

Instytut Podstawowych Problemów Techniki
Polska Akademia Nauk

**Generowanie modeli widokowych
wielościanów monottonnych
do identyfikacji wizualnej**

mgr Andrzej Salamończyk

Rozprawa doktorska

Promotor: dr hab. n. t. Wojciech S. Mokrzycki

Warszawa 2011

Streszczenie

Przedmiotem rozprawy są metody generowania reprezentacji widokowej wielościanów monotonicznych z przeznaczeniem do systemów rozpoznawania wizualnego. Reprezentacje tego typu znajdują zastosowanie między innymi w zadaniach identyfikacji np. jako element kanału wizualnego w automatyce lub robotyce w zadaniach typu: na początku jest dany pewien zbiór obiektów, które następnie należy rozpoznać w otoczeniu robota.

Celem rozprawy jest opracowanie metod i algorytmów stosownych do uzyskania kompletnej reprezentacji widokowej obiektu przy pomocy koncepcji sfery widokowej z rzutowaniem perspektywicznym, porównanie i wybór najlepszego algorytmu do tego zadania oraz przedstawienie sposobu pasowania modeli widokowych obiektów z wyglądami obiektów pozyskanymi z otoczenia za pomocą map głębi. Wykazano też tezy rozprawy: że jest możliwe wytworzenie $3W$ przybliżonej reprezentacji widokowej wielościanu monotonicznego z jego reprezentacji powłokowej B_{rep} za pomocą algorytmu o złożoności $O(n^2)$ oraz że minimalna liczba pasowań danych i modelu wielościanu monotonicznego w oparciu o dane z mapy głębi i $3W$ reprezentację widokową ma złożoność $O(n^2)$.

Nowym elementem, które wnosi niniejsza praca jest przedstawienie oraz omówienie metod generowania reprezentacji widokowej dla szerszej klasy wielościanów niż wielościany wypukłe (dla których istniały już odpowiednie sposoby otrzymywania widoków). W szczególności są to spiralne metody rozmieszczenia punktów widokowych (nad ścianą obiektu lub na całej sferze widokowej). W pracy udowodniono kompletność tak generowanej reprezentacji widokowej.

Rozprawa składa się z pięciu rozdziałów. W pierwszym z nich przedstawione są motywacje oraz sformułowane cele i tezy rozprawy. W rozdziale drugim dokonywana jest analiza literatury na temat reprezentacji obiektów $3W$ oraz ocena ich przydatności w systemach rozpoznawania. Rozdział trzeci (główny rozdział rozprawy) dotyczy nowych metod generowania reprezentacji widokowej. W tym rozdziale udowodniono pierwszą tezę rozprawy, omówiono implementację algorytmów oraz oszacowano ich złożoność obliczeniową. W czwartym rozdziale udowodniono drugą tezę rozprawy oraz przedstawiono przykład metody z wykorzystaniem reprezentacji widokowej do pasowania z danymi z map zasięgu. W ostatniej części omówiono uzyskane wyniki oraz przedstawiono perspektywę dalszych prac w tej dziedzinie.

Abstract

This dissertation investigates methods of generating view representations of monotone polyhedra for the purpose of visual recognition systems. Representations of this type are used in many areas including identification tasks e.g. as an element of visual channel in automation and robotics for tasks such as: at the beginning a set of objects is given, which then must be identified by a robot. 3D view representation of objects can be constructed offline and saved in a database.

The goal of this dissertation is to develop appropriate methods and algorithms in order to obtain a complete view representation of the object using the idea of the viewing sphere with perspective projection (compare and select the best algorithm for this task) and to show how viewing models can be matched with objects extracted from range images. It was proved that it is possible to obtain the approximate view representation of a monotone polyhedron using the algorithm with the complexity of $O(n^2)$ and that the minimum number of matching view representation of monotone polyhedron and data from range images is $O(n^2)$.

The major contribution of this work is presentation and discussion of methods of generating a view representation for a wider class of polyhedra than the convex ones (there exist methods for convex case). In particular, these are the spiral methods of viewpoints location (over the polyhedron face or on the whole view sphere). It was proved that views from these points yield the complete view representation (with given accuracy).

The dissertation consists of five chapters. In the first one, motivations and goals are formulated. The bibliography on 3D representations of objects and their usefulness in object recognition is reviewed in the second part. In the third main part, new spiral methods are presented and their implementations are discussed. Some idea of how to match view representation with the range images is presented in the fourth part. In the last chapter, the results are discussed and new research directions in this area are suggested.

Chciałbym podziękować:

*Panu Profesorowi Wojciechowi Mokrzyckiemu,
za opiekę naukową oraz cenne wskazówki i sugestie.*

*Żonie i Córkom,
za cierpliwość i wyrozumiałość.*

*Rodzicom,
za ciągłe wsparcie.*

Spis treści

1. Wprowadzenie	1
1.1. Przedmiot rozważań	1
1.2. Podstawowe pojęcia i definicje	4
1.3. Cel, zakres i tezy rozprawy	7
2. Przegląd literatury	9
2.1. Reprezentacje obiektów do identyfikacji wizualnej - wymagania	9
2.2. Reprezentacje obiektów 3W skupione na obiekcie	10
2.2.1. Reprezentacje punktowe	10
2.2.2. Reprezentacje objętościowe	11
2.2.3. Reprezentacje powierzchniowe	12
2.2.4. Deskrytory kształtu	13
2.3. Reprezentacje skupione na widoku	18
2.3.1. Grafy aspektowe	18
2.3.2. Reprezentacja widokowa ze sfery widokowej	23
2.4. Metody generowania reprezentacji widokowej	23
2.4.1. Iteracyjne metody generowania widoków	24
2.4.2. Nieiteracyjne metody generowania widoków	28
2.4.3. Bezpośrednie generowanie widoków	35
3. Przybliżone metody generowania reprezentacji widokowej	45
3.1. Skanowanie obszaru widoczności ściany torem spiralnym	46
3.2. Metody z wykorzystaniem wielościanu foremego	48
3.2.1. Metoda oparta na podziale dwudziestościanu foremego	48
3.2.2. Metoda korzystająca z wypukłości obszaru jednowidokowego	49
3.3. Rozmieszczenie punktów na spirali z określoną rozdzielczością	50
3.3.1. Opis metody	50
3.3.2. Analiza kompletności generowanej reprezentacji	52
3.4. Implementacja algorytmu	55
3.4.1. Przygotowanie wielościanu i generowanie zbioru punktów widokowych	55
3.4.2. Wyznaczanie widoku z punktu widokowego	55
3.4.3. Otrzymane rezultaty	57
3.5. Ocena metody	58
3.5.1. Złożoność obliczeniowa metody	58
3.5.2. Wady i zalety proponowanego rozwiązania	59

4. Przykład zastosowania modeli widokowych	61
4.1. Przykład SIW z wykorzystaniem reprezentacji widokowej	61
4.1.1. Składowe reprezentacji widokowej	61
4.1.2. Opis metody	62
4.1.3. Ocena metody	64
4.2. Metody pasowania obiektów 3W	65
4.2.1. Metryki używane w pasowaniu	65
4.2.2. Wybrane metody pasowania geometrycznego	66
5. Podsumowanie	69
Spis rysunków	79
Wykaz ważniejszych oznaczeń	81
Wykaz ważniejszych skrótów	83
Dodatki	87
A. Algorytmy	87
B. Wyniki	89
Wielościan o 8 ścianach (46 widoków)	90
Wielościan monotony o 10 ścianach (66 widoków)	91
Graniastosłup testowy o 8 ścianach (56 widoków)	92

1. Wprowadzenie

1.1. Przedmiot rozważań

Rozprawa dotyczy metod reprezentowania obiektów wielościennych i generowania ich modeli widokowych z przeznaczeniem do systemów rozpoznawania wizualnego. Zostały w niej przedstawione modele aspekt-grafowe dla pełnej przestrzeni trójwymiarowej (3W) punktów widokowych, modele aspekt-grafowe dla sfery jako zbioru punktów widokowych: z rzutowaniem równoległym oraz środkowym, modele generowane z map głębi, modele histogramowe oraz modele widokowe, generowane z użyciem różnych reprezentacji: wektorowej V_{rep} , naturalnej (ostroslupowej) N_{rep} oraz pełnej naturalnej Np_{rep} (wektorowo-ostroslupowej), wywodzących się z reprezentacji powłokowej B_{rep} . Szczególną uwagę skupiono na modelach widokowych, w zakresie których opisano nowe metody i algorytmy generowania widoków oraz przeprowadzono dogłębną analizę uzyskanych rozwiązań (rozdziały: trzeci i czwarty).

Główne zastosowanie omawianych modeli to **Systemy Identyfikacji Wizualnej (SIW)**. Działanie systemu SIW może polegać na: analizowaniu 3W sceny (wycinka 3W przestrzeni), wyodrębnianiu jej elementów (pozyskaniu wyglądnów obiektów), porównywaniu ich z przygotowanymi wcześniej widokami modeli i decydowaniu, czym są widziane przez system elementy sceny. Zatem, (każdy) system identyfikacyjny wymaga najpierw zapisania w jego pamięci reprezentacji obiektów, które będą rozpoznawane. Akt rozpoznawania następuje później. Tak więc, po zakończeniu pierwszego etapu istnieje w systemie baza danych o obiektach w postaci modeli widokowych, które system będzie mógł rozpoznać.

System SIW działa na ogół dwuetapowo:

1. **W pierwszym etapie** w pamięci systemu są **zapisywane reprezentacje obiektów**, które później system będzie w stanie rozpoznawać.
2. **W drugim etapie** zachodzi **właściwe rozpoznawanie** obiektów na 3W scenie.

Wyróżnia się dwa podstawowe typy baz danych modeli, różniące się rodzajem gromadzonej informacji i trwałością:

1. **baza „ucząca się”**, jak wyglądają poszczególne obiekty, przez „obejrzenie” ich i „zapamiętanie” ich wyglądu; jest to baza o charakterze tymczasowym, w której tworzenie modelu przebiega równoległe z fazą rozpoznawania i polega na poznaniu wyglądu określonego obiektu przez „obejrzenie go” z różnych stron i zapamiętanie. Po wykonaniu zadania rozpoznawania wyglądnów obiektów sceny nie są już więcej

wykorzystywane. Do ich tworzenia potrzebny jest podsystem wizualny systemu rozpoznającego, ruchomy względem rozpoznawanej bryły. Konieczne jest też fizyczne występowanie obiektu. System wykonuje kolejne ujęcia (tego) obiektu, przemieszczając się tak, by wykonać ujęcie z każdej strony. Uzyskane obrazy są analizowane, by wydobyć z nich elementy charakterystyczne obiektu (np. ściany, krawędzie, wierzchołki) i z nich jest odtwarzana 3W postać obiektu. Takie rozpoznawanie może być mniej lub bardziej dokładne i zależy od zastosowania systemu. Użycie bazy tego typu wymaga czasowego wykorzystania pewnej ilości pamięci na informacje o obiektach, które w konkretnym zadaniu rozpoznawania mogą się pojawić.

2. **baza o charakterze trwałym.** Jest tworzona automatycznie na bazie formalnego opisu określonych obiektów. Informacje o obiektach są zapisywane w sposób trwały w odpowiednich strukturach, które mogą być wykorzystywane wielokrotnie i do różnych celów. W fazie rozpoznawania wygląd rzeczywistego obiektu jest analizowany, poszczególne jego elementy wyodrębniane i jest wypełniona odpowiednia struktura opisująca wygląd - analogiczna do opisującej model. Następnym krokiem jest porównanie struktury wyglądu obiektu rzeczywistego i struktur widoków z bazy. Taką bazę można modyfikować zależnie od bieżącego zastosowania, np. powiększając reprezentowany w niej zbiór obiektów. Można ją też stworzyć niezależnie od systemu rozpoznającego, a także niezależnie od obecności obiektów (o ile znane są ich opisy formalne, np. B_{rep}).

Negatywny wynik poszukiwania pasującego modelu oznaczać będzie, że model obiektu nie istnieje w bazie danych systemu. Natomiast brak jednoznacznego wskazania na konkretny model świadczyć będzie o tym, że w bazie istnieje kilka modeli (czyli, że w bazie istnieje kilka obiektów), wyglądających podobnie do obiektu rozpoznawanego, przy pewnym ustawieniu go względem punktu widokowego. Wówczas konieczne jest dostarczenie systemowi rozpoznającemu dodatkowych informacji o obiekcie, np. jego nowych ujęć.

Rozpoznawanie to problem pasowania między wyglądem sceny i opisem modelu obiektu i jest klasycznym problemem sztucznej inteligencji. Opracowano wiele metod jego rozwiązania, jednak każda z nich jest kosztowna obliczeniowo. Przegląd zagadnień reprezentacji obiektów 3W do celów identyfikacji wizualnej umieszczono w rozdziale drugim, zaś zagadnienia pasowania modeli i widoków obiektu omówiono i zanalizowano w rozdziale czwartym.

W pracy **przyjęto następujący układ:**

W rozdziale 1 „Wprowadzenie” omówiono główne zagadnienia związane z tworzeniem odpowiednich reprezentacji użytecznych do rozpoznania obiektów oraz kolejne etapy w procesie identyfikacji wizualnej obiektów na bazie reprezentacji widokowej. Następnie przedstawiono koncepcję sfery widokowej z rzutowaniem perspektywicznym oraz

wprowadzono potrzebne pojęcia i definicje. Przy ich pomocy sformułowano założenia pracy oraz zadanie badawcze.

W rozdziale 2 „Przegląd literatury” przeprowadzono analizę źródeł literaturowych opisujących reprezentacje obiektów oraz ich ocenę w kontekście sformułowanego zadania badawczego. Również, omówiono widoki generowane na sferze widokowej z rzutowaniem środkowym, a następnie algorytmy oparte na koncepcji sfery widokowej z rzutowaniem perspektywicznym, które były bazą dla zaproponowanych (nowych) algorytmów tworzenia modeli widokowych.

W rozdziale 3 (głównym rozdziale rozprawy) „Przybliżone metody generowania reprezentacji widokowej” są prezentowane nowe sposoby generowania reprezentacji widokowych dla tytułowej klasy brył, ich omówienie oraz porównanie ze znanymi algorytmami w tej dziedzinie. W szczególności są to algorytmy z omiotem stożkiem widokowym:

- spiralnie nad ścianą widokową,
- spiralnie nad sferą widokową z zadaniem równomiernym spacjiowaniem między punktami widokowymi.

