

POLITECHNIKA ŚLĄSKA W GLIWICACH
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ

Sprawozdanie z projektu z przedmiotu
Programowanie Aplikacji z Interfejsem Graficznym

Shape from shading

Martyna Kamieniorz Karolina Kowol Paulina Klimanek

Zabrze, 9 czerwca 2020

Spis treści

1. Wstęp teoretyczny	3
2. Cel projektu	4
3. Specyfikacja zewnętrzna	5
4. Specyfikacja wewnętrzna	11
4.1 Przycisk „Wczytaj” zdjęcie	11
4.2 Przycisk „Wykonaj” Shape from Shading	11
4.3 Przycisk „Wyświetl” szczegóły transformacji	12
4.4 Przycisk „Wygeneruj” chmurę punktów	14
4.5 Przycisk „Wczytaj” chmurę punktów	15
4.6 Przycisk „Porównaj” chmury punktów	16
4.7 Przygotowanie chmury punktów w programie Agisoft Metashape Professional	16
5. Podsumowanie i wnioski	19

Spis rysunków

3.1	Wygląd aplikacji po uruchomieniu	5
3.2	Wybór zdjęcia do przetwarzania	6
3.3	Wstępnie przetworzony obraz wyświetlony w aplikacji	6
3.4	Prezentacja zdjęcia po procesie Shape from Shading	7
3.5	Wyświetlone szczegóły transformacji	8
3.6	Chmura punktów powstała na podstawie zdjęcia	9
3.7	Wybór pliku z zapisaną chmurą punktów	9
3.8	Wyświetlenie chmury punktów wygenerowanej w programie <i>Agisoft Metashape</i>	10
3.9	Porównanie dwóch wygenerowanych chmur punktów	10
4.1	Zaimportowana seria zdjęć	16
4.2	Zdjęcie obiektu rzeczywistego	17
4.3	Chmura punktów wygenerowana w programie Agisoft Metashape Professional	17
4.4	Zrekonstruowany model 3D a) 3D model oraz b) ang. <i>tiled model</i>	18

1. Wstęp teoretyczny

Shape From Shading jest procesem obliczania trójwymiarowego kształtu powierzchni z jednego obrazu tej powierzchni. W przeciwieństwie do większości innych problemów związanych z rekonstrukcją trójwymiarową (np. stereofotometrycznych), w problemie Shape From Shading dane są minimalne – rekonstrukcja powstaje na podstawie jednego obrazu. W konsekwencji, problem ten jest z natury trudny [2].

Shape from Shading (SFS) polega na obliczeniu trójwymiarowego kształtu powierzchni z jasności jednego monochromatycznego obrazu tej powierzchni. W latach 70-tych, Horn jako pierwszy sformułował problem Shape From Shading w sposób prosty i rygorystyczny. Polegał na znalezieniu rozwiązania nieliniowego równania różnicowego pierwszego rzędu, zwanego równaniem jasności [4].

Napotymane trudności to często wklęsłe lub wypukłe niejednoznaczności. W rzeczywistości tego rodzaju niedokładności mogą być szeroko uogólnione. Jeżeli kierunek oświetlenia i refleksyjność Lamberta (albedo) powierzchni są nieznane, to ten sam obraz może być uzyskany przez ciągłą rodzinę powierzchni (w zależności liniowej od trzech parametrów). Innymi słowy, cieniowanie obiektu widzianego z jednego punktu widzenia, nie ujawnia jego dokładnej trójwymiarowej struktury [4].

Shape from Shading wprowadzone przez Horna prowadzi do wyprowadzenia równania jasności [4]:

$$I(x_1, x_2) = R(n(x_1, x_2)), \quad (1.1)$$

gdzie x_1, x_2 są współrzędnymi punktu x na obrazie. Równanie jasności łączy mapę refleksyjności (R) z obrazem o jasności (I).

W tym przypadku założenia, że scena jest lambertiańska to mapa refleksyjności jest cosinusem kąta między wektorem światła $L(x)$ a wektorem normalnym $n(x)$ do powierzchni [4]:

$$R = \cos(L, n) = \frac{L}{|L|} \cdot \frac{n}{|n|}, \quad (1.2)$$

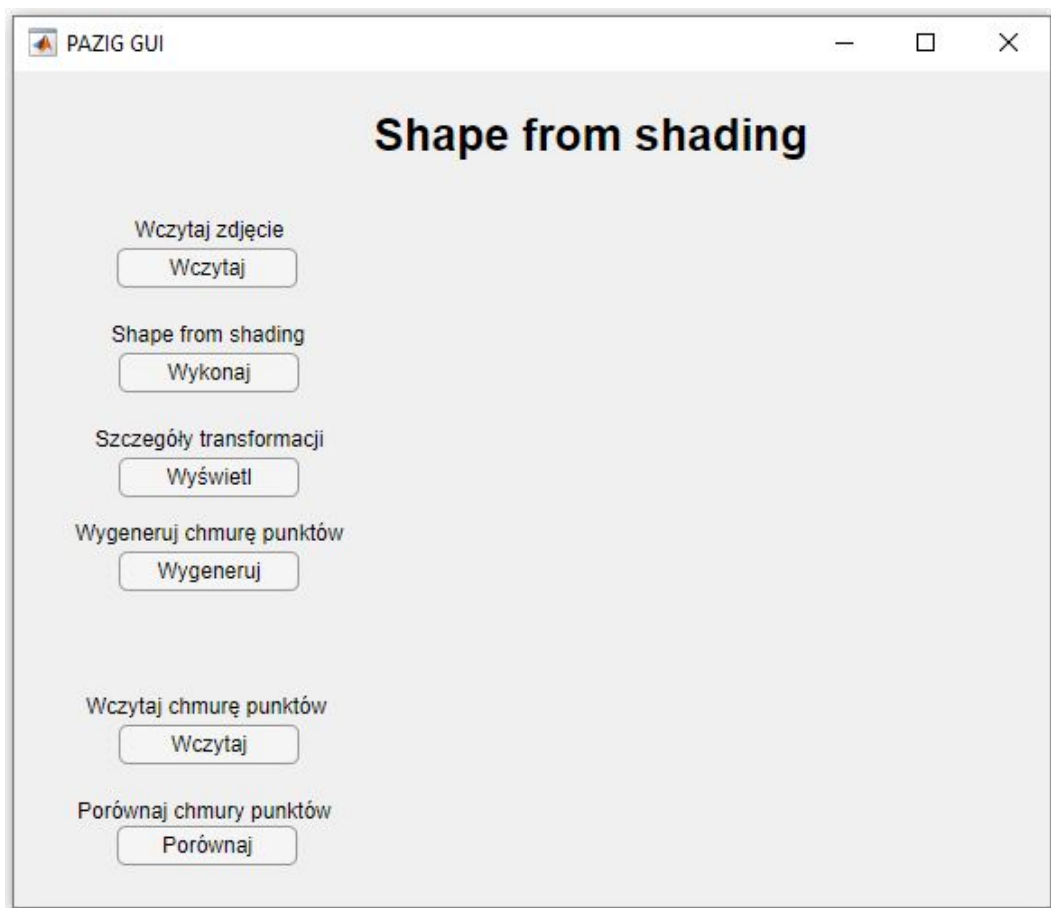
gdzie R, L oraz n zależą od (x_1, x_2) .

