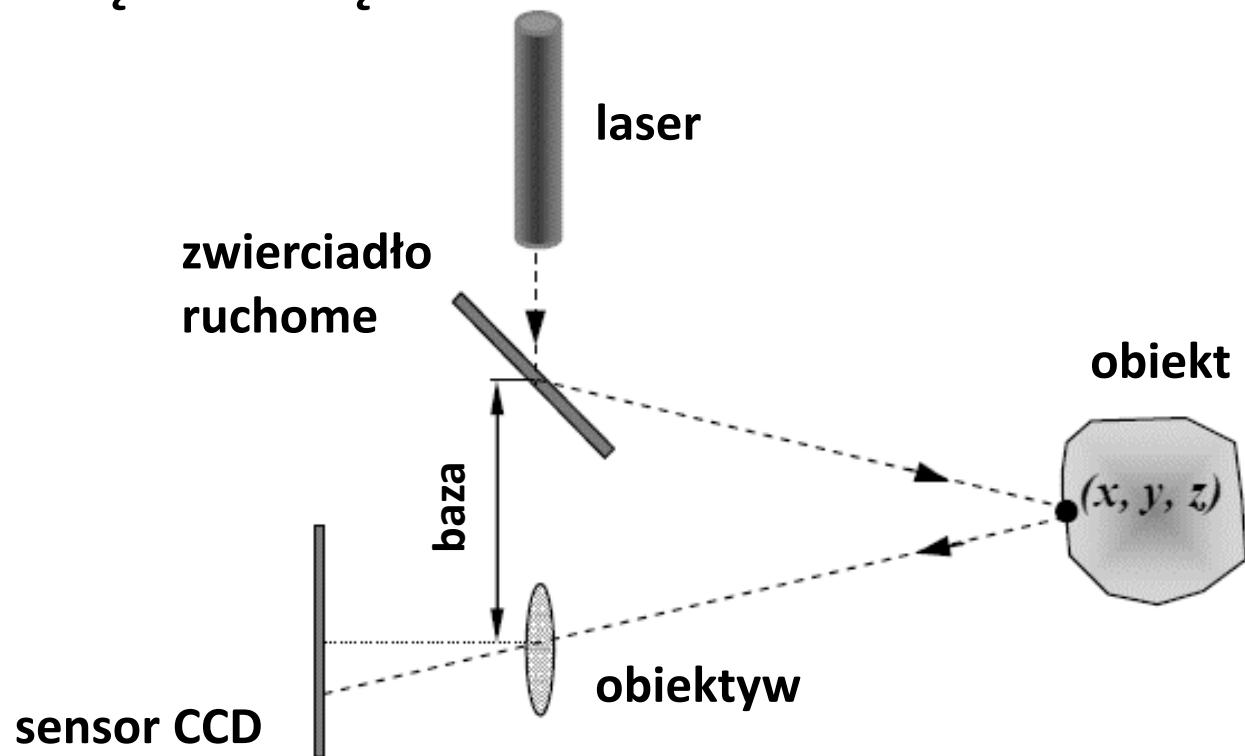
Pozwala to na wyznaczenie odległości badanego punktu od zwierciadła. Znając aktualne położenie zwierciadła, można wyliczyć (x, y, z) .



b) skanery triangulacyjne z jedną kamerą

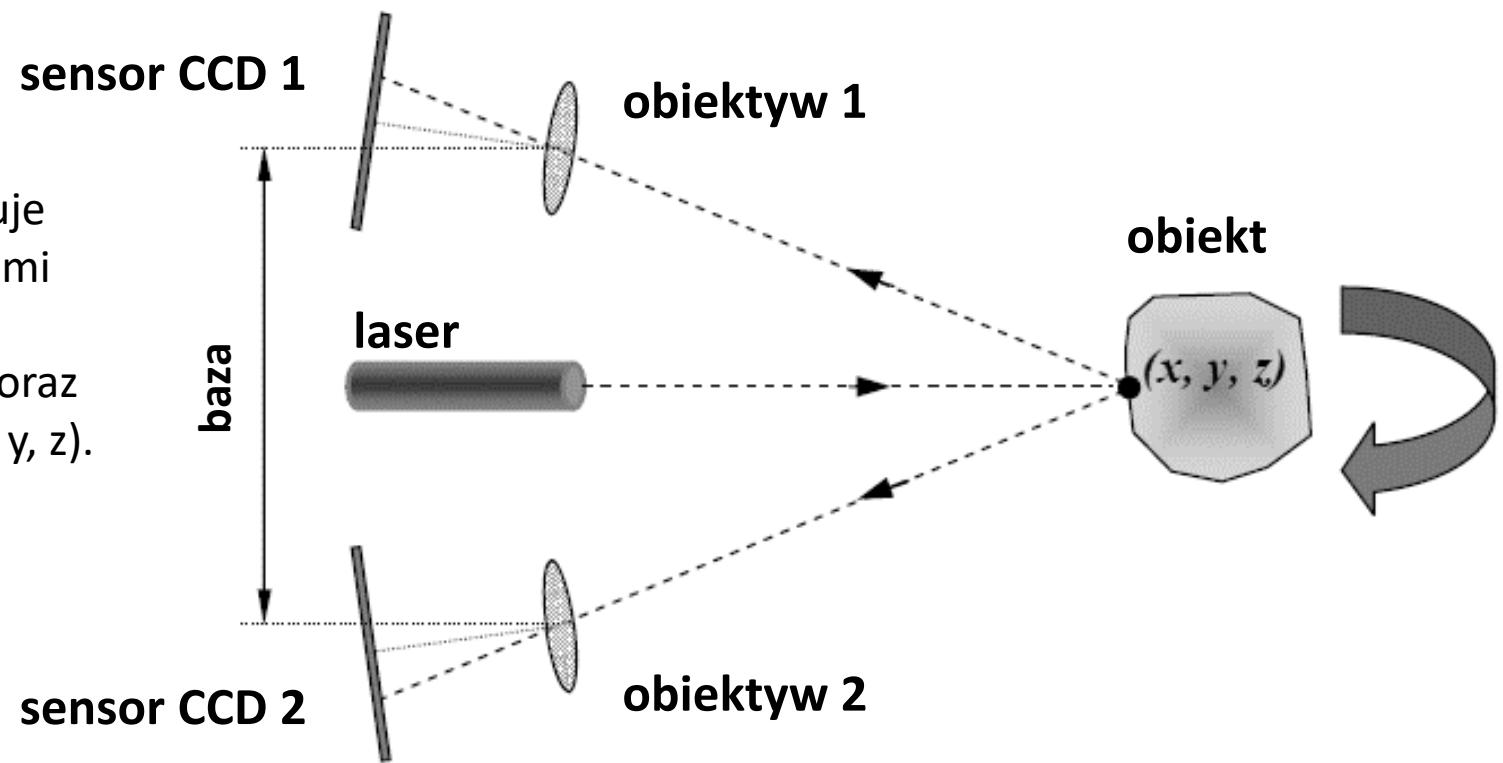
Laser wysyła wiązkę światła. Mierzone jest odchylenie obrazu oświetlonego punktu od osi sensora CCD.

Przy znajomości położenia zwierciadła i bazy (jest zawsze stała) można wyliczyć (x, y, z) .



c) skanery triangulacyjne z dwiema kamerami

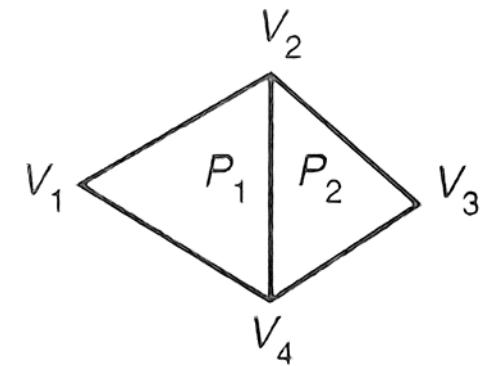
Laser wysyła wiązkę światła. Rejestruje się jednocześnie dwie klatki z obrazami mierzonego punktu. Przesunięcie obrazów punktu pomiędzy klatkami oraz znajomość bazy pozwala wyliczyć (x, y, z) .



...wracając do tematu

Siatka wielokątowa (model szkieletowy) jest zbiorem krawędzi, wierzchołków i wielokątów tak połączonych, że każda krawędź jest wspólna przynajmniej dla dwóch wielokątów.

Krawędź łączy dwa wierzchołki, a wielokąt jest zamkniętą sekwencją krawędzi. Krawędź może być wspólna dla dwóch sąsiednich wielokątów, wierzchołek jest wspólny dla przynajmniej dwóch krawędzi i każda krawędź jest częścią jakiegoś wielokąta.



Siatka wielokątowa może być reprezentowana na kilka sposobów, z których każdy ma swoje zalety i wady.

Zadaniem programisty piszącego program użytkowy jest wybranie najlepszej reprezentacji. W jednym programie można wykorzystać kilka reprezentacji: jedną dla pamięci zewnętrznej, inną dla potrzeb wewnętrznych i jeszcze inną, za pomocą której użytkownik interakcyjnie tworzy siatkę.

W celu porównania różnych reprezentacji można użyć dwóch podstawowych kryteriów: miejsce w pamięci i czas.

Typowe operacje związane z siatką wielokątową umożliwiają znalezienie wszystkich krawędzi związanych z węzłem, wielokątów mających wspólną krawędź albo wierzchołek, wierzchołków połączonych krawędzią, krawędzi wielokąta, wyświetlenie siatki i zidentyfikowanie błędów reprezentacji (na przykład brakująca krawędź, wierzchołek albo wielokąt).

Ogólnie, im bardziej bezpośrednio są reprezentowane relacje między wielokątami, wierzchołkami i krawędziami, tym operacje są szybsze i tym więcej miejsca w pamięci ta reprezentacja potrzebuje.

Reprezentowanie siatek wielokątowych

Można wymienić trzy reprezentacje siatek wielokątowych (modeli szkieletowych):

- bezpośrednią,
- wskaźników na listę wierzchołków,
- wskaźników na listę krawędzi.

W reprezentacji bezpośredniej każdy wielokąt jest opisany przez listę współrzędnych wierzchołków:

$$P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

Wierzchołki są zapamiętane w kolejności, w jakiej napotkaliśmy je posuwając się wokół wielokąta.

Między kolejnymi wierzchołkami na liście i między pierwszym a ostatnim wierzchołkiem są krawędzie.

Dla jednego wielokąta taka reprezentacja jest oszczędna, jeśli chodzi o zajętość pamięci.

Dla siatki wielokątowej traci się dużo miejsca, ponieważ współrzędne wspólnych wierzchołków są powielane.

Jeszcze większym problemem jest to, że nie ma bezpośredniej reprezentacji dla wspólnych krawędzi i wierzchołków.

Na przykład, w celu interakcyjnego przesunięcia wierzchołka ze wszystkimi jego krawędziami, musimy znaleźć wszystkie wielokąty wspólne dla wierzchołka. Takie przeszukanie wymaga porównania trójkę współrzędnych jednego wielokąta z trójkami współrzędnych innych wielokątów.

Najefektywniejszą metodą wykonania tego byłoby posortowanie wszystkich N trójk wstępnych.

Ten proces jest w najlepszym przypadku o złożoności $N \log_2 N$, a nawet wtedy istnieje niebezpieczeństwo, że ten sam wierzchołek mógłby, ze względu na niedokładności obliczeń, mieć nieco inne wartości wstępnych w każdym wielokącie i poprawnego dopasowania można by nigdy nie znaleźć.

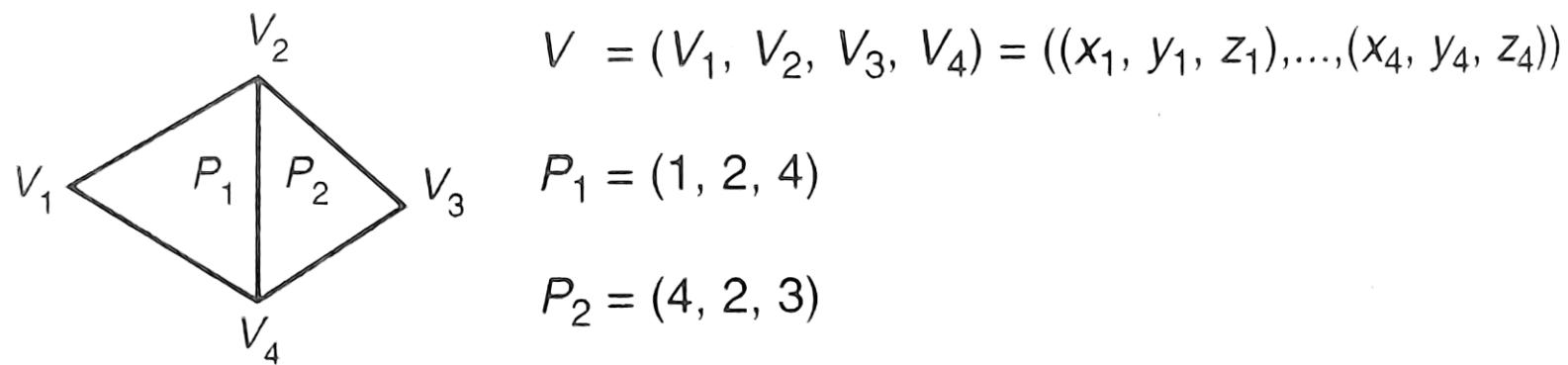
Przy takiej reprezentacji wyświetlanie siatki albo jako wypełnionych wielokątów, albo jako obwodów wielokątów wymaga przekształcenia każdego wierzchołka i obcięcia każdej krawędzi każdego wielokąta.

W czasie rysowania krawędzi każda wspólna krawędź jest rysowana dwukrotnie, co stwarza problem np. w przypadku algorytmów rastrowych wtedy, kiedy krawędzie są rysowane w przeciwnych kierunkach i mogą w efekcie zostać wyświetcone dodatkowe piksele.

Reprezentacja bezpośrednia – podsumowanie:

- wierzchołki na listach ścian powtarzają się,
- modyfikacja opisu jest trudna (przesunięcie wierzchołka wymaga znalezienia wszystkich ścian wspólnych dla wierzchołka i modyfikacji odpowiedniej trójki współrzędnych),
- przy rysowaniu krawędzie wspólne dla dwóch ścian będą rysowane dwukrotnie.

Wielokąty zdefiniowane za pomocą **wskaźników na listę wierzchołków** mają każdy wierzchołek siatki wielokątowej zapamiętany tylko raz na liście wierzchołków $V = ((x, y_1, z_1), \dots (x_n, y_n, z_n))$. Wielokąt P jest zdefiniowany jako lista wskaźników na listę wierzchołków. Wielokąt zawierający wierzchołki 3, 5, 7 i 10 na liście wierzchołków byłby więc reprezentowany jako $P=(3, 5, 7, 10)$.



Przykład takiej reprezentacji jaką opisano na poprzednim slajdzie ma kilka zalet w porównaniu z bezpośrednią reprezentacją wielokątową.

Ponieważ każdy wierzchołek jest zapamiętany tylko raz, oszczędza się miejsce w pamięci. Ponadto łatwo można zmienić współrzędne wierzchołków (każdy jest tylko raz, chociaż może być użyty wiele razy).

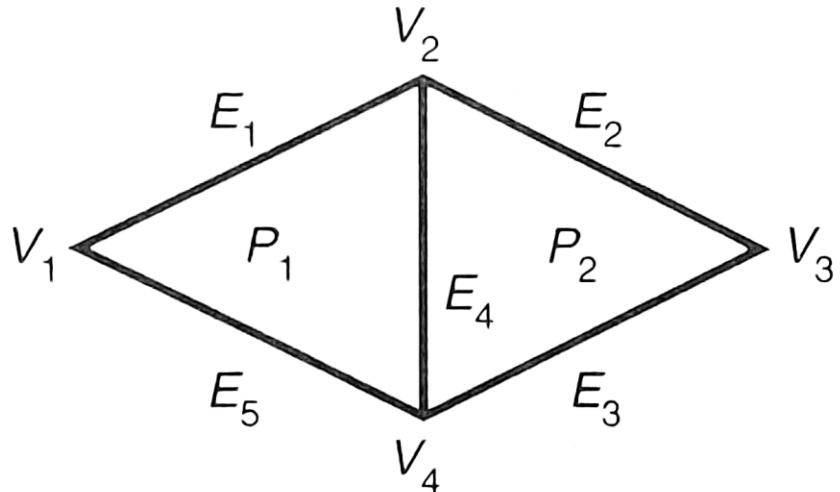
Wciąż jest trudno znaleźć wielokąty o wspólnej krawędzi i krawędzie wspólne dla wielokątów są wciąż rysowane dwukrotnie wówczas, gdy są wyświetlane wszystkie wielokąty. Możemy te dwa problemy wyeliminować, jeżeli krawędzie będą reprezentowane bezpośrednio, tak jak w następnej metodzie.

Wskaźniki na listę wierzchołków – podsumowanie:

- wierzchołek jest zapisywany tylko jeden raz,
- modyfikacja współrzędnych wierzchołka jest łatwa,
- trudno znaleźć ściany o wspólnej krawędzi,
- przy rysowaniu krawędzie wspólne dla dwóch ścian będą rysowane dwukrotnie,
- trudno wypełniać obrazy ścian.

Jeżeli wielokąty definiujemy przez **wskaźniki na listę krawędzi**, to znowu mamy listę wierzchołków V , ale wielokąt jest reprezentowany jako lista wskaźników z tym, że nie na listę wierzchołków, a na listę krawędzi, w której każda krawędź występuje tylko raz. Z kolei każda krawędź z listy krawędzi wskazuje dwa wierzchołki z listy wierzchołków definiujące krawędź, a także jeden albo dwa wielokąty, do których należy krawędź. Dlatego opisujemy wielokąt jako $P = (E_1, \dots, E_n)$ i krawędź jako $E = (V_1, V_2, P_1, P_2)$. Gdy krawędź należy tylko do jednego wielokąta, wówczas P_1 , albo P_2 jest zerem. Na rysunku dalej pokazano przykład takiej reprezentacji.

Wielokąty definiujemy przez wskaźniki na listę krawędzi, mamy listę wierzchołków V, ale wielokąt jest reprezentowany jako lista wskaźników z tym, że nie na listę wierzchołków, a na listę krawędzi, w której każda krawędź występuje tylko raz. Z kolei każda krawędź z listy krawędzi wskazuje dwa wierzchołki z listy wierzchołków definiujące krawędź, a także jeden albo dwa wielokąty, do których należy krawędź. Dlatego opisujemy wielokąt jako $P = (E_1, \dots, E_n)$ i krawędź jako $E = (V_1, V_2, P_1, P_2)$. Gdy krawędź należy tylko do jednego wielokąta, wówczas P_1 , albo P_2 jest zerem.



$$V = (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \dots, (x_4, y_4, z_4))$$

$$E_1 = (V_1, V_2, P_1, \lambda)$$

$$E_2 = (V_2, V_3, P_2, \lambda)$$

$$E_3 = (V_3, V_4, P_2, \lambda)$$

$$E_4 = (V_4, V_2, P_1, P_2)$$

$$E_5 = (V_4, V_1, P_1, \lambda)$$

$$P_1 = (E_1, E_4, E_5)$$

$$P_2 = (E_2, E_3, E_4)$$

Siatka wielokątowa zdefiniowana za pomocą listy krawędzi dla każdego wielokąta (Lambda reprezentuje zero)

Kontury wielokątów pokazujemy wyświetlając wszystkie krawędzie, a nie wszystkie wielokąty (unika się wielokrotnego odrysowywania krawędzi) - w ten sposób omija się nadmiarowe obcinanie, przekształcenia i konwersje.

Również łatwo wyświetla się wypełnione wielokąty.

Czasami, na przykład przy opisie struktury 3D przypominającej plaster miodu, niektóre krawędzie są wspólne dla trzech wielokątów. W takich przypadkach opis krawędzi może być tak rozszerzony, żeby zawierał dowolną liczbę wielokątów: $E = (V_1, V_2, P_1, P_2, P_3)$.

W żadnej z trzech reprezentacji (to znaczy bezpośrednich wielokątów, wskaźników na listę wierzchołków, wskaźników na listę krawędzi) nie jest łatwo określić, które krawędzie łączą się z wierzchołkiem: trzeba sprawdzić wszystkie krawędzie.

Można dodać bezpośrednią informację, która umożliwi określanie takich zależności. Na przykład można rozszerzyć opis krawędzi tak, żeby zawierał wskaźniki na krawędzie następne w opisie każdej ze ścian. Opis wierzchołka będzie zawierał wtedy wskaźnik na (dowolną) krawędź zawierającą ten wierzchołek.

Wskaźniki na listę krawędzi – podsumowanie:

- wierzchołek jest zapisywany tylko jeden raz,
- modyfikacja współrzędnych wierzchołka jest łatwa,
- trudno znaleźć ściany o wspólnej krawędzi.

Wielobok a płaszczyzna

Elementy rachunku wektorowego:

Punkt w przestrzeni 3-D będzie dalej traktowany jako wektor.

- Suma wektorów

$$\mathbf{P}_1 + \mathbf{P}_2 = [x_1 + x_2 \ y_1 + y_2 \ z_1 + z_2]$$

- Iloczyn wektora i liczby

$$a\mathbf{P}_i = [ax_i \ ay_i \ az_i]$$

- Iloczyn skalarny

$$\mathbf{P}_1 \cdot \mathbf{P}_2 = x_1x_2 + y_1y_2 + z_1z_2$$

- Iloczyn wektorowy

$$\mathbf{u}_x = [1 \ 0 \ 0] \quad \mathbf{u}_y = [0 \ 1 \ 0] \quad \mathbf{u}_z = [0 \ 0 \ 1]$$

$$\begin{aligned} \mathbf{P}_1 \times \mathbf{P}_2 &= \det \begin{bmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_z \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix} = \\ &= [y_1z_2 - z_1y_2, \quad z_1x_2 - x_1z_2, \quad x_1y_2 - y_1x_2] \end{aligned}$$

Wynik działania w sposób istotny zależy od doboru bazy przestrzeni.

W przypadku, gdy baza trójwymiarowej przestrzeni kartezjańskiej nie jest sprecyzowana, przyjmuje się bazę kanoniczną złożoną z wektorów

Równanie płaszczyzny wyznaczanej przez wierzchołki wieloboku i wektor normalny dla tej płaszczyzny

Równanie płaszczyzny to: $Ax + By + Cz + D = 0$

1. Wyznaczanie wektora normalnego $[A, B, C]$ do płaszczyzny

Dla trzech wierzchołków wieloboku P_1, P_2, P_3 obliczyć:

$$[A \ B \ C] = (P_2 - P_1) \times (P_3 - P_1)$$

Jeżeli $[A, B, C] = [0, 0, 0]$ (patrz równanie płaszczyzny) to wierzchołki są współliniowe i nie określają płaszczyzny.

Należy w takim przypadku wybrać inne wierzchołki i ponownie obliczyć wektor normalny.

2. Obliczenie współczynnika D

Obliczone wcześniej współczynniki A , B , C i współrzędne dowolnego wierzchołka wstawić do równania płaszczyzny i wyliczyć D .

Dla różnych trójków wierzchołków można otrzymać różne równania płaszczyzny w przypadku gdy wielobok nie jest płaski.

Abytrzymać wielobok lub układ wieloboków płaskich można:

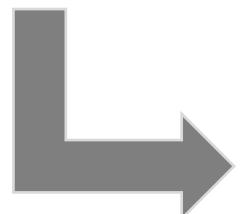
1. Podzielić wielobok na mniejsze wieloboki płaskie.
2. Zmodyfikować współrzędne wierzchołków tak, aby nowe wierzchołki leżały na płaszczyźnie możliwie najbliższej (w sensie pewnego kryterium) wierzchołkom przed modyfikacją.

Postać uwikłana równania powierzchni:

$$f(x, y, z) = 0$$

Powierzchnie drugiego stopnia (kwadryki):

$$\begin{aligned} f(x, y, z) &= Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + \\ &+ 2Fxz + 2Gx + 2Hy + 2Jz + K = 0 \end{aligned}$$



$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$

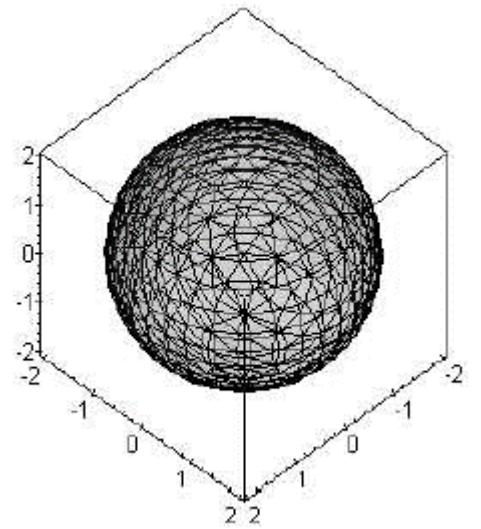
Przykłady:

1. Płaszczyzna: istnieje dla $A = B = C = \dots = F = 0$

$$f(x, y, z) = 2Gx + 2Hy + 2Jz + K = 0$$

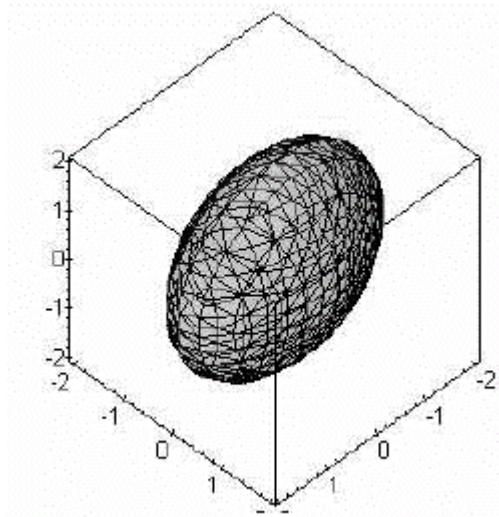
2. Sfera:

$$f(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$$



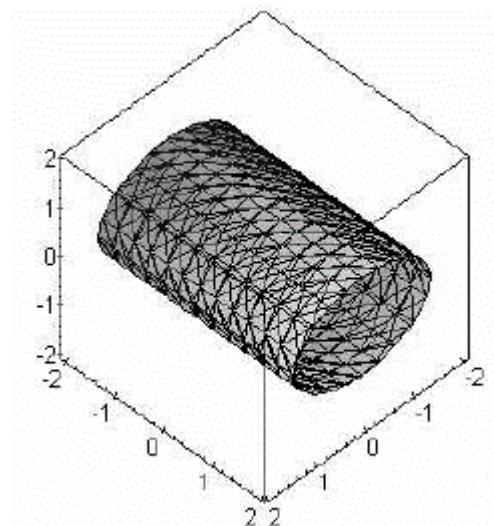
3. Elipsoida

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$



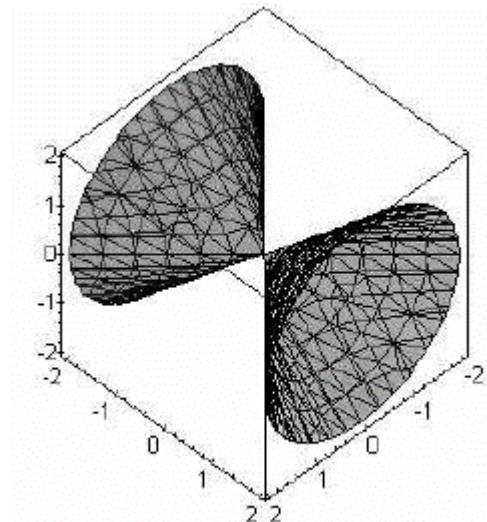
4. Cylinder

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$$



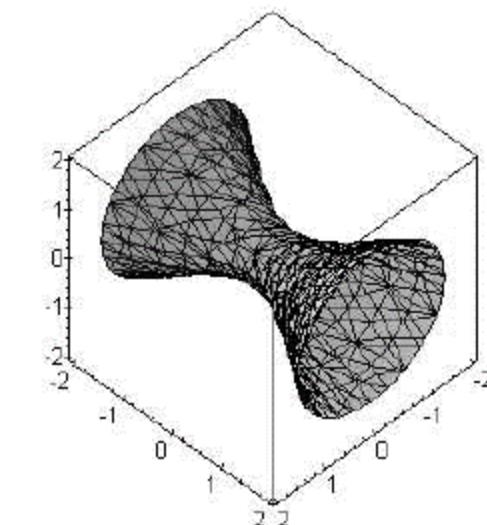
5. Stożek

$$f(x, y, z) = \frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$$



6. Hiperboloida

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} - 1 = 0$$



Zalety stosowania kwadryk w grafice komputerowej:

- Łatwe obliczanie wektora normalnego do powierzchni

$$\vec{N} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix}$$

Wektor normalny można wyznaczyć analitycznie.