Rozdział ten zawiera omówienie implementacji algorytmów oraz oszacowanie ich złożoności obliczeniowej. **Tu też została wykazana prawdziwość pierwszej tezy rozprawy.**

W rozdziale 4 „Przykład zastosowania modeli widokowych” są omówione rozwiązania do pozyskiwania i przetwarzania danych z otoczenia robota oraz sposoby ich pasowania z $3W$ reprezentacją widokową, a także przedstawiono pewne nowe sugestie w tej tematyce. Jest też zaprezentowana koncepcja identyfikacji wizualnej pewnej klasy $3W$ obiektów (wielościannów monottonnych - szerszej klasy od wielościannów wypukłych). **W tym rozdziale została też wykazana druga teza rozprawy.**

Jako sposób reprezentowania obiektów w systemie SIW proponowane są widoki, a mapy zasięgu są głównym sposobem akwizycji danych z otoczenia. Taki system może znaleźć szerokie zastosowanie na przykład w automatyce lub robotyce w zadaniach typu: na początku jest dany pewien zbiór obiektów, które następnie należy rozpoznać w otoczeniu robota. Ponieważ ten zbiór jest wcześniej znany, jest dostępna pełna informacja o szukanych obiektach (w szczególności o cechach geometrycznych obiektu). Można określić, jakie cechy wyróżnione tego obiektu (np. ściany, krawędzie) mogą być jednocześnie widoczne, a tym samym skonstruować model widokowy obiektu. Następnie taką reprezentację umieszcza się w bazie danych i robot pozyskując informacje (np. za pomocą wspomnianych już map zasięgu) po odpowiednim przetworzeniu danych (segmentacji), może sprawdzić, czy taki obiekt ma w swojej bazie danych. Używając reprezentacji widokowej obiektu można szybciej dokonać pasowania, co ma duże znaczenie, jeśli chce się, by rozpoznawanie odbywało się w czasie rzeczywistym. W tym rozdziale jest też przedstawione pewne podejście do rozwiązania problemów typu: obiekty

są rozpoznawane niezależnie od wyboru punktu widokowego oraz jest uwzględniany problem częściowych zasłonięć (samoprzesłaniania), które mogą się pojawić dla obiektów niewypukłych.

W ostatnim rozdziale „Podsumowanie” są omówione uzyskane wyniki oraz jest zakreślona perspektywa dalszych prac w tej dziedzinie.

1.2. Podstawowe pojęcia i definicje

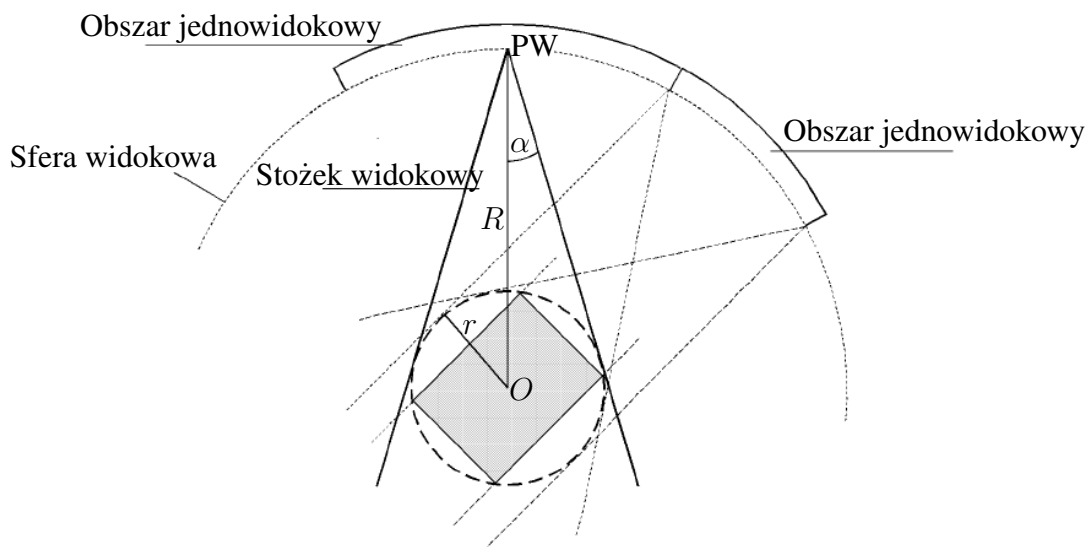
Idea generowania widoków korzystając ze sfery widokowej (z rzutowaniem perspektywicznym) została przedstawiona w pracach [23, 24].

Wielościan, którego widoki mają być wygenerowane, umieszczamy w lokalnym kartezjańskim układzie współrzędnych tak, aby początek tego układu $O(0, 0, 0)$ pokrywał się ze środkiem geometrycznym bryły. Możemy na tym wielościanie opisać sferę (zwaną „małą sferą”) o promieniu

$$r = \max_{i=1..n} \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad (1.1)$$

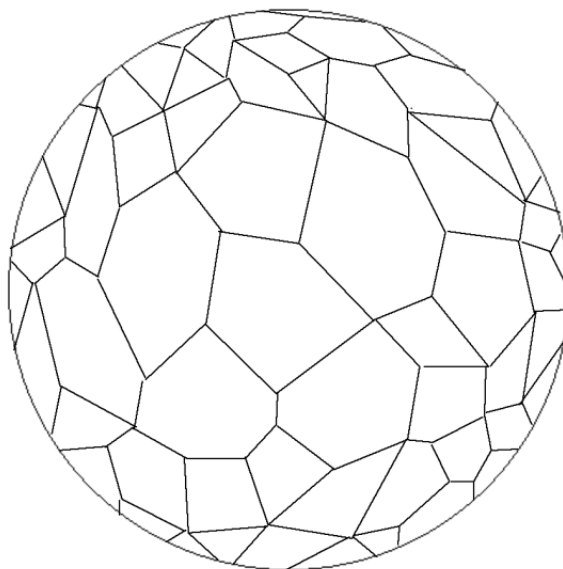
której środek pokrywa się ze środkiem geometrycznym bryły. Na tej sferze umieszczamy **stożek widokowy (SW)** (odzwierciedlający pole widzenia urządzenia pozyskującego dane) o kącie rozwarcia 2α , tak aby ten stożek był styczny do sfery. Kąt rozwarcia SW jest parametrem technicznym urządzenia (lasera, kamery, ...). Wierzchołek tego stożka to **punkt widokowy (PW)**, a swobodny ruch tego punktu (ale taki, że stożek jest styczny do małej sfery, wyznacza **sferę widokową z perspektywą (SWP)** (rys. 1.1). Kierunkiem rzutowania (osią widokową) jest prosta przechodząca przez środek sfery SWP oraz wierzchołek (środek) rzutowania PW. Promień sfery widokowej (oznaczany dalej przez R) wynosi

$$R = \frac{r}{\sin(\alpha)}. \quad (1.2)$$



Rys. 1.1. Sfera widokowa z perspektywą

Patrząc z punktu widokowego na obiekt widzi się oczywiście nie cały wielościan, a jedynie pewną jego część (elementy widoczne). Jeżeli przyjąć jako cechy wyróżnione obiektu np. ściany, krawędzie, wierzchołki (naturalne cechy dla wielościanu), to taki zbiór jednocześnie widocznych cech nazywa się **widokiem** (pojęcie bardzo blisko związane z aspektem, szerzej omówionym w rozdziale 2.3.1). Obszar na sferze widokowej, dla którego widok nie ulega zmianie, to **obszar jednowidokowy**. Na rysunku 1.2 znajduje się przykład sfery widokowej podzielonej na obszary jednowidokowe.



Rys. 1.2. Sfera podzielona na obszary jednowidokowe

Warunkiem koniecznym widoczności ściany jest spełnienie równania

$$\theta = \angle(\vec{N}, \overrightarrow{PW}) < \frac{\pi}{2} - \alpha, \quad (1.3)$$

gdzie \vec{N} oznacza wektor normalny danej ściany (zaczepiony w środku układu współrzędnych), \overrightarrow{PW} wektor o wierzchołku w punkcie widokowym (pokrywający się z kierunkiem rzutowania), a α , jak już wspomniano wcześniej, to połowa kąta rozwarcia indywidualnego¹ stożka SW. Do sprawdzania, czy dana ściana znajduje się w widoku jest użyteczne pojęcie **stożka dopełniającego (SD)**, zdefiniowanego jako stożek współosiowy ze stożkiem widokowym, o wierzchołku w punkcie $O = (0, 0, 0)$, tworzącej prostopadłej do tworzącej stożka widokowego i podstawie wspólnej ze stożkiem widokowym. Warunkiem koniecznym widoczności ściany (dla brył wypukłych jest to również warunek wystarczający) jest zawieranie się wektora normalnego ściany w stożku dopełniającym $\vec{N} \subset SD$.

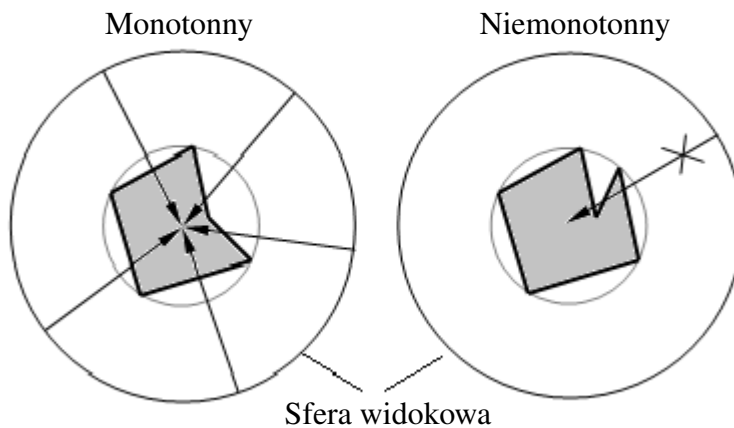
Ten model został początkowo zaplanowany do otrzymywania widoków dla wielościanów wypukłych. Warto jednak zastosować ten model do generowania widoków dla szerszej klasy brył. Przy tak przyjętym sposobie rzutowania jeżeli chce się, by dla każdej ściany był

¹ W stożku indywidualnym obejmujemy ścianę a nie całą sferę opinającą.

taki widok, w którym jest ona widoczna w całości stawia się warunek, by odcinek \overline{OPW} (fragment osi widokowej) przecinał powłokę wielościanu W w dokładnie jednym punkcie (rys. 1.3)

$$\forall \vec{PW} \exists! p \in R^3 \quad W \cap \overline{OPW} = p, \quad (1.4)$$

a ściany wielościanu były o kształcie gwiazdy. Tak zdefiniowaną klasę wielościanów nazywa się **wielościanami monotonnymi 1-rodzaju** (w dalszej części rozprawy nazywane krótko monotonnymi).



Rys. 1.3. Monotonność 1-rodzaju

Innym równoważnym sposobem opisu wielościanu monotonnego 1-rodzaju jest warunek aby dla każdej płaszczyzny zawierającej ścianę wielościanu o kształcie gwiazdy punkt widokowy był położony po przeciwnej stronie ściany niż środek geometryczny bryły.

Pojęcia tego nie należy mylić z definiowanymi przez Toussainta w [103] pojęciami słabej monotoniczności² (ang. *weakly monotonic*) wielościanu i słabej monotoniczności w sensie wypukłym³. Każdy wielościan monotorny jest słabo monotoniczny, natomiast nie musi być słabo monotoniczny w sensie wypukłym. Monotonność jest raczej uogólnieniem pojęcia wielokąta o kształcie gwiazdy (ang. *star-shaped*) gdzie wielokąt ma kształt gwiazdy gdy istnieje taki punkt p należący do wnętrza wielokąta prostego, że dla dowolnego punktu q należącego do wielokąta odcinek pq zawiera się w tym wielokącie (zbiór takich punktów p nazywamy jądrem wielokąta). Pojęcie to jest używane między innymi w określaniu widoczności na płaszczyźnie (np. [4]) oraz w geometrii obliczeniowej [27]. W literaturze spotyka się także wielościany o kształcie gwiazdy [11], w wielościanie monotonnym 1-rodzaju żądamy jednak aby dowolny punkt wielościanu był widoczny ze środka geometrycznego bryły.

Jeżeli zrezygnujemy z tego warunku i zostawimy warunki aby wielościan oraz jego ściany były o kształcie gwiazdy, to mówimy wtedy o **wielościanie monotonnym 2-rodzaju**.

² Wielościan jest słabo monotoniczny, gdy istnieje kierunek l , dla którego przekrój każdej płaszczyzny ortogonalnej do l daje wielokąt prosty.

³ Gdy przekrój wielościanu jest zbiorem wypukłym.

W wielościanach monottonnych **dla każdej ściany** istnieje punkt widokowy, z którego **otrzymuje się widok w którym jest ona widoczna w całości** (gdy linia rzutowania przechodzi przez punkt widokowy, środek bryły i jądro wielokąta będącego tą ścianą).

Przez **widok** w tej pracy będziemy rozumieć zbiór płatów powierzchniowych f_k (całych ścian lub ich części powstałych w wyniku samoprzesłoneń) widocznych z punktu widokowego

$$W = \{f_k : f_k \subset \acute{S}CW \vee f_k \subset \acute{S}CzW\}, \quad (1.5)$$

gdzie $\acute{S}CW$ oznacza zbiór ścian całkowicie widocznych a $\acute{S}CzW$ zbiór ścian częściowo widocznych.

Reprezentacja widokowa W_{rep} wielościanu to kompletny zbiór wszystkich możliwych widoków (dla ustalonego kąta α)

$$W_{rep} = \{W_i : W_i - \text{widok wielościanu}\}. \quad (1.6)$$

Kompletność reprezentacji oznacza, że są w niej wszystkie możliwe widoki badanego wielościanu lub, że z każdego punktu widokowego otrzymuje się widok, który jest już w utworzonej reprezentacji

$$\forall PW \exists W \in W_{rep}. \quad (1.7)$$

Baza modeli widokowych to zbiór wszystkich widoków wraz z informacją, do którego wielościanu (lub wielościanów) należy dany widok.

Mając już definicje potrzebnych pojęć można zdefiniować cel, zakres i tezy rozprawy.

1.3. Cel, zakres i tezy rozprawy

Celem rozprawy jest opracowanie metod i algorytmów stosownych do uzyskania kompletnej reprezentacji widokowej obiektu spełniającego podane niżej założenia przy pomocy koncepcji sfery widokowej z rzutowaniem perspektywicznym, porównanie i wybór najlepszego algorytmu do tego zadania oraz przedstawienie sposobu pasowania modeli widokowych obiektów z wyglądami obiektów pozyskanymi z otoczenia.

Założenia dotyczące obiektów, które są rozpoznawane:

- obiekty są wielościanami monottonnymi;
- obiekty nie są przezroczyste⁴;
- szukany obiekt nie jest przesłonięty przez inne obiekty⁵.

W pracach [40, 85] podano, że dla obiektu o n ścianach liczba podziałów przestrzeni widokowej oraz liczba widoków w rzutowaniu równoległym wynosi $O(n^2)$ dla wielościanu wypukłego i $O(n^6)$ dla wielościanu z wklęsłościami. Zaś złożoność obliczeniowa podziału

⁴ tzn. można pozyskiwać informacje o obiekcie za pomocą map zasięgu

⁵ co nie znaczy, że nie można wykorzystać modeli widokowych w tych systemach, tu przyjmujemy to założenie dla uproszczenia i skupieniu się na innych ważnych aspektach postawionego celu

przestrzeni widokowej i obliczenia aspekt-grafu wynosi odpowiednio: $O(n^2)$ i $O(n^6 \log n)$.
W przypadku rzutowania perspektywicznego liczba widoków wynosi $O(n^3)$ (wypukły) i $O(n^9)$ (z wklęsłościami) a złożoność konstrukcyjna odpowiednio $O(n^3)$ i $O(n^9 \log n)$.