2. Cel projektu

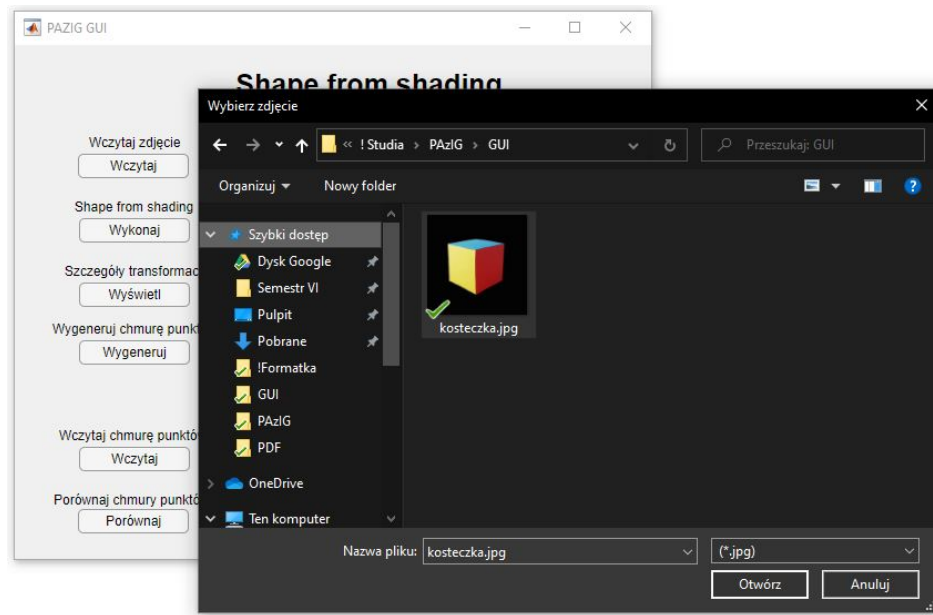
Celem projektu było stworzenie aplikacji z intuicyjnym graficznym interfejsem użytkownika, która pozwoliła na trójwymiarową rekonstrukcję obiektu z wykorzystaniem jego zdjęcia. Aplikacja powstała w programie MatLab 2019 z wykorzystaniem App Designer. W celu weryfikacji zaproponowanego algorytmu Shape from Shading skorzystano z chmury punktów wygenerowanej w programie Agisoft Metashape Professional. Jest to samodzielne oprogramowanie, które wykonuje obróbkę fotogrametryczną obrazów cyfrowych i generuje trójwymiarowe dane przestrzenne do wykorzystania w aplikacjach, dokumentacji dziedzictwa kulturowego i produkcji efektów wizualnych, jak również do pośrednich pomiarów obiektów o różnych skalach [3].

3. Specyfikacja zewnętrzna

Zaprojektowany interfejs graficzny składa się z jednego okna aplikacji, widocznego na rysunku 3.1. w pierwszej kolejności użytkownik powinien wybrać zdjęcie, które zostanie przetworzone w aplikacji – taka funkcja dostępna jest poprzez użycie przycisku „Wczytaj”. Umożliwia on otwarcie eksploratora plików, w którym dostępny jest wybór plików o rozszerzeniu .jpg lub .png (Rys. 3.2).