- Łatwe obliczanie punktów przecięcia powierzchni z prostą (ważne w algorytmach metody śledzenia promieni).
- Łatwe testowanie czy dany punkt leży na powierzchni (podstawić współrzędne punktu do równania i sprawdzić czy wynik jest bliski zera).
- Łatwe obliczanie z dla danych x, y (ważne w algorytmach realizujących usuwanie powierzchni niewidocznych)
- Stosunkowo łatwe obliczanie krzywej przecięcia jednej kwadryki z drugą.

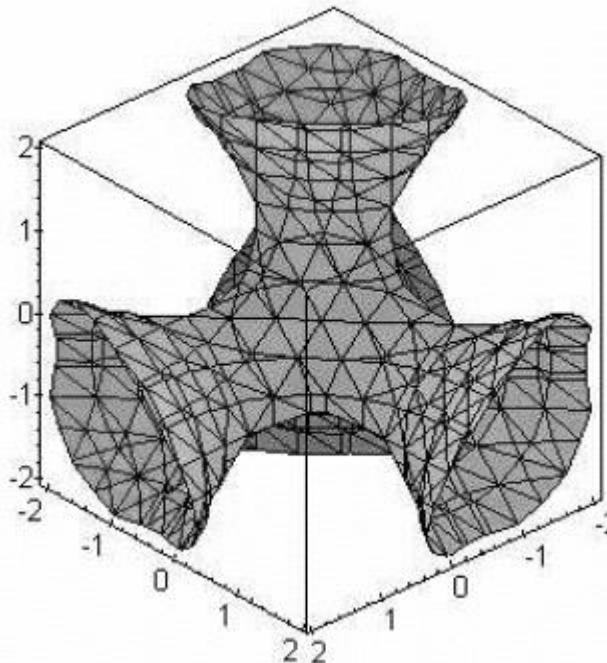
Wady opisu w postaci kwadryk:

- Trudna generacja punktu leżącego na powierzchni
- Trudna generacja fragmentu powierzchni np. połowy czy ćwiartki sfery

Powierzchnia opisane równaniami innymi niż kwadryki

Powierzchnia trzeciego stopnia

$$f(x, y, z) = x^3 + y^3 + z^3 - (x + y + z)^3 = 0$$



Powierzchnie opisane parametrycznie

Powierzchnia opisana parametrycznie

$$x(u,v) = f_x(u,v)$$

$$y(u,v) = f_y(u,v)$$

$$z(u,v) = f_z(u,v)$$

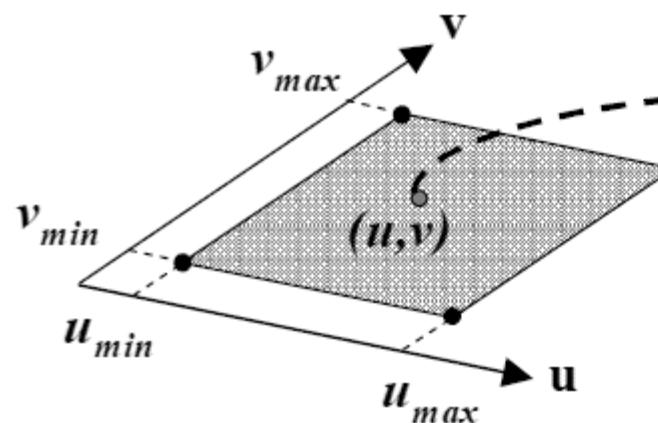
$$x = f_x(u,v)$$

$$y = f_y(u,v) \quad 0 \leq u, v \leq 1$$

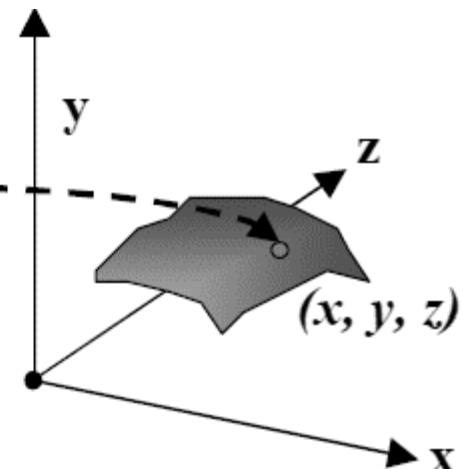
$$z = f_z(u,v)$$

$$u_{min} \leq u \leq u_{max}$$

$$v_{min} \leq v \leq v_{max}$$



dziedzina
parametryczna



przestrzeń 3-D

Przykład:

Sfera o promieniu r i środku $(0, 0, 0)$

$$\begin{aligned}x(u, v) &= r \sin(\pi u) \cos(2\pi v) & 0 \leq u \leq 1 \\y(u, v) &= r \sin(\pi u) \sin(2\pi v) & 0 \leq v \leq 1 \\z(u, v) &= r \cos(\pi u)\end{aligned}$$

- Łatwa generacja punktu leżącego na powierzchni (dla pary parametrów u, v łatwo jest wyznaczyć $x(u, v), y(u, v), z(u, v)$).
- Łatwa generacja fragmentu powierzchni np. prowadzimy obliczenia punktów dla przedziałów $0 \leq u \leq \frac{1}{2}, 0 \leq v \leq \frac{1}{2}$ i uzyskujemy tylko część powierzchni sfery.

Przykład - powierzchnia o równaniu:

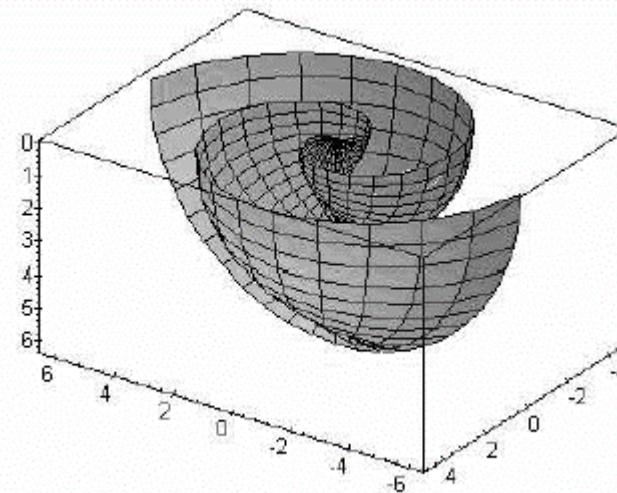
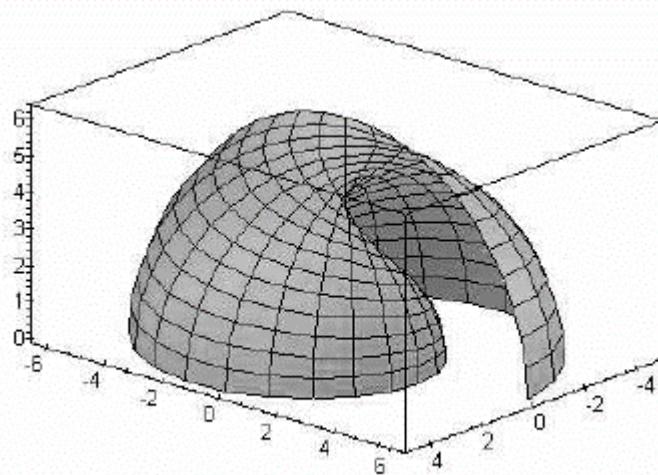
$$x(u, v) = u \sin(u) \cos(v)$$

$$y(u, v) = u \cos(u) \cos(v)$$

$$z(u, v) = u \sin(v)$$

$$0 \leq u \leq 2\pi$$

$$0 \leq v \leq \pi$$



Krzywe parametryczne w 3-D

Równania parametryczne krzywej:

$$x(u) = f_x(u)$$

$$y(u) = f_y(u) \quad u_{min} \leq u \leq u_{max}$$

$$z(u) = f_z(u)$$

Przykład:

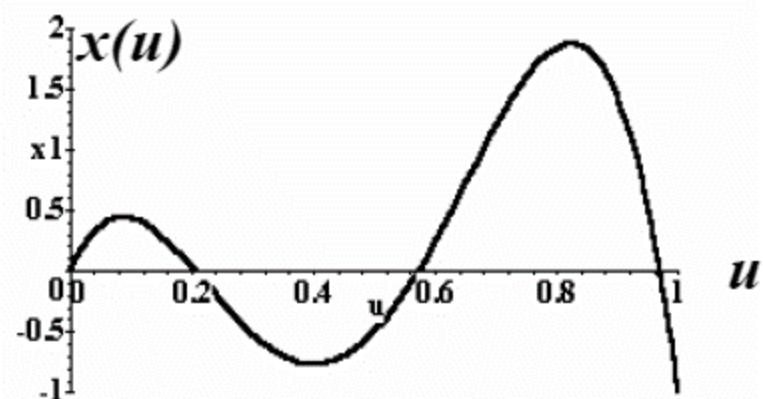
Krzywa opisana wielomianami 4-go stopnia.

$$x(u) = -100u^4 + 175u^3 - \frac{175}{2}u^2 + \frac{23}{2}u$$

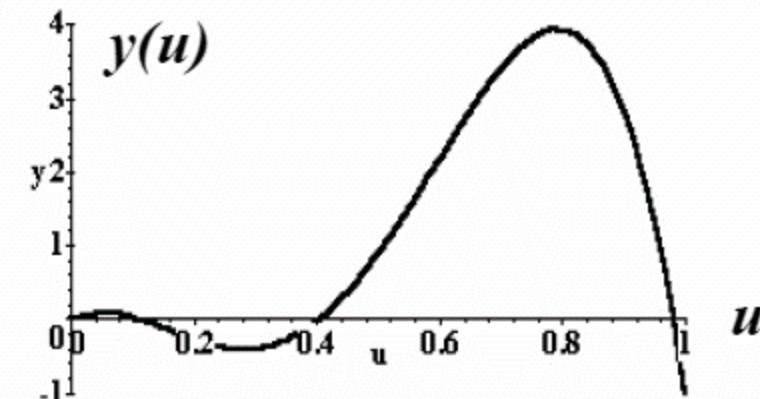
$$y(u) = -100u^4 + 150u^3 - \frac{500}{9}u^2 + \frac{41}{9}u \quad 0 \leq u \leq 1$$

$$z(u) = 0$$

Każda ze współrzędnych wyznaczana jest za pomocą odrębnej funkcji dla tego samego u. Czyli miejsce na krzywej jest określone przez u.



$$x(u) = f_x(u)$$

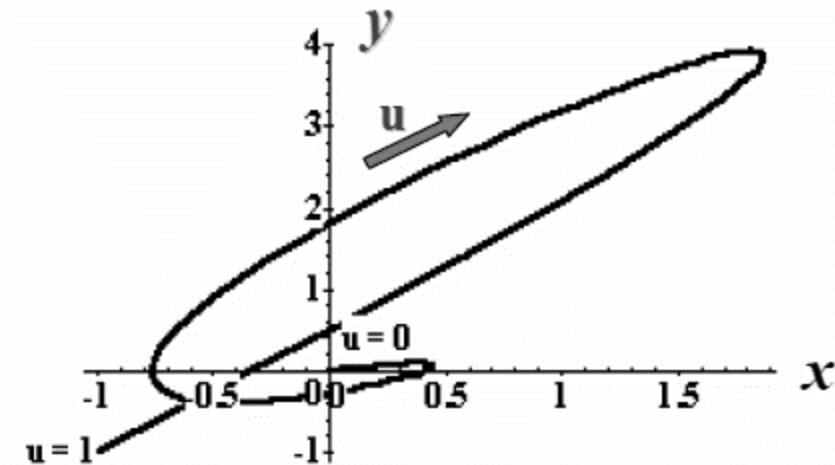


$$y(u) = f_y(u)$$

$$x(u) = f_x(u)$$

$$y(u) = f_y(u) \quad 0 \leq u \leq 1$$

$$z(u) = 0$$



Wynikowa krzywa (pierwszy element krzywej $\rightarrow u=1$, $z=0 \rightarrow$ brak zmian w osi z)

Parametryczne krzywe trzeciego stopnia

Krzywa zadana jest równaniem:

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T$$

gdzie:

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad 0 \leq u \leq 1$$

$$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$$

Jeżeli podstawić:

$$U = [u^3 \quad u^2 \quad u \quad 1]^T$$

oraz

$$C = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

to

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T = C \cdot U$$

Jednoznaczne określenie krzywej wymaga podania wartości 12 parametrów.

...krzywe będą kontynuowane

Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Krzywe parametryczne cd

Edward Półrolniczak, dr inż.,
Katedra Systemów Multimedialnych,
Wydział Informatyki, ZUT w Szczecinie

Grafika i wizualizacja

Zagadnienia

- Krzywa Hermite'a
- Krzywa Bezier'a
- Krzywe sklejane

Krzywe Hermite'a

Krzywe parametryczne trzeciego stopnia (w 3D)

Krzywa zadana jest równaniem:

$$Q(u) = [\begin{array}{ccc} x(u) & y(u) & z(u) \end{array}]^T$$

gdzie

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad 0 \leq u \leq 1$$

$$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$$

Jeżeli sformułujemy wektor U:

$$\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1]^T$$

oraz macierz C współczynników:

$$\mathbf{C} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

//przypomnienie:

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad 0 \leq u \leq 1$$

$$z(u) = a_z u^3 + b_z u^2 + c_z u + d_z$$

To Q(u) można zapisać zwięźle w formie iloczynu macierzy i wektora:

$$Q(u) = [\ x(u) \ y(u) \ z(u)]^T = \mathbf{C} \cdot \mathbf{U}$$

Należy zauważyć, że krzywą taką definiuje 12 współczynników (zebranych tutaj w macierzy C):

$$Q(u) = [x(u) \ y(u) \ z(u)]^T = C \cdot U$$


$$C = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

Staje się to przyczynkiem do poszukiwania innych sposobów na opisywanie krzywych w 3D.

Krzywa Hermite'a

Charles Hermite (ur. 24 grudnia 1822, zm. 14 stycznia 1901) – matematyk francuski.

W swych pracach zajmował się teorią liczb, algebra i analizą matematyczną. Jako pierwszy dowódł, że liczba e jest liczbą przestępna. Jego prace wykorzystał potem Ferdinand Lindemann przy dowodzeniu, że liczba π jest również liczbą przestępna. Takie pojęcia matematyczne jak Wielomiany Hermite'a czy Sprzężenie hermitowskie są nazwane na jego cześć.

Jego uczniem był inny znany matematyk,
Henri Poincaré.

* Źródło: Wikipedia

(czyt. pi), *Iudolfinia*, stała Archimedesa – stosunek obwodu koła (czyli długości okręgu) do długości jego średnicy; stosunek ten jest niezależny od wyboru koła, bowiem każde dwa koła są podobne.

Krzywa Hermite'a określona jest przez cztery składowe:

- dwa punkty krzywej (początkowy i końcowy),
- dwa wektory styczne do krzywej (pochodne) w tych punktach.





W przypadku krzywej Hermite'a opis krzywej 3D wprowadzony na poprzednich slajdach przybiera postać:

$$Q(u) = C \cdot U = G \cdot M \cdot U$$

Przy czym:

$$Q(u) = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} \underbrace{\begin{bmatrix} m_{11} & m_{21} & m_{31} & m_{41} \\ m_{12} & m_{22} & m_{32} & m_{42} \\ m_{13} & m_{23} & m_{33} & m_{43} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix}}_{M} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

gdzie

M - macierz bazowa Hermite'a

G - macierz geometrii (zawiera wymienione składowe definiujące krzywą)



G - macierz geometrii (zawiera wymienione składowe definiujące krzywą: punkty i styczne)

$$G = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} = \begin{bmatrix} P_{0x} & P_{1x} & R_{0x} & R_{1x} \\ P_{0y} & P_{1y} & R_{0y} & R_{1y} \\ P_{0z} & P_{1z} & R_{0z} & R_{1z} \end{bmatrix}$$

gdzie:

współrzędne punktu początkowego i końcowego:

$$(P_{0x}, P_{0y}, P_{0z}), (P_{1x}, P_{1y}, P_{1z})$$

wektory styczne (pochodne) w punktach, początkowym i końcowym:

$$(R_{0x}, R_{0y}, R_{0z}), (R_{1x}, R_{1y}, R_{1z})$$

*w obrębie jednej macierzy znajdują się wartości określające punkty i wektory



Macierz M jest wyznaczona następująco – zauważmy, że np. :

$$\mathbf{x}(u) = a_x u^3 + b_x u^2 + c_x u + d_x = \mathbf{G}_x \cdot M \cdot [u^3 \ u^2 \ u \ 1]^T$$

Gdzie:

$$\mathbf{G}_x = [P_{0x} \quad P_{1x} \quad R_{0x} \quad R_{1x}]$$

Jeśli tak to styczna: $\mathbf{x}'(u) = \mathbf{G}_x \cdot M \cdot [3u^2 \ 2u \ 1 \ 0]^T$

Uwzględniając nałożone wcześniej ograniczenia uzyskuje się:

$$\left. \begin{array}{lcl} \mathbf{x}(0) & = & P_{0x} \\ \mathbf{x}(1) & = & P_{1x} \\ \mathbf{x}'(0) & = & R_{0x} \\ \mathbf{x}'(1) & = & R_{1x} \end{array} \right\} \begin{array}{l} * \text{początek i koniec} \\ * \text{styczne} \end{array} \quad \left. \begin{array}{l} \mathbf{G}_x \cdot M \cdot [0 \ 0 \ 0 \ 1]^T \\ \mathbf{G}_x \cdot M \cdot [1 \ 1 \ 1 \ 1]^T \\ \mathbf{G}_x \cdot M \cdot [0 \ 0 \ 1 \ 0]^T \\ \mathbf{G}_x \cdot M \cdot [3 \ 2 \ 1 \ 0]^T \end{array} \right\} * \text{wzdłuż współrzędnej } x$$



Sformułowane wyżej cztery równania, zapisane w postaci macierzowej, przyjmują formę:

$$[P_{0x} \quad P_{1x} \quad R_{0x} \quad R_{1x}] = G_x = G_x \cdot M \cdot \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Aby równanie było spełnione musi zachodzić:

$$M = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$



Kompletne równanie krzywej Hermite'a (czyli dla x, y i z) można zapisać następująco:

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T = G \cdot M \cdot U$$

Co można rozwinać jako:

$$\begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \begin{bmatrix} P_{0x} \\ P_{0y} \\ P_{0z} \end{bmatrix} (2u^3 - 3u^2 + 1) + \begin{bmatrix} P_{1x} \\ P_{1y} \\ P_{1z} \end{bmatrix} (-2u^3 + 3u^2) +$$

$$+ \begin{bmatrix} R_{0x} \\ R_{0y} \\ R_{0z} \end{bmatrix} (u^3 - 2u^2 + u) + \begin{bmatrix} R_{1x} \\ R_{1y} \\ R_{1z} \end{bmatrix} (u^3 - u^2)$$

Wielomiany – funkcje bazowe Hermite'a



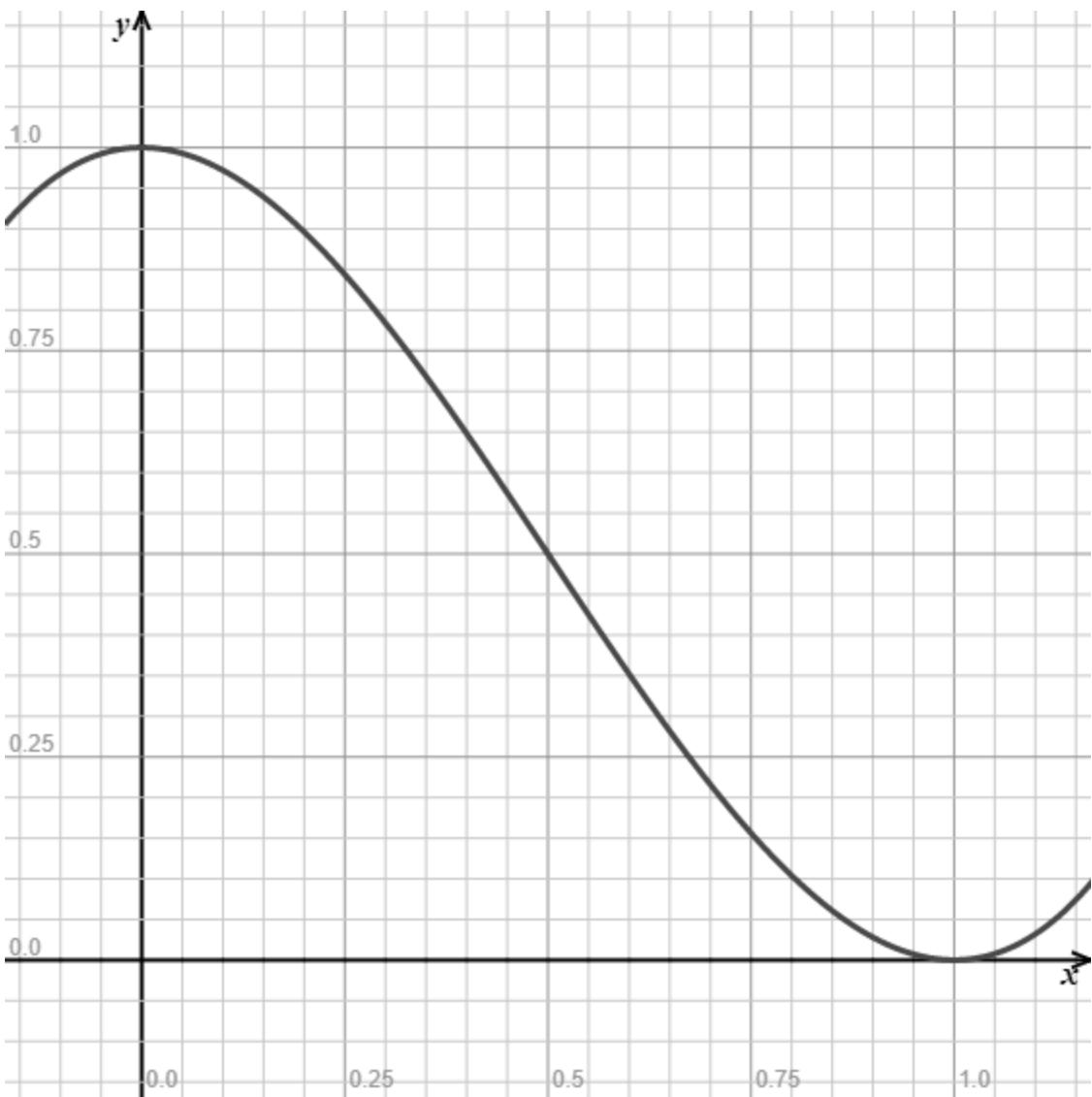
Krzywa Hermite'a może więc być definiowana jako kombinacja liniowa wektorów składowych macierzy geometrii G i funkcji bazowych Hermite'a (*oznaczona tutaj jako $B(u)$*).