Główne tezy rozprawy to:

1. Jest możliwe wytworzenie $3W$ przybliżonej reprezentacji widokowej wielościanu monotonnego z jego reprezentacji powłokowej B_{rep} za pomocą algorytmu o złożoności $O(n^2)$.
2. Minimalna liczba pasowań danych i modelu wielościanu monotonnego w oparciu o dane z mapy głębi i $3W$ reprezentację widokową zapewniająca identyfikację ma złożoność $O(n^2)$.

2. Przegląd literatury

Niniejsza praca dotyczy dwóch dziedzin: przetwarzania obrazów i widzenia maszynowego. W każdej z nich istnieje obszerna literatura. Ten rozdział dotyczy zagadnień odnoszących się do tematu rozprawy: porównaniu reprezentacji widokowej z innymi reprezentacjami oraz metod generowania reprezentacji widokowej, jako użytecznego sposobu reprezentacji obiektu 3W.

2.1. Reprezentacje obiektów do identyfikacji wizualnej - wymagania

Sposób reprezentacji obiektów jest istotnym krokiem, który rzutuje na proces rozpoznawania. Jain i współautorzy w [47] określają kluczowe elementy projektowania systemu rozpoznawania obiektów:

1. **Wybór odpowiedniej reprezentacji obiektów i utworzenie bazy danych zawierających te obiekty.**
2. Pozyskiwanie danych i wyodrębnienie cech wyróżnionych.
3. Wybór odpowiedniego sposobu pasowania danych z modelem.
4. Określenie sposobu wyboru obiektu (obiektów) pasującego do danych z największym prawdopodobieństwem.
5. Wyznaczenie metody weryfikacji postawionej w poprzednim kroku hipotezy.

Grimson w [41] grupuje cechy charakteryzujące SIW w następujący sposób:

1. **Cel** - jaki rodzaj obiektów ma być rozpoznawany.
2. **Niezawodność** - m. in. odporność na szum w sygnale wejściowym, a także na zasłonięcia obiektów na scenie.
3. **Wydajność** - szybkość oraz zapotrzebowanie na pamięć i moc obliczeniową.
4. **Poprawność** - dokładność systemu rozpoznającego.

Arman i Aggarwal w [3] precyzują użyteczność reprezentacji do pasowania: **jednoznaczność, unikalność, mała wrażliwość na zakłócenia i wygoda w użyciu.** W praktyce niektóre ze sposobów reprezentacji obiektów spełniają tylko część wymienionych kryteriów (np. wydajność), inne są lepsze, bo są np. bardziej dokładne i niezawodne.

W literaturze znajduje się wiele sposobów reprezentacji obiektów geometrycznych. Niektóre z nich są stosowane do rozpoznawania w systemach wizualnych, inne zaś (mimo że powszechnie stosowane np. do renderowania) nie znalazły zastosowania w systemach rozpoznających. W poniższym przeglądzie skupiono się na pracach dotyczących głównie

reprezentacji obiektów wielościennych i metod ich pasowania jako najważniejszych z punktu widzenia rozpatrywanych w rozprawie zagadnień. Najczęściej spotykany w literaturze podział to **reprezentacje skupione na obiekcie** (ang. *object-centered*), opisujące obiekt w pewnym lokalnym układzie współrzędnych skojarzonym z obiektem, oraz **skupione na widoku**, gdzie układ współrzędnych jest określony przez obserwatora (ang. *viewer-centered*) i trójwymiarowy obiekt jest najczęściej reprezentowany przez wiele jego dwuwymiarowych widoków (w rozprawie jest proponowana trójwymiarowa struktura widoków).

2.2. Reprezentacje obiektów 3W skupione na obiekcie

Reprezentacje obiektów trójwymiarowych można podzielić na: **reprezentacje objętościowe**, w których definiuje się obiekt przez pewien zbiór w przestrzeni 3W oraz **reprezentacje powłokowe** obiektu przez określenie jego powierzchni¹. Wyróżnić też można **reprezentacje punktowe**, w których obiekt jest reprezentowany za pomocą punktów na jego powierzchni.

2.2.1. Reprezentacje punktowe

Te rodzaje reprezentacji brył² są otrzymywane głównie podczas pozyskiwania danych. Ten rodzaj reprezentacji nie jest jednak jednoznaczny i nie nadaje się do generowania reprezentacji widokowej. **Chmura punktów** (ang. *point cloud*) jest niestrukturalnym zbiorem punktów 3W. Uporządkowany zbiór punktów znajdziemy w danych pozyskiwanych przez skanery zasięgu i przechowywanych w strukturach zwanych **obrazami zasięgu** (ang. *range images*) lub **mapami głębi** (ang. *depth maps*)³. Dane są przechowywane najczęściej w macierzach, w których wiersze i kolumny odpowiadają odpowiednim kątom w pionie i poziomie, a wartości to zmierzone odległości do obiektów. Umożliwia to często (po segmentacji) odtworzenie geometrii bryły. Dokładność współcześnie używanych urządzeń wynosi 0,5mm, [64].

Mapy zasięgu można generować wspomnianymi już metodami aktywnymi (skanery laserowe) lub metodami pasywnymi, głównie stereogrammetrycznymi, które nie będą tu omawiane. Przykładowe zastosowania sterowizji do rozpoznawania obiektów znajdują się w [18] i [21]. Zbiór punktów leżących na powierzchni trójwymiarowego obiektu⁴ jest często jednym z pierwszych kroków do wyznaczenia deskryptorów kształtu. Zwykle uzyskuje się to przez równomierny rozkład punktów na powierzchni. Ale Nehab i Shilane w [78]

¹ Zwykle zbiór dwumiarowych obiektów w przestrzeni, oddzielający wnętrze obiektu od części zewnętrznej.

² Całych lub pewnych ich fragmentów.

³ Często używa się tych dwóch pojęć wymiennie, chociaż można spotkać definicje, w których mapa głębi przechowuje więcej informacji o scenie niż tylko głębokość, lub że termin obrazu zasięgu określa dwuwymiarowy obraz zasięgu reprezentowany przez intensywność koloru zamiast wartości liczbowej.

⁴ Lub wierzchołków siatki wielokątów przybliżających kształt.

przedstawili technikę warstwowego rozkładu punktów i wykazali większą skuteczność dla zestawu różnych deskryptorów wygenerowanych przy pomocy tak rozłożonych punktów. Warstwowy model został uzyskany przez podział obiektu na woksle i umieszczanie punktów w wokslach brzegowych, a następnie usunięciu punktów położonych zbyt blisko siebie.

Jak wiadomo wksel jest pojęciem wolumetrycznym. Kolejnym omówionym typem reprezentacji będą właśnie reprezentacje z wykorzystaniem trójwymiarowych struktur posiadających pewną objętość.

2.2.2. Reprezentacje objętościowe

Reprezentacje objętościowe można podzielić na **konstruktywne** oraz **dekompozycyjne**. W podejściu **CSG** (ang. *Constructive Solid Geometry*) są one definiowane za pomocą pierwotników (takich jak sześcian, walec, sfera, stożek⁵) oraz regularyzowanych operacji boolowskich⁶ (zdefiniowanych w [87]) na nich. Obiekt jest reprezentowany jako drzewo z operatorami⁷ w węzłach wewnętrznych i pierwotnikami w liściach.

Wśród reprezentacji dekompozycyjnych czyli związanych z podziałem przestrzeni wymienić należy **reprezentację woksłową**, w której obiekt jest dekomponowany na identyczne komórki (woksle) uporządkowane według pewnej regularnej siatki. Hierarchicznym wariantem tej metody są **drzewa ósemkowe**, [16], w których obiekt otacza się prostopadłościanem dzielonym następnie rekurencyjnie na osiem mniejszych. Korzeń drzewa ósemkowego reprezentuje kostkę trójwymiarową (prostopadłościan lub w szczególności sześcian), a jego wierzchołki kolejne mniejsze prostopadłościany uzyskane w procesie rekurencji. W **drzewach binarnego podziału przestrzeni** (ang. *BSP - binary space partition*) dzieli się całą przestrzeń reprezentowaną przez korzeń drzewa za pomocą hiperpłaszczyzn. Po wstawieniu pierwszej ściany uzyskuje się dwa poddrzewa, które reprezentują półprzestrzenie. Dobierając odpowiednio kolejne hiperpłaszczyzny⁸ podziału można w sposób efektywny reprezentować całą bryłę. Drzewa BSP mają zastosowanie w algorytmach widoczności i wyznaczania cieni, a także w różnych zadaniach, w których ich celem jest obniżenie kosztu algorytmu w wyniku zmniejszenia ilości wykonywanych obliczeń. Te typy reprezentacji, mimo że są powszechnie używane w zastosowaniach CAD, inżynierskich i wizualizacji, nie nadają się wprost do identyfikacji obiektów i nie znalazły szerszego zastosowania w systemach rozpoznających.

Do metod objętościowych można także zaliczyć **reprezentacje z przesunięciem** (ang. *sweep representation*), w których jest definiowany dwuwymiarowy przekrój bryły, a następnie jest on przesuwany wzdłuż zadanej ścieżki. W czasie przesuwania jest wyznaczana powłoka bryły. Można też przesuwanie zastąpić obrotem, co umożliwi łatwe

⁵ W innym podejściu pierwotnikami są wyłącznie półprzestrzenie.

⁶ Takich jak suma, różnica, część wspólna.

⁷ Oprócz operacji boolowskich mogą to być także przesunięcia, obroty i skalowanie.

⁸ O dowolnej orientacji i położeniu.

definiowane brył obrotowych. Przykładem takiego podejścia jest **metoda uogólnionych walców** (ang. *generalized cylinders*), koncepcyjnie zbliżona do tej poprzedniej, polega ona na przesuwaniu dwumiarowej obwiedni wzdłuż zadanej ścieżki. W czasie przesuwania przekrój bryły można parametrycznie zmieniać (uogólnienie poprzedniego sposobu). Ta reprezentacja znalazła zastosowanie w zadaniach identyfikacji, choćby w opracowanym przez Brooksa systemie rozpoznawania ACRONYM, [13]. Badano też jej zastosowania w obrazach zasięgu, [28], podobnie jak superkwadryki, [48, 62]. Istotną wadą tego sposobu reprezentowania obiektów jest fakt, że tylko pewna grupa obiektów daje się łatwo w ten sposób opisać (szczególnie trudno jest uwzględnić lokalne deformacje bryły).

Rozpoznawanie obiektów reprezentowanych przez zajmowaną objętość nie jest najbardziej efektywną techniką. Patrząc na nieprzeźroczysty obiekt widzi się tylko jego powierzchnię, stąd też w rozpoznawaniu 3W często wykorzystuje się reprezentacje w których obiekt jest opisany przez jego powłokę.

2.2.3. Reprezentacje powierzchniowe

Wśród reprezentacji powłokowych wyróżnia się reprezentację B_{rep} (ang. *boundary representation*) definiującą obiekt określeniem wierzchołków, krawędzi i ścian (wraz z określeniem „wewnętrznej” i „zewnętrznej” strony) oraz ich wzajemnych przestrzennych zależności.

Jak podaje Stroud [99]⁹, pierwsze podstawy B_{rep} opracował Braid, które przedstawił w swojej pracy doktorskiej, [12]. Kolejnym badaczem, który wniósł znaczący wkład w podwaliny tej reprezentacji to Baumgart, [7]. Jego wkład to m.in.: struktura uskrzydłonych krawędzi (ang. *winged-edge*), [6], oraz wprowadzenie pojęcia zbioru operatorów Eulera¹⁰, które działają na obiektach spełniających regułę Eulera. Przekształcają je w nowe obiekty, dla których ta reguła obowiązuje. Dla przypomnienia, reguła Eulera dotyczy zależności pomiędzy ilością ścian, krawędzi i wierzchołków w wielościanach i można ją zapisać następującym wzorem:

$$V - E + F = 2(B - H), \quad (2.1)$$

gdzie: V jest liczbą wierzchołków i dalej odpowiednio: E - krawędzi, F - ścian, B - spójnych obszarów tworzących brzeg figury, H - dziur. Dla obiektu o jednej powierzchni bez dziur, wzór 2.1 można zapisać prościej $V - E + F = 2$.

W reprezentacji brzegowej bryła jest traktowana w sposób hierarchiczny - obiekt jest opisany za pomocą powierzchni brzegowych (dla wielościanu jest to zbiór ścian); ściany są reprezentowane za pomocą krawędzi; z kolei krawędzie są reprezentowane za pomocą wierzchołków. Jest to pełna i dokładna reprezentacja (opisana w pewnym lokalnym układzie współrzędnych) znajdująca szerokie zastosowanie, m.in. reprezentacja widokowa obiektów

⁹ Książka zawiera wyczerpujący opis reprezentacji brzegowej.

¹⁰ Należą do nich m.in. operatory dodawania i usuwania wierzchołków, krawędzi i ścian.

wielościennych jest generowana na podstawie tej struktury. Reprezentacje widokowe omawiane w tej rozprawie są generowane z reprezentacji brzegowej.

Przykładami innych reprezentacji powierzchniowych są: **siatka wielokątów**, **zbiór wielokątów** (ang. *polygon soups*) oraz w szczególności **model szkieletowy** (ang. *wireframe representation*) czyli zbiór wierzchołków, krawędzi i ścian (wieloboków) połączonych tak, że każda krawędź jest wspólna dla przynajmniej dwóch ścian. Reprezentację tego typu można zapisać bezpośrednio w oparciu o opis każdej z powierzchni (ścian), współrzędnymi wierzchołków ściany (co jest składowaniem nadmiarowych informacji) oraz wykorzystując wskaźniki na liście wierzchołków (co często prowadzi do utrudnionego odnajdywania wspólnych krawędzi). Innym sposobem zapisu jest reprezentacja za pomocą wskaźników na liście krawędzi (do listy wierzchołków dochodzi lista krawędzi związanych z wierzchołkami), która razem z listą wierzchołków opisuje ścianę.

Ten sposób reprezentacji (siatka wielokątów) jest popularny gdyż stosunkowo łatwo można przekształcać dane z map zasięgu do siatek wielokątów oraz można przybliżać siatką wielokątów obiekty o dowolnym kształcie. Był on jednak długo unikany w systemach rozpoznających ze względu na duże zapotrzebowanie na moc obliczeniową.