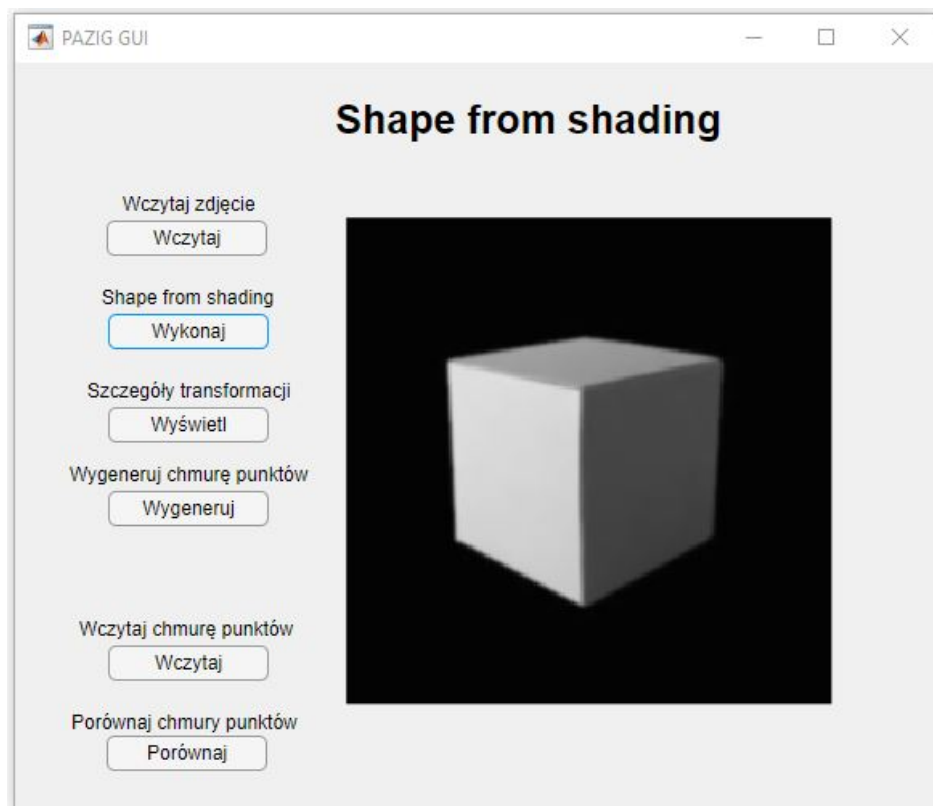


Rys. 3.1: Wygląd aplikacji po uruchomieniu



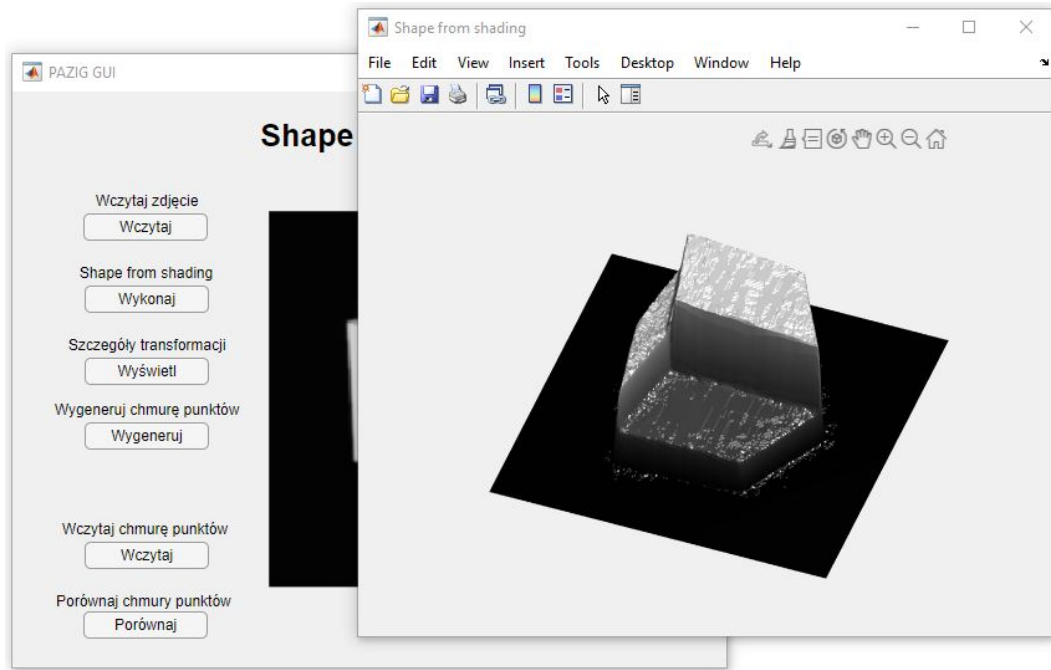
Rys. 3.2: Wybór zdjęcia do przetwarzania

Po wczytaniu, wybrany obraz zostaje wstępnie przetworzony, aby możliwe było wykorzystanie go jako obrazu wejściowego do metody Shape from Shading (Rys. 3.3).



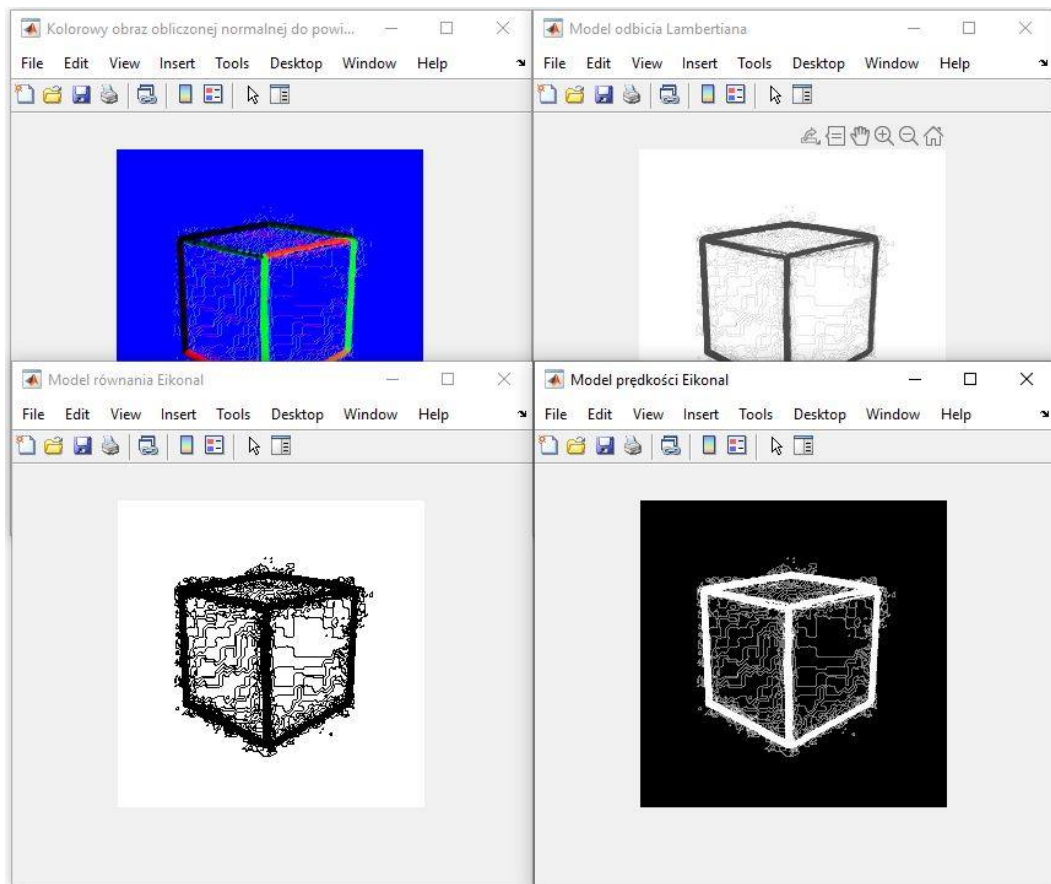
Rys. 3.3: Wstępnie przetworzony obraz wyświetlony w aplikacji

Działanie metody SFS następuje po wciśnięciu przycisku „Wykonaj”. Wskutek jego działania uzyskujemy obraz wynikowy, przedstawiony na rysunku 3.4.