Jeśli wykreślić wielomiany bazowe to mają one następujący przebieg: (nast. slajd)

*pamiętamy, że są 4 wielomiany bazowe

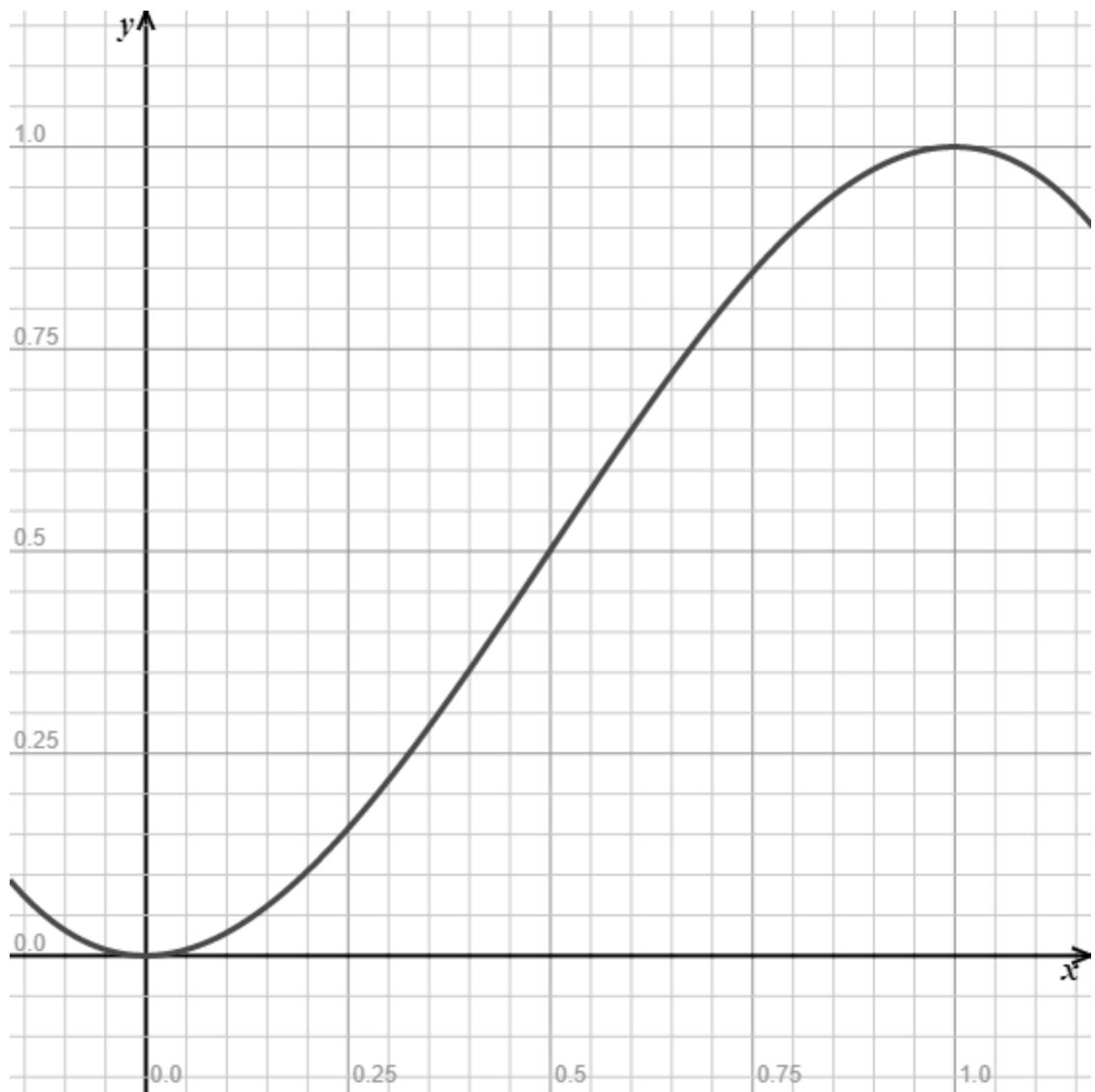
$$B_0(u) = 2u^3 - 3u^2 + 1$$

*wykres wygenerowano narzędziem:
<https://www.matemaks.pl/program-do-rysowania-wykresow-funkcji.html>



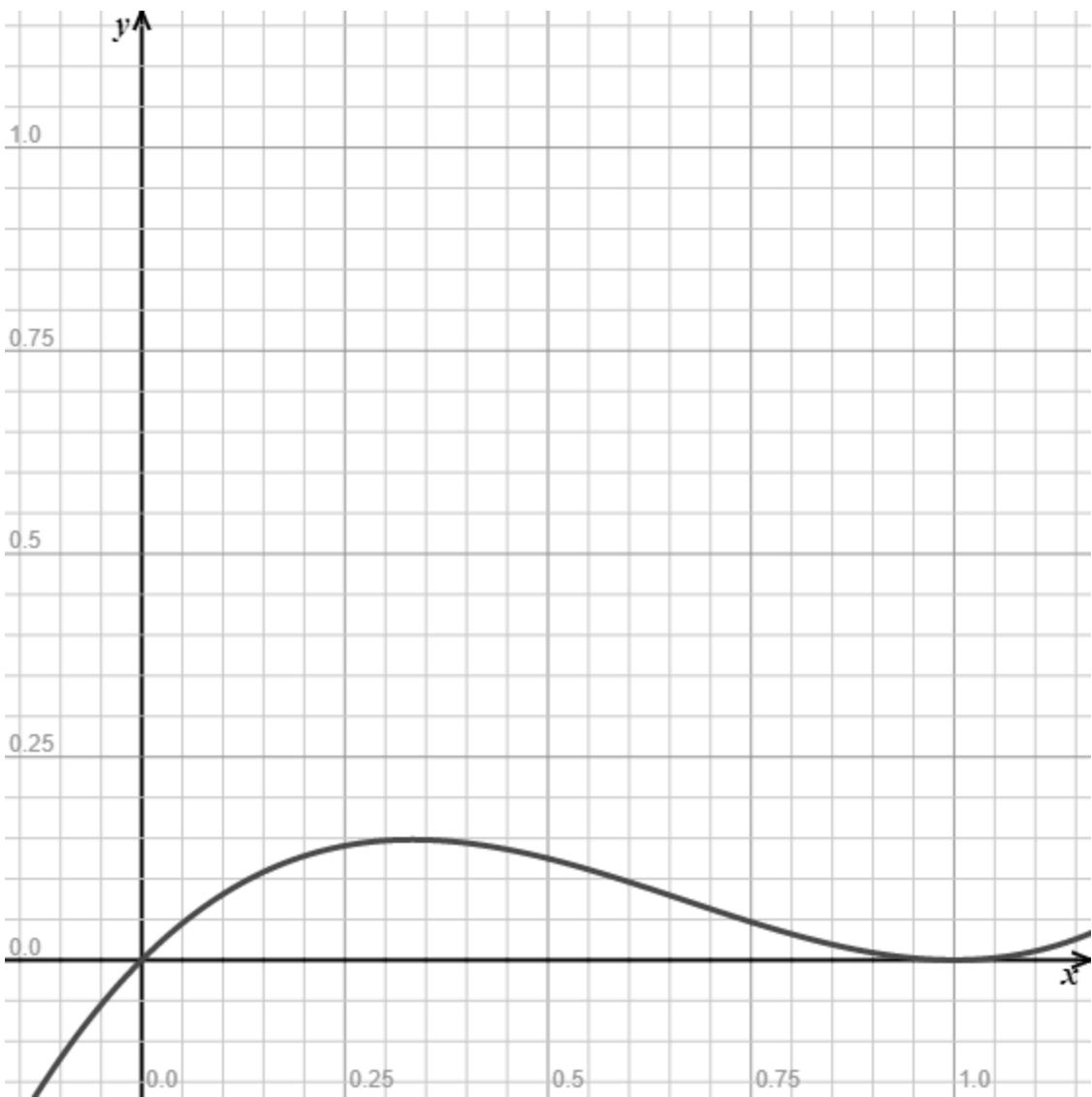
$$B_1(u) = -2u^3 + 3u^2$$

*wykres wygenerowano narzędziem:
<https://www.matemaks.pl/program-do-rysowania-wykresow-funkcji.html>



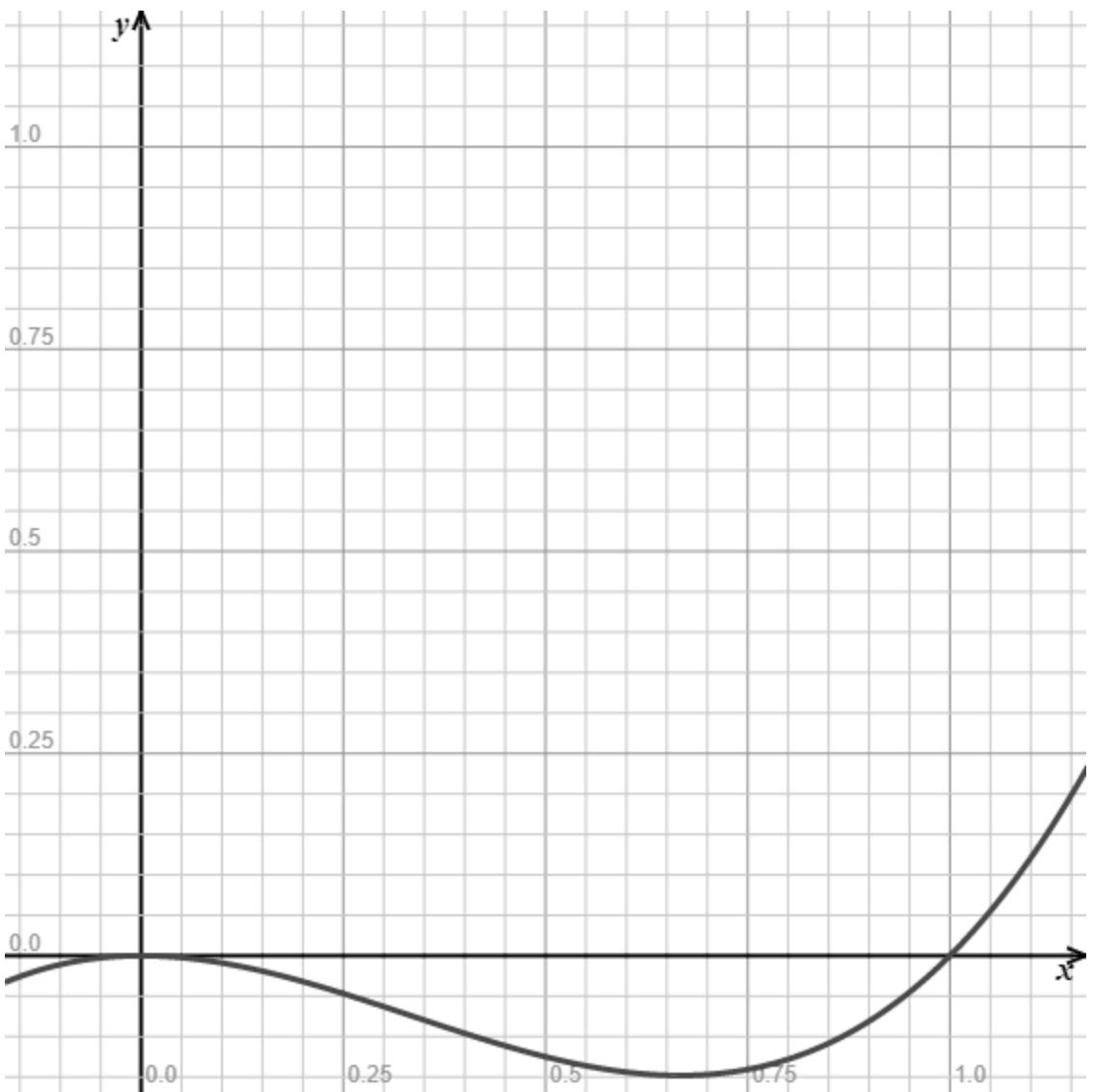
$$B_2(u) = u^3 - 2u^2 + u$$

*wykres wygenerowano narzędziem:
<https://www.matemaks.pl/program-do-rysowania-wykresow-funkcji.html>



$$B_3(u) = u^3 - u^2$$

*wykres wygenerowano narzędziem:
<https://www.matemaks.pl/program-do-rysowania-wykresow-funkcji.html>



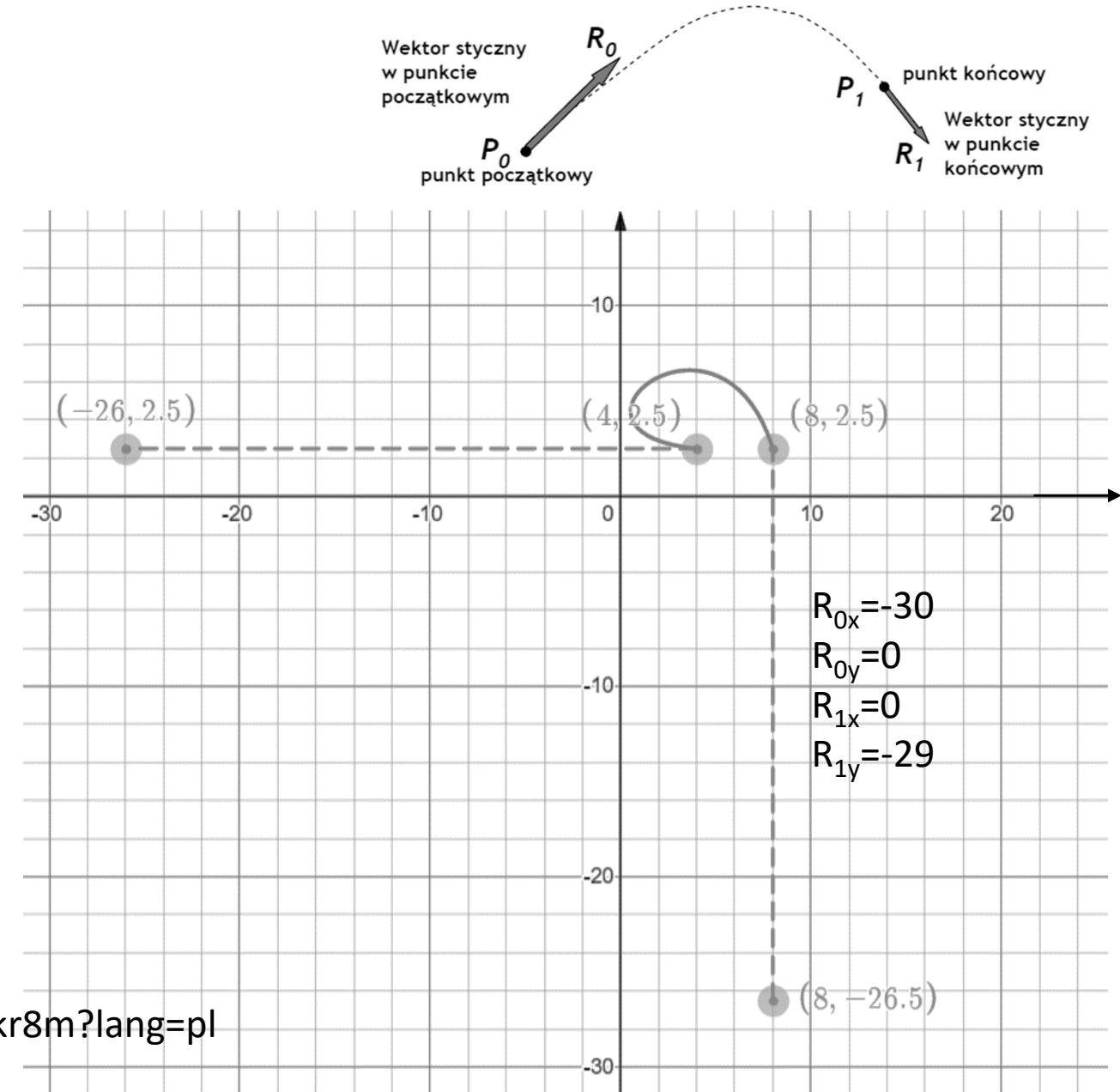
Dla krzywej o początku w punkcie (P_{0x}, P_{0y}, P_{0z}) i końcu w (P_{1x}, P_{1y}, P_{1z}) , sterowanie przebiegiem krzywej realizuje się przez dobór wektorów stycznych (R_{0x}, R_{0y}, R_{0z}) i (R_{1x}, R_{1y}, R_{1z}) /*pochodnych*/ „zaciszonych” w tych punktach .

Widać, że wektory muszą być „długie”, żeby krzywa stała się „bardziej” wypukła.

//na wykresie

*wykres wygenerowano narzędziem:

<https://www.desmos.com/calculator/5knm5tkr8m?lang=pl>



Inny przykład

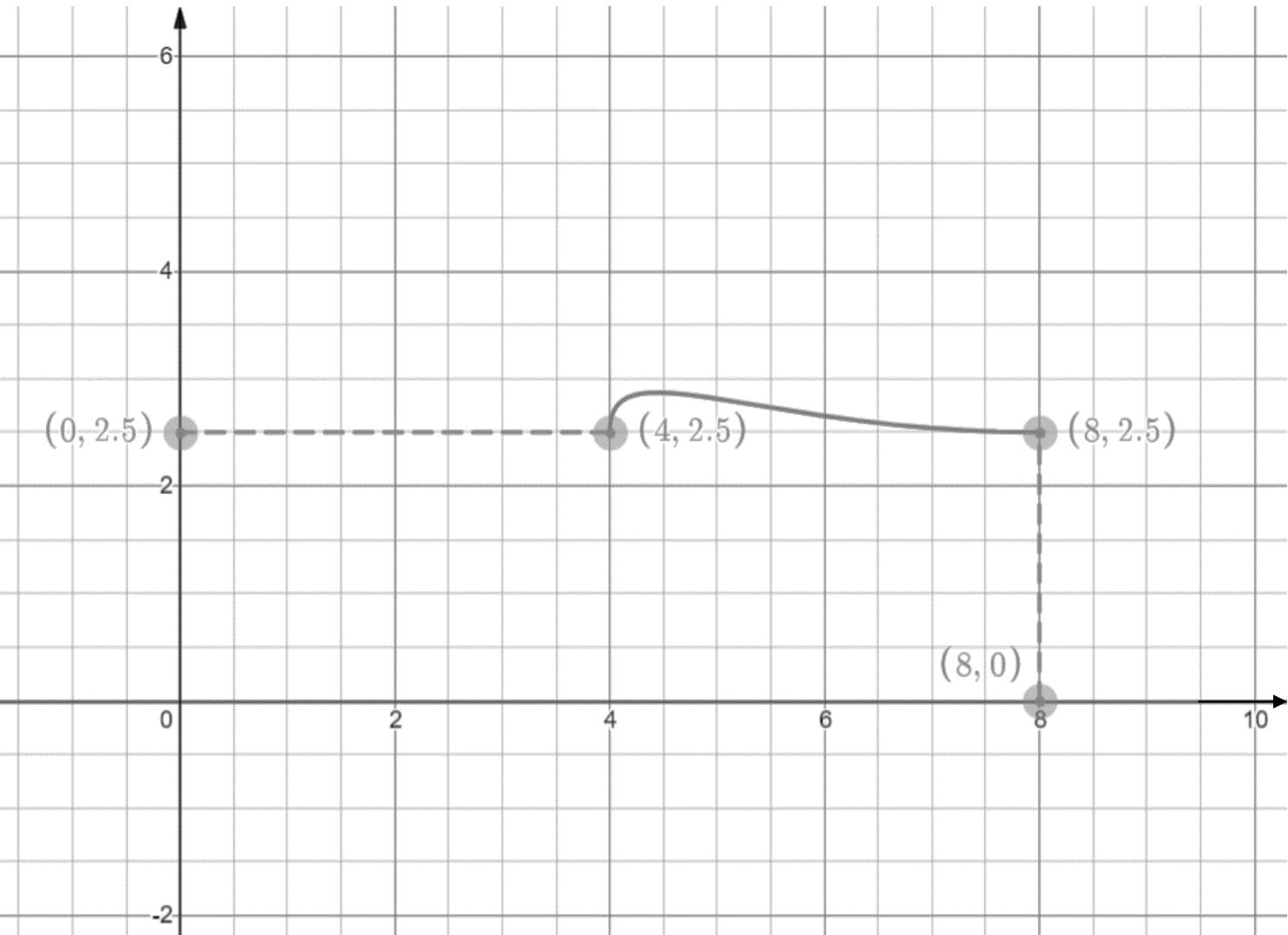
Jakie są tutaj
współrzędne wektorów?

$$R_{0x}=?$$

$$R_{0y}=?$$

$$R_{1x}=?$$

$$R_{1y}=?$$



Inny przykład

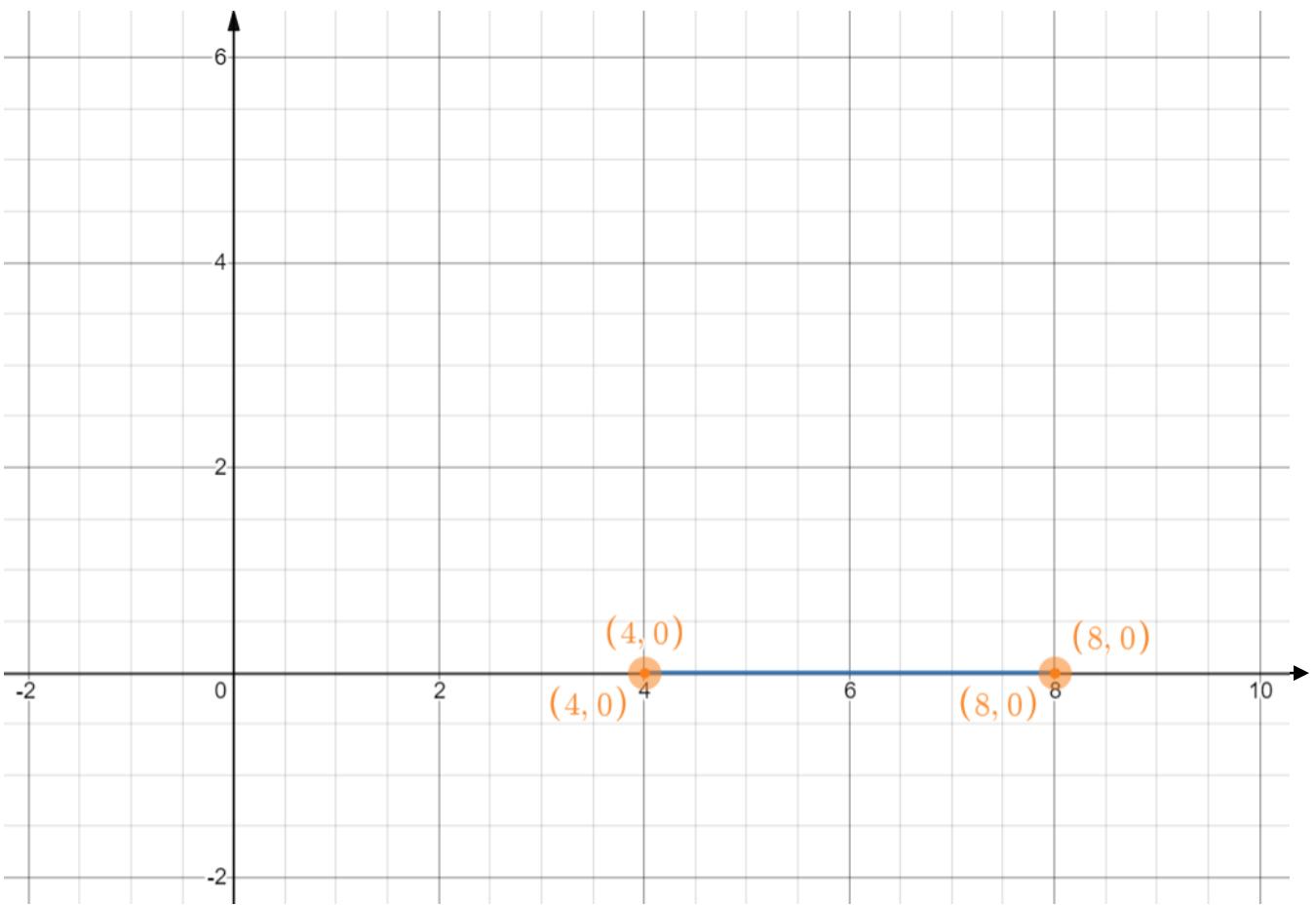
Jakie są tutaj
współrzędne wektorów?

$$R_{0x} = ?$$

$$R_{0y} = ?$$

$$R_{1x} = ?$$

$$R_{1y} = ?$$



Podsumowanie:

1. Krzywa Hermite'a jest określona przy pomocy wielomianów 3-go stopnia.
2. Jako dane, projektant podaje dwa punkty (początkowy i końcowy) i dwa wektory (łącznie cztery liczby) określające pochodne w tych punktach.
3. Sterowanie przebiegiem krzywej odbywa się przez manipulacje wektorami pochodnej.
4. Nie zawsze łatwo jest uzyskać pożądany kształt krzywej.

Krzywe Bezier'a

Krzywe Bezier'a

Pierre Étienne Bézier (ur. 1 września 1910 w Paryżu, zm. 25 listopada 1999; wym. ['pjɛʁ e'tjɛn be'zje]) – francuski matematyk i inżynier.

Od roku 1933 pracował w przedsiębiorstwie Renault, gdzie przepracował 42 lata. W latach 60. XX wieku opisał pewien typ krzywych nazwanych krzywymi Béziera, które były używane przy projektowaniu nadwozi samochodów tego przedsiębiorstwa. Opracował również system CAD/CAM o nazwie UNISURF.

* Źródło: Wikipedia

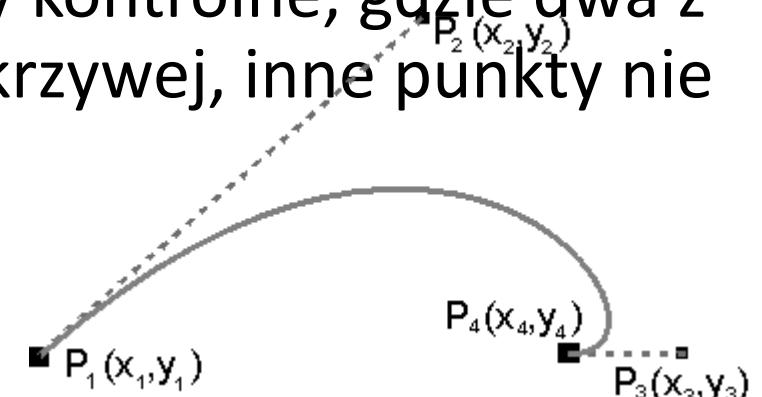
Swoją koncepcję wyznaczania krzywych opisał w materiale:

Bézier, Pierre. "Numerical control: mathematics and applications." (1970).

Krzywa Beziera jest określona jest przez punkty kontrolne, gdzie dwa z nich (punkt początkowy i końcowy) należą do krzywej, inne punkty nie należą do krzywej.