Nie wszystkie obiekty warto przybliżać siatką wielokątów. Niektóre z nich można przybliżać parametrycznie (powierzchnie Beziera, NURBS, itp.) lub opisywać w sposób algebraiczny. Pasko i inni w [83] zaproponowali, by cały obiekt próbować reprezentować pojedynczą funkcją o wartościach rzeczywistych (ang. F_{rep} - *function representation*). Dziedzina tej funkcji są punkty przestrzeni, przyjmując ona wartości dodatnie dla punktów wewnątrz bryły, ujemne dla punktów zewnętrznych i wartości zerowe dla punktów na powierzchni. Podejście takie jest użyteczne w renderowaniu brył, jednak ciężko je wykorzystać w systemie rozpoznawania. W tym zakresie większą popularność zyskały sposoby opisu bryły klasyfikowane jako **deskryptory kształtu**.

2.2.4. Deskryptory kształtu

Pojęcie deskryptora zyskało na znaczeniu wraz z upowszechnieniem standardu MPEG-7, [10]. Ten termin odnosi się nie tylko do obiektów dwuwymiarowych¹¹, ale jest też używany do opisu trójwymiarowych obiektów, gdzie jest pewnym odwzorowaniem z przestrzeni obiektów 3W do struktury zwanej **wektorem cech** (ang. *feature vector*), w którym różne cechy są w różny sposób przechowywane, w tym mogą różnić się wymiarem¹²(np. funkcje, histogramy, grafy,...). Deskryptor kształtu koduje pewne geometryczne i topologiczne właściwości obiektu, co umożliwia uproszczoną klasyfikację obiektów¹³.

¹¹ Wyróżniano tu deskryptory konturu i powierzchni.

¹² Z reguły im niższy wymiar, tym szybszy proces pasowania, ale traci się pewne informacje o obiekcie, gdyż najczęściej jest to odwzorowanie nieodwracalne.

¹³ Nie ma potrzeby pasowania całego trójwymiarowego obiektu oraz małe lokalne deformacje obiektu nie przeszkadzają w jego poprawnym sklasyfikowaniu.

W pracy [52] są sformułowane **pożądane cechy deskryptora kształtu 3W**. Są to:

- **łatwość wyznaczenia**,
- mały rozmiar i **łatwość przechowania** w bazie danych,
- **łatwość indeksowania**,
- **niezmienniczość względem pewnych przekształceń**,
- **mała wrażliwość na zaszumienie**,
- **niezależność** od sposobu opisu bryły lub przyjętego sposobu podziału,
- **odporność na małe** topologiczne **zmiany kształtu** i
- wystarczająca **dystynktywność**.

Deskryptory punktów należą do metod statystycznych i to czyni je dobrym narzędziem do analizy zaszumionych danych. Wykazują duży potencjał w użyciu do analizy obiektów dowolnego kształtu. Wyznacza się je najczęściej ze zbioru punktów¹⁴. Budując deskryptory zwykle korzysta się z położenia względnych typu centroid obiektu lub punkt, dla którego jest wyznaczany histogram, co czyni je często niezmienniczymi względem obrotów i przesunięć. Deskryptory można podzielić na następujące grupy: ze względu na sposób opisywania bryły na deskryptory globalne i lokalne, ze względu na sposób przechowywania danych na np. deskryptory liczbowe, histogramowe i grafowe, ze względu na precyzję opisywanych obiektów na dokładne (najczęściej są to deskryptory geometryczne) i przybliżone (statystyczne). Przykłady różnych typów deskryptorów są w następujących paragrafach.

Deskryptory globalne

Farias i de Carvalho w [32] do rozpoznawania obiektów 3W wykorzystują atrybuty typu powierzchnia, wektory normalne, liczba wierzchołków i centroid. Te cechy jednak silnie zależą od przyjętego sposobu segmentacji mapy zasięgu, co jest wadą takiego sposobu rozpoznawania. Konderink w [55] ostrzega przed systemami wykorzystującymi obliczanie powierzchni w systemach rozpoznających. Wykorzystując tego typu metody trzeba być świadomym problemu, że powierzchnia zależy od przyjętego sposobu triangulacji.

Jednym z pierwszych deskryptorów globalnych są **rozszerzone obrazy gaussowskie**. W deskrytorze EGI, (ang. *Extended Gaussian Image*), [43], obiekt jest reprezentowany w taki sposób, że jego wektory normalne są mapowane na sferę jednostkową (mapowanie gaussowskie) w taki sposób, że początki wektorów leżą w środku sfery a końce wektorów - na sferze. Deskryptor EGI rozszerza obraz gaussowski o informacje o rozmiarze powierzchni reprezentowanej przez ten wektor¹⁵. Tak zdefiniowany deskryptor jest unikalny (a nawet odwracalny)¹⁶ tylko dla obiektów wypukłych. W przypadku obiektów niewypukłych istotną wadą jest brak możliwości odtworzenia informacji o położeniu ścian obiektu. Problem ten rozwiązuje **reprezentacja CEGI**, (ang. *Complex Extended Gaussian Image*), [51], przez

¹⁴ Np. wierzchołki siatki wielokątów przybliżającej obiekt, kierunki wektorów normalnych, punkty równomiernie rozłożone na powierzchni obiektu.

¹⁵ Rozmiar powierzchni można reprezentować przed długość wektora lub przypisując mu pewną wagę.

¹⁶ Co oznacza, że można odtworzyć obiekt na podstawie deskryptora.

przypisanie każdemu wektorowi normalnemu wagi w postaci liczby zespolonej, której część rzeczywista reprezentuje powierzchnię ściany, a część urojona - odległość od ustalonego punktu¹⁷ do powierzchni ściany. W wyniku tego zabiegu CEGI może reprezentować większą bazę obiektów niż EGI. Problem ten rozwiązują także **reprezentacje DEGI** (ang. *Distributed Extended Gaussian Image*), **BGI** (ang. *Branched Gaussian Image*) i **HEGI** (ang. *Hierarchical Extended Gaussian Images*), [107]. Odwzorowanie obiektów na sferę nie musi dotyczyć wyłącznie wielościanów, ale wartości niezmiennicze muszą być tu inne (np. lokalna krzywizna), [95]. Ikeuchi i Herbert w [45] opracowali pojęcie powierzchni deformowalnej i wzajemnie jednoznacznego przekształcenia między tą powierzchnią a sferą. W ten sposób pewne lokalne atrybuty (np. krzywizna) mogą być przeniesione na sferę, stąd nazwa **sferyczny obraz atrybutu** (ang. *Spherical Attribute Image SAI*). Dokładny opis tej reprezentacji wraz z przeglądem innych sferycznych reprezentacji znajduje się w pracy [45].

Jedną z istotnych cech geometrycznych jest symetryczność obiektu. Jednym z popularnych deskryptorów opisujących symetryczność obiektu jest **RSD**, (ang. *Reflective Symmetry Descriptor*), [52], będący funkcją ciągłą, opisującą miarę symetryczności względem każdej płaszczyzny poprowadzonej przez środek ciężkości obiektu. Ze względu na fakt, że symetria obiektu jest własnością globalną ten typ reprezentacji jest użyteczny przy pasowaniu całych obiektów.

Innym przykładem globalnego deskryptora jest **dystrybuanta kształtu**, (ang. *shape distribution*), [81]. Opisuje ona kształt przez podanie rozkładu prawdopodobieństwa parametru, będącego funkcją losowo wybranych punktów należących do tej bryły (np: odległości pomiędzy dwoma losowo wybranymi punktami na powierzchni obiektu, odległości pomiędzy środkiem geometrycznym obiektu a punktem na powierzchni, kąt pomiędzy trzema losowymi punktami na powierzchni i inne podobne zależności, [80]). Pasowanie z użyciem tej reprezentacji sprowadza się często do znalezienia odległości pomiędzy dwiema funkcjami analitycznymi. Metoda jest dosyć szybka, jednak reprezentacja obiektu - niejednoznaczna. Dobrze są klasyfikowane pewne grupy obiektów, jednak ta metoda zawodzi, gdy zgrubny kształt obiektu jest podobny lecz różni się detalami. Skuteczność tego deskryptora zależy od wybranej funkcji kształtu.

Histogram kształtu

Ankerst i inni w [1] liczą **histogram kształtu** dzieląc przestrzeń 3W następująco:

- na komórki wyznaczone przez współśrodkowe sfery o różnych promieniach¹⁸;
- na rozłączne sektory o wspólnym wierzchołku w środku bryły. W przestrzeni 3W jest osiągnięte to w ten sposób, że na początku korzysta się z wierzchołków wielościanów regularnych, następnie w sposób rekurencyjny dzieli się ścianę na mniejsze

¹⁷ Najczęściej od środka geometrycznego bryły.

¹⁸ Maksymalna długość promienia zależy od rozmiaru największego obiektu w bazie danych.

części, następnie wyznacza się podział przestrzeni konstruując diagram Woronoja dla wyznaczonych punktów;

- uzyskuje się model mieszany używając jednocześnie dwóch poprzednich metod, umożliwiającą przechowanie bardziej precyzyjnej informacji o obiekcie.

Deskrytory statystyczne (np. histogramy) często umożliwiają klasyfikację i identyfikację obiektu w warunkach niepełnej informacji o bryle¹⁹. Jeszcze lepsze wyniki w tym zakresie są możliwe, gdy zamiast informacji o całej bryle (co nie zawsze jest możliwe do pozyskania) będą wykorzystywane pewne jej lokalne właściwości.

Deskrytory lokalne

Przykładem deskryptora lokalnego jest **sygnatura punktu** (ang. *point signature*). Chua and Jarvis w [19] używają tego pojęcia do opisu sąsiedztwa punktu. Sygnatura punktu jest funkcją tworzoną następująco: sfera o środku w testowanym punkcie i zadanim promieniu tworzy w przecięciu z powierzchnią pewną krzywą, którą można rzutować na płaszczyznę styczną do badanego punktu. Wykonując pełny obrót dookoła wektora prostej pokrywającej się z kierunkiem wektora normalnego (zaczepionego w tym punkcie), wyznaczamy odległość ze znakiem punktu na krzywej od jego rzutu na wyznaczoną płaszczyznę. Zatem sygnatura punktu jest funkcją przedstawiającą odległość powierzchni od płaszczyzny stycznej zależnie od kąta między wektorem przechodzącym przez punkt na rzutowanej krzywej i wektorem odniesienia²⁰. Dodatkowo wartości minimalne i maksymalne są używane jako indeksy pomagające w szybszym przeszukiwaniu bazy danych. Jest to deskryptor niezmienniczy względem przekształceń sztywnych.

Bardzo popularnym deskryptorem i szeroko opisywanym w literaturze są **obrazy spinowe**, [50] (ang. *spin images*). W celu ich wyznaczenia najpierw obiekt pokrywa się pewną regularną siatką. Następnie dla każdego wierzchołka w tej siatce wyznacza się obraz spinowy, będący histogramem liczebności pozostałych punktów z siatki w pewnym lokalnym, zmodyfikowanym walcowym²¹ układzie współrzędnych. Środkiem tego układu jest właśnie badany punkt, a oś walca pokrywa się z wektorem normalnym wyznaczonym w tym punkcie. Dwa parametry dwuwymiarowego histogramu to odległość radialna od osi walca i odległość od płaszczyzny stycznej, przechodzącej przez badany punkt, natomiast wartości, to liczebność wierzchołków w komórkach wyznaczonych przez zadaną rozdzielczość. Obiekt jest reprezentowany przez wiele takich histogramów, co wpływa na czas rozpoznawania i co jest wadą tego rozwiązania. Można ten czas skrócić przez wybór tylko części deskryptorów do pasowania. Metody te były szeroko badane w ostatnim dziesięcioleciu i doczekały się wielu modyfikacji, na przykład **kontekst trójwymiarowego**

¹⁹ Np. z powodu zaszumienia lub zasłonięć.

²⁰ Wektor odniesienia przechodzi przez punkt o maksymalnej odległości, oba wektory mają początek w testowanym punkcie i leżą na płaszczyźnie stycznej.

²¹ Pomija się tu współrzędną kątową i wszystkie punkty równo oddalone od wektora normalnego oraz leżące na tej płaszczyźnie o zadanim wektorze normalnym, które uznaje się za tożsame.

kształtu, [35] (ang. *3D shape contexts*) i **sferyczne obrazy spinowe** (ang. *spherical spin images*) [89] w którym obrazy obrotowe są odwzorowane na sferę jednostkową.

Innym przykładem lokalnego deskryptora są **lokalne płyty powierzchni**, [15] (ang. *Local Surface Patch, LSP*), w którym deskryptor pewnego wycinka powierzchni składa się z: centroidu płyta, typu powierzchni i dwuwymiarowego histogramu, określającego rozkład cech sąsiadów badanego punktu. Na jednej z osi histogramu są odkładane wartości indeksu kształtu, a na drugiej znajdują się kąty między kierunkami normalnymi do powierzchni w punkcie i w punktach sąsiednich. W celu przyspieszenia przeszukiwania LSP są indeksowane w tablicy mieszającej.

Wspólną wadą deskryptorów lokalnych jest duża liczba histogramów potrzebna do opisu obiektu, jeżeli chce się to zrobić w dokładny sposób²². Zaletą jest uwzględnianie w ten sposób przesłonięć (ponieważ kształt jest opisywany lokalnie) oraz stosunkowo duża odporność na zasłonięcia i zaszumienie danych.

Deskryptory grafowe

Przykładem grafowego deskryptora kształtu 3W jest **wielorozdzielczościowy graf Reeba MRG** (ang. *Multiresolutional Reeb Graph*) bazujący na koncepcji grafu Reeba, w którym za pomocą pewnej funkcji dzieli się obiekt na obszary. Węzeł grafu Reeba reprezentuje składnik danego obszaru połączony krawędziami ze składnikami obszarów sąsiednich. W podejściu MRG dzieli się na mniejsze i po ponownym wyznaczeniu dla nich grafu Reeba łączy z poprzednim. Istotnym krokiem budowy deskryptora MRG, jest wybór funkcji dzielącej obiekt 3W na obszary, w pracy [42] użyto funkcję odległości geodezyjnej. Ważną cechą deskryptora MRG jest fakt, że umożliwia hierarchiczne pasowanie od zgrubnego modelu do zawierającego więcej detali (ang. *coarse-to-fine*). Innym przykładem może być **graf szkieletowy**, (ang. *skeleton graph*) [100]. Zaletą deskryptora szkieletowego jest stała struktura grafu, także dla poruszającego się obiektu. Graf szkieletowy umożliwia też częściowe pasowanie, dodatkowo węzły grafu przechowują pewne lokalne deskryptory, co jeszcze usprawnia ten proces.