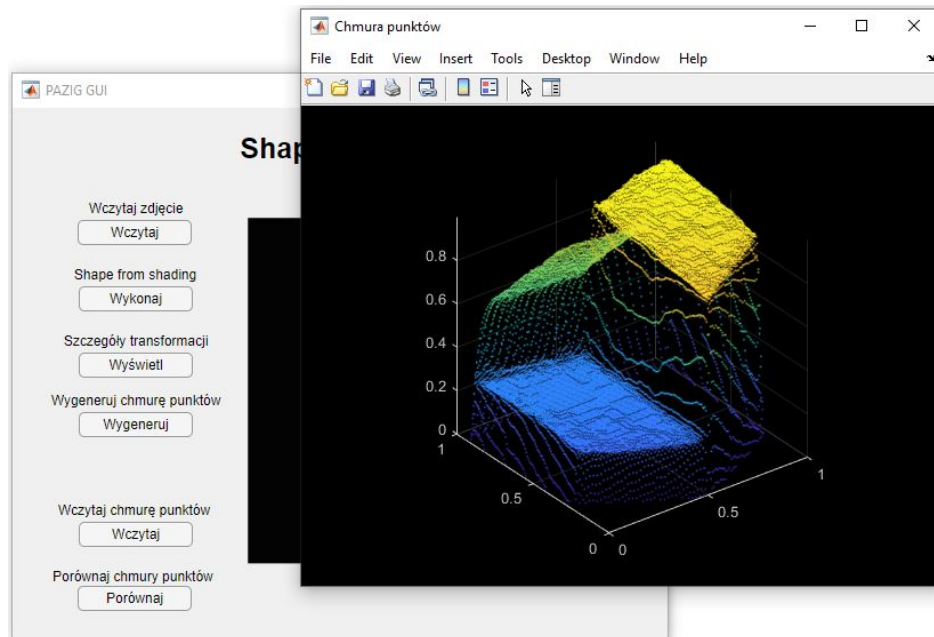
Rys. 3.4: Prezentacja zdjęcia po procesie Shape from Shading

Szczegóły transformacji, a więc obrazy przedstawiające normalną do powierzchni oraz model odbicia Lambertiana (Rys. 3.5), dostępne są po wciśnięciu przycisku „Wyświetl”.



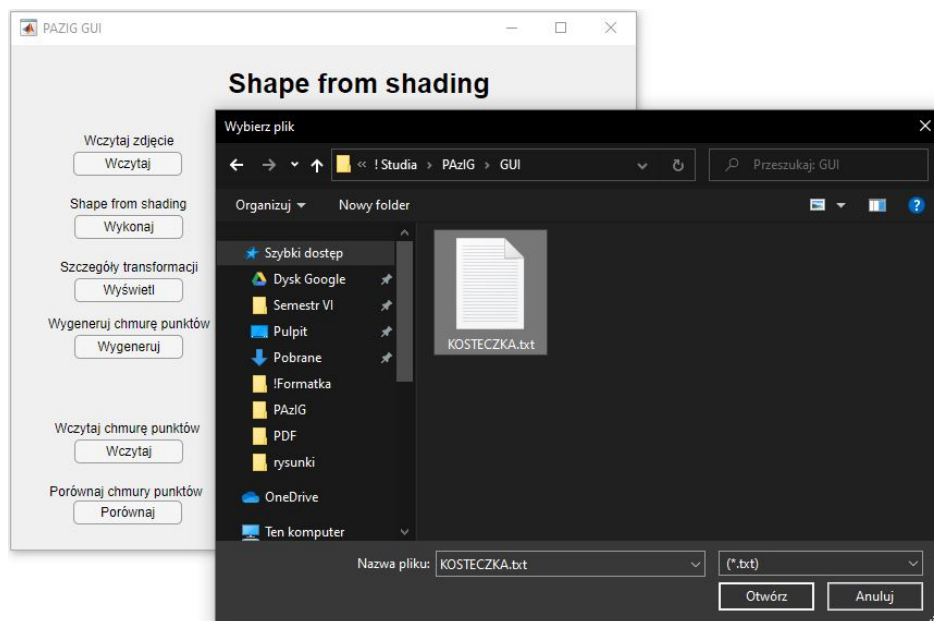
Rys. 3.5: Wyświetlone szczegóły transformacji

Jednym z kluczowych etapów działania programu jest porównanie chmur punktów stworzonych dwiema metodami. W tym celu konieczne jest stworzenie chmury punktów na podstawie figury uzyskanej w Shape from Shading – taką funkcjonalność zapewnia przycisk „Wygeneruj”. Stworzona chmura widoczna jest na rysunku 3.6.

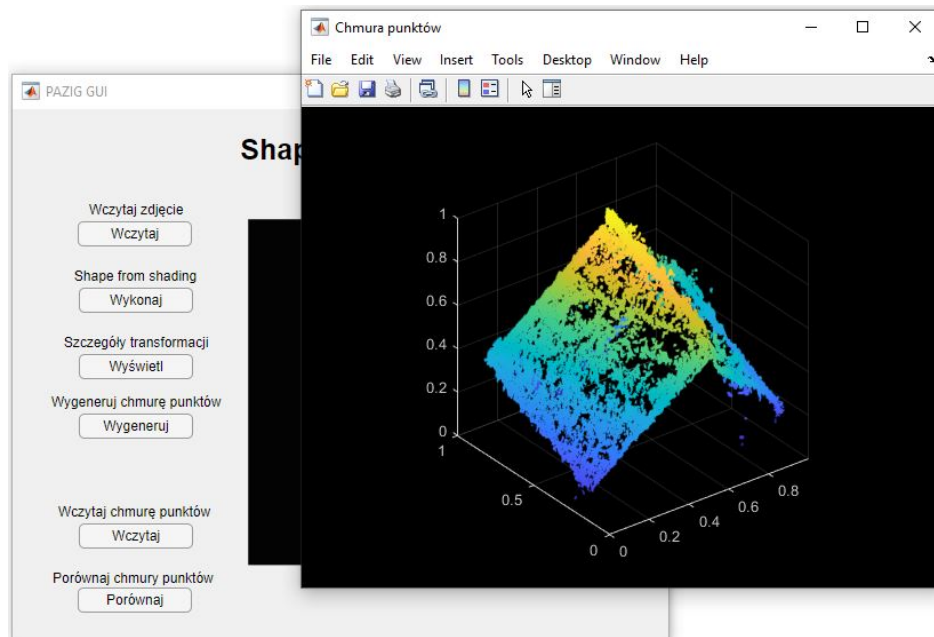


Rys. 3.6: Chmura punktów powstała na podstawie zdjęcia

Do porównania zostanie wykorzystana chmura punktów wygenerowana na podstawie zdjęć kolorowej kostki za pomocą programu Agisoft Metashape Professional, przedstawiona na rysunku 3.8. Punkty zapisane są w pliku o rozszerzeniu .txt, który może zostać wczytany do aplikacji przy użyciu klawisza „Wczytaj”. Otwarty w ten sposób eksplorator plików pozwala na wczytanie plików tekstowych jedynie o tym rozszerzeniu (Rys. 3.7).