*źródło:

http://www.interklasa.pl/portal/dokumenty/corel/krzywe_bezi era/krzywe_beziera.htm





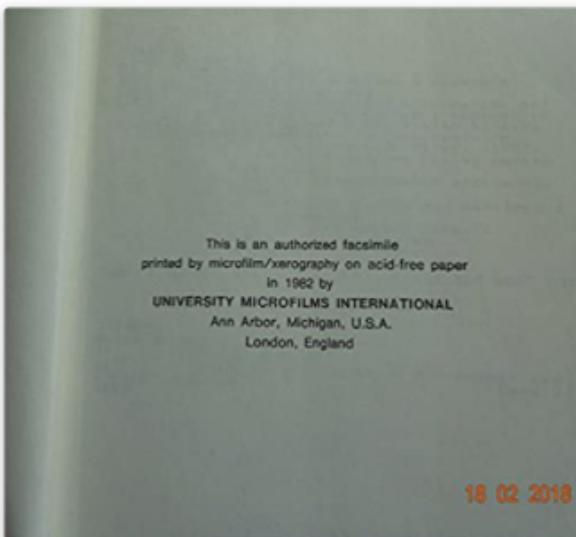
All Best Sellers Mobiles Fashion Prime ▾ Today's Deals New Releases Electronics Customer Service Amazon Pay Home & Kitchen Computers

Books Advanced Search New Releases & Pre-orders Best Sellers ▾ Browse Genres ▾ Children's & Young Adult ▾ Textbooks ▾ Exam Central ▾ All Indian Languages ▾

Books > Higher Education Textbooks > Engineering Textbooks > Bezier *numerical Control* – Mathematics And Applications (Wiley Series in

Bezier *numerical Control* – Mathematics And Applications (Wiley Series in Computing) Paperback – Import, 1 January 1972

by P BEZIER (Author)



› See all formats and editions

Paperback
from ₹ 15,496.00

1 Used from ₹ 15,496.00



10 Days

Replacement

Only

Dokładniej kształt krzywej Béziera jest określony czterema punktami: dwoma punktami krańcowymi krzywej (tzw. węzłami) (P_1, P_4) oraz dwoma punktami kontrolnymi (P_2, P_3). Krzywa interpoluje dwa krańcowe punkty krzywej i aproksymuje dwa punkty kontrolne. Jeżeli oznaczymy współrzędne tych czterech punktów jako:

$$P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3), P_4(x_4, y_4)$$

to kształt krzywej Béziera (w 2D) określa równanie parametryczne:

$$P(t) = P_1(1-t)^3 + 3P_2t(1-t)^2 + 3P_3t^2(1-t) + P_4t^3$$

Gdzie stosując oznaczenia zgodne z rysunkiem

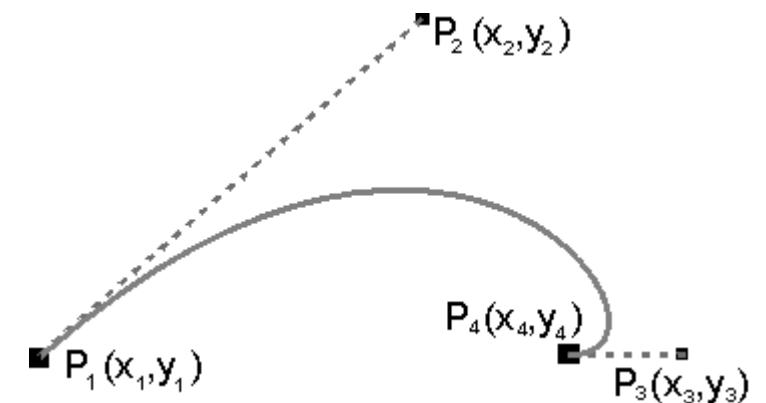
$$x(t) = x_1(1-t)^3 + 3x_2t(1-t)^2 + 3x_3t^2(1-t) + x_4t^3$$

$$y(t) = y_1(1-t)^3 + 3y_2t(1-t)^2 + 3y_3t^2(1-t) + y_4t^3$$

gdzie parametr t przybiera wartości z przedziału $<0,1>$

(w 3D będzie jeszcze jedno równanie $z(t)$)

$$P(t) = A(1-t)^3 + 3Bt(1-t)^2 + 3Ct^2(1-t) + Dt^3 \quad \text{dla } 0 \leq t \leq 1,$$

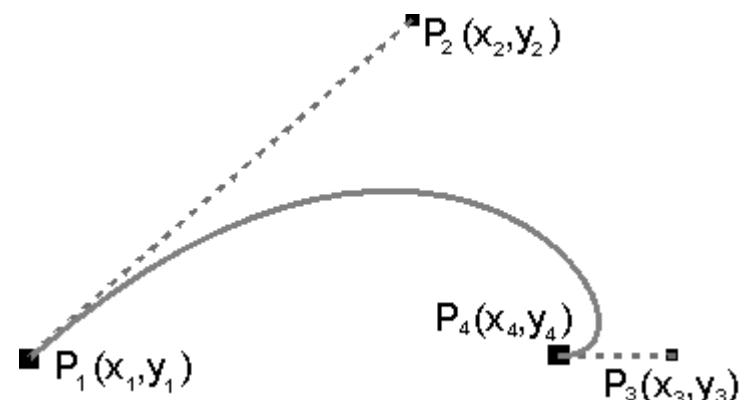


Alternatywnie można krzywą Bezier'a zapisać macierzowo (stosując oznaczenia A B C D dla punktów):

$$P(t) = [A, B, C, D] \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

*dla przypomnienia

$$P(t) = A(1-t)^3 + 3Bt(1-t)^2 + 3Ct^2(1-t) + Dt^3 \quad \text{dla } 0 \leq t \leq 1,$$



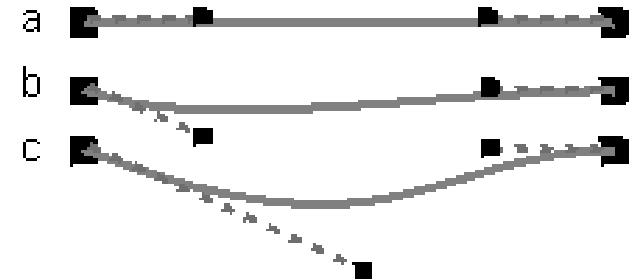
Połączone krzywe

W programie do grafiki wektorowej każdą krzywą (krzywą jest także okrąg, kwadrat, itp.) definiuje się podając węzły i punkty kontrolne. Istnieje także pojęcie segmentu w skład którego wchodzą dwa węzły (na jego końcach) i dwa punkty kontrolne.

Ponieważ segmenty sąsiadują ze sobą, dlatego z każdym węzłem związane są tylko dwa punkty kontrolne.

Na krzywej można wykonać następujące czynności:

- przesunąć węzeł - zmianie ulegnie wygląd jednego lub dwóch segmentów (to, gdy węzeł należał do dwóch segmentów);
- przesunąć punkt kontrolny - zmieni się kształt jednego segmentu;
- dodać węzeł - jeden segment zostanie podzielony na dwa segmenty, pomiędzy którymi znajdzie się dodany węzeł;
- usunąć węzeł - zostaną usunięte także dwa punkty kontrolne, a dwa sąsiednie segmenty zostaną połączone w jeden segment, którego kształt będą określały pozostałe-sąsiednie punkty kontrolne;
- połączyć dwa końcowe węzły - powstanie jeden węzeł z punktami kontrolnymi tak ustawnionymi, aby przejście krzywej przez ten węzeł było "gładkie";
- przekształcić segment na prostą, krzywą, itp. - powoduje to automatyczne ustawienie punktów kontrolnych w ten sposób, aby uzyskać żądany kształt.



W praktyce:

- a) chcąc uzyskać linię prostą należy tak ułożyć punkty kontrolne, aby leżały na linii łączącej oba węzły;
- b) gdy zostanie przesunięty jeden z punktów kontrolnych tak, aby nie leżał na prostej łączącej dwa węzły, to wtedy segment "wybrzuszy" się w taki sposób, by w węźle segment był styczny do linii łączącej węzłów z punktem kontrolnym;
- c) gdy punkt kontrolny zostanie oddalony od węzła, to krzywa będzie "łagodniej" przechodzić przez węzeł.

Krzywe Bezier'a 3D

Definicja krzywej Beziera:

Dany jest zbiór $n+1$ punktów kontrolnych P_k

$$P_k = \begin{pmatrix} P_{kx} & P_{ky} & P_{kz} \end{pmatrix} \quad k = 0, 1, \dots, n$$

gdzie P_{kx} , P_{ky} , P_{kz} są współrzędnymi x, y, z punktu kontrolnego.

Krzywa Beziera opisana jest układem równań parametrycznych

$$x(u) = \sum_{k=0}^n P_{kx} B_{kn}(u)$$

$$y(u) = \sum_{k=0}^n P_{ky} B_{kn}(u) \quad 0 \leq u \leq 1$$

$$z(u) = \sum_{k=0}^n P_{kz} B_{kn}(u)$$

Gdzie

$$B_{kn}(u) = \binom{n}{k} u^k (1-u)^{n-k} \quad k = 0, 1, \dots, n$$

Przy czym

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Wielomiany $B_{kn}(u)$ noszą nazwę wielomianów Bernsteina.

Stopień wielomianów $B_{kn}(u)$ zależy od liczby punktów kontrolnych.

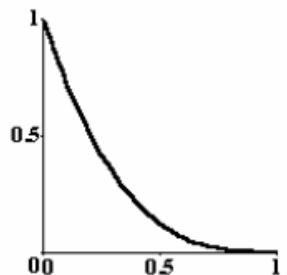
Wektory styczne do krzywej w punktach początkowym i końcowym spełniają warunki:

$$Q'(0) = n(P_1 - P_0)$$

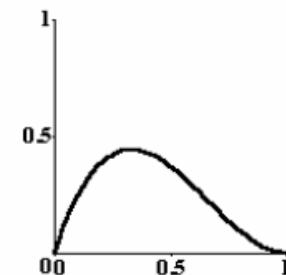
$$Q'(1) = n(P_n - P_{n-1})$$

Przykład: $n = 3$ (cztery punkty kontrolne)

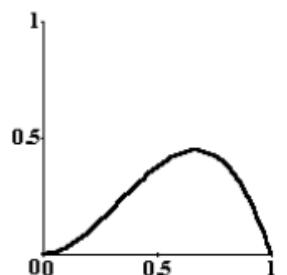
$$Q(u) = P_0 B_{03}(u) + P_1 B_{13}(u) + P_2 B_{23}(u) + P_3 B_{33}(u)$$



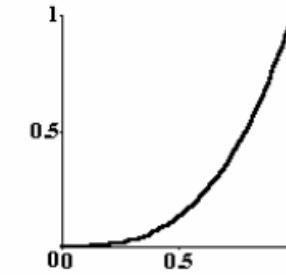
$$B_{03}(u) = (1-u)^3$$



$$B_{13}(u) = 3u(1-u)^2$$



$$B_{23}(u) = 3u^2(1-u)$$

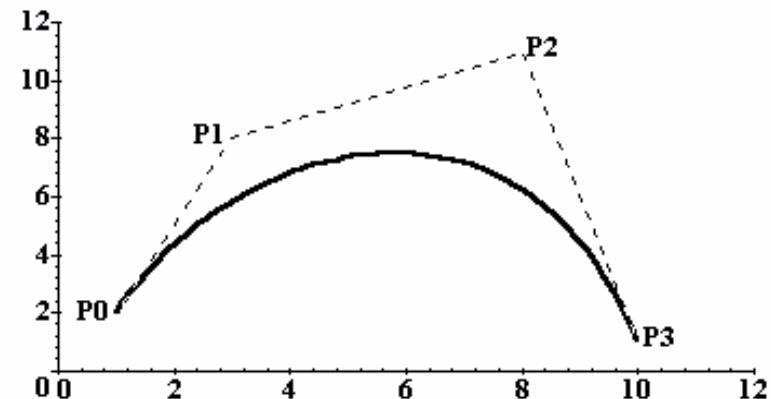
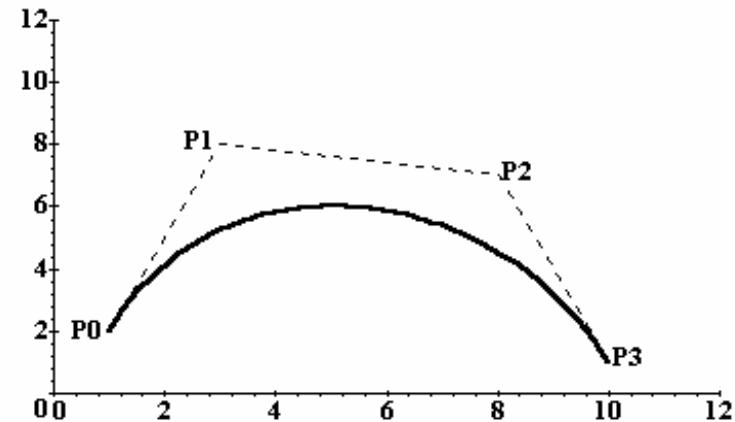


$$B_{33}(u) = u^3$$

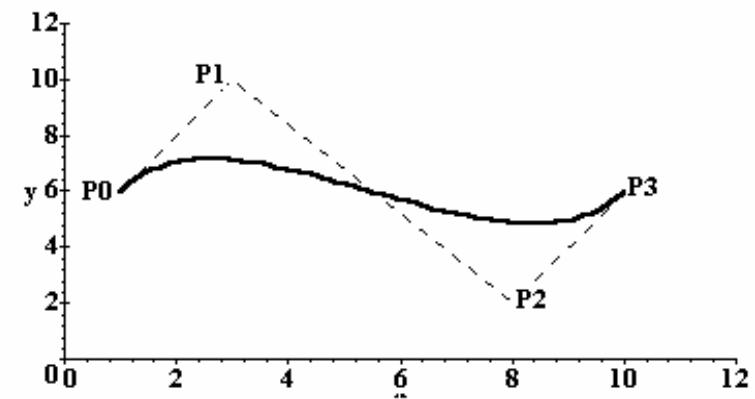
Własności krzywych Bezier'a:

Krzywa dla pewnego układu punktów kontrolnych, $n = 3$
(od 0 do 3, czyli 4 punkty)

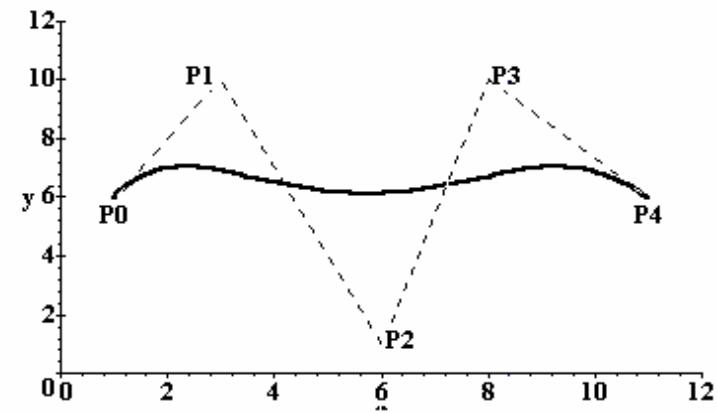
Po zmianie położenia punktu kontrolnego P_2



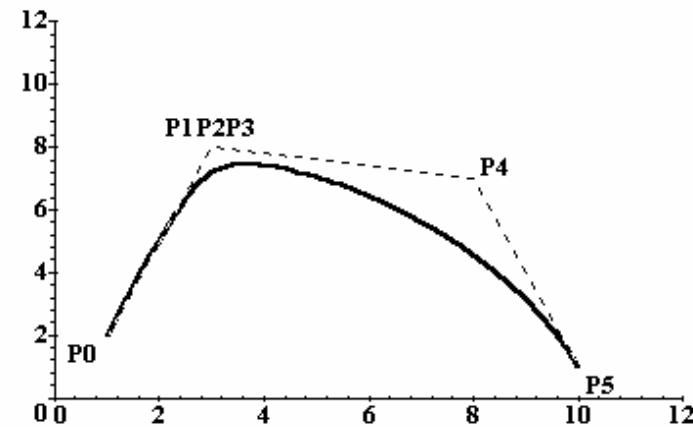
Krzywa dla innego układu punktów kontrolnych, n = 3



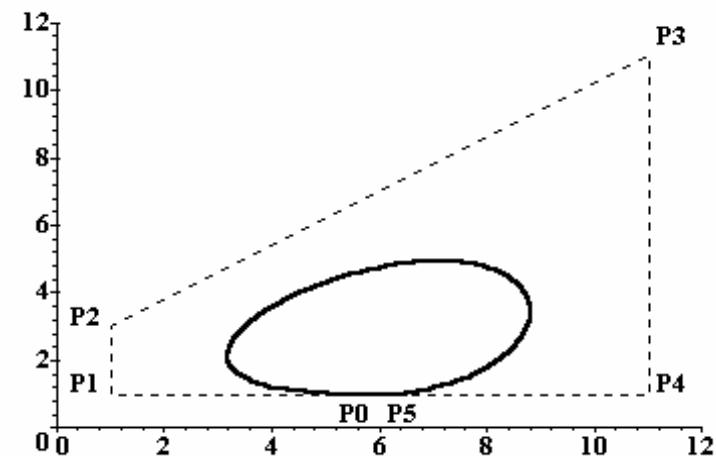
Krzywa dla większej liczby punktów kontrolnych, n = 4



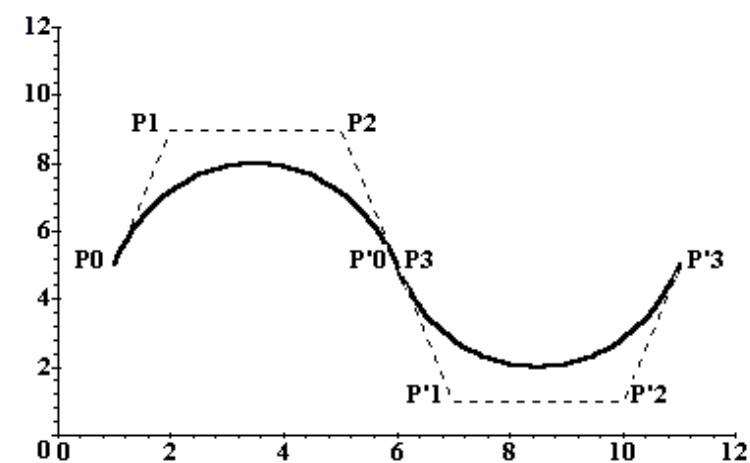
„Wielokrotny” punkt kontrolny, n = 5



„Zamknięty” układ punktów kontrolnych, n = 5



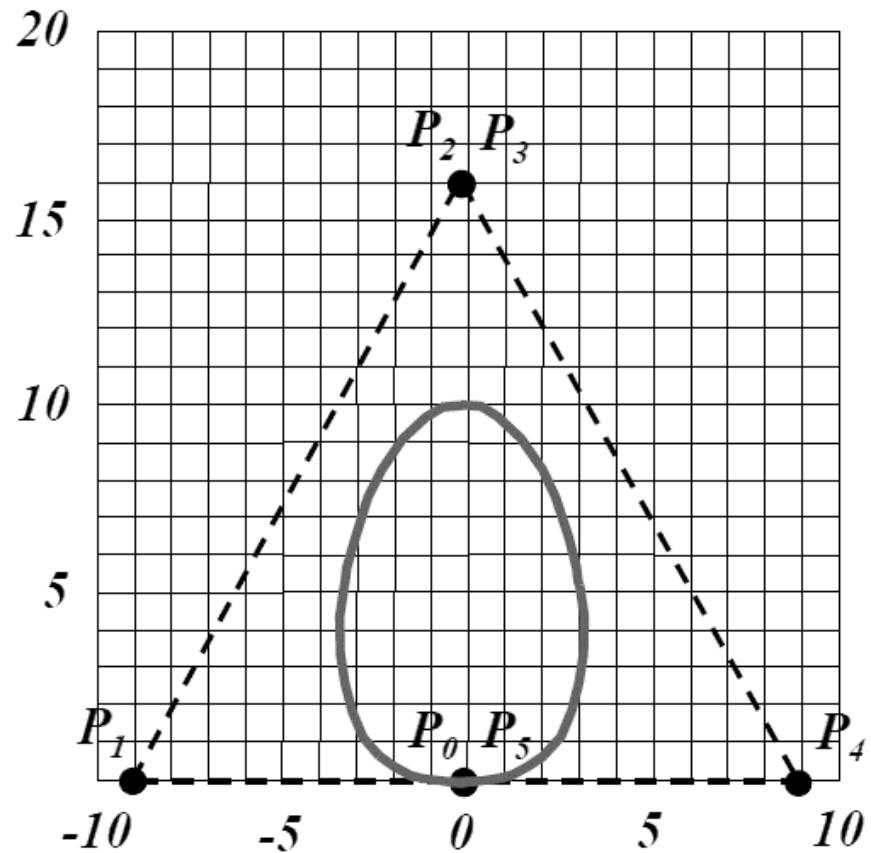
„Gładkie” połączenie dwóch krzywych Bezier'a:



Podsumowanie:

1. Krzywa Bezier'a jest określona wielomianami o stopniu zależnym od liczby punktów kontrolnych.
2. Jako dane projektant podaje tylko punkty kontrolne.
3. Sterowanie przebiegiem krzywej odbywa się przez manipulacje położeniem punktów kontrolnych.
4. Przesunięcie jednego punktu kontrolnego, zmienia kształt całej krzywej (jest to poniekąd wada tego rodzaju modelowania).

Przykład zastosowania krzywej Beziera:



Równania tej krzywej mają postać:

$$x(u) = -90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u$$

$$y(u) = 160u^4 - 320u^3 + 160u^2 \quad 0 \leq u \leq 1$$

$$z(u) = 0$$

Jeśli obrócić krzywą wokół osi y, można otrzymać równania parametryczne powierzchni:

$$x(u,v) = (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u)\cos(2\pi v)$$

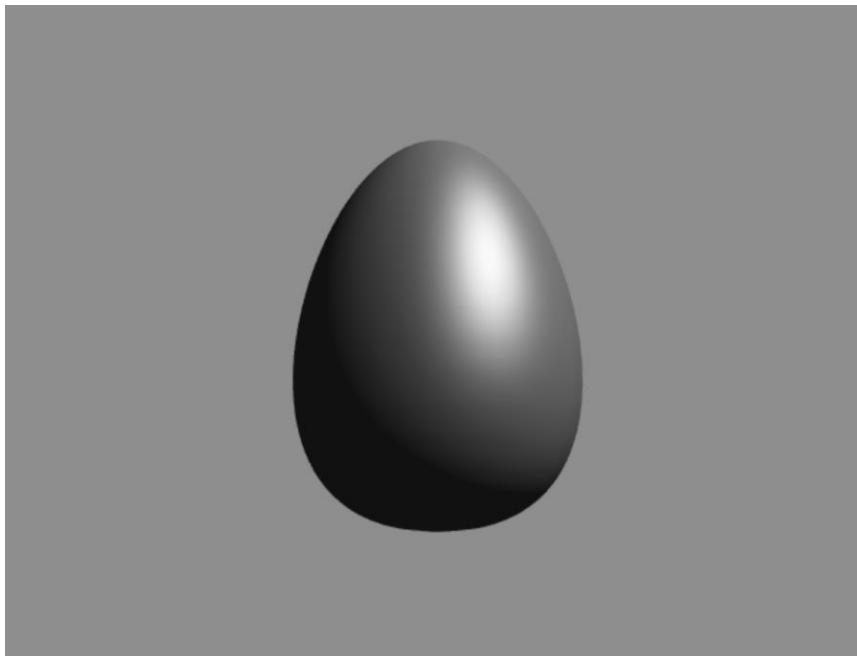
$$y(u,v) = 160u^4 - 320u^3 + 160u^2$$

$$z(u,v) = (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u)\sin(2\pi v)$$

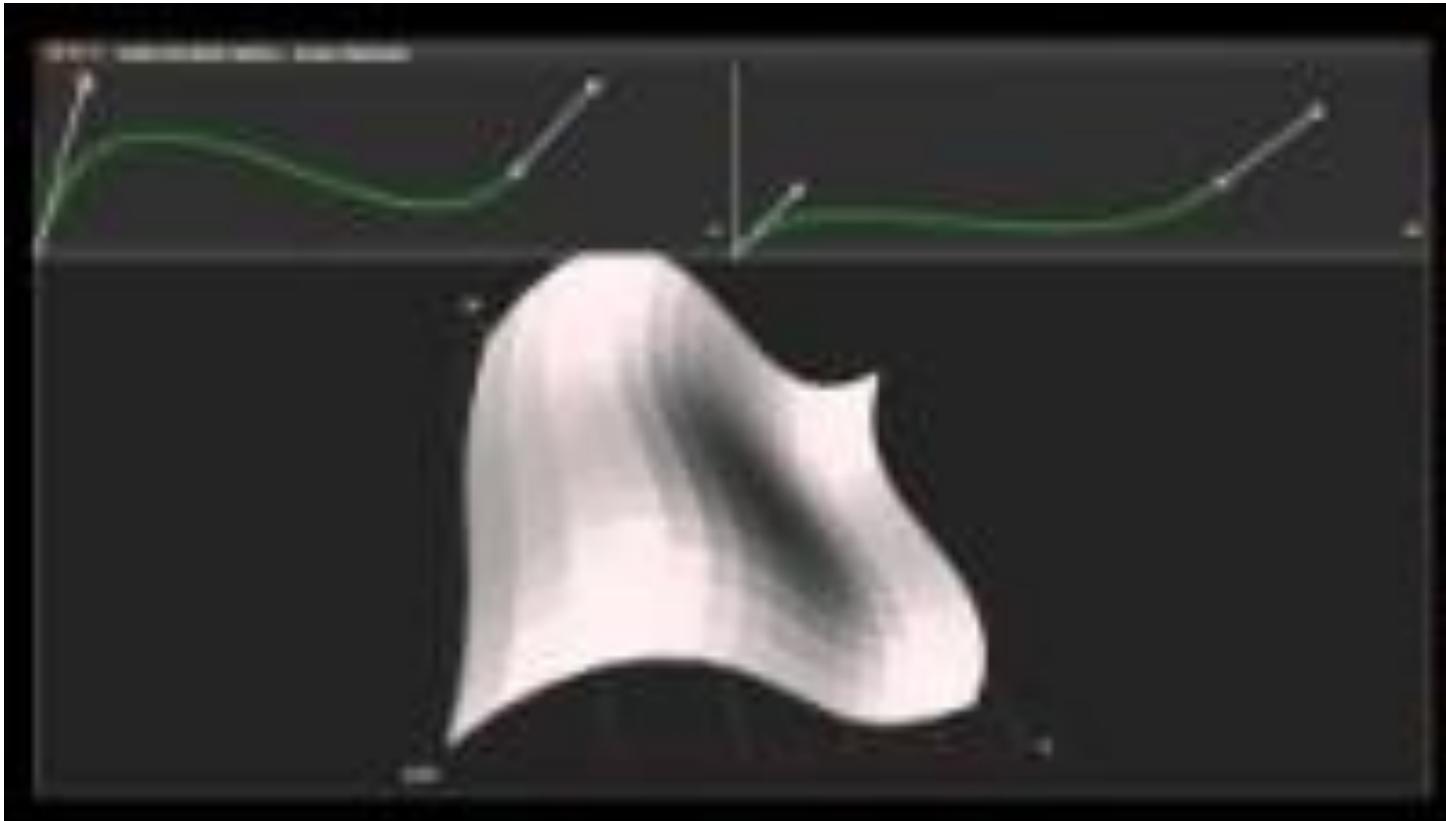
$$0 \leq u \leq 1$$

$$0 \leq v \leq 1$$

Po zastosowaniu renderingu z użyciem ustawionego źródła światła model wygląda jak poniżej:



Porównanie krzywej Hermite'a i Bezier'a



https://www.youtube.com/watch?v=vwT_5RGlxY

Krzywe sklejane

Wadą krzywej Beziera był fakt, że manipulacja położeniem jednego punktu kontrolnego powodowała zmianę kształtu całej krzywej.

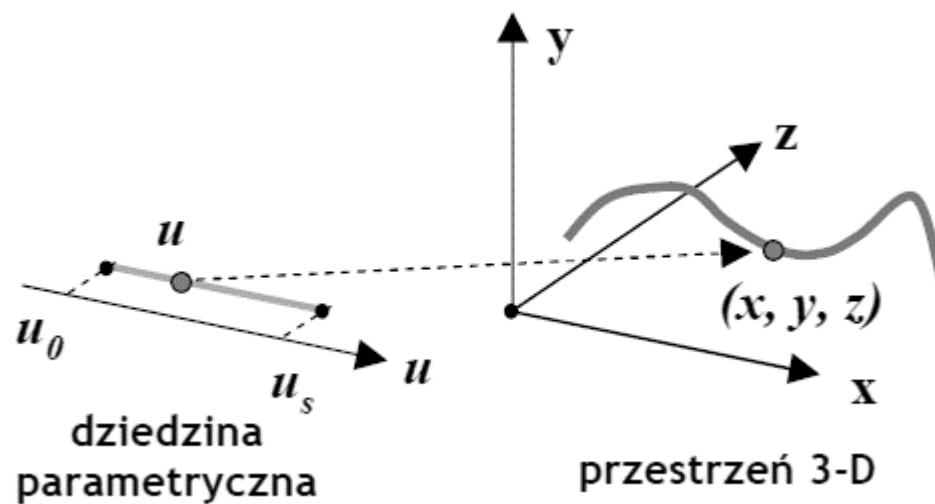
Można tą niedogodność usunąć, budując krzywą z segmentów połączonych (sklejonych) tak, że punkty ich połączenia nie są dla obserwatora widoczne.