Reprezentacje omawiane w tym rozdziale w większości opisują bryłę w lokalnym układzie odniesienia, w którym z reguły bryła jest umieszczona w środku układu współrzędnych, a sposób opisu kształtu nie zależy od położenia obserwatora (ang. *object-centered*). Ten sposób reprezentacji z reguły nie nadaje się wprost do identyfikacji wizualnej. Innym sposobem jest reprezentowanie obiektów w układzie, którego środek wyznacza położenie obserwatora (ang. *viewer-centered*). Polemika, który z tych sposobów jest lepszy toczyła się już w latach osiemdziesiątych. Niektórzy (np. Marr w [68]) twierdzą, że reprezentacje „object-centered” są małe, niezależne od punktu widokowego i dlatego lepiej się nadają do identyfikacji wizualnej. Inni prezentują przeciwny pogląd, np. Rosenfeld

²² Wraz ze wzrostem liczby punktów i zmniejszeniem rozmiaru komórek wzrasta czas potrzebny na przeszukania bazy danych.

w [88] z kolei przedstawia ideę widoków charakterystycznych (zbiór widoków uzyskanych z pewnych charakterystycznych punktów widokowych, jak np. przedłużenie wierzchołków, krawędzi) i prezentuje pogląd, że reprezentacje „viewer-centered” odpowiadają sposobowi postrzegania człowieka i są użyteczne w systemach rozpoznawania w czasie rzeczywistym.

Autor tej rozprawy podziela pogląd, że reprezentacje oparte na widoku mogą przyspieszać rozpoznawanie. Dlatego w proponowanym sposobie identyfikacji używa modeli widokowych (widok jest ściśle powiązany z układem obserwatora), zaś do generowania widoków używa reprezentacji obiektu-centricznych ze względu na pełny opis geometryczny bryły.

Deskryptory oparte na widokach

Główną idea deskryptora LFD (ang. *Light Field Descriptor*), [14], jest reprezentowanie obiektu przez zestaw wyrenderowanych dwuwymiarowych obiektów, otrzymanych w wyniku rzutowania badanego obiektu pod różnymi kątami. Następnie tak uzyskane reprezentacje modeli mogą być ze sobą porównywane. Przy czym, wychodzi się z uproszczonego założenia, że dwa modele trójwymiarowe są do siebie podobne, gdy ich rzuty dwuwymiarowe z różnych punktów są podobne. LFD jest bardzo wydajnym deskryptorem zdobywającym szeroką popularność w SIW, gdyż używa się tu sfery widokowej (punkty są rozmieszczone w wierzchołkach wielościanu regularnego).

Omówiono tu tylko wybrane deskryptory kształtu, szerszy ich przegląd jest w [34].

Kontynuując sposób opisu obiektu z obserwatorem w środku układu współrzędnych, w następnym rozdziale zostaną omówione kolejne reprezentacje obiektów 3W.

2.3. Reprezentacje skupione na widoku

2.3.1. Grafy aspektowe

Idea grafów aspektowych (konceptyjnie bardzo bliska reprezentacji widokowej) została zaproponowana przez Koenderinka i van Doorna, [54]. Aspekt to jakościowa struktura obiektu widzialnego z pewnego punktu przestrzeni widokowej, rys. 2.1²³.

W strukturze grafu węzły reprezentują aspekty²⁴ zaś krawędzie - zdarzenia wizualne²⁵. Reprezentacja przez grafy aspektowe określa obszary o tym samym aspekcie (widoku²⁶) oraz przez krawędzie, które definiują relacje sąsiedztwa między kolejnymi widokami²⁷.

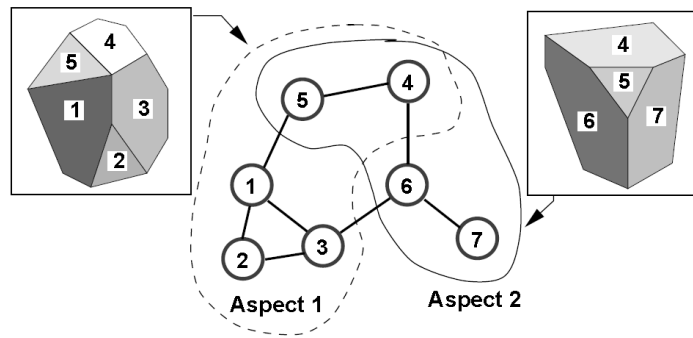
²³ Zbiór takich aspektów jest określany potencjałem wizualnym obiektu.

²⁴ Czasem zwane widokami ogólnymi (ang. *general views*), w których mały ruch punktu PW nie powoduje zmiany widoku.

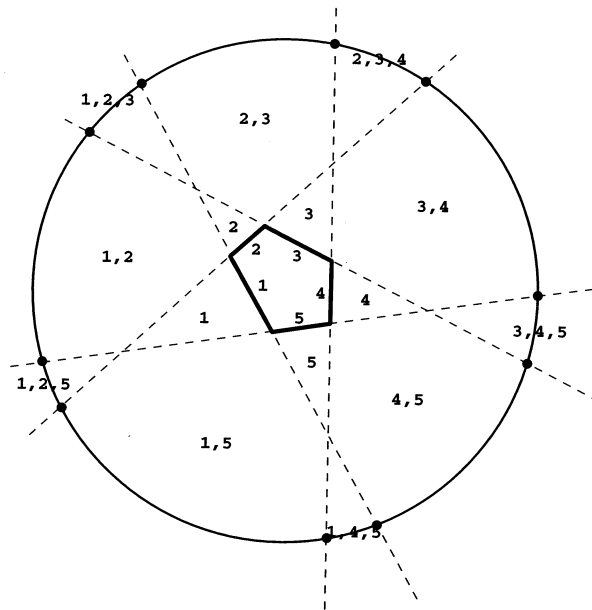
²⁵ Zwane też widokami szczególnymi (granicznymi) (ang. *accidental views*), gdzie najmniejszy ruch punktu PW spowoduje zmianę widoku.

²⁶ Ponieważ widok, rys. 2.2, to także pewna jakościowa struktura takich samych widocznych cech wyróżnionych, pojęć tych używa się wymiennie.

²⁷ Najczęściej różniące się jedną cechą dystynktywną.



Rys. 2.1. Graf aspektowy z wyróżnionymi dwoma aspektami, [76]



Rys. 2.2. Widoki wielokąta w 2W przestrzeni z rzutowaniem środkowym, [102]

Algorytmy generowania grafów aspektowych można podzielić na trzy zasadnicze grupy: ze względu na dziedzinę obiektów (np. wielokąty, wielościany, bryły obrotowe), sposób rzutowania w przestrzeni widokowej oraz sposób reprezentacji widoków. Najczęściej spotykanym sposobem reprezentacji to grafowa struktura **ISG** (ang. *Image Structure Graph*) dwuwymiarowych widoków o następujących cechach:

Składowe i właściwości ISG

- węzeł w grafie to (jeden) aspekt obiektu;
- przestrzeń widokowa jest podzielona na obszary;
- każdemu punktowi obszaru odpowiada dokładnie jeden i ten sam aspekt obiektu;
- różnym obszarom odpowiadają różne aspekty;
- graf aspektowy jest również grafem podziału przestrzeni widokowej;
- każda krawędź między węzłami grafu to możliwe przejście granicy między dwoma obszarami podziału;
- **zdarzenie wizualne** - zmiana aspektu, która następuje w chwili przekroczenia granicy.

Kluczowym elementem generowania reprezentacji skupionych na widoku jest określenie zbioru cech obiektu widzialnych z położenia obserwatora (gdzie jest teraz środek układu współrzędnych). Podstawową klasyfikacją przestrzeni widokowej jest podział ze względu na sposób rzutowania: wyróżniamy rzutowanie ortograficzne lub perspektywiczne. W pierwszym przypadku punkty widokowe są rozmieszczone na sferze o nieskończonym promieniu i reprezentują kierunek patrzenia. W rzutowaniu perspektywicznym punkt widokowy jest umieszczony w przestrzeni R^3 . Kierunek patrzenia zawsze przechodzi przez punkt widokowy i jest ograniczony polem widzenia, które przybiera najczęściej kształt stożka lub ostrosłupa o prostokątnej podstawie.

Punkty widokowe, w których wygląd obiektu przy nieskończonej małym ruchu obserwatora nie jest jakościowo różny, są uważane za stabilne, w przeciwieństwie do jakościowych zmian w wyglądzie (zdarzeń wizualnych), występujących na granicach obszarów widoczności cechy. Wszystkie sąsiednie, stabilne widoki mogą być grupowane w obszary, w których jakościowy opis obiektu (zwany często aspektem) nie ulega zmianie i razem z relacją sąsiedztwa między obszarami może być użyty do zdefiniowania struktury zwanej grafami aspektowymi.

Plantinga i Dyer w [84] zaproponowali, by zamiast zbioru topologicznie różnych widoków zapisanego w formie grafu, opisać obiekt jako funkcję położenia punktu widokowego. Reprezentacja ta zwana *asp*²⁸ jest zdefiniowana jako produkt płaszczyzny obrazu i przestrzeni widokowej. Jeżeli modelem przestrzeni widokowej jest sfera widokowa (jak w rzutowaniu ortogonalnym), to dla wielościanu *asp* jest strukturą czterowymiarową w przestrzeni aspektowej o następujących właściwościach:

- opisuje on pojawianie się wielościanu z usuniętymi niewidocznymi powierzchniami. Każda ściana jest reprezentowana przez strukturę, w której zasłonięte części są usunięte;
- przekrój reprezentacji aspektowej dla ściany daje informację o pojawianiu się tej ściany z usuniętymi niewidocznymi częściami;
- powierzchnie i krzywe graniczne w *asp*-ie odpowiadają zdarzeniom wizualnym, które nie są związane jedynie z obiektem (jak np. wierzchołki), ale również zawierają zdarzenia spowodowane przez rzutowanie, np. dodatkowe przecięcie się pary krawędzi.

Reprezentacja *asp* może być wykorzystana jako forma pomocnicza przy konstrukcji grafu aspektowego.

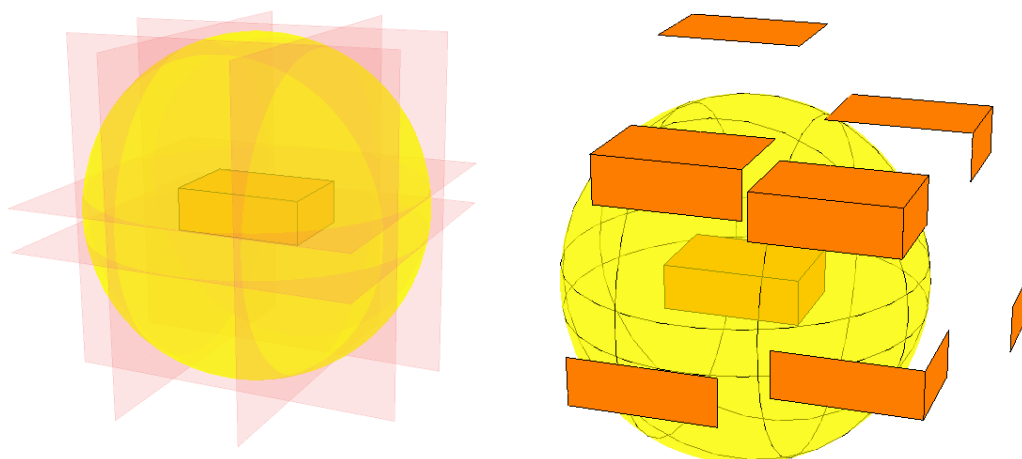
Jako rozwiązanie problemu dużej liczby węzłów w grafie aspektowym zaproponowano między innymi **aspekt-grafy ograniczonej rozdzielczości** (ang. *finite resolution aspect graph*), [94], rozróżniające obiekty różniące się o zadaną wartość progową; **aspekt-grafy skalo-przestrzenne** (ang. *scale-space aspect-graph*), [31], w których uznajemy, że aspekt-graf zawiera szczegóły nieistotne w pewnej skali dla obserwatora; **aspekt-grafy z przypisaną wagą** (ang. *weighted aspect graphs*), [2], przypisujące widokom pewną

²⁸ Lub reprezentacją aspektową.

wagę (prawdopodobieństwo wystąpienia w zależności od wielkości zajmowanego obszaru na sferze widokowej) i **hierarchiczne aspekt-grafy** (ang. *hierarchical aspect graph*), [8], łączące niektóre widoki w ten sam węzeł grafu.

Sposób wyznaczenia grafów aspektowych dla wielościanów jest przedstawiony m.in. w pracach [39, 40, 94, 98]. Ogólna myśl polega na wyznaczaniu obszarów zdarzeń wizualnych, a następnie podziału przestrzeni widokowej za pomocą tych hiperpłaszczyzn.

W wielościanie wypukłym zdarzenia wizualne związane są tylko z położeniem wektora normalnego ściany względem obserwatora. Obszary związane ze zdarzeniami wizualnymi to płaszczyzny zawierające ściany obiektu (rys. 2.3).



Rys. 2.3. Sfera widokowa - płaszczyzny widoczności i widoki

W wielościanie niewypukłym ściana leżąca we wklęsłości bryły może zostać przesłonięta przez inną ścianę. Jak wykazano w [40] dla obiektów wielościennych istnieją dwa podstawowe wizualne zdarzenia: gdy rzut krawędzi i wierzchołka pokrywają się (**zdarzenie** kw , ang. *ev*, *edge-vertex event*) oraz gdy rzuty trzech nierównoległych krawędzi przecinają się w punkcie (**zdarzenie** kkk , (ang. *eee*, *edge-edge-edge event*). Zatem szukając zdarzeń wizualnych związanych z przesłonięciami ściany, powodowanymi przez inne ściany, można się ograniczyć do zdarzeń typu kw i kkk . Obszar zdarzenia kw to płaszczyzna, która w przecięciu ze sferą widokową daje okrąg, natomiast obszar zdarzenia kkk to trójwymiarowa krzywoliniowa płaszczyzna drugiego stopnia.

Warto zauważyć, że liczba zdarzeń kw jest równa n^2 a zdarzeń kkk - $\leq n^3$. Zatem liczba obszarów na sferze widokowej wyznaczone przez powierzchnię zdarzeń kkk jest rzędu n^6 . Niektórzy autorzy, [33, 102], sugerują zrezygnowanie z badań zdarzeń kkk jako znacznie podwyższających koszt obliczeniowy algorytmów, a dodatkowe obszary jednowidokowe uzyskane z uwzględnieniem zdarzeń kkk są przeważnie małe i można je pominąć w praktyce.

Innym przykładem alternatywy dla grafów aspektowych w rzutowaniu perspektywicznym jest „**kompleks widoczności**” (ang. *visibility complex*). Pocchiola

i Vegter, [86], korzystając z obszarów o tej samej widoczności w przestrzeni 2W wprowadzili pojęcie dwuwymiarowego kompleksu widoczności²⁹. Ta topologiczna struktura pomagała określać widoczność z poszczególnych punktów sceny. Rozszerzenie tej koncepcji na przestrzeń 3W zaproponował Durand w [29], przedstawiając strukturę widokowych kompleksów (struktura dualna do linii używanych w metodach śledzenia promieni) do opisu widoczności, zdarzeń widokowych i relacji sąsiedztwa w przestrzeni. Złożoność tej metody $O(n^4)$ jest mniejsza niż w grafach aspektowych.