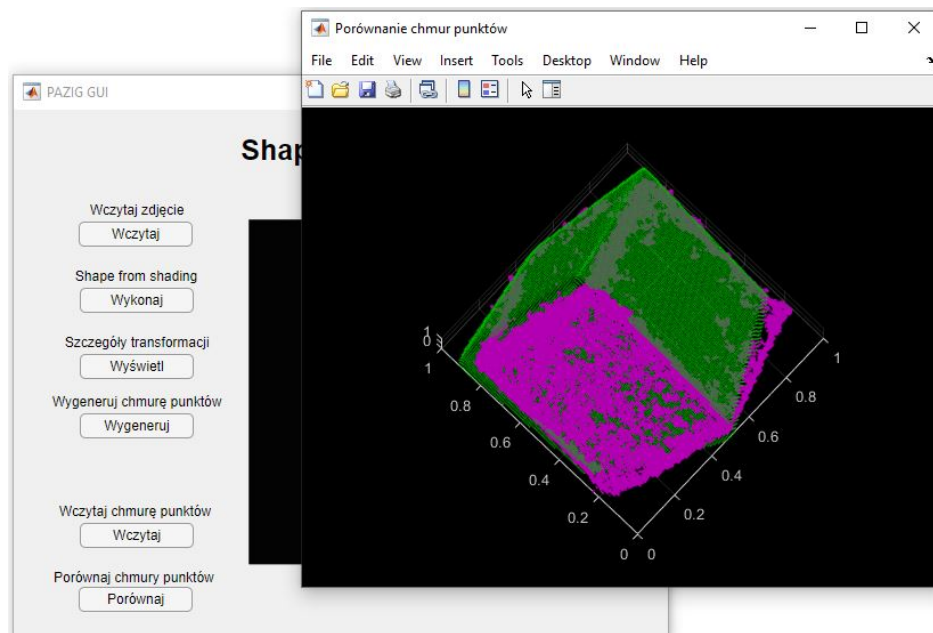


Rys. 3.7: Wybór pliku z zapisaną chmurą punktów



Rys. 3.8: Wyświetlenie chmury punktów wygenerowanej w programie *Agisoft MetaShape*

Dostępny w aplikacji przycisk „*Porównaj*” umożliwia porównanie wyżej wymienionych chmur punktów. Wynikiem działania jest wyświetlenie figury z nałożonymi na siebie chmurami, co wizualizuje dokładność odwzorowania poszczególnych metod, a także różnice w sposobie generowania punktów (Rys. 3.9).



Rys. 3.9: Porównanie dwóch wygenerowanych chmur punktów

4. Specyfikacja wewnętrzna

4.1 Przycisk „Wczytaj” zdjęcie

Po uruchomieniu aplikacji i naciśnięciu pierwszego przycisku, wyświetla się okno do wyboru odpowiedniego zdjęcia do przetworzenia. Funkcją odpowiedzialną za tą czynność jest `uigetfile()`. Następnie do wczytania zdjęcia zastosowano `imread()` oraz przeprowadzono wstępne przetwarzanie polegające na wygładzeniu krawędzi (`imgaussfilt()`) i zamiany obrazu na odcienie szarości (`rgb2gray()`). Wynikiem działania tej funkcji jest wyświetlenie zdjęcia poprzez `imshow()` oraz zapisanie macierzy do zmiennej globalnej.

```
1 function WczytajZdjecieButtonPushed(app, event)
    file = uigetfile({'*.jpg'; '*.png'}, 'Wybierz
        zdjęcie ');
    A = imread(file);
    A = imgaussfilt(A,1);
5    A = rgb2gray(A);
    imshow(A, 'parent', app.UIAxes, ...
        'XData', [1 app.UIAxes.Position(3)], ...
        'YData', [1 app.UIAxes.Position(4)]);
9
    app.ZDJECIE = A;
end
```

4.2 Przycisk „Wykonaj” Shape from Shading

W celu wczytania obrazu zawartość macierzy jest zaimportowana, by w następnej kolejności pobierane były jej wymiary (`size()`) oraz dopasowana szerokość i wysokość (`rescale()`), aby uzyskać obraz kwadratowy. W nowym oknie wyświetlony zostaje trójwymiarowy wykres powierzchni wykorzystując funkcję `surf()` wraz z odpowiednimi funkcjami dostosowującymi wygląd wykresu.

```
1 function WykonajSFSButtonPushed(app, event)
    A = app.ZDJECIE;

    n = size(A,1);
5    h = n*.4;
    A = rescale(A,0,h);
```

```

9         figure('Name','Shape from shading','NumberTitle','
13         off');
        surf(A);
        colormap(gray(256));
        shading interp;
        axis equal;
        view(110,45);
        axis('off');
        camlight;
end