Krzywa parametryczna zbudowana z segmentów

$$x(u) = f_x(u)$$

$$y(u) = f_y(u) \quad u_0 \leq u \leq u_s$$

$$z(u) = f_z(u)$$



Funkcje $f_x(u)$, $f_y(u)$, $f_z(u)$ zostaną zapisane w następujący sposób

$$f_x(u) = \begin{cases} f_{x1}(u) & u_0 \leq u < u_1 \\ \dots \\ f_{xk}(u) & u_{k-1} \leq u < u_k \quad f_y(u), f_z(u) \text{ podobnie} \\ \dots \\ f_{xs}(u) & u_{s-1} \leq u \leq u_s \end{cases}$$

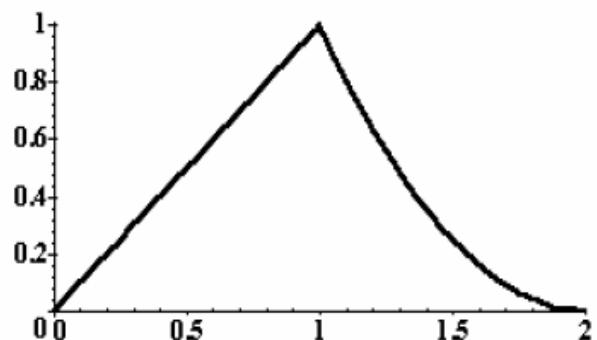
Dziedzinę parametryczną, czyli przedział $[u_0, u_s]$ podzielono na podprzedziały $[u_0, u_1]$, $[u_1, u_2]$... $[u_{s-1}, u_s]$ i dla każdego z nich zdefiniowano funkcje $f_x(u)$, $f_y(u)$, $f_z(u)$ osobno.

Punkty podziału $u_0, u_1, u_2, \dots, u_{s-1}, u_s$ nazywamy węzłami krzywej.

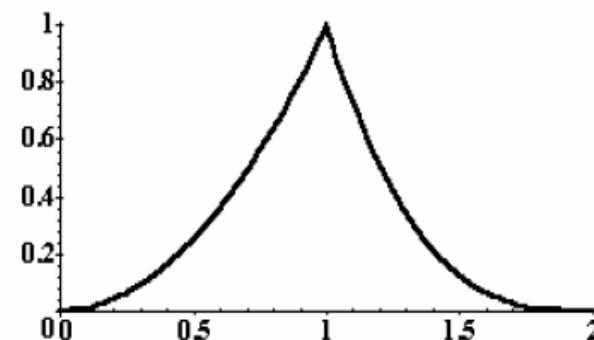
Przykładowe krzywe sklejane (sklejone ;-)):

$$x(u) = \begin{cases} u & 0 \leq u < 1 \\ (u-2)^2 & 1 \leq u \leq 2 \end{cases}$$

$$y(u) = \begin{cases} u^2 & 0 \leq u < 1 \\ -(u-2)^3 & 1 \leq u \leq 2 \end{cases} \quad z(u) = 0$$



$$x(u) = f_x(u)$$

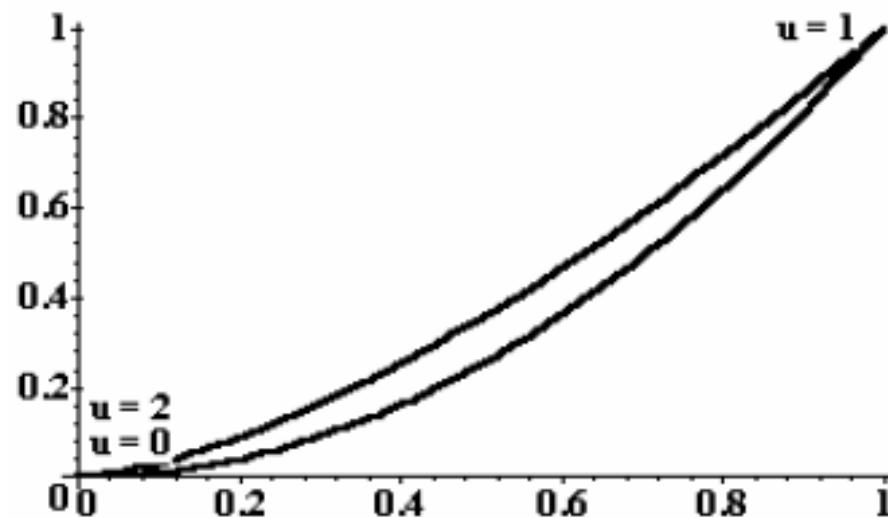


$$y(u) = f_y(u)$$

Kolejny przykład – to jest jedna krzywa sklejana (nie dwie osobne)

$$\begin{aligned}x(u) &= f_x(u) \\y(u) &= f_y(u)\end{aligned}\quad 0 \leq u \leq 2$$

*Dokładna definicja f_x i f_y nie jest w tym przykładzie istotna



Ciągłość między segmentami krzywej

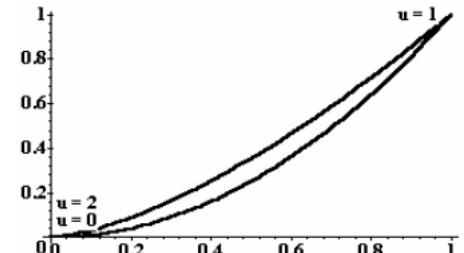
Krzywa opisana jest równaniem

$$Q(u) = [x(u) \quad y(u) \quad z(u)]^T \quad u_0 \leq u \leq u_s$$

Pochodna $Q'(u)$ zwana jest parametrycznym wektorem stycznym do krzywej – zapisuje się to w następujący sposób

$$Q'(u) = [x'(u) \quad y'(u) \quad z'(u)]^T \quad u_0 \leq u \leq u_s$$

Jak zachowuje się parametryczny wektor styczny krzywej w punkcie połączenia segmentów
(dla poprzedniego przykładu w punkcie $u = 1$) ?



Ciągłość geometryczna

1. Jeżeli dwa segmenty krzywej łączą się ze sobą, to krzywa ma ciągłość geometryczną G_0 .
2. Jeżeli kierunki (nie długości) wektorów stycznych segmentów w punkcie połączenia są równe, to krzywa ma ciągłość geometryczną G_1 .

Ciągłość parametryczna

1. Jeżeli wektory styczne dwóch segmentów w punkcie połączenia są równe (kierunki i długości wektorów są równe), to krzywa ma w tym punkcie ciągłość parametryczną C_1 .
2. Jeżeli kierunki i długości wektorów stycznych segmentów ($\frac{d^n}{du^n}[Q(u)]$) w każdej n-tej pochodnej są równe (wektory w każdej pochodnej są równe), to krzywa ma ciągłość parametryczną C_n .

Jednorodne krzywe B-sklejane

Dany jest zbiór $n + 1$ ($n \geq 3$) punktów kontrolnych P_k

$$P_k = (P_{kx} \quad P_{ky} \quad P_{kz}) \quad k = 0, 1, \dots, n$$

gdzie P_{kx} , P_{ky} , P_{kz} są współrzędnymi x , y , z punktu kontrolnego.

Krzywa B-sklejana składa się z $n-2$ segmentów wielomianowych trzeciego stopnia

$$Q_3(u), Q_4(u), \dots, Q_i(u), \dots, Q_n(u)$$

Punkty u_i połączenia segmentów $Q_i(u)$ i $Q_{i+1}(u)$ oraz punkty u_3 i u_n nazywa się węzłami krzywej.

Krzywa B-sklejana jest jednorodna jeśli węzły znajdują się w jednakowych odstępach, czyli

$$u_3 = 0 \quad \text{oraz} \quad u_{i+1} - u_i = 1$$

Konstrukcja krzywej B-sklejanej:

Każdy segment krzywej określony jest przez cztery punkty kontrolne:

- segment $Q_3(u)$, przez punkty P_0, P_1, P_2, P_3 dla $0 \leq u < 1$,
- segment $Q_4(u)$, przez punkty P_1, P_2, P_3, P_4 dla $1 \leq u < 2$,
- ...
- segment $Q_i(u)$, przez punkty $P_{i-3}, P_{i-2}, P_{i-1}, P_i$ dla $i - 3 \leq u < i - 2$.

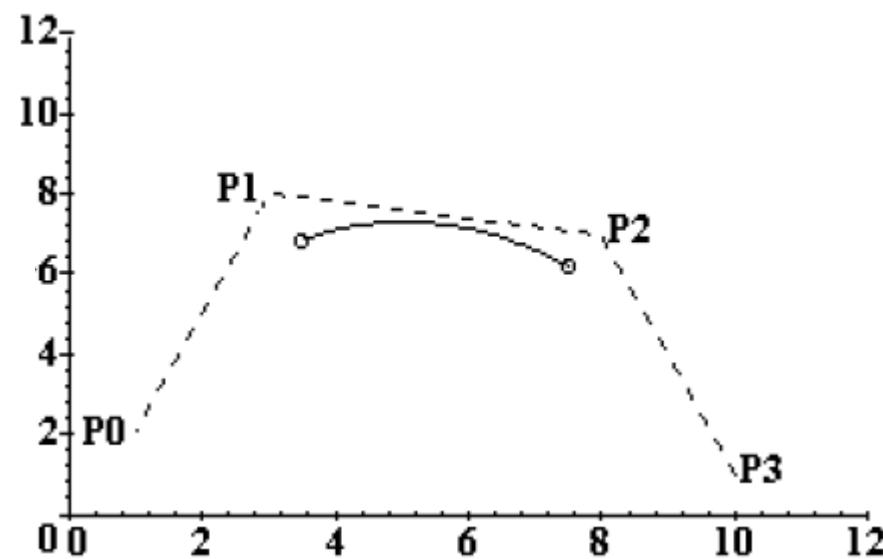
Segment i-ty krzywej jest opisany (po zastąpieniu $u-u_i$ przez u) wzorem

$$\begin{aligned} Q_i(u - u_i) &= G_{Bi} \cdot M_B \cdot U_i = \\ &= P_{i-3} \frac{(1-u)^3}{6} + P_{i-2} \frac{3u^3 - 6u^2 + 4}{6} \\ &\quad + P_{i-1} \frac{-3u^3 + 3u^2 + 3u + 1}{6} + P_i \frac{u^3}{6} \end{aligned}$$

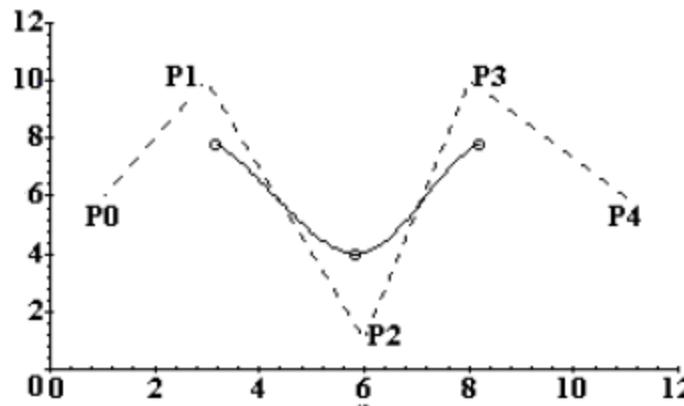
$$0 \leq u < 1$$

Własności jednorodnych krzywych B-sklejanych

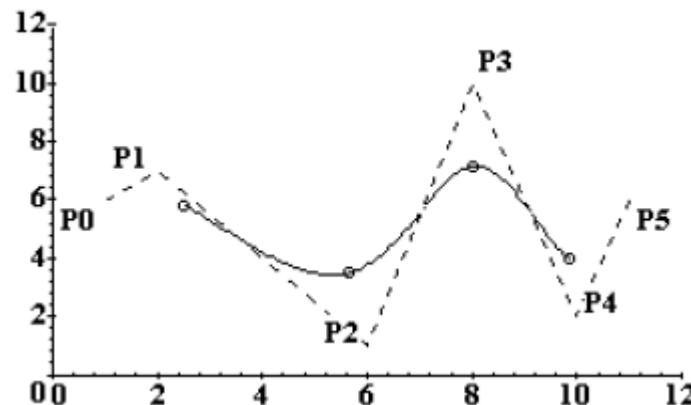
Wygląd krzywej dla przykładowego układu punktów kontrolnych, $n = 3$



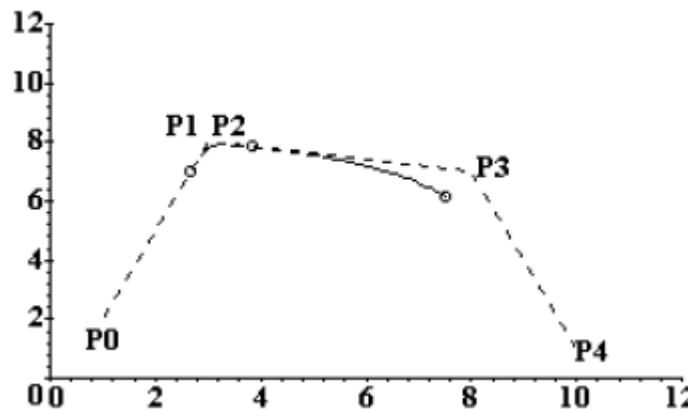
Krzywa zbudowana z dwóch segmentów, $n = 4$



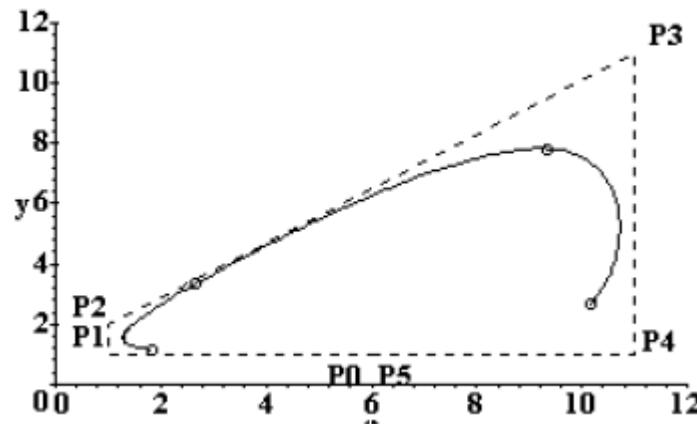
Krzywa zbudowana z trzech segmentów, $n = 5$



Podwójny punkt kontrolny, n = 4



Zamknięty układ punktów kontrolnych, n = 5



Podsumowanie:

1. Krzywa B - sklejana jest określona przy pomocy wielomianów 3-go stopnia.
2. Jako dane, projektant podaje punkty kontrolne.
3. Manipulacja punktem kontrolnym nie zmienia kształtu całej krzywej (w przypadku większej liczby punktów).
4. Krzywa nie zaczyna się i nie kończy w pierwszym i w ostatnim punkcie kontrolnym (wada!).

Niejednorodne krzywe B-sklejane

Dany jest zbiór $n + 1$ ($n \geq 3$) punktów kontrolnych P_k

$$P_k = (P_{kx} \quad P_{ky} \quad P_{kz}) \quad , \quad k=0,1,\dots,n$$

gdzie P_{kx} , P_{ky} , P_{kz} są współrzędnymi x , y , z punktu kontrolnego.

Niejednorodna krzywa B-sklejana, zbudowana z wielomianów stopnia t zadana jest równaniami

$$x(u) = \sum_{k=0}^n P_{kx} N_{k,t}(u)$$

$$y(u) = \sum_{k=0}^n P_{ky} N_{k,t}(u) \quad 0 \leq u \leq n - t + 2$$

$$z(u) = \sum_{k=0}^n P_{kz} N_{k,t}(u)$$

przy czym, wielomiany bazowe $N_{k,t}(u)$ określone są rekurencyjnie równaniami

$$N_{k,1}(u) = \begin{cases} 1 & u_k \leq u < u_{k+1} \\ 0 & \text{w przypadku przeciwnym} \end{cases}$$

$$N_{k,t}(u) = \frac{u - u_k}{u_{k+t-1} - u_k} N_{k,t-1}(u) + \frac{u_{k+t} - u}{u_{k+t} - u_{k+1}} N_{k+1,t-1}(u)$$

Wartości parametru u_j (węzły krzywej) generowane są natomiast według reguły

$$u_j = \begin{cases} 0 & j < t \\ j - t + 1 & t \leq j \leq n \\ n - t + 2 & j > n \end{cases} \quad j = 0, 1, \dots, n + t$$

Przykład:

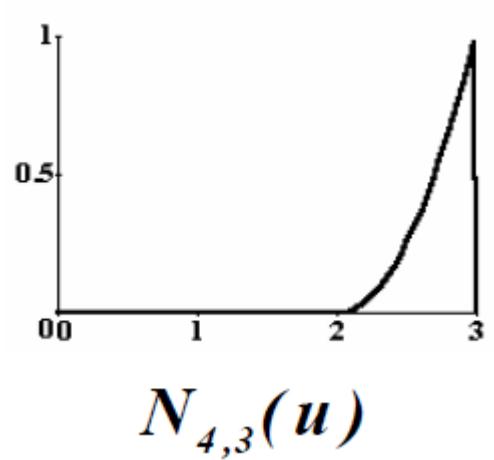
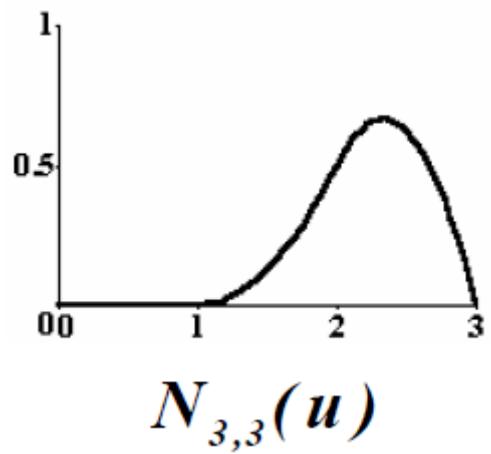
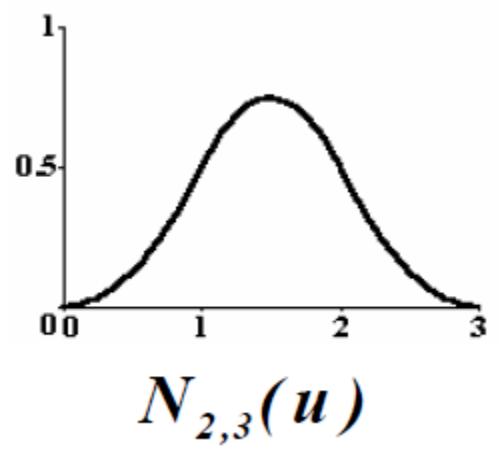
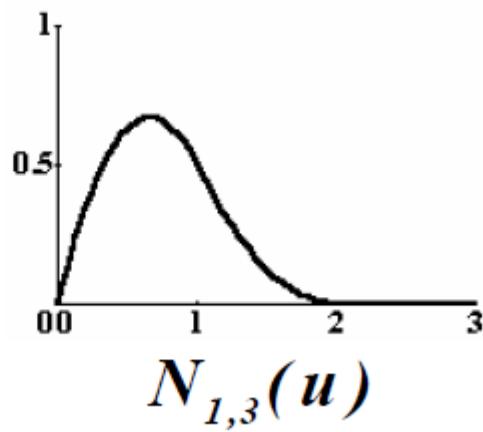
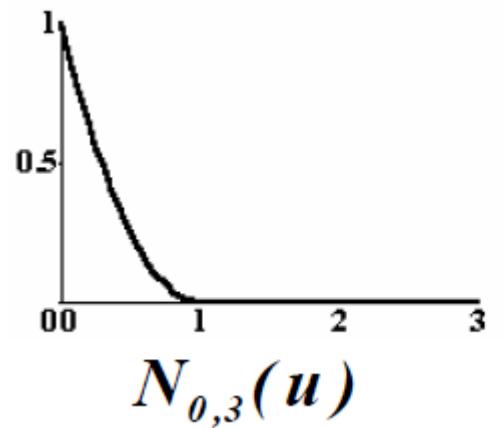
Pięć punktów kontrolnych $n = 4$.

Wielomiany trzeciego stopnia $t = 3$.

Układ węzłów, generowany na podstawie powyższej zależności ma postać

$$\{ u_0, u_1, \dots, u_7 \} = \{ 0, 0, 0, 1, 2, 3, 3, 3 \}$$

Obliczone dla tych węzłów na podstawie zależności rekurencyjnej, wielomiany bazowe mają następujący przebieg...



Podsumowanie:

1. Stopień wielomianów opisujących niejednorodną krzywą B-sklejaną jest dobierany (dodatkowy stopień swobody).
2. Jako dane, projektant podaje punkty kontrolne, punkt początkowy, końcowy i punkty pośrednie.
3. Krzywa rozpoczyna się w punkcie początkowym i kończy w punkcie końcowym.
4. Sterowanie przebiegiem krzywej odbywa się przez manipulacje położeniem punktów kontrolnych.

Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę

Synteza obrazów

Grafika i wizualizacja

Zagadnienia

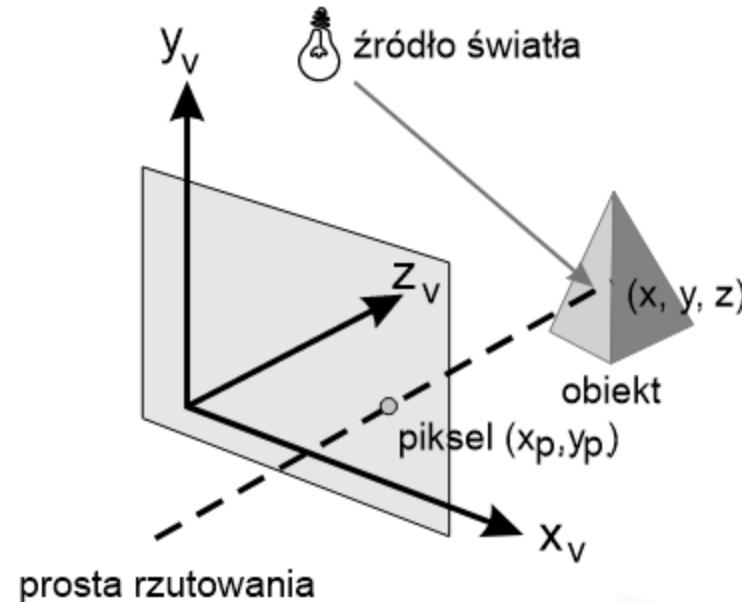
- Modelowanie oświetlenia lokalnego scen 3D
- Metoda śledzenia promieni – Ray tracing
- Metoda energetyczna - Radiosity

Modelowanie oświetlenia lokalnego scen 3D

Wstęp

Generalną ideą służącą modelowaniu oświetlenia lokalnego jest śledzenie drogi światła od źródeł i obiektów.

Jakie czynniki (składowe) wpływają na jasność lub kolor punktu (piksela) (x_p, y_p) , będącego rzutem punktu (x, y, z) , w kontekście źródła światła ?



Stopień jasności lub kolor punktu (piksla) (x_p, y_p) zależy m. in. od następujących czynników:

- geometrii układu (wzajemnego usytuowania obiektu i źródła światła, kształtu obiektu, sposobu rzutowania),
- charakterystyki źródła światła (intensywności świecenia, koloru, tłumienia światła w przestrzeni, kierunkowości),
- charakterystyki powierzchni obiektu (odbijania, rozpraszania, pochłaniania, przepuszczania, koloru powierzchni),
- charakterystyki rozchodzenia się światła odbitego,
- oświetlania obiektu światłem odbitym (np. od innych obiektów znajdujących się na scenie).

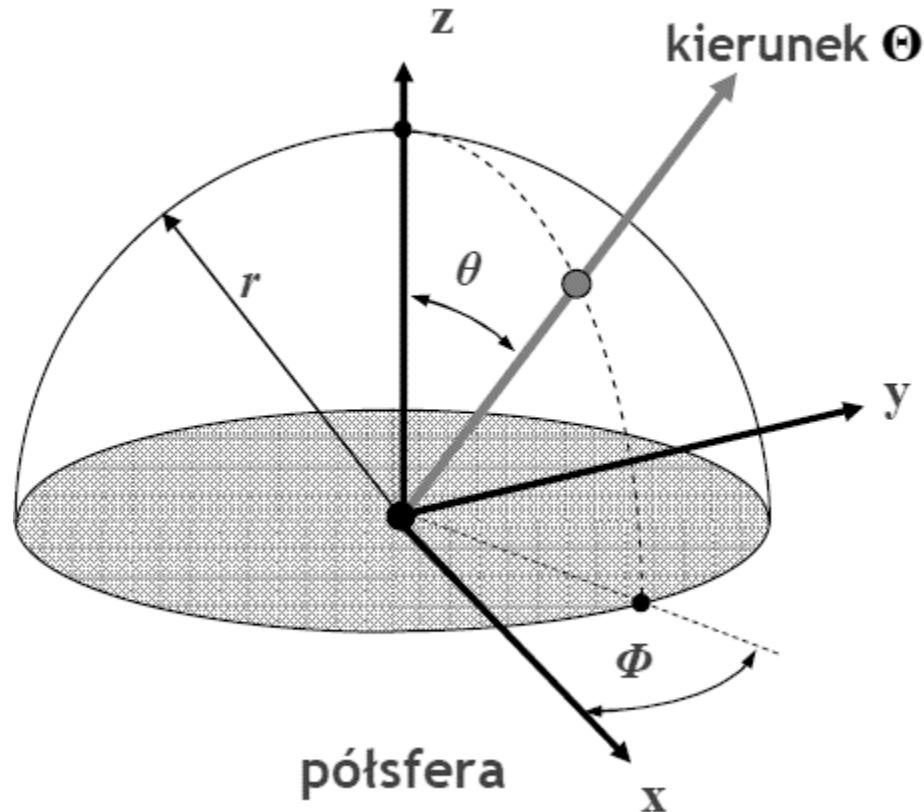
Biorąc pod uwagę powyższe należy odpowiedzieć na pytania:

Jak obliczyć stopień jasności lub kolor punktu (piksla) (x_p, y_p) ?

Jak zmniejszyć ilość obliczeń?