Uproszczone badanie obszarów widocznych występuje w przypadku wielościanów. Liu i inni w [67] dzielą ścianę wielościanu siatką trójkątną i dla każdego z wyznaczonych obszarów³⁰ określają obszar jego całkowitej widoczności na sferze³¹. W ten sposób zostaje wyznaczona mapa pełnej widoczności dla obszarów na granicach wielościanu z użyciem sum Minkowskiego, [66].

Obszerna analiza zagadnień widoczności znajduje się w pracach [20, 30].

Grafy aspektowe (wraz z całą grupą metod opartych na widoku) to jedna z najintensywniej rozwijanych i najbardziej rozbudowanych struktur obiektów. Do czynników przeszkadzających w szerszym wykorzystywaniu aspekt-grafów należy:

1. mała zmiana geometrii bryły może znacząco wpływać na liczbę zdarzeń wizualnych, a w konsekwencji na liczbę węzłów w grafie,
2. obszar, w którym są wyznaczone widoki, może być bardzo mały, co w konsekwencji daje małą szansę na otrzymanie takiego widoku,
3. przy dużej liczbie cech wyróżnionych zmiana między kolejnymi węzłami w grafie może być bardzo mała, w praktyce nierozróżnialna przez urządzenia pozyskujące,
4. rozmiar aspekt-grafów może być bardzo duży, co utrudni wyszukiwanie i pasowanie,
5. pasowanie konturów w przypadku reprezentacji dwuwymiarowej wymaga dokładnego wydzielenia cech i segmentacji,
6. trudności w dokładnym wyznaczeniu wszystkich widoków stabilnych,
7. widoki dwuwymiarowe nie pozwalają odróżniać obiektów pod względem skali.

Cyr i Kimia w [22] wykorzystują koncepcję grafów aspektowych jako krok pośredni do reprezentowania obiektu 3W przez dwuwymiarowe widoki. Do grupy reprezentacji widokowych należy też zaliczyć reprezentacje korzystające z sylwetki obiektu, [105]. Przegląd dotyczący grafów aspektowych można także znaleźć w pracy Schiffenbauera, [93]. Szersze rozważania na temat tego, co składa się na widok oraz wyszczególnienie zalet stosowania reprezentacji opartych na widokach znaleźć można w [102].

²⁹ W matematyce kompleksem sympleksyjnym nazywamy taki układ sympleksów $\{S_1, \dots, S_n\}$, że dla dowolnych $i, j = 1, \dots, n$ zbiór $S_i \cap S_j$ jest pusty lub jest wspólną ścianą tych dwóch sympleksów.

³⁰ Trójkątny fragment płaszczyzny, (ang. *facet*).

³¹ Przy pomocy obiektów 0, 1 i 2-komórkowych - zależnie od wymiaru obszaru) na powierzchni sfery.

Widoki 2.5W

Chcąc przyspieszyć proces rozpoznawania obiektów, Dąbkowska i Mokrzycki w pracach [23, 24, 26] zaproponowali koncepcję wykorzystania widoków 2.5W.

W opisywanym systemie rozpoznającym z użyciem 2.5W modeli widokowych zaplanowali następujące etapy:

- określenie klasy obiektów, jakie system będzie w stanie rozpoznawać i zbudowanie dla nich modeli (pierwotnie wielościany wypukłe),
- określenie zadania identyfikacji i wybranie cech wyróżnionych do identyfikacji,
- wygenerowanie 2.5W dokładnych modeli widokowych,
- zbudowanie bazy wszystkich widoków wszystkich modeli,
- pozyskanie danych wizualnych o scenie i segmentacja,
- spasowanie każdego z obiektów występujących na scenie z jednym (lub wieloma) widokami z bazy widoków.

Jak widać generowanie reprezentacji widokowej jest kluczowym krokiem w tym systemie. Sposoby generowania reprezentacji widokowych zostaną omówione w rozdziale 3, gdzie proponowana jest trójwymiarowa struktura widoku.

2.3.2. Reprezentacja widokowa ze sfery widokowej

Widoki 3-wymiarowe są czymś pośrednim między reprezentacją trójwymiarową a widokami płaskimi. Widoki 2.5W opisywane są w układzie współrzędnych obserwatora, natomiast widoki 3W są opisywane w układzie współrzędnych związanym z bryłą (ale wyznaczone są z punktu położenia obserwatora). Są one zdefiniowane przez widoczne płyty powierzchni obiektu, ale prócz numerów (identyfikatorów) płyt zawierają też informacje o 3-wymiarowych współrzędnych widocznych wyróżnionych punktów (w przypadku wielościanów będą to wierzchołki) w układzie globalnym. Zatem są to dane 3-wymiarowe, ale ograniczone tylko do widocznych elementów powłoki obiektu. Przy rozpoznawaniu można szybko kwalifikować widoki (jako pasujące lub nie), a przy wyświetlaniu cała potrzebna informacja jest zgromadzona w jednym miejscu i można ją szybko przetworzyć (przeorganizować) i wykorzystać.

2.4. Metody generowania reprezentacji widokowej

W tym rozdziale są przedstawione metody generowania reprezentacji widokowych z wykorzystaniem koncepcji sfery widokowej z rzutowaniem środkowym. Metody te bezpośrednio poprzedzały osiągnięcia autora pracy w tej dziedzinie i były dalej przez niego rozwijane. Wcześniejsze wyniki w tej dziedzinie zostały opisane w pracy doktorskiej Kowalczyk, [56]. Kryterium podziału tych metod może być klasa brył, dla których są generowane komplety widoków (można wyróżnić m.in. wielościany wypukłe i monotonne),

jak też zastosowane w nich algorytmy, które generalnie można podzielić na **iteracyjne** (2.4.1) oraz **nieiteracyjne** (podrozdział 2.4.2). Metody generowania widoków można podzielić też na te, w których jest wyznaczany obszar jednowidokowy, odpowiadający danym widokom oraz te, w których są wyznaczane tylko widoki (w tych metodach nie wykorzystuje się obszarów jednowidokowych). Stąd już bardzo blisko do następnego kryterium: czy jest w danej metodzie zagwarantowane generowanie wszystkich widoków³² czy też kompletność reprezentacji jest ograniczona do pewnej z góry założonej rozdzielczości³³. Tabela 2.1 przedstawia podział omawianych algorytmów biorąc pod uwagę dwa z wymienionych kryteriów. Organizacja tego rozdziału także wynika z tego podziału.

	Algorytmy iteracyjne	Algorytmy nieiteracyjne
Wyznaczanie obszaru jednowidokowego	Algorytmy wykorzystujące: wianki, PP i SD	przekraczanie granicy obszaru jednowidokowego, KM^\cap , KM^\cup ,
Bez wyznaczania obszaru jednowidokowego	metody z podziałem wielościanów regularnych	$FM^{V_{rep}^\cap}$, $MS^{N_{prep}^\cup}$, ruch spiralny wokół ściany, ruch spiralny po całej sferze

Tab. 2.1. Klasyfikacja algorytmów generowania widoków

2.4.1. Iteracyjne metody generowania widoków

Istotnym kryterium oceny algorytmów generowania widoków jest kompletność generowanej przez nie reprezentacji. Kompletność reprezentacji oznacza, że wszystkie widoki możliwe do uzyskania ze sfery widokowej zostały przez dany algorytm wygenerowane (czyli cała sfera jest pokryta obszarami jednowidokowymi odpowiadającymi uzyskanym widokom). Jednym ze sposobów generowania kompletnych reprezentacji widokowych jest iterowanie generowania widoku aż do pokrycia całej sfery obszarami jednowidokowymi (stąd też określenie „algorytm iteracyjny”).

Koncepcja algorytmów iteracyjnych

Zachowanie tej grupy algorytmów można zilustrować tak, jak pokazano to na rys. 2.4. Jako widoki podstawowe autorzy [23] proponują widoki ścienne³⁴, krawędziowe³⁵ lub wierzchołkowe³⁶. Dokładne algorytmy generowania tych widoków znajdują się

³² Również takich, w których obszar widokowy jest zredukowany do jednego punktu.

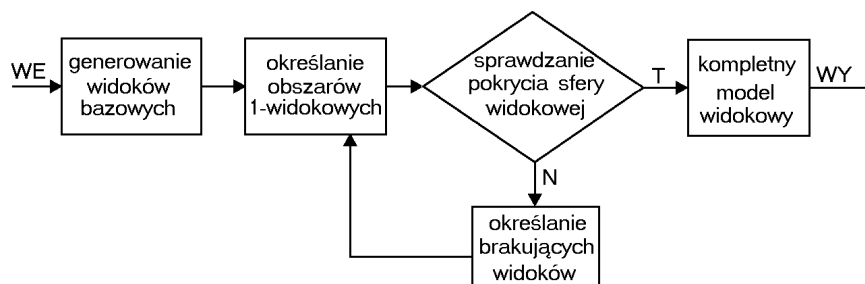
³³ Uzasadnionej np. rozdzielczością urządzeń pozyskujących dane wizualne o otoczeniu lub prawdopodobieństwem uzyskania danego widoku.

³⁴ Pozyskane z punktu, w którym oś stożka widokowego przechodzi przez środek ściany.

³⁵ Oś SW przechodzi przez środek krawędzi.

³⁶ Oś SW przechodzi przez wierzchołek.

w [23]. Kolejnym krokiem jest wyznaczenie obszarów widokowych odpowiadających uzyskanym widokom (dokładniejszy opis znajduje się w podrozdziale 2.4.1 niniejszej rozprawy). Następnie sprawdza się, czy cała sfera widokowa jest już pokryta obszarami jednowidokowymi. Autorzy korzystają tu ze spostrzeżenia, że każdy obszar jednowidokowy ma tylu sąsiadów, ile ma krawędzi³⁷. Do wyznaczania sąsiadów obszaru jednowidokowego (i kontroli kompletności reprezentacji) są wykorzystywane dwie tablice: T_{OOS} ³⁸ - tablica obszarów jednowidokowych ograniczonych przez daną ścianę (oddzielna dla każdej ściany) i T_{OSO} ³⁹ - tablica obszarów jednowidokowych sąsiadujących z danym obszarem jednowidokowym (oddzielna dla każdego obszaru jednowidokowego). Jeżeli w obu tych tablicach zostanie odnaleziona para obszarów, które mają wspólną krawędź to zostają uznane za sąsiednie, a jeżeli nie ma takiej pary dla ustalonej krawędzi to przyjmuje się, że ta krawędź „graniczy z dziurą”. Algorytm uzupełniania zbioru widoków dla tej metody nie został szczegółowo opracowany ani zaimplementowany. Jednym z ważniejszych kroków w tym algorytmie jest dokładne wyznaczenie obszaru jednowidokowego.



Rys. 2.4. Koncepcja algorytmów iteracyjnych, [23]

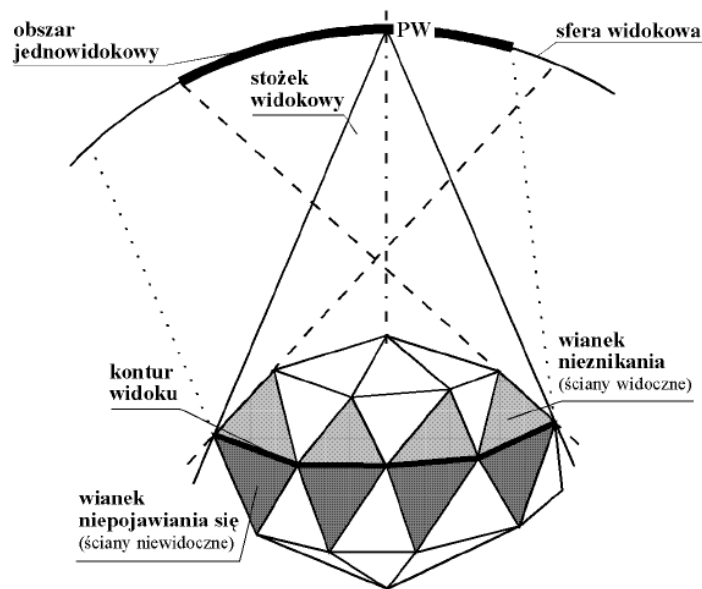
Wyznaczanie obszaru jednowidokowego

Mając widok można wyznaczyć kontur widoku KW , czyli linie (zbiór krawędzi) oddzielające widoczną część obiektu od części niewidocznej. Następnie wyznacza się wianek widoczny (ściany widoczne graniczące z KW) oraz wianek niewidoczny (ściany niewidoczne graniczące z KW). Płaszczyzny ścian wianka widocznego wycinają w sferze widokowej wielokąt otaczający obszar wianka widocznego. Obszar wianka niewidocznego jest z kolei wyznaczony przez płaszczyzny ścian wianka niewidocznego, rys. 2.5. Wyznaczenie granic obszaru widokowego opiera się na spostrzeżeniu, że część wspólna tych obszarów to właśnie obszar jednowidokowy zawierający widok początkowy. Obszary wianka widocznego i niewidocznego są wyznaczone za pomocą wierzchołków tych

³⁷ By być precyzyjnym, należy uściślić, że chodzi o sąsiadów różniących się o jedno zdarzenie wizualne (jedną ścianę w widoku).

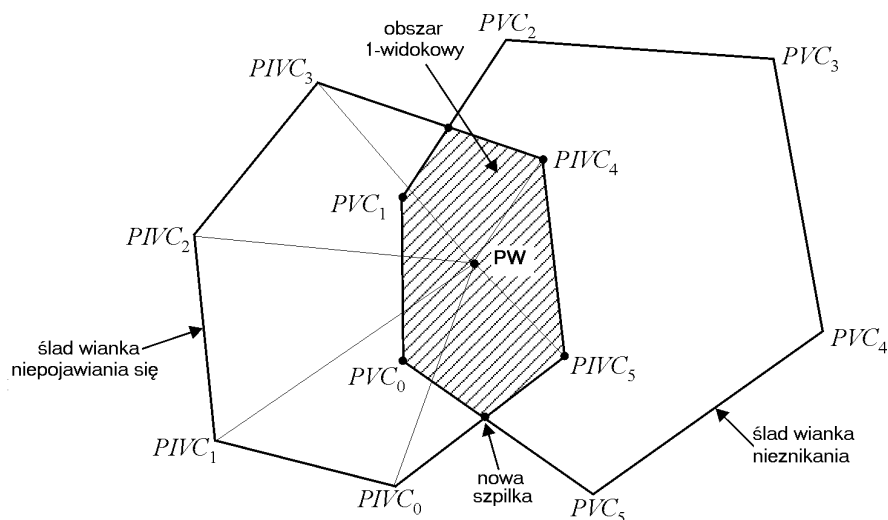
³⁸ Zawierająca numery obszarów jednowidokowych, tworzących granicę dla danej ściany oraz odpowiednie wierzchołki definiujące krawędź obszaru (zwane szpilkami).