```

4.3 Przycisk „Wyświetl” szczegóły transformacji

Podobnie jak w poprzednio, całość obrazu zostanie zaimportowana oraz odpowiednio dostosowany wymiar. W celu dalszych obliczeń utworzona zostaje macierz wypełniona zerami o wymiarach zgodnych z wymiarami zdjęcia. W celu przeciwdziałania utracie ostatniego wiersza i kolumny macierzy, poprzez funkcję *cat()* dodane zostaną podwojone wartości na końcach macierzy. Do obliczenia pochodnej wykorzystano funkcję *diff()*, którą wykorzystuje się aby uzyskać znormalizowaną macierz. Wyświetlenie figury z obrazem normalnej do powierzchni, możliwy jest poprzez wyliczenie zmiennej *N0* (nieznormalizowane), którą następnie posłuży do obliczenia normalnej *N*. W celu wyświetlenia modelu odbicia Lambertiana zapisane zostały współczynniki oraz wyliczone zmienne na podstawie wartości normalnej. Z zsumowanych współczynników wyznaczona zostaje maksymalna wartość (*max()*), na podstawie której zostaje przetworzony obraz. Uzyskanie modelu równania Eikonal możliwe jest dzięki zapisaniu odpowiednich wartości współczynnikom, które wraz z informacją o wymiarach zdjęcia pozwalają na prawidłowe obliczenie danych. Normalna obrazu zostaje pomnożona przez wartości w odpowiednich punktach macierzy oraz zsumowana. Wyświetlanie obrazu następuje, tylko jeśli odpowiednie piksele posiadające wartość większą niż ustalona tolerancja. Ostatnią wyświetlona figurą jest model prędkości Eikonal, który jest pierwiastkiem z elementów macierzy mapy luminancji (*sqrt()*) oraz wartościami większymi od tolerancji.

```

function WywietlSzczegButtonPushed(app, event)
    A = app.ZDJECIE;
    n = size(A,1);
    n2 = size(A,2);
    h = n*.4;
    A = rescale(A,0,h);
    B = ones(n,n2);
8
    doklejenie1 = cat(2,A, A(:,n2));

```

```

12     doklejenie2 = cat(1,A, A(n,:));
    pochodnax = diff(doklejenie2,1,1);
    pochodnay = diff(doklejenie1,1,2);

16     N0 = cat(3, -pochodnax,-pochodnay, B);
    s = sqrt(sum(N0.^2,3) );
    Q = repmat( s, [1 1 3] );
    N = N0 ./ Q;
    figure( 'Name','Kolorowy obraz obliczonej normalnej
        do powierzchni','NumberTitle','off')
    imshow(N);

20     d = [0 0 1];
    R = reshape(d, [1 1 3]);
    W = repmat(R, [n n2 1]);

24     T = N .* W;
    Y = sum( T,3 );
    L = max(0,Y);

28     vmin = .3;
    P = max(L,vmin);
    figure( 'Name','Model odbicia Lambertiana','
        NumberTitle','off')
    imshow(P);

32     d = [0 0 1];
    M = reshape(d,[1 1 3]);
    V = repmat(M,[n n2 1]);
    Z = N .* V ;
    L = sum( Z,3 );

36     epsilon = 1e-9;
    A2 = L>1-epsilon;
    figure( 'Name','Model równania Eikonal','
        NumberTitle','off');
    imshow(A2,[]);

40     W = sqrt(1./L.^2-1);
    W = max(W,epsilon);
    figure( 'Name','Model prędkości Eikonal','
        NumberTitle','off');
    imshow(min(W,3) );

48 end

```

4.4 Przycisk „Wygeneruj” chmurę punktów

Uzyskanie chmury punktów z jednej fotografii obiektu rzeczywistego możliwe jest dzięki obliczeniu wektorów współrzędnych. Po pobraniu wymiarów obrazu *size()*, pierwszy wektor X zostaje wygenerowany jako wartości od 1 do wartości wymiaru zdjęcia, aby następnie dodać kolejne takie same kolumny aż do uzyskania macierzy kwadratowej z której po złączeniu do jednego długiego wektora otrzymujemy gotową współrzędną. Wymiar w osi Y jest skopiowanym wektorem X, który został posortowany rosnąco. Wektor Z zawiera wartości intensywności każdego z piksela obrazu, który poddany został połączony w jeden ciąg. Aby zapisać wektory w postaci obiektu *pointCloud()* wszystkie liczby należało zmienić w precyzję *single()* oraz znormalizować (*normalize()*) w zakresie [0 1]. Wartości mniejsze lub równe 10 zostały pomnożone. Wyświetlenie obiektu w odpowiedniej figurze możliwe jest poprzez funkcję *pcshow()*. Dodatkowo do dalszych obliczeń zapisano gotową chmurę punktów do zmiennej globalnej.

```

function WygenerujButtonPushed(app, event)
    A = app.ZDJECIE;

3
    A = imrotate(A,180);
    n = size(A,1);
    n2 = size(A,2);
7
    z = n*n2;

    wektX = (1:1:n)';

11
    w = wektX;
    for i = 1:n-1
        wektX = [wektX w];
    end

15
    wektX = wektX(:);
    wektY = sort(wektX);

19
    transposeA = A';
    wektZ = transposeA(:);

    xyzPoints = [wektX wektY wektZ];
23
    xyzPoints = single(xyzPoints);

    for i = 1:z
        if(wektZ(i)<=10)
27
            xyzPoints(i,:) = 100;
        end
    end
end

```

```
31         xyzPoints = normalize(xyzPoints, 'range');
        ptCloud_obj = pointCloud(xyzPoints);
        app.OBJ_PC = ptCloud_obj;
        figure('Name', 'Chmura punktów', 'NumberTitle', 'off'
35         )
        pcshow(ptCloud_obj)
        axis tight
end
```

4.5 Przycisk „Wczytaj” chmurę punktów

Aby porównać różne sposoby uzyskania punktów w przestrzeni z zdjęcia zastosowany został program Agisoft Metashape, z którego wyeksportowane zostały punkty do pliku tekstowego (.txt). Plik ten możliwy jest do wczytania dzięki funkcji *dlmread()*, z której poszczególne kolumny zapisane zostały do odpowiednich wektorów współrzędnych. Podobnie jak poprzednio niezbędne było znormalizowanie oraz zmiana dokładności. Po zapisaniu do obiektu (*pointCloud()*) wyświetlona została chmura punktów (*pcshow()*).

```
function WczytajPCButtonPushed(app, event)
    file = uigetfile({'*.txt'}, 'Wybierz plik');
3    PC = dlmread(file);

    wektX = PC(:,1);
    wektY = PC(:,2);
7    wektZ = PC(:,3);

    xyzPoints = [wektX wektY wektZ];
    xyzPoints = single(xyzPoints);
11    xyzPoints = normalize(xyzPoints, 'range');

    ptCloud_agi = pointCloud(xyzPoints);
    app.AGI_PC = ptCloud_agi;
15

    figure('Name', 'Chmura punktów', 'NumberTitle', 'off'
        )
    pcshow(ptCloud_agi)
end
```