Wyjaśnienie pojęcia funkcji BRDF – Bidirectional Reflectance Distribution Function

Półsfera i kierunek



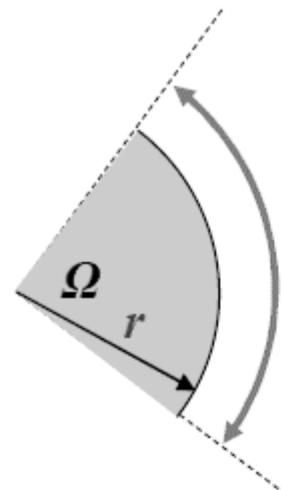
Φ - azymut

Θ - elewacja (nachylenie)

Kąt bryłowy, radiancja

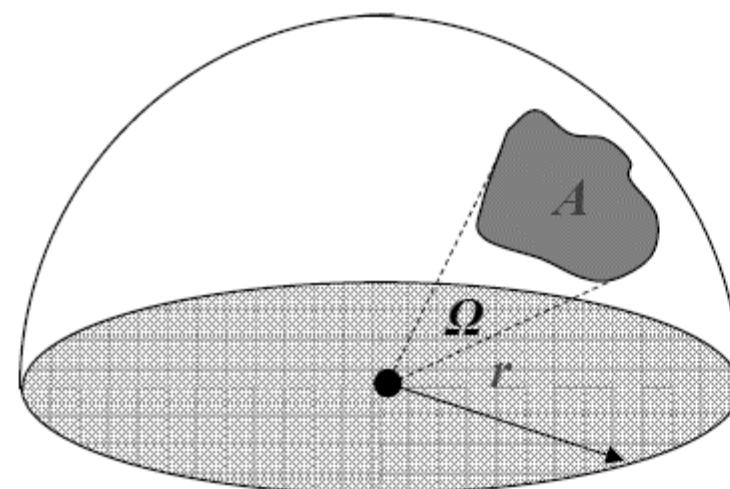
Radiancja (luminancja energetyczna) – moc dostarczana przez jednostkę powierzchni źródła promieniowania do obszaru przestrzeni określonego przez kąt bryłowy [W/(sr*m²)]

kąt płaski



$$\Omega = \frac{l}{r} \text{ [rd]}$$

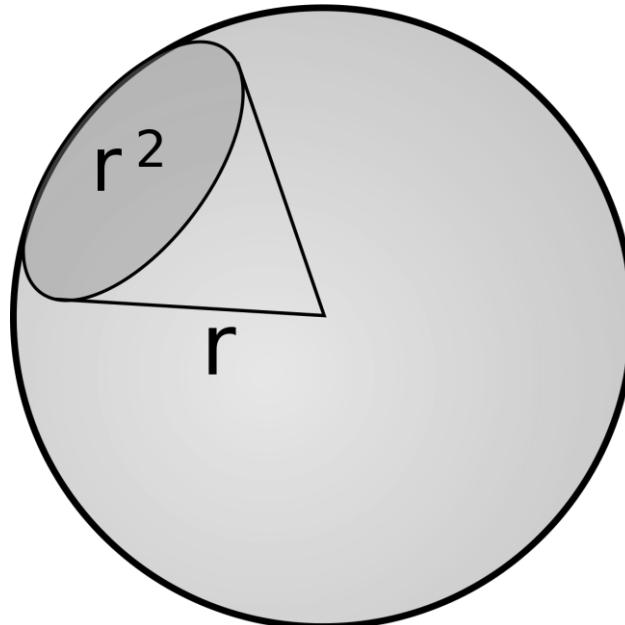
kąt bryłowy



$$\Omega = \frac{A}{r^2} \text{ [sr]}$$

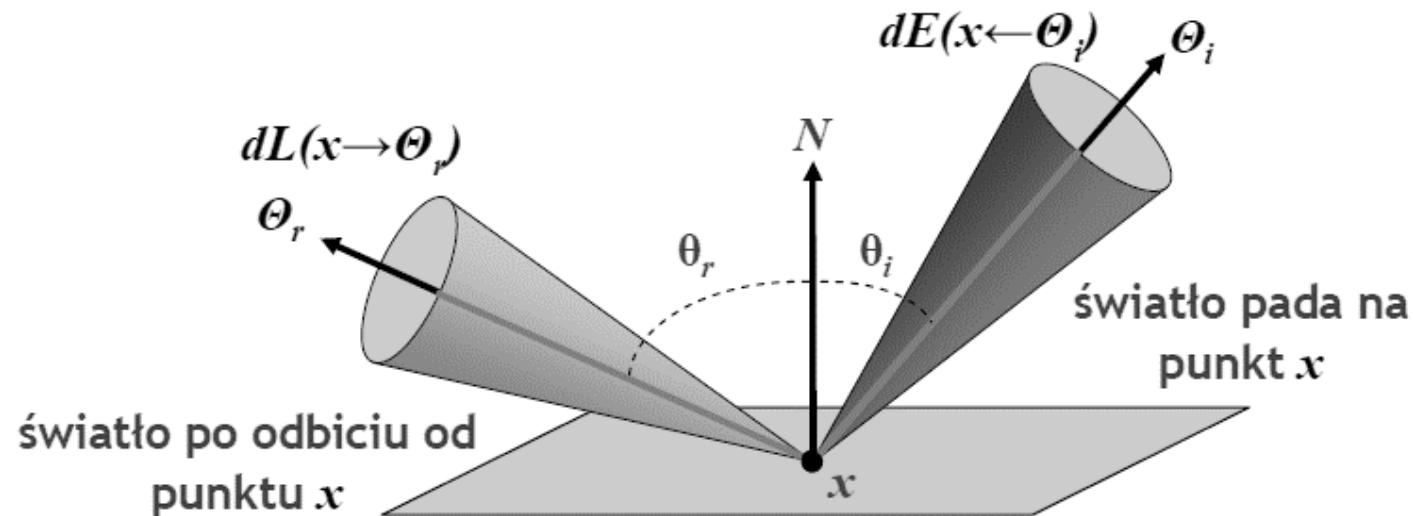
* Steradian [sr]

Steradian (sr od gr. στέρεος – stężały, twardy, masywny, przestrzenny i łac. radius – promień) – niemianowana jednostka pochodna układu SI określająca miarę kąta bryłowego. Jest to kąt bryłowy o wierzchołku w środku kuli, wycinający z powierzchni tej kuli pole równe kwadratowi jej promienia.



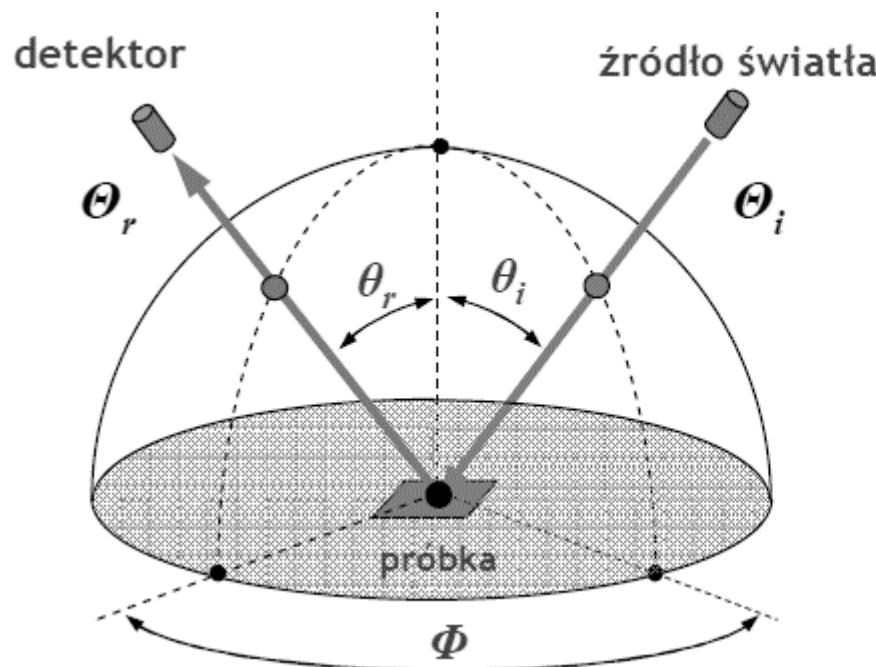
*wyjaśnienie dot. [sr]

Funkcja BRDF – stosunek radiancji odbitej w punkcie x w kierunku Θ_r do radiancji dochodzącej do punktu x z kierunku Θ_i



$$f_r(x, \Theta_i \rightarrow \Theta_r) = \frac{dL(x \rightarrow \Theta_r)}{dE(\Theta_i \rightarrow x)}$$

Gonioreflektometr – przyrząd do pomiaru funkcji BRDF



θ_i - $0 - 90^\circ$
 θ_r - $0 - 90^\circ$
 Φ - $0 - 180^\circ$
kolor - *RGB (3 składowe)*

Zakładając pomiar katów co 1° , do zapisania funkcji BRDF dla punktu na powierzchni potrzeba tablicy o $90 \times 90 \times 180 \times 3 = 4.374.000$ elementach.

Idee modeli oświetlenia

- Model dla oświetlania światłem otoczenia
- Model dla obiektów o odbiciu dyfuzyjnym
- Model dla obiektów o odbiciu zwierciadlanym
- Model dla obiektów przezroczystych

– model oświetlenia światłem otoczenia

Założenia:

- Na scenie występuje jedynie światło rozproszone (bezkierunkowe)
- Powierzchnie obiektów odbijają światło

Model oświetlenia: $I = I_a * k_a$

I - intensywność oświetlenia punktu powierzchni,

I_a - intensywność dla światła rozproszonego,

k_a - współczynnik odbicia światła rozproszonego przez powierzchnię ,
współczynnik zależy od materiału z jakiego wykonany jest obiekt, $k_a \in [0,1]$

– model dla powierzchni dyfuzyjnych

Założenia:

- Na scenie występuje punktowe źródło światła emitujące światło tak samo we wszystkich kierunkach.
- Powierzchnie obiektów rozpraszą światło (są matowe).

Model oświetlenia:

Model wynika z prawa cosinusów Lamberta.

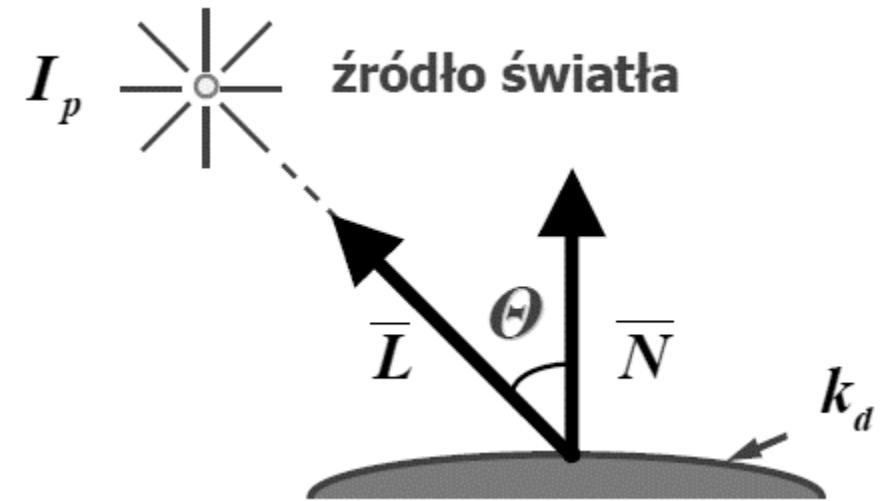
$$I = I_p \cdot k_d \cdot \cos \Theta$$

I - intensywność oświetlenia punktu powierzchni,

I_p - intensywność świecenia punktowego źródła światła,

k_d - współczynnik odbicia przez powierzchnię, współczynnik zależy od materiału z jakiego wykonany jest obiekt, $k_d \in [0,1]$

θ - kąt między kierunkiem padania światła a prostopadłą do oświetlanej powierzchni w badanym punkcie, $\theta \in [-90^\circ, 90^\circ]$



$$I = I_p \cdot k_d \cdot \cos \Theta$$

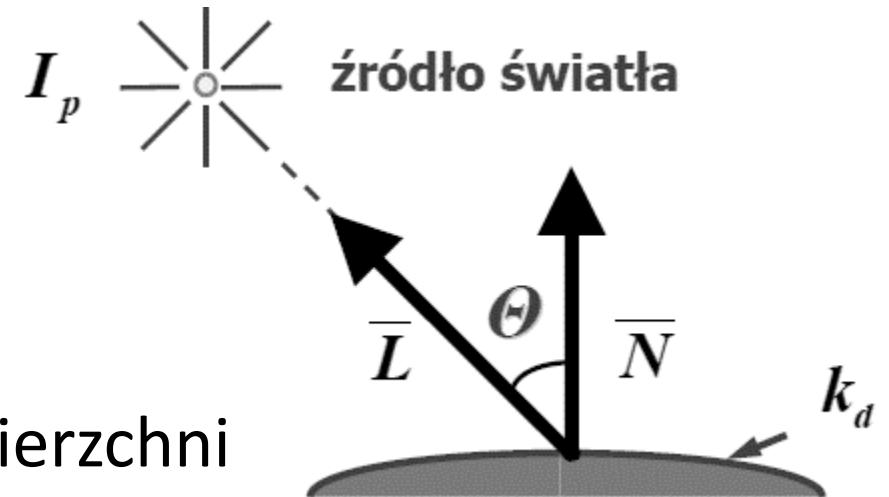
Jeśli odpowiednie kierunki opisać znormalizowanymi wektorami, to model można zapisać jako:

$$I = I_p \cdot k_d \cdot (\bar{N} \cdot \bar{L})$$

gdzie:

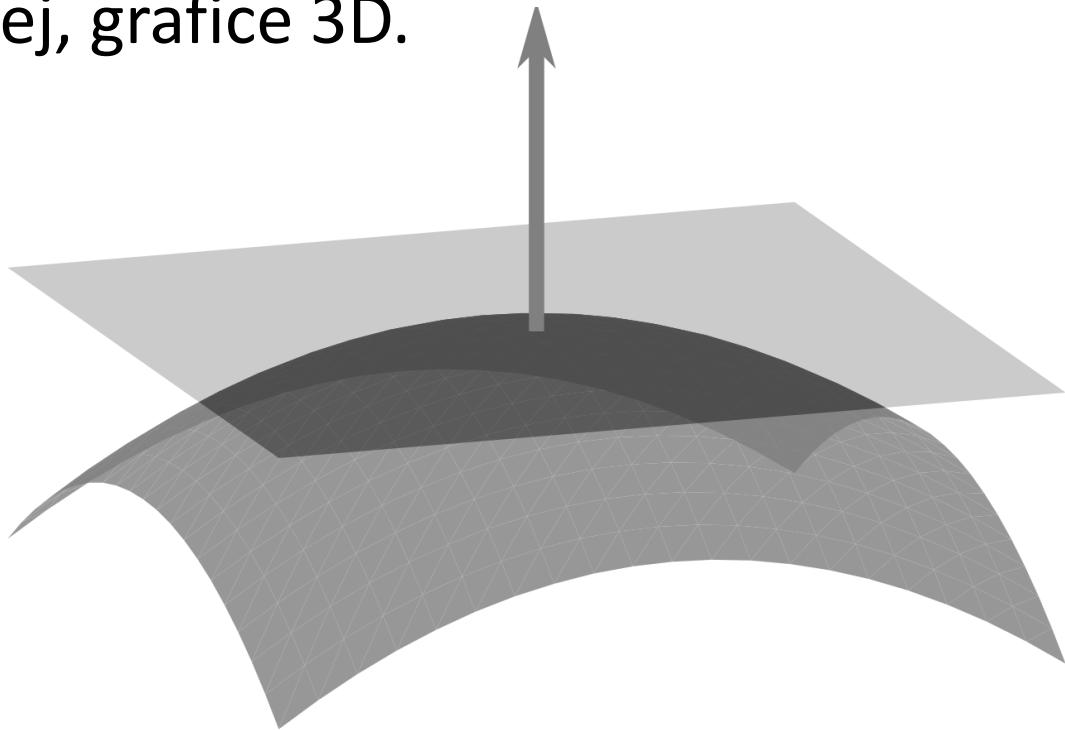
\bar{N} - jednostkowy wektor normalny do powierzchni w badanym punkcie,

\bar{L} - jednostkowy wektor opisujący kierunek padania światła.



*Wektor normalny

Wektor normalny jest to wektor prostopadły do płaszczyzny, lub w wypadku innych powierzchni prostopadły do płaszczyzny stycznej do powierzchni w danym punkcie. Pojęcie to używane jest w matematyce, fizyce, biologii molekularnej, grafice 3D.



Model nie uwzględnia kilku istotnych czynników.

Modyfikacja 1: Uwzględnienie światła rozproszonego

Te elementy sceny, na które nie padają bezpośrednio promienie wysyłane przez punktowe źródło światła nie będą widoczne.

Modyfikacja polega na połączeniu modelu opartego na prawie cosinusów z modelem dla światła rozproszonego.

$$I = I_a \cdot k_a + I_p \cdot k_d \cdot (\bar{N} \cdot \bar{L})$$

gdzie

I_a - intensywność dla światła rozproszonego,

k_a - współczynnik odbicia światła rozproszonego przez powierzchnię .

Modyfikacja 2: Uwzględnienie tłumienia światła emitowanego przez źródło

Z doświadczenia wiadomo, że obiekty położone dalej od źródła światła, są oświetlane słabiej.

Modyfikacja polega na uwzględnieniu zjawiska tłumienia przez wprowadzeniu współczynnika tłumienia f_{att} .

$$I = I_a \cdot k_a + f_{att} \cdot I_p \cdot k_d \cdot (\bar{N} \cdot \bar{L})$$

Jak **uzależnić** współczynnik f_{att} , od odległości między źródłem światła a badanym punktem powierzchni ?

Z fizyki wiadomo, że:

$$f_{att} = \frac{1}{d_L^2}$$

gdzie d_L jest odlegością pomiędzy źródłem światła a punktem oświetlanej powierzchni.

W praktyce powyższy wzór nie daje zbyt dobrych wyników, bowiem:

- jeśli d_L jest duże, f_{att} zmienia się nieznacznie nawet dla daleko położonych od siebie powierzchni,
- jeśli d_L jest małe, f_{att} zmienia się bardzo znacznie nawet dla blisko położonych od siebie powierzchni.

W grafice komputerowej stosuje się wiec bardziej ogólną zależność w postaci:

$$f_{att} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$$

gdzie c_1, c_2, c_3 są stałymi dobieranymi empirycznie.

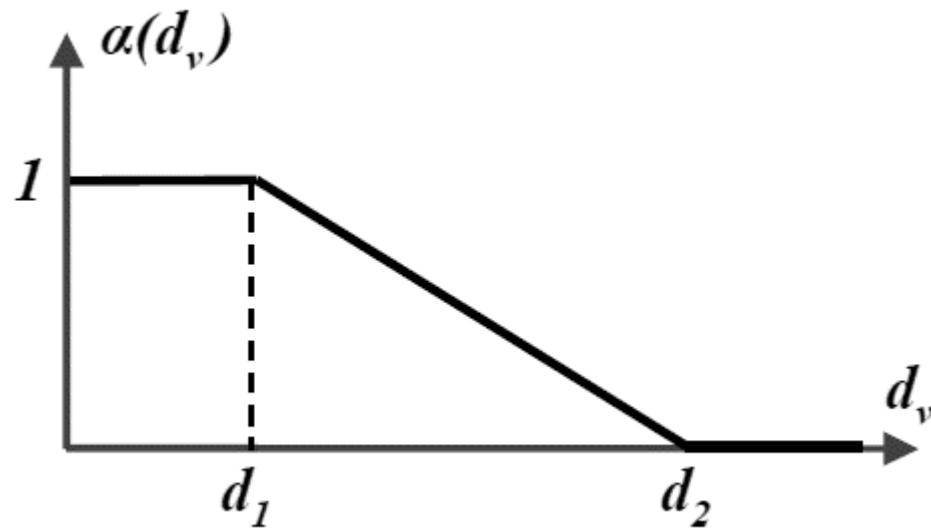
Modyfikacja 3: Uwzględnienie odległości oświetlonego obiektu od obserwatora

Z doświadczenia wiadomo, że obiekty położone dalej od obserwatora, są postrzegane jako oświetlane słabiej.

Zjawisko to uwzględnia się w prosty sposób, modyfikując wyznaczoną przy pomocy poprzednio omówionych modeli intensywność i następująco:

$$I' = I \cdot \alpha(d_v)$$

d_v - odległość oświetlanego punktu powierzchni od obserwatora,
 $\alpha(d_v)$ - funkcja o przebiegu pokazanym na rysunku (przykład takiej funkcji)

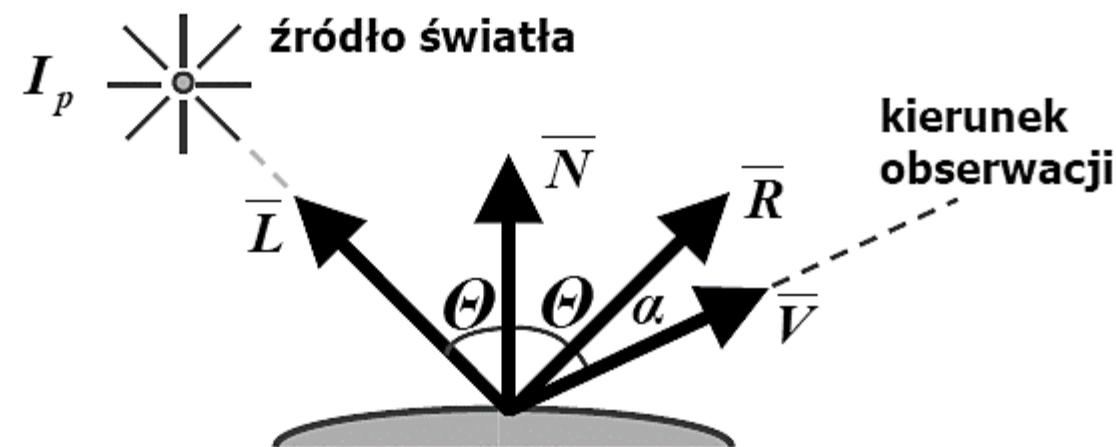


– model dla powierzchni o odbiciu kierunkowym

Założenia:

- Na scenie występuje punktowe źródło światła emitujące światło tak samo we wszystkich kierunkach.
- Powierzchnie obiektów odbija światło (różnie w różnych kierunkach).

Przykład: Idealne zwierciadło



Wydaje się sensowne poszukiwanie modelu powierzchni, który łączy własności rozpraszania i odbijania światła.

Model oświetlenia Phonga: (Phong Bui Tuong)

$$I = I_a \cdot k_a + f_{att} \cdot I_p [k_d \cdot \cos \Theta + W(\Theta) \cdot \cos^n \alpha]$$

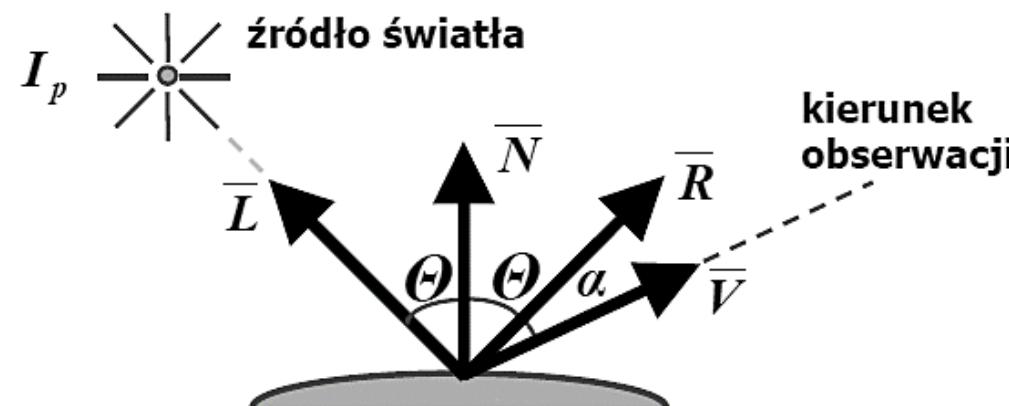
$W(\Theta)$ - pewna funkcja kąta (Θ) (zależy od własności materiału),

n – stała, gdzie n należy do przedziału [1, 200]

Często funkcję tą zastępuje się stałą k_s , czyli parametr nie zależy wtedy od kąta pod jakim światło pada na analizowany punkt powierzchni.

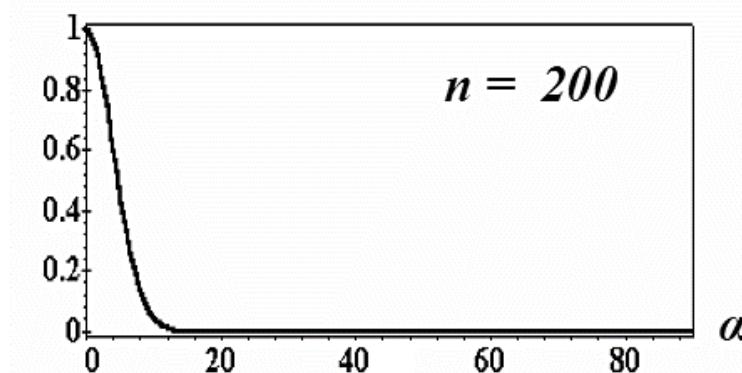
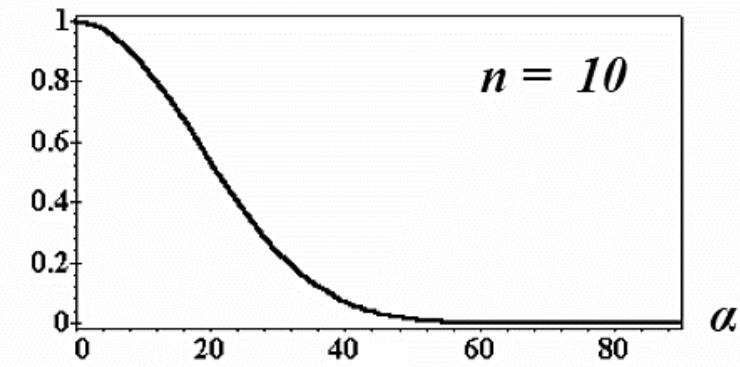
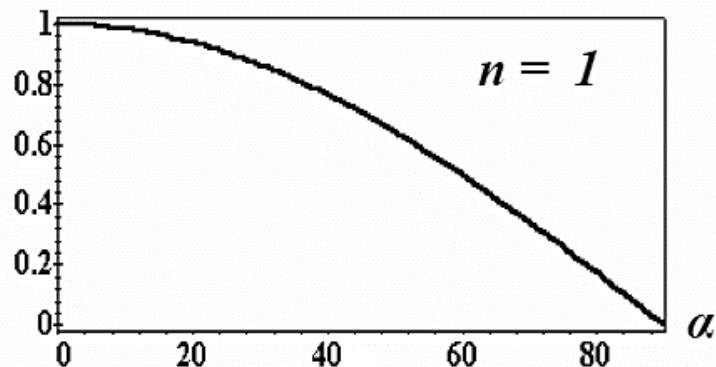
Model Phonga można wtedy zapisać w postaci:

$$I = I_a \cdot k_a + f_{att} \cdot I_p [k_d \cdot (\bar{N} \cdot \bar{L}) + k_s \cdot (\bar{V} \cdot \bar{R})^n]$$

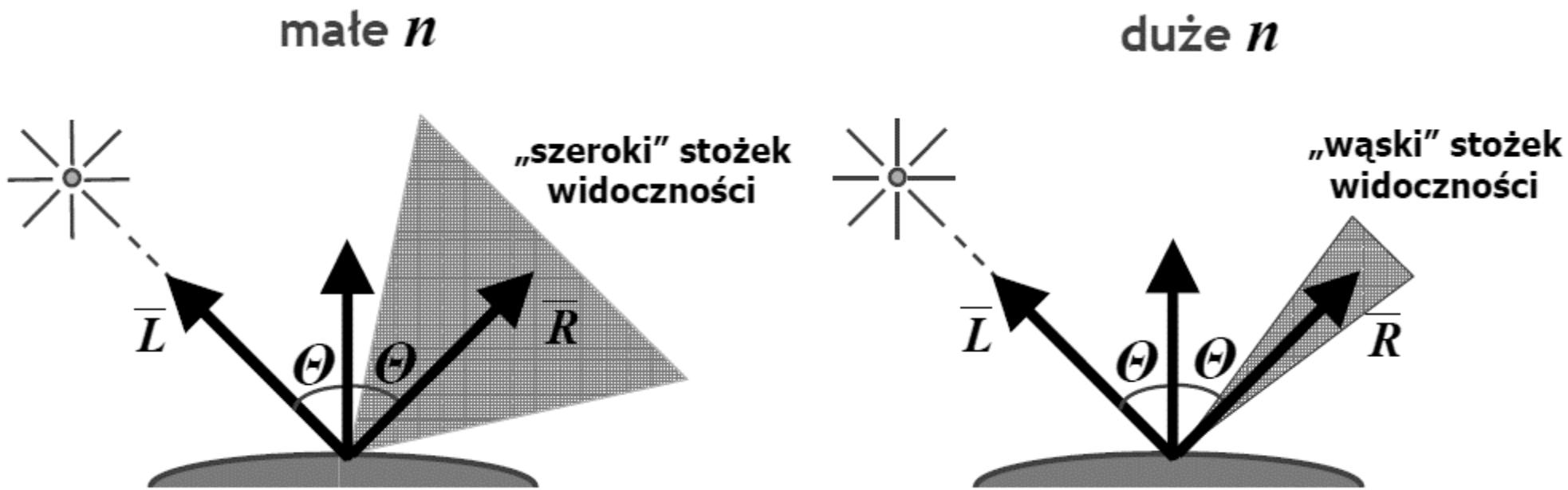


W modelu podstawową rolę odgrywa składnik $\cos^n \alpha$, który uzależnia intensywność oświetlenia punktu powierzchni od kąta obserwacji α . Wyjaśnić to można badając przebieg funkcji $\cos^n \alpha$.