³⁹ Zawierająca numery sąsiednich obszarów jednowidokowych oraz numery ścian, przez które to sąsiedztwo zachodzi.



Rys. 2.5. Wyznaczanie obszaru jednowidokowego przez wianki ścian, [26]

obszarów. Leżą one w punkcie przecięcia przedłużenia krawędzi wielościanu (część wspólna sąsiednich płaszczyzn) ze sferą widokową (wersory wystawione w tych punktach o kierunku zgodnym z tą prostą są zwane szpilki wianka widocznego lub niewidocznego, odpowiednio). Wspólny obszar jest wyznaczony przez punkty przecinania się tych obszarów oraz przez szpilki z jednego obszaru zawierające się w drugim obszarze (rys. 2.6). Zaletą

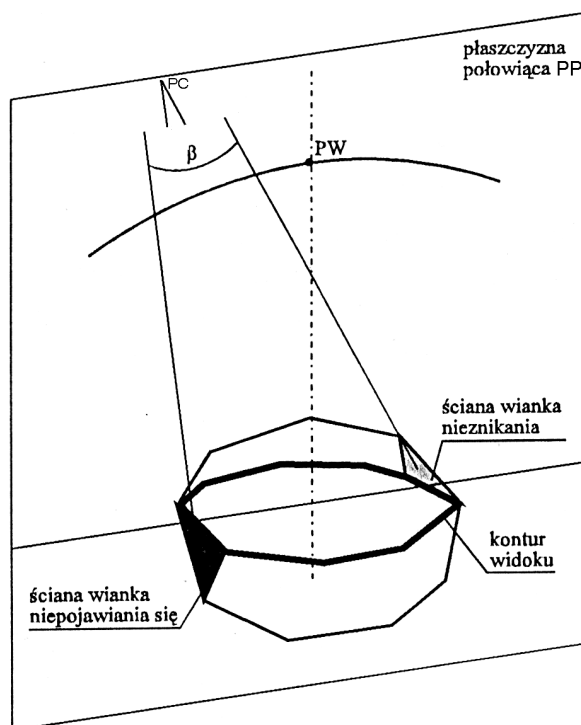


Rys. 2.6. Wyznaczanie obszaru jednowidokowego przez szpilki wianka widocznego i niewidocznego, [26]

tego rozwiązania jest precyzyjne wyznaczenie obszaru jednowidokowego zawierającego dany punkt widokowy. Wśród wad należy wymienić fakt, że znalezienie ścian konturowych (wianka widocznego) nie musi być zadaniem trywialnym (autorzy nie podają w jaki sposób się to odbywa). W dodatku wyznaczenie części wspólnej obszarów odbywa się w dość rozbudowanym, jak przyznają sami autorzy, czaso- i pamięcio- chłonnym algorytmie, dość

wrażliwym na błędy numeryczne. Można to uprościć, rzutując wianki na płaszczyznę dwuwymiarową i sprowadzając tym samym wyznaczenie obszaru jednowidokowego do klasycznego problemu geometrii obliczeniowej, mianowicie, znalezienia części wspólnej wielokątów wypukłych.

Pojęcie płaszczyzny połowiącej PP, rys. 2.7, [25], zostało wprowadzone po to, by proces wyznaczenia obszaru jednowidokowego był bardziej efektywny. Terminem tym określa się płaszczyznę przechodzącą przez środek krawędzi konturu widoku oraz oś widokową i wykorzystuje się ją w algorytmie zwanym *metodą bezpośredniego wyznaczenia obszaru jednowidokowego*. Polega ona na wybraniu pewnej krawędzi z konturu widoku, uzyskanego z punktu widokowego. Dla tej krawędzi wyznacza się PP a następnie obraca stożek widokowy (w płaszczyźnie PP względem początku układu współrzędnych), aż punkt widokowy dotrze do płaszczyzny ściany widocznej, przyległej do tej krawędzi ($s_{W_{w1}}$). Jeśli w tym czasie nie pojawi się nowa ściana w stożku, to aktualna pozycja punktu widokowego należy do krawędzi obszaru widokowego, będącej częścią wspólną płaszczyzny ściany widocznej i sfery widokowej. Jeżeli natomiast pojawia się jakaś nowa ściana ($s_{W_{nj}}$ z przeciwnej strony konturu widoku), to właśnie ona wyznacza krawędź graniczną obszaru jednowidokowego. Następnie punkt widokowy wraca do położenia początkowego i jest



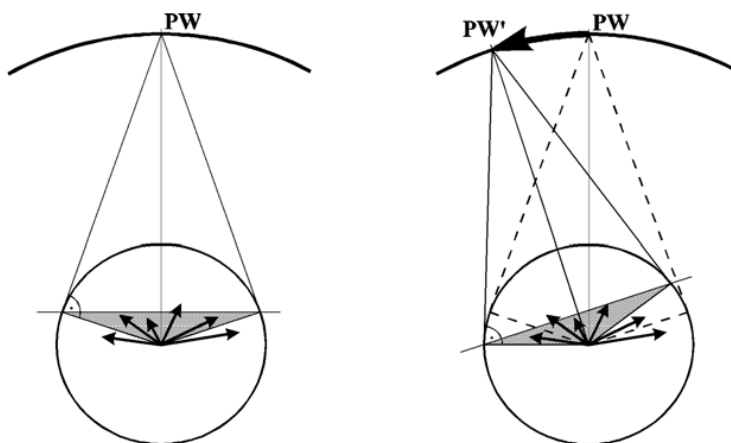
Rys. 2.7. Koncepcja płaszczyzny połowiącej, [25]

powtarzany ten proces dla wszystkich krawędzi konturu widoku, a wygenerowane krawędzie tworzą kształt wielokąta wypukłego, będącego szukanym obszarem widokowym. Opisane czynności są powtarzane dla wszystkich wygenerowanych widoków. W praktyce rozstrzyga się, czy płaszczyzny $s_{W_{w1}}$ i $s_{W_{nj}}$ są zbieżne lub rozbieżne w obszarze stożka widokowego

i na podstawie odpowiednich zależności kątowych określa, która z tych dwóch płaszczyzn ogranicza obszar jednowidokowy.

Metoda ta jest prostsza i szybsza od metody ze szpilkami. Wyznacza ona krawędzie obszaru jednowidokowego umożliwiające precyzyjnie ustalić obszar jednowidokowy.

Korzystając z koncepcji PP, wyznaczenie obszaru jednowidokowego można dodatkowo przyspieszyć, wykorzystując koncepcję stożka dopełniającego SD. Przypomnijmy, że dla wielościanów wypukłych spełniony jest warunek mieszczoności kierunków wektorów normalnych ścian w kącie rozwarcia SD⁴⁰. Dla określonego ruchu punktu PW (np. po wielkim kole, wyznaczonym przez PP), można wskazać wektor normalny, który pierwszy przetnie się z powierzchnią stożka dopełniającego (rys. 2.8). Przecięcie płaszczyzny ściany odpowiadającej temu wektorowi wyznaczy pierwszą krawędź obszaru jednowidokowego. Mając dany sposób wyznaczania obszaru jednowidokowego, można szukać coraz lepszych



Rys. 2.8. Ruch PW w kierunku najbliższego zdarzenia wizualnego, [58]

sposobów wyszukania wszystkich możliwych widoków.

Przedstawione zostaną teraz pewne metody, które robią to szybciej niż już omówione.

Opisane w tym podrozdziale metody wymagały sprawdzania czy sfera widokowa jest już całkowicie pokryta obszarami jednowidokowymi. Jeśli tak nie było to powtarzano procedurę generowania widoków (rys. 2.4), stąd też pochodzi określenie tych algorytmów jako iteracyjne, [59]. Inne sposoby generowania widoków poprzez konstrukcję algorytmu nie wymagały sprawdzania kompletności generowania reprezentacji, algorytm najczęściej działał bez pętli i klasyfikowane były jako algorytmy nieiteracyjne, [60].

2.4.2. Nieiteracyjne metody generowania widoków

Metoda przekraczania granic obszaru zarodkowego

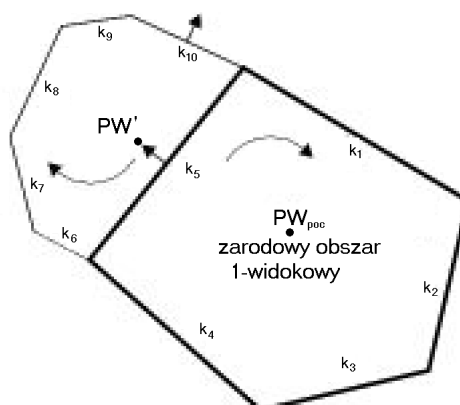
Metoda przekraczania granic obszaru zarodkowego, [60], jest następująca: wylicza się pierwszy widok przy dowolnym punkcie PW oraz jego obszar jednowidokowy, np. metodą

⁴⁰ Czasem zachodzi potrzeba uwzględniania indywidualnych stożków dopełniających.

z użyciem płaszczyzny PP. Ten pierwszy obszar jednowidokowy (ZOW, rys. 2.9) staje się miejscem początkowym na sferze SWP, do którego będą sukcesywnie dołączane w sposób uporządkowany (spiralny) kolejne obszary jednowidokowe sąsiadujące wzdłuż wszystkich krawędzi obszaru zarodkowego.

Generowanie drugiego i dalszych widoków opiera się na spostrzeżeniu, że obszary jednowidokowe na sferze SWP dla wielościanów wypukłych ściśle pokrywają powierzchnię tej sfery i przylegają wzdłuż całych krawędzi do siebie, tzn. każda krawędź konturu dowolnego obszaru jednowidokowego pokrywa się z dokładnie jedną krawędzią innego obszaru jednowidokowego. Zatem, mając jeden (zarodkowy) obszar jednowidokowy, następnie można wyliczyć przesuwając punkt PW poza granice pierwszego obszaru jednowidokowego na obszar sąsiedni i obliczenie nowego widoku oraz wyznaczenie jego obszaru jednowidokowego (na którym znajduje się aktualnie punkt PW). Ten nowy obszar będzie graniczył z poprzednim wzdłuż krawędzi, którą przekroczył punkt PW .

Do powiększającego się w ten sposób obszaru sfery SWP, pokrytego przez obszary jednowidokowe, są przyłączane następne, sąsiadujące obszary, aż do pełnego pokrycia sfery SWP. Pełne pokrycie sfery SWP przez obszary jednowidokowe jest stwierdzane na podstawie stanu rejestru granicznego RG.



Rys. 2.9. Przekraczanie granic obszaru jednowidokowego, [60]

Rejestr RG to rejestr, w którym są zapisywane zewnętrzne krawędzie grupy dotąd wyznaczonych obszarów jednowidokowych, czyli krawędzie oddzielające przebadaną (pokrytą) część sfery SWP od niepokrytej. Gdy cała sfera SWP zostanie pokryta obszarami jednowidokowymi, rejestr RG stanie się pusty. Będzie to znaczyło, że każdy obszar jednowidokowy ma określonego sąsiada wzdłuż każdej swojej krawędzi. Wówczas, dla każdego punktu na sferze SWP jest znany widok modelu, czyli jest znana kompletna baza widoków standardowych modelu.

Metoda ta nie zależy od wyboru punktu widokowego. Kompletność generowanej reprezentacji wynika wprost ze sposobu przemieszczania punktu widokowego (do

wszystkich swoich sąsiednich punktów widokowych). Jeśli by istniał obszar jednowidokowy nie wygenerowany tą metodą to on również miałby pewne granice i swoich sąsiadów i rekurencyjnie znaleźlibyśmy się w pierwszym obszarze jednowidokowym, który na pewno został wygenerowany. Zaznaczyć przy tym należy, że metoda ta generuje jedynie komplet widoków stabilnych (takich, które mają niezerową⁴¹ powierzchnię i nieskończenie mały ruch nie powoduje zmiany widoku), natomiast nie generuje widoków niestabilnych (np. redukujących się do jednego punktu). Słabością tej metody jest także fakt, że nie jest określony krok, o jaki możemy przekraczać granicę obszaru jednowidokowego bez ryzyka, że znaleźlibyśmy się „za daleko” i przeoczyli pewien widok.

Metoda śledzenia granicy obszaru jednowidokowego

W ogólności metoda śledzenia granicy obszaru jednowidokowego ([61], algorytm KM^n) polega na wyliczaniu widoków sąsiadujących z już wyznaczonymi widokami, korzystając z płaszczyzn granicznych wyznaczonych wcześniej obszarów jednowidokowych, a następnie znalezienie odpowiadających im obszarów jednowidokowych. Poszukiwania widoków i obszarów jednowidokowych odbywają się spiralnie wokół pierwszego (zarodkowego) obszaru jednowidokowego i prowadzą do pełnego pokrycia sfery, czyli do wygenerowania pełnego zbioru widoków.