4.6 Przycisk „Porównaj” chmury punktów

Ostatnim elementem programu jest możliwość wyświetlenia na jednej osi współrzędnych obydwu chmur punktów. Po wczytaniu zmiennych globalnych wyświetlone zostają dzięki funkcji *pcshowpair()*.

```

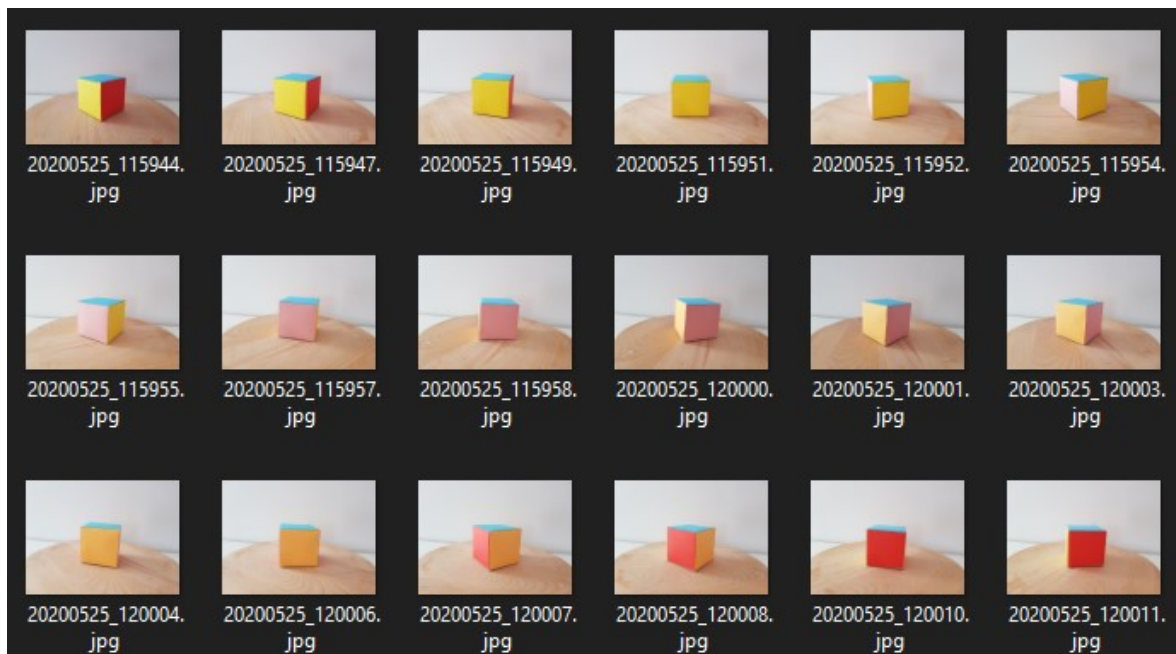
function PorwnajButtonPushed(app, event)
    ptCloud_agi = app.AGI_PC;
    ptCloud_obj = app.OBJ_PC;

    figure('Name','Porównanie chmur punktów','
        NumberTitle','off')
    pshowpair(ptCloud_agi,ptCloud_obj)
end

```

4.7 Przygotowanie chmury punktów w programie Agisoft Metashape Professional

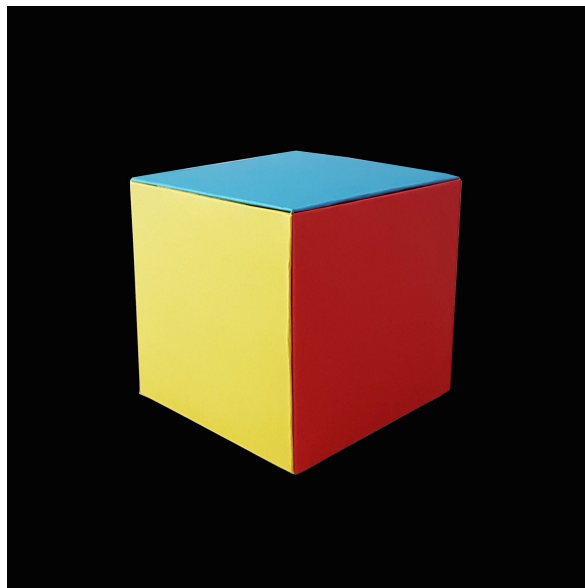
W celu wygenerowania chmury punktów konieczne było zaimportowanie serii zdjęć obiektu (Rys. 4.1), (tj. kolorowej kostki, przedstawionej na rysunku 4.2), które zostały wykonane pod różnymi kątami. Program wyrównał (ang. *align*) wprowadzone zdjęcia, aby lepiej odwzorować położenie punktów względem kamery.