Przebieg funkcji $\cos^n \alpha$:



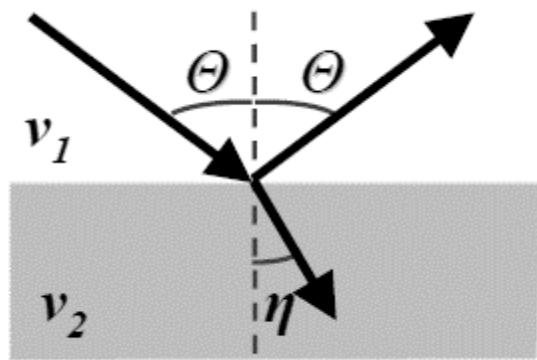
Wniosek jest następujący:



Jeżeli $n \rightarrow \infty$ uzyskuje się prawie idealne zwierciadło.

– model dla obiektów przezroczystych

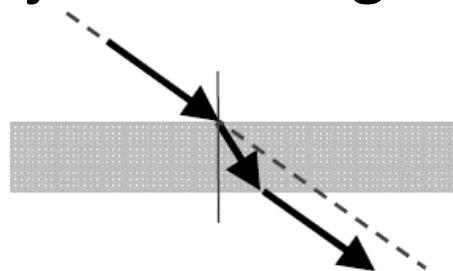
Przy przejściu z jednego ośrodka przezroczystego do drugiego promień świetlny ulega załamaniu (rys.). Zasadę tą opisuje prawo Snella.



$$\frac{\sin \Theta}{\sin \eta} = \frac{v_1}{v_2}$$

gdzie: v_1 i v_2 są prędkościami rozchodzenia się światła w pierwszym i drugim ośrodku.

Przy przejściu przez przezroczystą płytę następuje podwójne załamanie, które powoduje równoległe przesunięcie promienia.

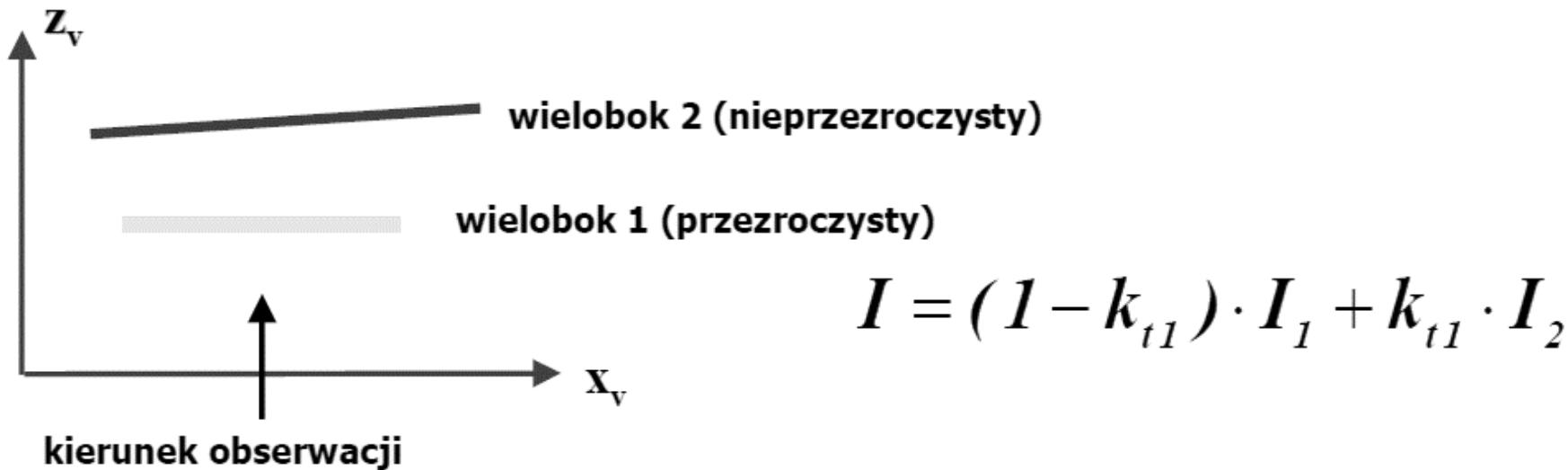


Przesunięcie promienia zależy od poprzednio wymienionych parametrów i grubości płytki.

Niniejsze podstawy fizyczne stosuje się do budowy modeli przechodzenia światła przez obiekty .

Przykładowo, modele opisujące przechodzenie światła przez szyby, są one wykorzystywane w symulatorach lotu.

Przykład zastosowania prostego modelu przezroczystości



Gdzie:

I_1 – intensywność oświetlenia wieloboku 1,

I_2 – intensywność oświetlenia wieloboku 2,

k_{t1} – współczynnik przezroczystości wieloboku 1, gdzie k_{t1} należy do przedz. [0,1] (jeśli $k_{t1} = 0$ – wielobok 1 jest nieprzezroczysty, jeśli 1 – wielobok 1 jest całkowicie przezroczysty)

Algorytmy generacji obrazów scen oświetlonych

Rendering

Proces cieniowania, obliczania jasności (koloru) poszczególnych pikseli obrazu (sceny) z uwzględnieniem efektów oświetlenia (i innych).

Algorytmy generacji obrazów scen oświetlonych

Algorytm bezpośredni renderingu:

1. Dla punktu obrazu (piksla) o współrzędnych (x_p, y_p) wyznaczyć odpowiadający punkt na widocznej powierzchni obiektu (x, y, z) .
2. Dla punktu (x, y, z) zastosować wybrany model oświetlenia i obliczyć intensywność i analizowanego punktu (operacja złożona).
3. Nadać pikselowi (xp, yp) kolor zgodnie z obliczoną intensywnością.

Zaletą powyższego postępowania jest dokładność odwzorowania fizycznych zjawisk, wadą jest znacząca liczba obliczeń.

Algorytm cieniowania jednotonowego (cieniowanie płaskie)

Założenia:

- Obiekty sceny opisane są jako siatki wieloboków,
- Dla widocznej ściany obiektu sceny, zakłada się stałą intensywność oświetlenia.

Algorytm:

1. Dla dowolnego punktu widocznej ściany wyliczyć intensywność oświetlenia.
2. Rzutować ścianę, wypełniając odpowiedni wielobok stałą wartością, zgodnie z obliczoną w punkcie 1. intensywnością.

Zaleta: stosunkowo mało obliczeń.

Wada: jeśli siatka wieloboków aproksymuje obiekt o płynnych kształtach – widoczne będą krawędzie (obiekt przestaje być gładki, wielokąty o różnej intensywności oświetlenia „odcinają” się od siebie na krawędziach).

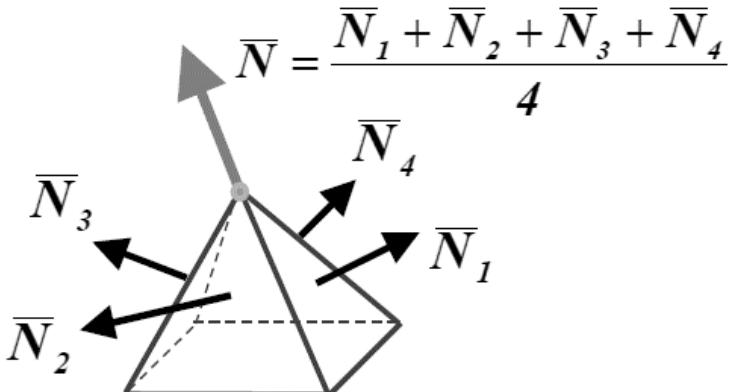
Algorytm interpolacji intensywności (algorytm Gouraud'a)

Założenie:

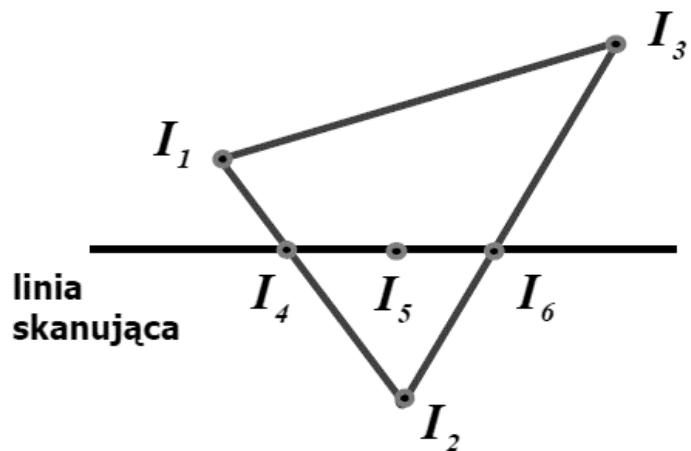
Obiekty sceny opisane są jako siatki wieloboków.

Algorytm:

1. Dla każdego wierzchołka siatki wieloboków wyliczyć „wektor normalny” jako średnią arytmetyczną wektorów normalnych dla ścian, do których należy analizowany wierzchołek.



2. Dla każdego wierzchołka siatki, stosując wybrany model oświetlenia i obliczony „wektor normalny”, obliczyć intensywność oświetlenia I_j .
3. Wypełniać rzuty widocznych wieloboków, używając algorytmy linii skanującej, w poniższy sposób (dzięki temu wielokąt będzie wypełniony nierównomiernie w sposób „zacierający” różnice na krawędziach):



$$I_4 = I_1 \cdot \frac{y_4 - y_2}{y_1 - y_2} + I_2 \cdot \frac{y_1 - y_4}{y_1 - y_2}$$

$$I_6 = I_3 \cdot \frac{y_6 - y_2}{y_3 - y_2} + I_2 \cdot \frac{y_3 - y_6}{y_3 - y_2}$$

$$I_5 = I_4 \cdot \frac{x_4 - x_5}{x_6 - x_4} + I_6 \cdot \frac{x_5 - x_4}{x_6 - x_4}$$

Zaleta cieniowania Gouraud'a:

Ograniczenie ilości obliczeń.

Obiekty aproksymowane siatkami wieloboków wyglądają „gładko” (krawędzie przestają być widoczne).

Wada:

Nienaturalne efekty w przypadkach odbicia światła od powierzchni lustrzanych.

Algorytm interpolacji wektorów normalnych (algorytm Phong'a):

Założenie:

Obiekty sceny opisane są jako siatki wieloboków.

Algorytm:

1. Obliczyć „średnio-arytmetyczne wektory normalne” dla wierzchołków wieloboków siatki tak samo jak w poprzednim algorytmie.

2. Wypełniać rzuty widocznych wieloboków używając algorytmu linii skanującej, lecz interpolując nie intensywności a „wektory normalne” obliczone dla wierzchołków.

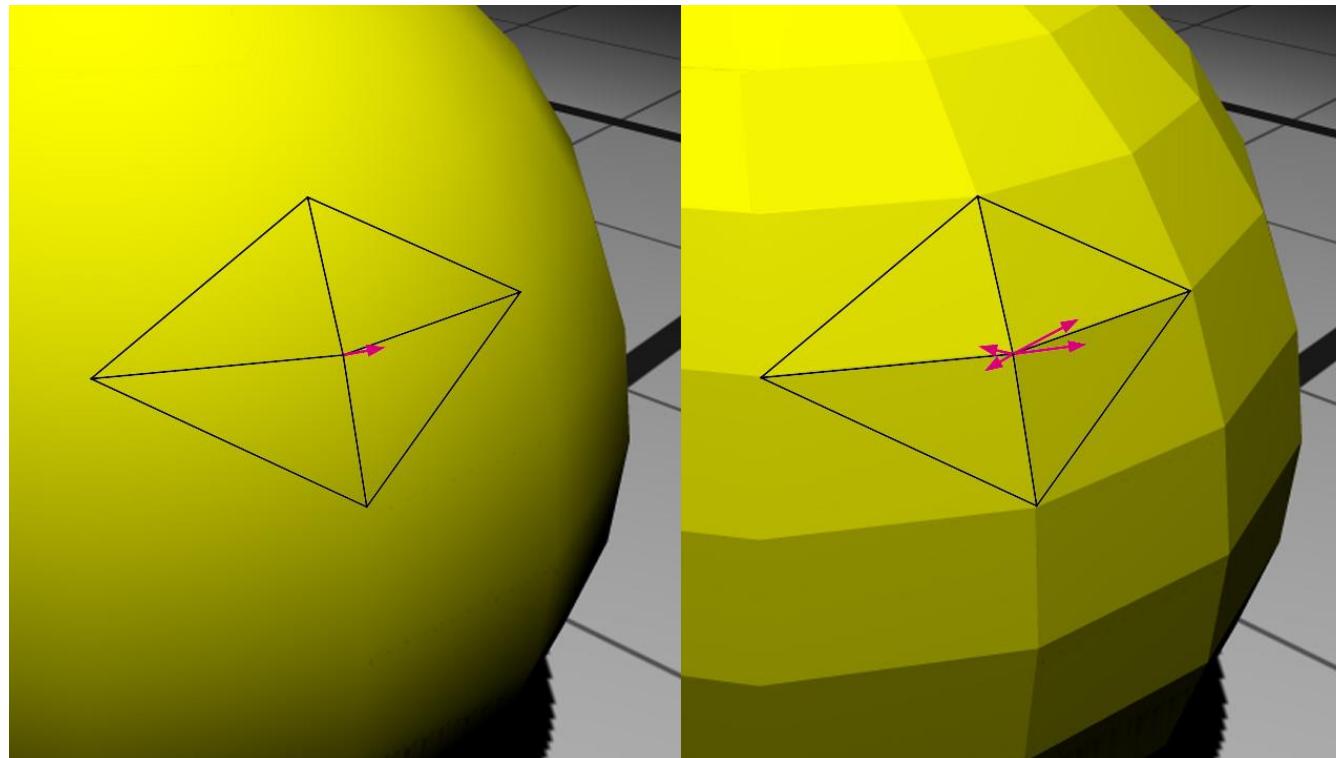
Zaleta: znacznie lepsze efekty niż w przypadku algorytmu Gouraud i cieniowaniu płaskim.

Wada: większa liczba obliczeń (dla każdego piksela wykorzystywany jest model oświetlenia).

*ilustracja różnic w modelowaniu oświetlenia



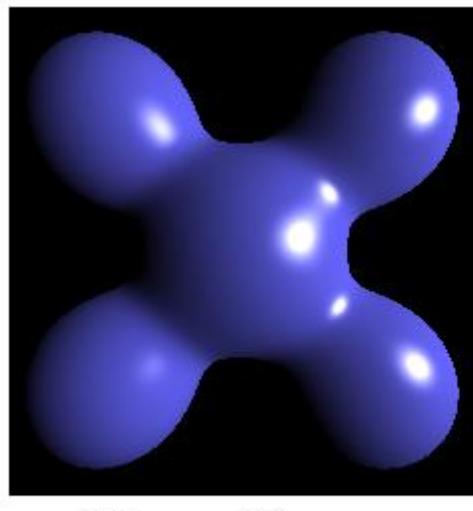
Wielokąty sprowadzone do trójkątów



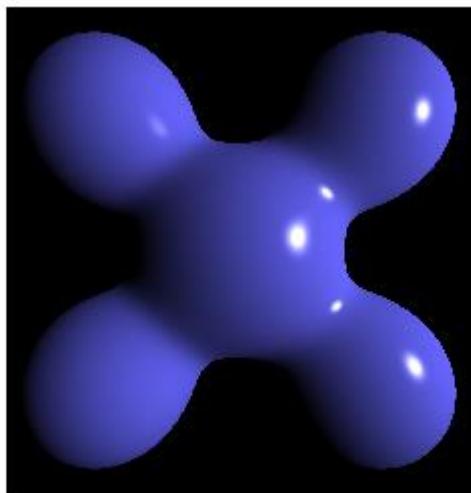
Smooth triangle
(model Phonga)

Trójkąty zwykłe
(cieniowanie płaskie)

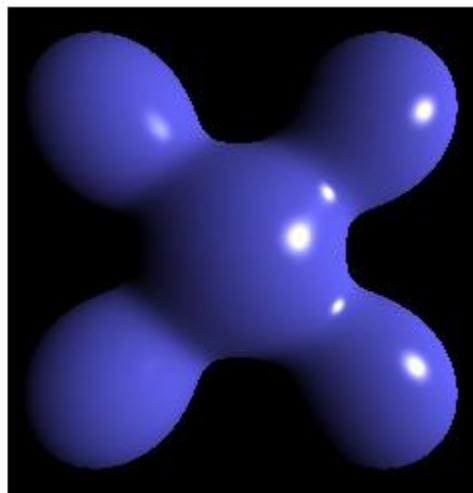
*wersje oświetlenia w modelu Phonga (po środku model podstawowy)



Blinn-Phong



Phong



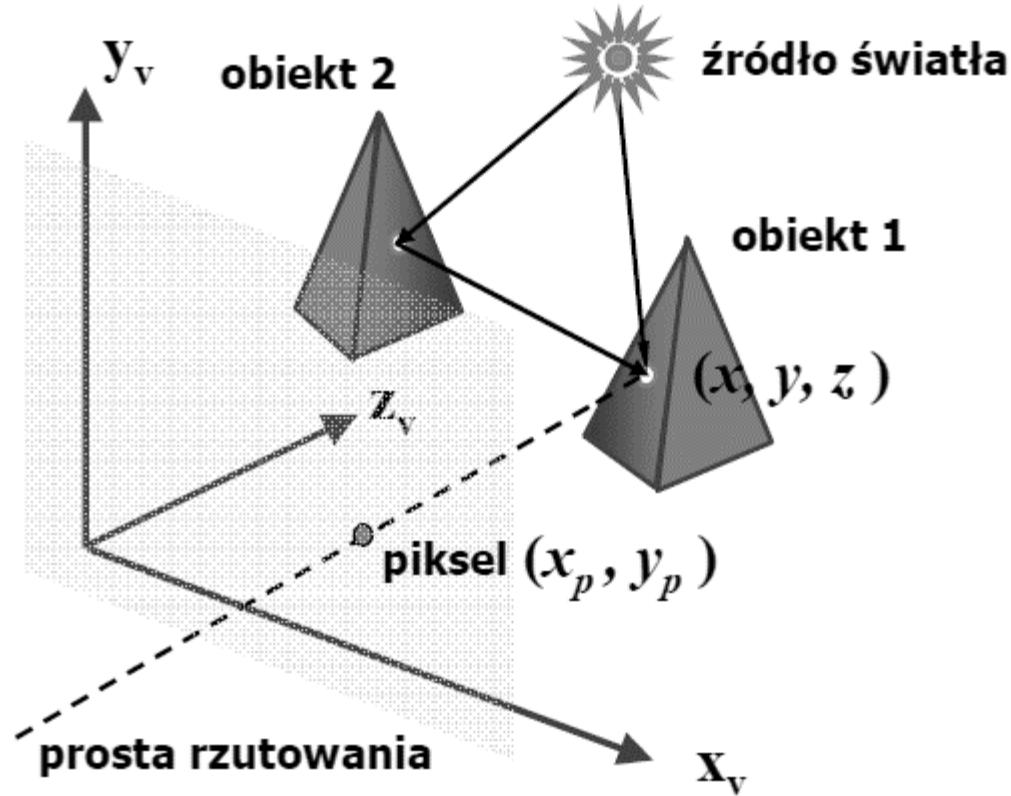
Blinn-Phong
(higher exponent)

Fizyczne realizacje modeli empirycznych

Metoda śledzenia promieni – Ray tracing

Sformułowanie problemu

Punkt (x, y, z) na ścianie obiektu 1 jest oświetlany bezpośrednio przez źródło światła a także przez źródło światła odbite od obiektu 2.



Jak obliczyć jasność lub kolor punktu (piksla) (x_p, y_p) będącego rzutem punktu (x, y, z) ?

Metoda próbkowania przestrzeni

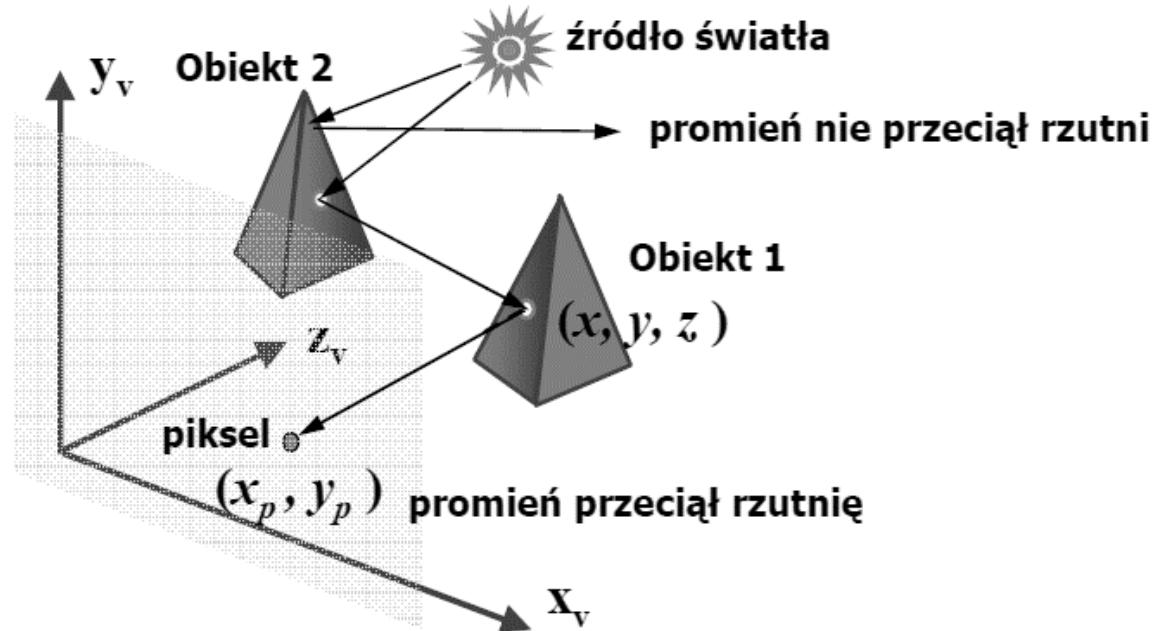
Założenie:

Źródło światła reprezentowane jest przez pęk promieni biegących w różnych kierunkach (dyskretyzacja źródła światła)

Algorytm:

1. Analizuje się poszczególne promienie emitowane przez źródło światła w kierunku od źródła światła do rzutni.
2. Wylicza się kolejne kierunki odbicia analizowanego promienia od ścian obiektów, aż do wyznaczenia kierunku ostatniego odbicia promienia.
3. Prosta wyznaczona przez ostatni kierunek odbicia analizowanego promienia przecina rzutnię, bądź nie.

Metoda próbkowania przestrzeni



Wady metody:

W miarę dokładne próbkowanie przestrzeni wymaga analizy wielu promieni wychodzących ze źródła (źródeł) światła.

Prawdopodobnie znaczna większość analizowanych promieni nie przetnie rzutni (i będzie badana niepotrzebnie).

Metoda śledzenia promieni - idea

Whitted T. „An illumination model for shaded display”, Communications of the ACM, vol. 23, No. 6, 1980

Założenia:

Na scenie może znajdować się wiele obiektów i wiele źródeł światła.

Możliwe jest obliczenie punktu przecięcia prostej (promienia) i powierzchni obiektu.

Opisany jest sposób oddziaływanego powierzchni obiektów i światła (zadane są modele oświetlenia lokalnego).

Inne metody: Bada się drogę fikcyjnego promienia biegącego od obserwatora przez piksel ekranu w głąb sceny.

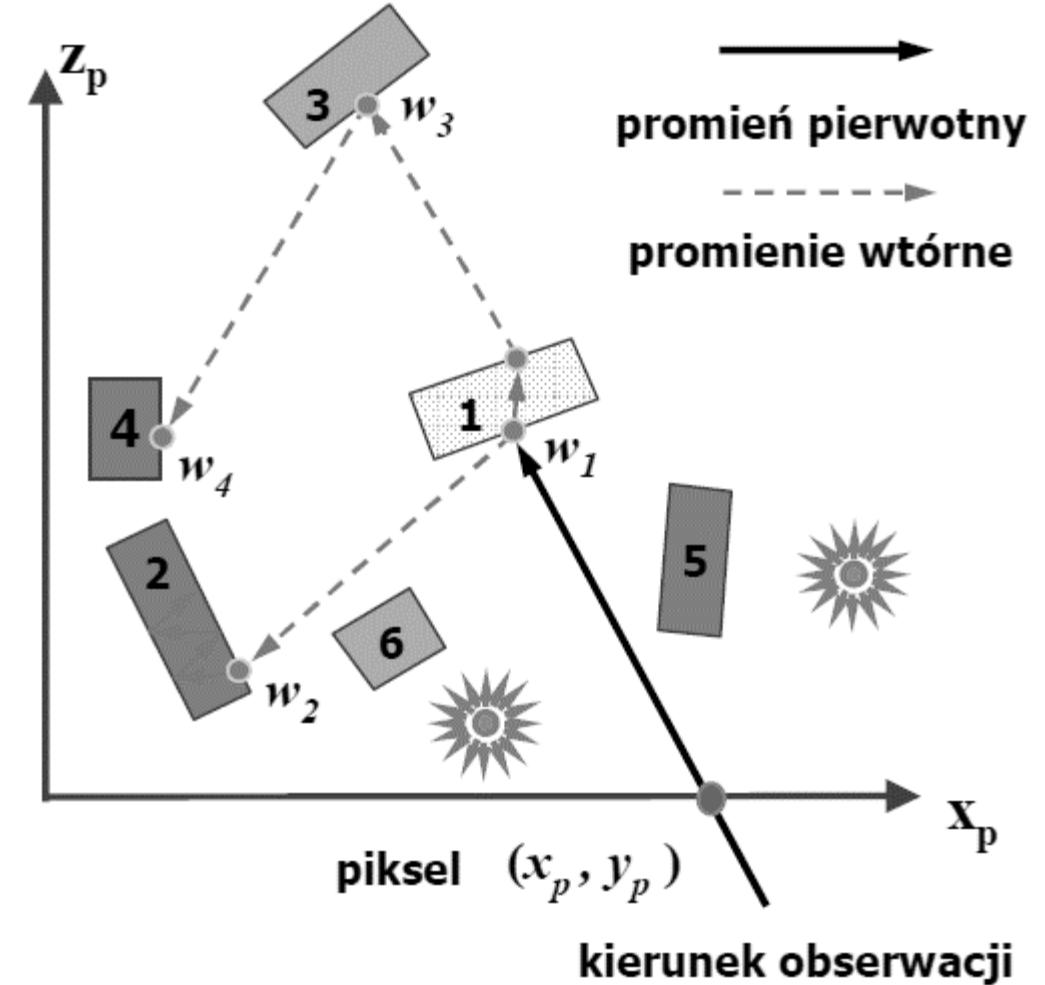
Śledzenie (wyliczanie kolejnych kierunków biegu promienia) prowadzone jest w oparciu o geometrię analizowanej sceny i określony z góry sposób oddziaływania obiektów i światła. Kończy się gdy:

- promień nie trafia w żaden obiekt na scenie, wtedy analizowany piksel jest wypełniany kolorem tła,
- następuje ostatnie odbicie promienia, czyli takie, po którym nie trafia on już w żaden obiekt analizowanej sceny,
- promień rozprasza się po osiągnięciu obiektu, którego powierzchnia jest matowa (ma charakterystykę dyspersyjną),
- promień trafia na obiekt, w którym następuje całkowite odbicie wewnętrzne.