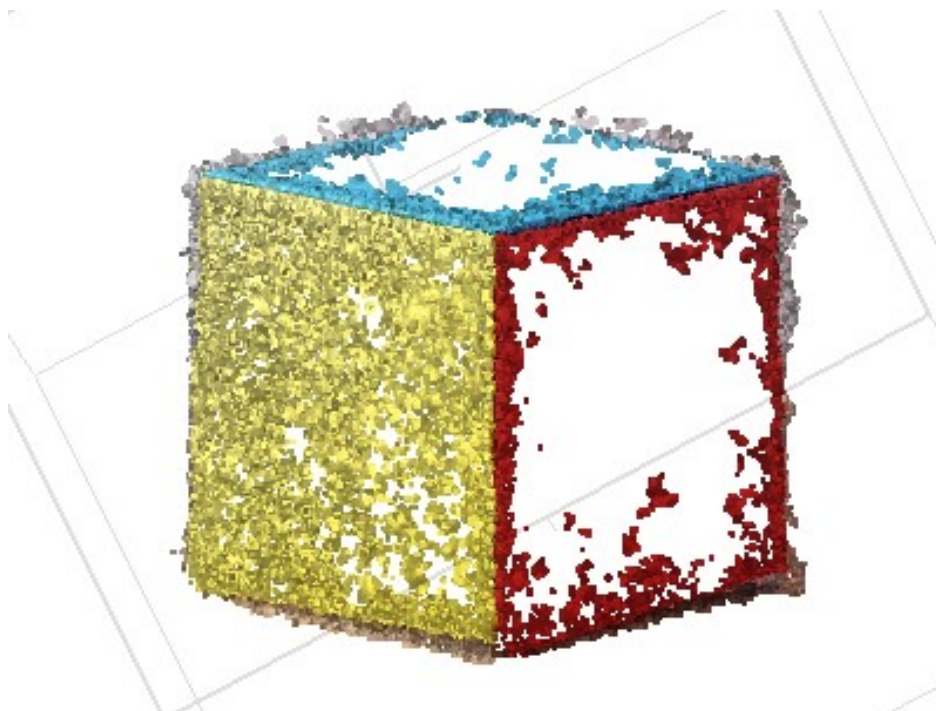
Rys. 4.1: Zaimportowana seria zdjęć

Na podstawie ułożonych zdjęć zostały wygenerowane kolejno gęsta chmura punktów

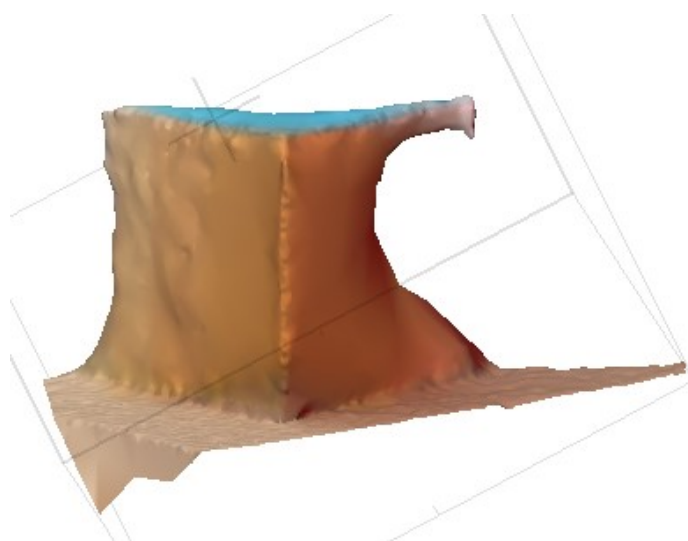
(ang. *dense cloud* – Rys. 4.3) oraz siatka (ang. *mesh* – Rys. 4.4a), przypominająca trójwymiarowy model rekonstruowanego obiektu.



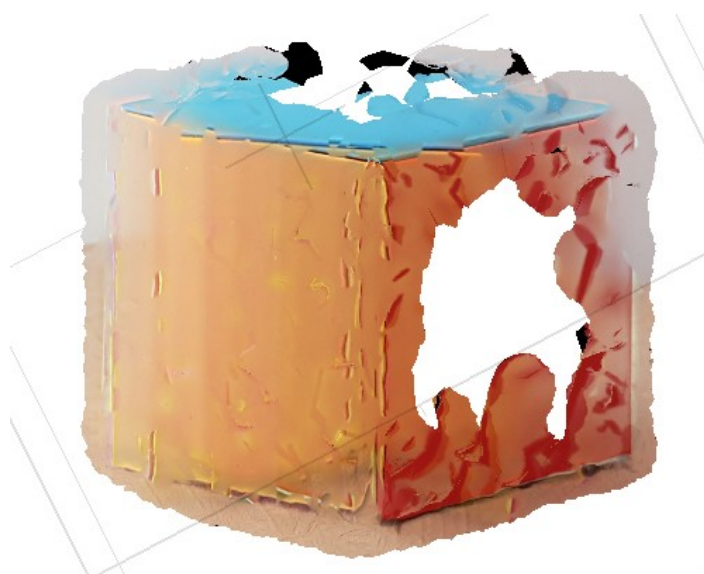
Rys. 4.2: Zdjęcie obiektu rzeczywistego



Rys. 4.3: Chmura punktów wygenerowana w programie Agisoft Metashape Professional



(a)



(b)

Rys. 4.4: Zrekonstruowany model 3D a) 3D model oraz b) ang. *tiled model*

5. Podsumowanie i wnioski

Stworzona aplikacja na podstawie wczytanego zdjęcia pozwala na jego trójwymiarową rekonstrukcję, poprzez wygenerowanie chmury punktów. Otrzymany na podstawie jednego obrazu model tylko częściowo oddaje kształt rzeczywistego obiektu. Chmura punktów otrzymana za pomocą programu Agisoft Metashape Professional, która powstała na podstawie serii zdjęć obiektu, wykonanych pod różnym kątem, posłużyła w procesie porównania z punktami uzyskanymi dzięki metodzie SFS.

Rysunek 3.9 przedstawia wynik porównania. Można więc wywnioskować, że uzyskane chmury punktów nie są wystarczająco dokładnym odwzorowaniem obiektu rzeczywistego. Niestety różnice widoczne w położeniu punktów wprowadzają liczne zniekształcenia, przez co model obiektu nie jest dokładny.

Metoda Shape from Shading jest metodą o niskiej dokładności, ponieważ wykorzystuje ona jedno zdjęcie w celu uzyskania trójwymiarowej rekonstrukcji, dlatego możliwe byłoby ulepszenie działania algorytmu o użycie kilku zdjęć jednocześnie. Dzięki temu możliwe będzie stworzenie modelu o lepszym odwzorowaniu.

Problemem jest również sama scena, na której powstaje obraz obiektu w związku z małym budżetem realizowanego projektu stanowisko, jak i sama kamera do akwizycji zdjęć nie pozwalają na rezultaty wysokiej jakości. Zarówno tło, jak i oświetlenie mają duży wpływ na wynik rekonstrukcji – w przypadku metody Shape from Shading, a także tej powstałej w programie Agisoft.

Zaprojektowana aplikacja daje najlepsze wyniki, gdy zdjęcie wejściowe jest przygotowane w odpowiedni sposób – ma rozmiar około 250×250 pikseli oraz przetwarzany obiekt jest wycięty, a jego tło jest zastąpione czarną powierzchnią. Ważnym aspektem jest również oświetlenie, które najlepiej, aby padało na centralny punkt fotografowanego obiektu.

Bibliografia

- [1] Shape from Shading – data dostępu 04.06.2020 http://www.numerical-tours.com/matlab/graphics_7_shape_shading/
- [2] Bernd Jähne „*Digital Image Processing*” Springer Science & Business Media, 2005, ISBN 3540240357, 9783540240358
- [3] Agisoft Metashape Professional – <https://www.agisoft.com/>
- [4] Emmanuel Prados, Olivier Faugeras „*Shape from Shading* ” – data dostępu 04.06.2020 <http://perception.inrialpes.fr/Publications/2006/PF06a/chapter-prados-faugeras.pdf>