Przykład:

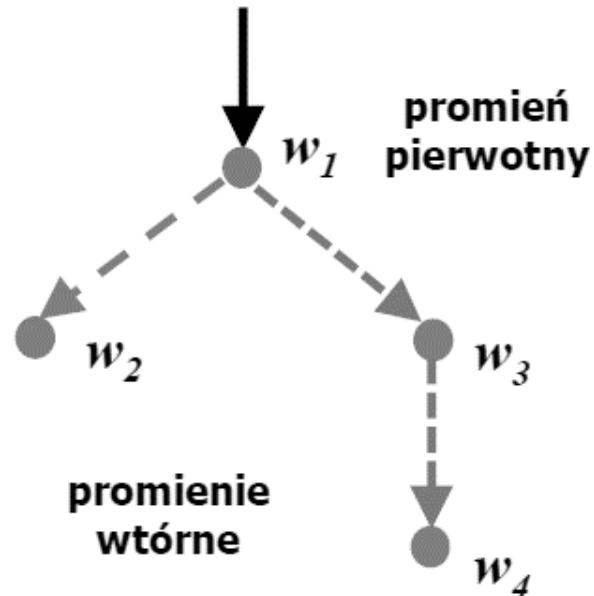
Na scenie jest 6 obiektów i dwa źródła światła:

- 1 – odbija i załamuje promień,
- 2 – występuje całkowite odbicie wewnętrzne,
- 3 – odbija promień,
- 4 – rozprasza promień,
- 5 – zasłania źródło światła,
- 6 – tłumii promień, ale nie zasłania źródła światła

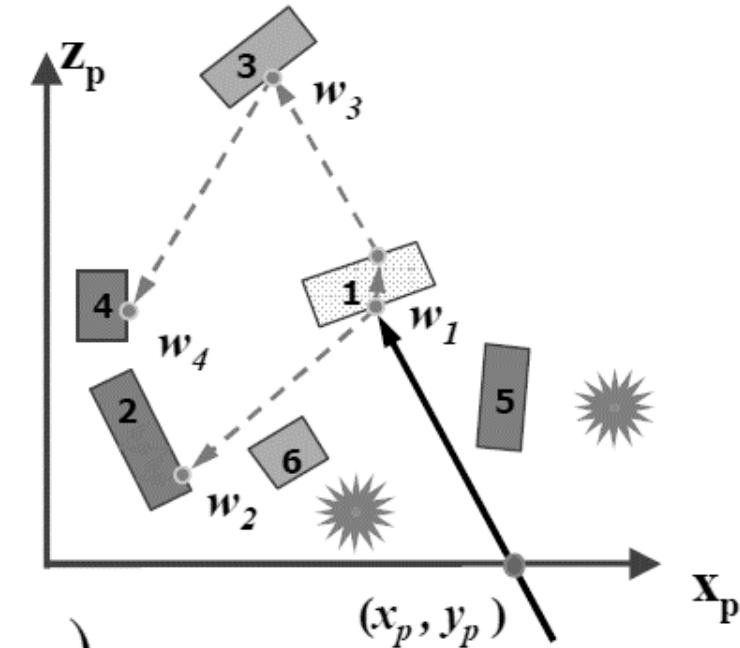


Metoda śledzenia promieni - algorytm

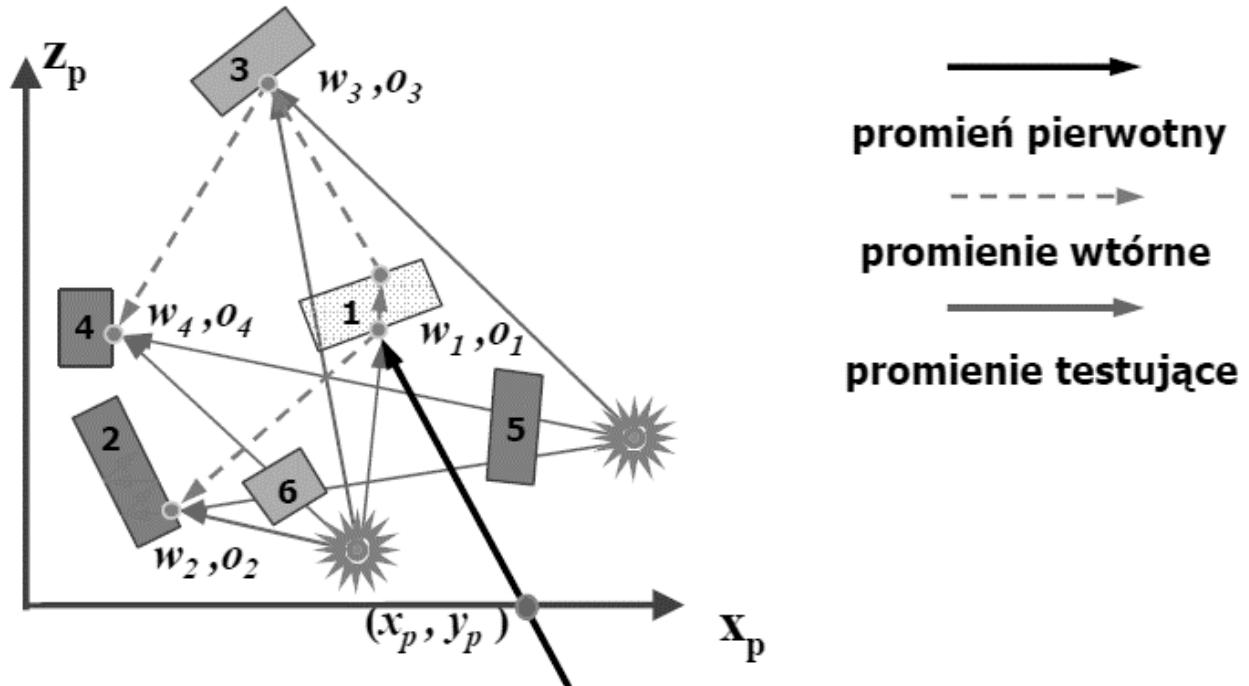
Krok 1: Przez każdy punkt (piksel) ekranu przeprowadza się promień pierwotny i wyznacza promienie wtórne (śledzenie). Dalej, dla analizowanego punktu ekranu buduje się graf w postaci drzewa.



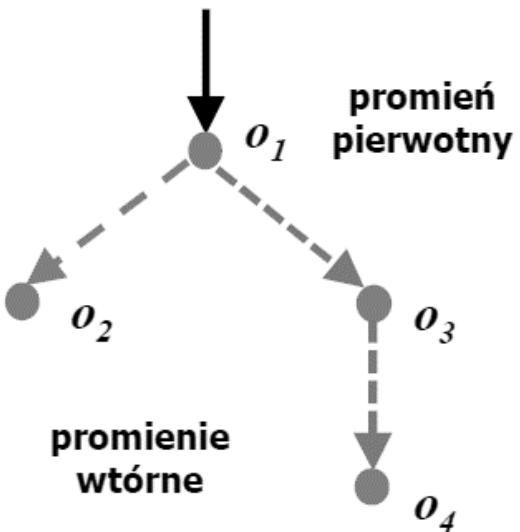
$$w_i \rightarrow (x_i, y_i, z_i)$$



Krok 2: Dla każdego węzła w_i drzewa, na podstawie analizy promieni testujących (łączących bezpośrednio źródło światła i węzeł), wyznacza się oświetlenie lokalne węzła o_i .

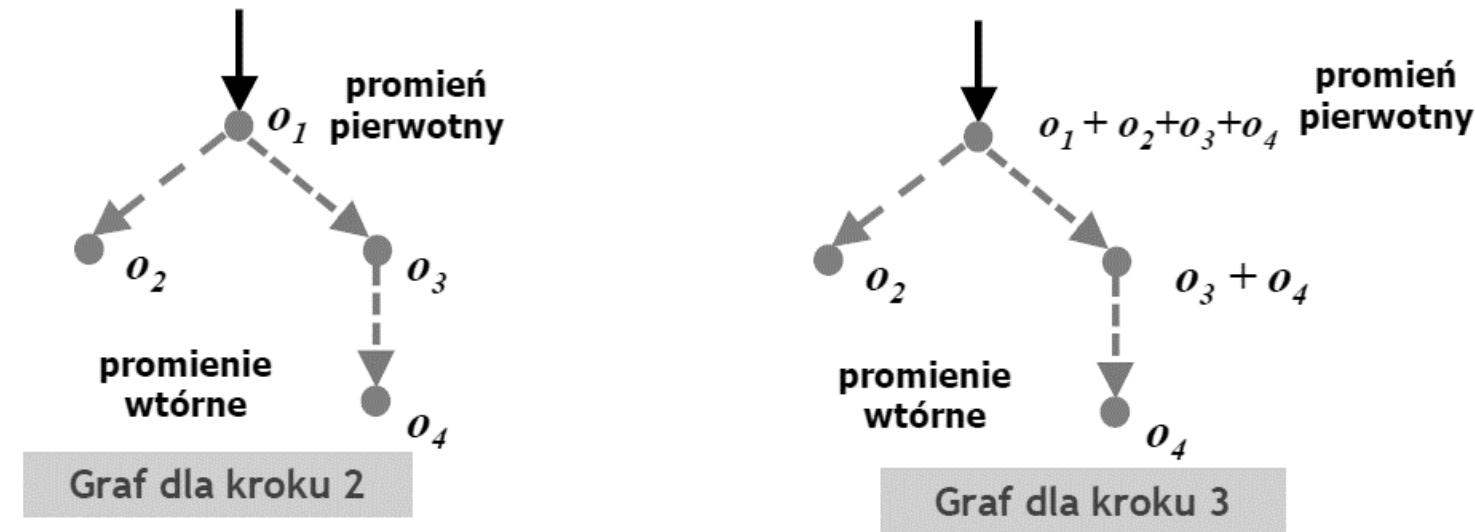


Modyfikuje się wyznaczony w kroku 1 graf, przypisując węzłom w_i obliczone oświetlenia lokalne o_i .



Oświetlenia lokalne o_i są sumami oświetleń pochodzących od wszystkich widocznych z węzła w_i źródeł światła.

Krok 3: Sumuje się oświetlenia lokalne przypisane węzłom grafu. Oświetlenie węzła „wyższego” uzyskuje się przez dodanie do jego oświetlenia lokalnego, oświetlenia węzłów „niższych”.



Punkt ekranu (piksel) wypełnia się odpowiednio do oświetlenia „najwyższego” węzła grafu.

Metoda śledzenia promieni – wady

1. Mogą powstać efekty aliasingowe, powodowane analizowaniem nieskończego wąskich promieni:

- „małe” obiekty mogą zostać pominięte,
- ostre krawędzie mogą być zniekształcone.

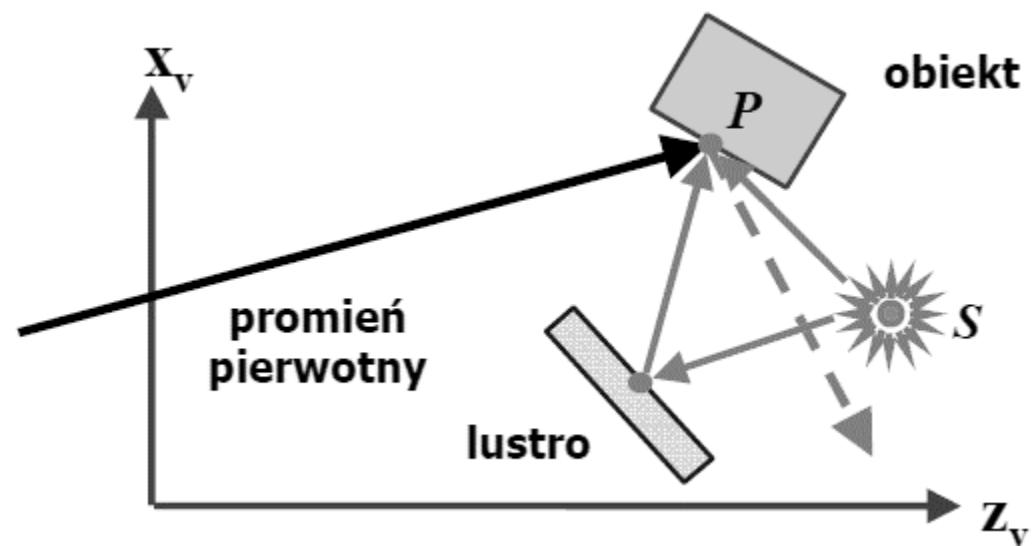
W celu uniknięcia wymienionych efektów stosuje się rozproszone śledzenie promieni, które polega na zastąpieniu nieskończego wąskiego promienia wiązką promieni.

2. Złożoność obliczeniowa metody jest znaczna.

3. Nie wszystkie kierunki padania światła na powierzchnie obiektów są rozpatrywane, co powoduje czasem błędy w wyznaczaniu oświetlenia.

Przykład:

Metoda prowadzi w tym przypadku do błędного wyznaczenia oświetlenia.

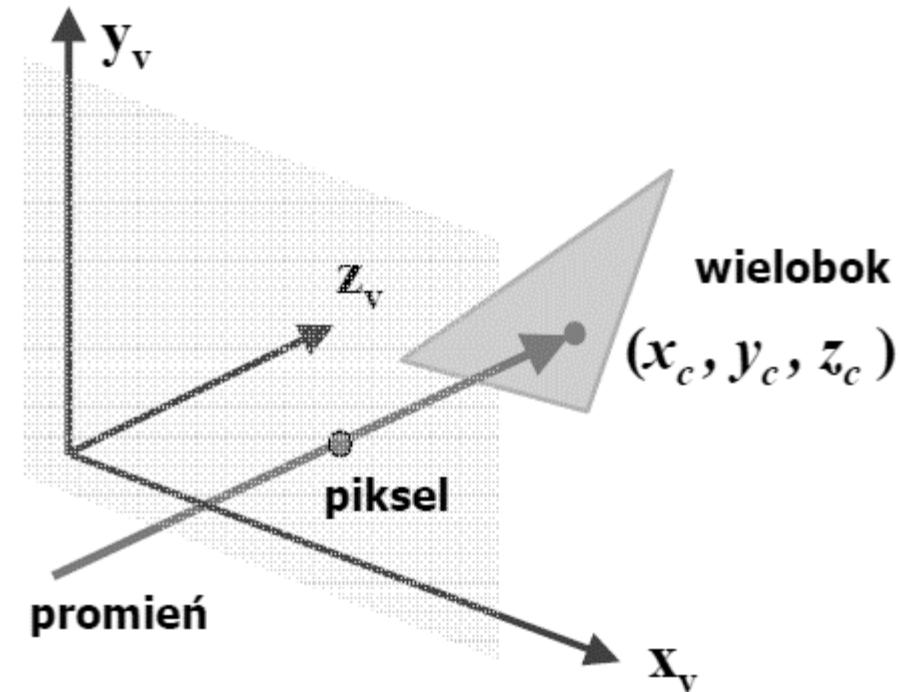


Jeśli zastosować metodę śledzenia promieni, to punkt P będzie oświetlony jedynie przez źródło światła S. W rzeczywistości, punkt P oświetlany jest przez źródło światła S i światło odbite od lustra.

Obliczenia geometryczne w metodzie śledzenia promieni

W metodzie śledzenia promieni, większość nakładu obliczeniowego pochłania wyznaczanie punktów przecięć promieni z powierzchniami obiektów.

Znając kierunek promienia i opis wieloboku, należy wyznaczać współrzędne punktu przecięcia (x_c , y_c , z_c).



Upraszczanie obliczeń w metodzie śledzenia promieni

Rozróżnia się trzy podstawowe rodzaje sposobów upraszczania obliczeń:

1. Redukcję kosztów śledzenia pojedynczego promienia,
2. Redukcję liczby przecięć na drodze promienia,
3. Redukcję gęstości próbkowania.

Ad. 1. Redukcja kosztów śledzenia pojedynczego promienia:

Dla skomplikowanego obiektu po wykonaniu wielu obliczeń może okazać się, że promień nie przecina obiektu, czyli wykonane rachunki przeprowadzone zostały niepotrzebnie.

Stosuje się więc różne sposoby na zmniejszenie szansy wykonywania niepotrzebnych obliczeń, np. Metodę brył otaczających

Metoda brył otaczających

Skomplikowane obiekty otacza się bryłami, które powinny:

- „ciasno” obejmować obiekt,
- mieć prosty opis matematyczny.

*rys.

Przykładami stosowanymi w praktyce są kula, czy sześciian.

Jeżeli stwierdzi się, że promień nie przecina bryły otaczającej to jest oczywiste, że nie przecina też powierzchni obiektu

Ad. 2. Redukcja liczby przecięć na drodze promienia:

Dąży się do wyeliminowania testów przecięcia z tymi obiektami, w które promień na pewno nie trafi, np.

Metoda struktur hierarchicznych brył otaczających

Obiekty sceny otacza się bryłami tworzącymi strukturę hierarchiczną (w sensie zawierania się w sobie).

Jeżeli promień nie przecina bryły „wyższej” w ustalonej hierarchii, to nie przecina również żadnej z brył „niższych”, lub...

Metoda podziału przestrzeni

Metoda wymaga wstępnego przetworzenia danych o geometrii sceny. Proces ten wymaga też oczywiście pewnego nakładu obliczeniowego, jednak w sumie jest to metoda bardzo wydajna.

Postępuje się następująco:

1. Określa się minimalny prostopadłościan tzw. domenę, w którym zawierają się wszystkie obiekty analizowanej sceny.
2. Dzieli się domenę na małe elementy prostopadłoscienne tzw. woksele (voxel).
3. Dla każdego woksela domeny tworzy się listę obiektów, które całkowicie lub częściowo w nim się znajdują.
4. W procesie śledzenia promienia, identyfikuje się woksele leżące na jego drodze i analizuje jedynie obiekty znajdujące się na właściwych listach.

Ad. 3. Redukcja gęstości próbkowania:

Analiza wszystkich promieni pierwotnych nie zawsze jest konieczna.

Stosowana jest tu np.

Metoda redukcji liczby promieni pierwotnych (przytoczona na
następnym slajdzie)

...

Metoda redukcji liczby promieni pierwotnych

- Promień pierwotny i obliczenia prowadzi się tylko dla niektórych pikseli ekranu np. dla jednego piksela na cztery.
- Dla punktu ekranu, przez które nie prowadzono promieni pierwotnych rezultat wyznacza się, interpolując wyniki obliczeń przeprowadzonych dla sąsiednich punktów.

Sposób taki jest często stosowany przy próbnej generacji obrazów skomplikowanych scen. Jeśli okaże się że wynik spełnia określone wymagania, dokonuje się generacji obrazu inną, dokładniejszą metodą.

Metoda energetyczna - Radiosity

Gorol C., Torrance K., Greeneberg D., Battaile B.,
„Modeling the Interaction of Light Between Diffusion Surfaces,”
SIGGRAPH 1984.

Metoda energetyczna (radiosity method)

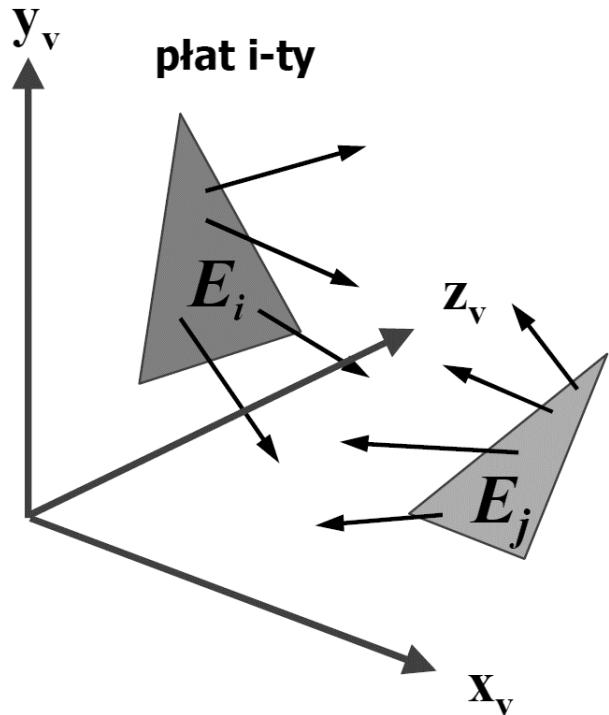
W metodzie śledzenia promieni obliczenia odbywały się przy założeniu ustalonego sposobu rzutowania (ustalonym położeniu obserwatora; położenie obserwatora miało znaczenie dla układu obliczeń).
W metodzie energetycznej nie czyni się takiego założenia.

Założenia:

1. Modelowana scena zbudowana jest ze skończonej liczby płatów powierzchni.
2. Płaty powierzchni emitują, absorbują i odbijają światło.
3. Emisja i odbicie są takie same we wszystkich kierunkach (dyfuzja).
4. Cała energia emitowana lub odbijana przez płaty powierzchni dociera do innych płatów, przez które jest absorbowana lub odbijana (zachowanie energii).

Z techniką **Radiosity** związane są pojęcia:

- Równanie energetyczne płyta powierzchni
- Równanie energetyczne sceny



Równanie energetyczne dla sceny jest to w rzeczywistości układ równań. Poszczególne równania układu są równaniami energetycznymi dla płyt powierzchni, niewiadomymi są natężenia promieniowania płytów i_{tej} i j_{tej} powierzchni (E_i i E_j).

Równań w układzie jest tyle, ile jest płyt powierzchni na scenie. Zapisany w postaci macierzowej układ równań wiąże promienistości dla poszczególnych płyt powierzchni.

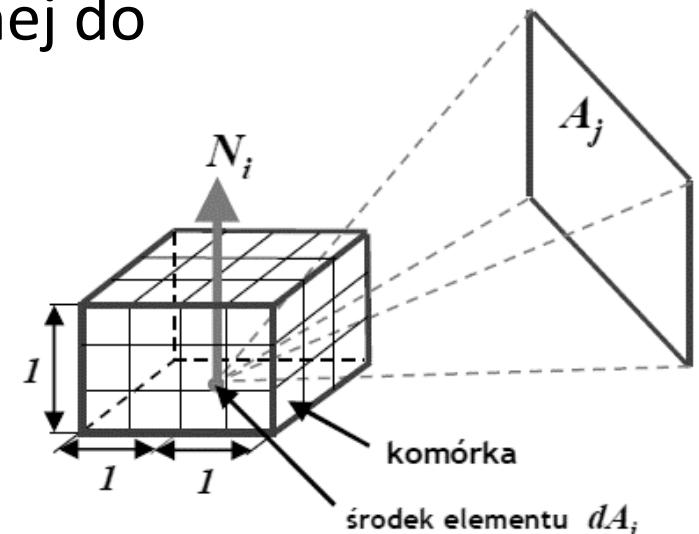
Jest to układ n równań liniowych z n niewiadomymi (n jest zazwyczaj duże).

W pierwszych pracach dotyczących metody energetycznej do rozwiązania układu stosowano specyficzny, iteracyjny algorytm Gaussa –Seidla (metoda analityczna).

Numeryczne wyznaczanie współczynników sprężenia optycznego opisano w artykule:

Cohen M., Greenberg D. „The Hemi-cube; a Radiosity Solution for Complex Environments” Computer Graphics, No 3, 1985.

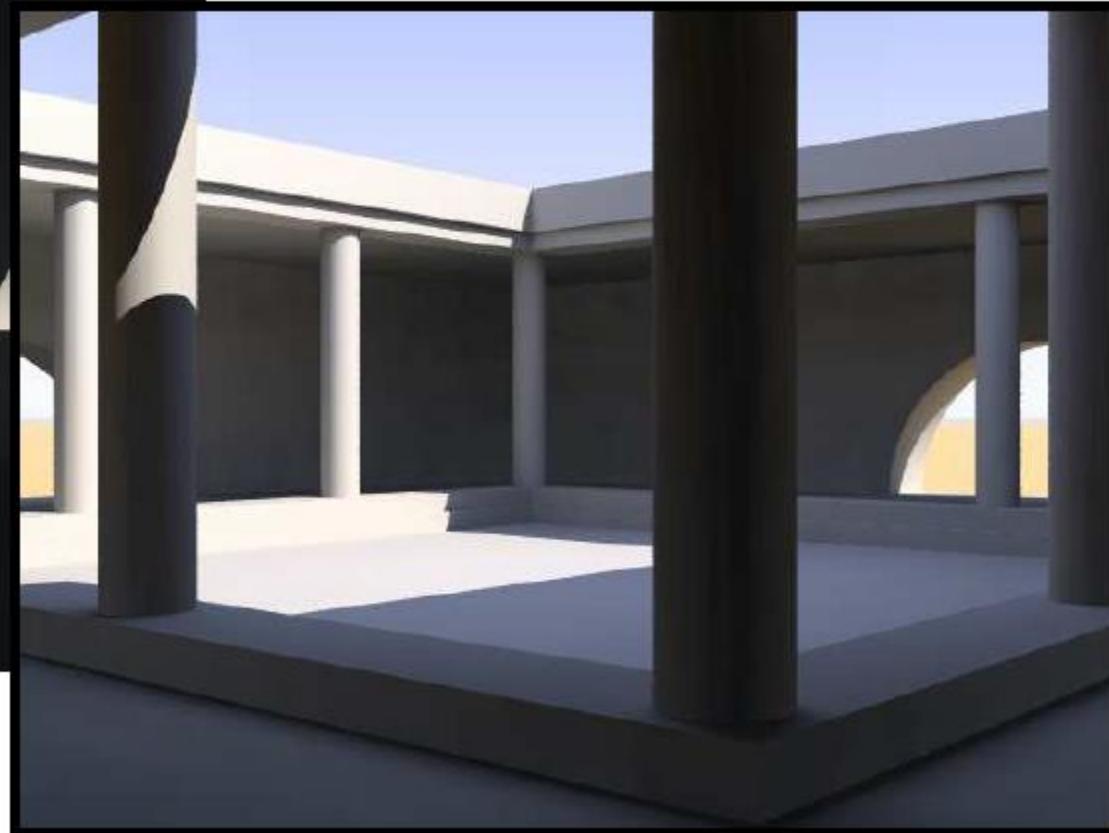
Metoda ta wykorzystuje tzw. hemi-cube (półsześcian).



Porównanie



Ray tracing



Radiosity

Podsumowanie

- Przypomnienie najważniejszych pojęć

Dziękuję za uwagę