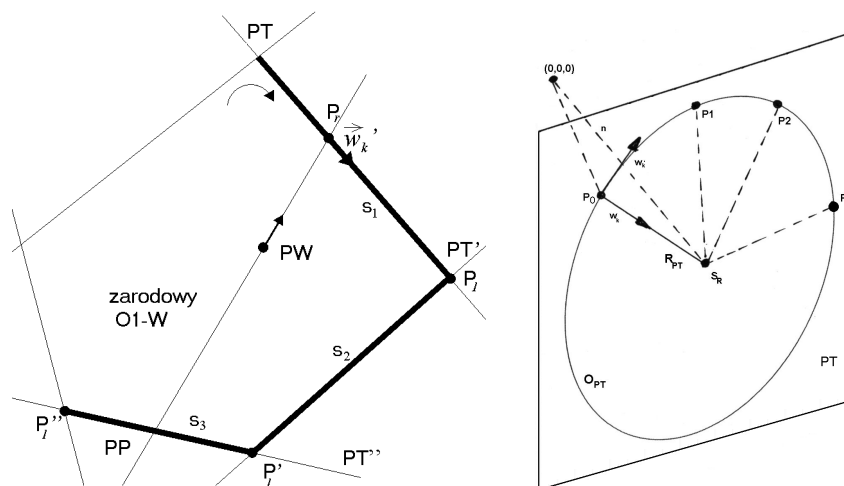
U podstaw niniejszej metody leży myśl, że w generowaniu kompletnej reprezentacji widokowej bryły pierwotnym pojęciem jest podział sfery SWP na obszary jednowidokowe. **Jest on wynikiem określonej geometrii bryły i układu na niej cech, które zostały „wytypowane” jako składowe modelu widokowego tej bryły.** Natomiast **widok jest pojęciem drugorzędym.** Z każdym obszarem jednowidokowym jest niewątpliwie sprzężony na stałe (jest mu przypisany) określony widok. Natomiast punkt PW wskazuje (przez swoje współrzędne) tylko ten obszar jednowidokowy, którego widok osiąga się. Z tego wynika, że szczelność pokrycia należy osiągać wprost dla (sąsiednich) obszarów jednowidokowych. Autorzy metody proponują następujący tego scenariusz:

Ustawia się oś widokową - jak dotychczas, by wyznaczyć widok. Ale obszar jednowidokowy wyznacza się śledząc jego granice. Pierwszą krawędź graniczną pierwszego (zarodkowego) obszaru jednowidokowego znajduje się przyjmując dowolny punkt PW i stosując mechanizm płaszczyzny PP oraz wykorzystując tę właściwość obszaru jednowidokowego, że **jeśli płaszczyzna PP przetnie którąś z najbliższych punktowi PW płaszczyzn ścian bryły, to będzie to na pewno płaszczyzna graniczna tego obszaru jednowidokowego, na którym leży wybrany punkt PW**, gdyż z definicji obszaru jednowidokowego (żaden) **obszar jednowidokowy nie jest przecinany przez jakąkolwiek ścianę bryły**, a więc i żadną ze ścian wianków. Zatem, mając punkt PW (który jest z definicji na pewnym obszarze jednowidokowym) oraz jedną krawędź poszukiwanego obszaru jednowidokowego (utworzoną przez płaszczyznę ściany bryły, np. s_1), można

⁴¹ W praktyce odpowiednio dużą.

wyznaczyć pozostałe ściany graniczne obszaru jednowidkowego następująco: Wybiera się początkowy punkt widokowy PW_{poc} (rys. 2.10a) i oblicza (początkowy) widok WID_{poc} , dokładniej, kompletuje się zbiór ścian wianków widoku WID_{poc} , będących kandydatkami na ściany graniczne i znajduje się pierwszą ścianę graniczną przy użyciu płaszczyzny PP wyznaczonej przez PW_{poc} i wektor normalny pierwszej ściany widoku. Płaszczyzna PP w przecięciu ze sferą SWP tworzy okrąg. Punkt PW , startując ze swego pierwotnego położenia PW_{poc} w kierunku składowej wektora normalnego pierwszej ściany WID_{poc} , porusza się po tym okręgu, aż napotka (w punkcie P_0) jedną ze ścian kandydatek, rys. 2.10b. Będzie to ściana graniczna obszaru jednowidkowego. Dalej, tzn. od punktu P_0 porusza się po okręgu wyznaczonym na sferze SWP przez pierwszą ścianę graniczną, w obranym kierunku obiegu po konturze, rys. 2.10b. Wyznacza się teraz punkty przecięć tego okręgu z płaszczyznami pozostałych kandydatek na ściany graniczne. Drugą ścianą graniczną będzie ta, której płaszczyznę najwcześniej spotka punkt PW w ruchu po okręgu.

Po napotkaniu drugiej ściany granicznej (co nastąpi w pierwszym znalezionym wierzchołku obszaru zarodowego), punkt PW porusza się już po nowym okręgu, wyznaczonym przez płaszczyznę tej (drugiej) ściany. Dalej, gdy napotka płaszczyznę kolejnej (trzeciej) ściany-kandydatki będzie się po niej poruszać w poszukiwaniu następnej (czwartej) ściany granicznej. Ten cykl poszukiwań ustaje, gdy punkt PW dotrze do płaszczyzny pierwszej ściany granicznej. Wówczas kończy się proces wyznaczania zarodowego obszaru jednowidkowego.



Rys. 2.10. Szukanie pierwszej i dalszych krawędzi obszaru zarodowego, [60]

Przejście do sąsiedniego obszaru widokowego i zmiana widoku odbywa się następująco:

Należy wziąć ostatnią krawędź z rejestru RG , która jest zarazem tzw. krawędzią wyjścia, i odczytać numer ściany s , która ją utworzyła. Widok sąsiedni (przylegający wzdłuż krawędzi wyjścia) będzie się różnił od ostatniego wyłącznie tą właśnie ścianą, czyli:

- jeśli ściana s była niewidoczna, to teraz zbiór widocznych ścian powiększy się o s ,
- jeśli ściana s była widoczna, to teraz zbiór widocznych ścian zmniejszy się o ścianę s .

Odpowiadający mu obszar jednowidokowy jako (pierwszą) ścianę graniczną będzie miał s (dotychczasowa krawędź wyjścia stanie się jego pierwszą krawędzią) - punkt widokowy PW stanie po przeciwnej stronie tej ściany, niż w ostatnim widoku. Wierzchołki obszaru wyznaczone przez pierwszą krawędź (krawędź wyjścia poprzedniego obszaru) należy teraz wziąć w odwrotnej kolejności niż, gdy tworzyły poprzedni obszar. Poczynając z pierwszego wierzchołka, który teraz jest punktem odniesienia, należy poruszać się po okręgu utworzonym przez płaszczyznę ściany s (obecna płaszczyzna testująca PT) wyznaczając kolejne punkty przecięcia z pozostałymi płaszczyznami ścian. Zachowuje się przy tym kierunek ruchu zgodny z ruchem wskazówek zegara. Dalej proces wyznaczania obszaru jednowidokowego przebiega tak, jak w przypadku obszaru zarodowego.

W tej metodzie na początku zostaje wybrany dowolny punkt widokowy (także wyznaczony widok początkowy) oraz wyznaczony odpowiadający mu obszar jednowidokowy. Korzysta się tu ponownie z koncepcji płaszczyzny połowiącej do wyznaczenia pierwszej ściany granicznej. Następnie poruszamy się po okręgu wyznaczonym przez sferę widokową i płaszczyznę zawierającą ścianę graniczną aż napotkamy następną ścianę graniczną⁴². Dalej poruszamy się w podobny sposób wzdłuż tej ściany oraz ścian kolejno wyznaczonych, aż dojedziemy do płaszczyzny pierwszej ściany granicznej kończąc wyznaczanie ZOW. Krawędzie tego obszaru zostają dodane do RG pełniącego ponownie rolę kontrolną kompletności generowanej reprezentacji. Ostatnia dodana krawędź do RG staje się krawędzią wyjścia a sąsiadujący z nim widok będzie się różnił o jedną ścianę. Odpowiadający mu obszar jednowidokowy wyznaczamy poruszając się wzdłuż krawędzi wyjścia w odwrotnym kierunku (będzie to krawędź wejścia nowego obszaru jednowidokowego) i postępując tak jak przy wyznaczaniu ZOW. Kolejno przechodząc wzdłuż krawędzi (a nie przekraczając krawędzie jak w poprzedniej metodzie) wyznaczamy kolejne obszary jednowidokowe (i w konsekwencji widoki). Dołączanie kolejnych obszarów kończymy gdy RG staje się pusty.

Metodę poruszania się po granicy obszaru jednowidokowego Kowalczyk uznaje za najlepszą opracowaną przez siebie⁴³ do generowania widoków i obszernie opisuje w swojej pracy doktorskiej, [56]. Koszt wyznaczenia obszaru jednowidokowego wielościanu wypukłego o n ścianach jest rzędu n^2 a ilość widoków jest proporcjonalna do kwadratu liczby ścian. Złożoność obliczeniowa omawianej metody wynosi zatem $O(n^4)$.

Mankamentem tej metody jest skupienie się na precyzyjnym wyznaczaniu obszarów jednowidokowych (gdzie głównym zadaniem badawczym jest wyznaczenie widoku), a ponadto do wyznaczenia obszarów jednowidokowych poruszamy się dwukrotnie wzdłuż tej samej krawędzi obszaru jednowidokowego (tylko w różnych kierunkach). Kolejny algorytm jest wolny od tej wady.

⁴² Przyjmuje się stały kierunek ruchu po konturze (np. ruch zgodny z ruchem wskazówek zegara).

⁴³ Jako współautor metody z Mokrzyckim.

Metoda śledzenia granicy obszaru jednowidokowego dla wielościanów monottonnych

Opisana w poprzednim paragrafie metoda śledzenia granicy obszaru jednowidokowego (zrealizowana w algorytmie KM^{\cap}) dotyczyła wielościanów wypukłych. Opisywana niżej metoda obliczania widoków (algorytm KM^{\cup} , [57]) jest uogólnieniem metody poprzedniej na pewną podklasę nieprzezroczystych wielościanów niewypukłych zwanych wielościanami monottonnymi 1-rodzaju.

W wypadku monottonnych wielościanów w widokach mogą wystąpić **niewidoczności** - gdy pewna ściana leżąca wewnątrz konturu widoku będzie zwrócona tyłem do kierunku osi widokowej (jej wektor normalny będzie poza stożkiem SD) - oraz **zasłonięcia** - gdy pewne ściany, leżące wewnątrz konturu widoku będą zasłonięte przez krawędź konturu wklęsłości.

Gdy wewnątrz konturu widoku będą niewidoczne ściany, wówczas - razem ze ścianami widocznymi, będą tworzyły krawędzie konturu niewidoczności w widoku (wewnątrz konturu podstawowego widoku). Tak więc, gdy na etapie łączenia krawędzi konturu widoku, nie wszystkie krawędzie zostaną w jeden kontur widokowy włączone oznaczać to będzie, że w widoku występują niewidoczności i te „niewykorzystane na kontur widokowy” krawędzie łączy się w inne kontury (niż kontur widoku).

Każdy kontur niewidoczności będzie brzegiem grupy niewidocznych ścian widoku, a ściany niewidoczne - ścianami wklęsłości, graniczącymi ze ścianami widocznymi: jedną z obszaru wypukłości, pozostałymi - ścianami wklęsłości.

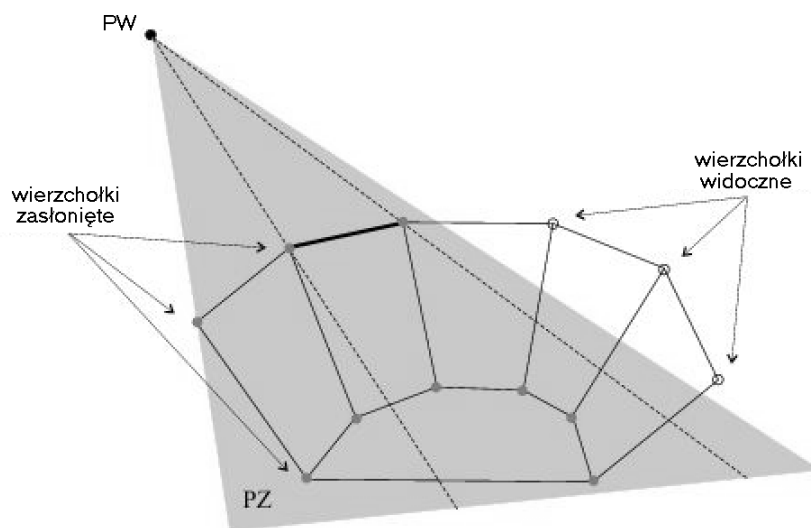
Przesłanki dotyczące **występowania wklęsłości** i ścian wewnątrz niej są następujące: Ściany w przestrzeni stożka SD z zewnętrznymi wektorami normalnymi są niewidoczne, a więc są ścianami wklęsłości. Gdy w widoku występuje ściana niewidoczna, to występuje również wklęsłość, wówczas pewne ściany z wnętrza konturu widoku są ścianami wklęsłości, więc leżą wewnątrz konturu wklęsłości. Natomiast **kryterium leżenia wewnątrz konturu wklęsłości** może być **wzajemne przecinanie się prostych pokrywających się z kierunkami wektorów normalnych ścian, zaczepionych w środku układu współrzędnych**, leżących wewnątrz konturu wklęsłości.

Mogą występować: zdarzenia wizualne, zasłonięcia i dodatkowe widoki. Zdarzenia wizualne rozumiane jako nowe widoki powstają, gdy:

- na obrzeżu widoku pojawiła się lub znikła jedna ściana,
- na obrzeżu widoku jedna ściana się pojawiła, a inna znikła,
- jakaś ściana z wklęsłości stała się niewidoczna,
- jakiś wierzchołek wewnątrz konturu wklęsłości pojawił się bądź zniknął (stał się widoczny lub niewidoczny),
- nastąpiła równoczesna zmiana na obrzeżu widoku i wewnątrz konturu wklęsłości;

Czy i które ściany wklęsłości będą zasłaniane (częściowo lub całkiem) zależy będzie od tzw. **płaszczyzny zasłaniania PZ**, rys. 2.11, należącej do znikniętej ściany s_i , mianowicie: wymieniona płaszczyzna PZ (która w momencie znikania ściany s_i zawiera ją), w czasie

ruchu PW_i będzie zakrywała stopniowo kolejne ściany wklęsłości – na obszarze zwanym **plachtą zasłaniania** Pl_z – czyli poczynając od ścian sąsiadujących ze ścianą znikniętą s_i , a kończąc na konturze przecinania się ściany PZ ze ścianami wklęsłości). Należy więc kontrolować położenie wierzchołków tych ścian względem płaszczyzny zasłaniania PZ . Wierzchołki, które znajdują się pod tą płaszczyzną, nie będą widoczne. Te, które pozostaną nad nią – będą widoczne. Za **zdarzenia wizualne spowodowane niewidoczną lub**



Rys. 2.11. Ilustracja do wyznaczania cieniowania wklęsłości, [57]

zasłonięciami uważa się zdarzenia polegające na całkowitym, bądź częściowym zasłonięciu ściany, w szczególności, każde kolejne zakrycie (pojedynczego) wierzchołka będzie takim zdarzeniem.

Aby określić **dotatkowe widoki spowodowane niewidocznymi i zasłonięciami**, prowadzi się przez bieżący punkt PW_i oraz krawędź zasłaniającą k_{wk_i} płaszczyznę PZ . Płachta zasłaniania Pl_z wycięta z tej płaszczyzny przez krawędzie przecinania się jej ze ścianami wklęsłości, spowoduje (częściowe lub całkowite) zasłanianie ścian wklęsłości (leżących po tej jej stronie, co i ściana zniknięta), a przez to powstawanie kolejnych zdarzeń wizualnych i dodatkowych widoków.

Autorzy nie podają wyników implementacji opisanego algorytmu. Może to wynikać między innymi z praktycznych trudności w poruszaniu się po granicy obszaru jednowidokowego w sytuacji gdy ten brzeg jest opisany powierzchnią zdarzeń kkk . Sądzić należy też, że ten algorytm byłby kosztowny obliczeniowo (przez to niezbyt szybki) gdyż jego elementem jest dokładne wyliczenie obszarów jednowidokowych (skoro poruszamy się po ich granicy). W innej grupie metod główny nacisk kładzie na otrzymanie widoku a nie na wyznaczenie jego obszaru widokowego.

2.4.3. Bezpośrednie generowanie widoków

Metoda *bezpośredniego generowania widoków* nie wymaga wyznaczania obszarów jednowidokowych. Jest więc bardziej wydajna obliczeniowo. Rezygnowanie z obliczania obszarów jednowidokowych wynika z możliwości zapewnienia kompletności reprezentacji za pomocą innych instrumentów niż kompletność pokrycia sfery widokowej obszarami jednowidokowymi. W przypadku wielościanów wypukłych stosunkowo łatwo można wyznaczyć zdarzenia wizualne, co wykorzystuje następująca metoda.

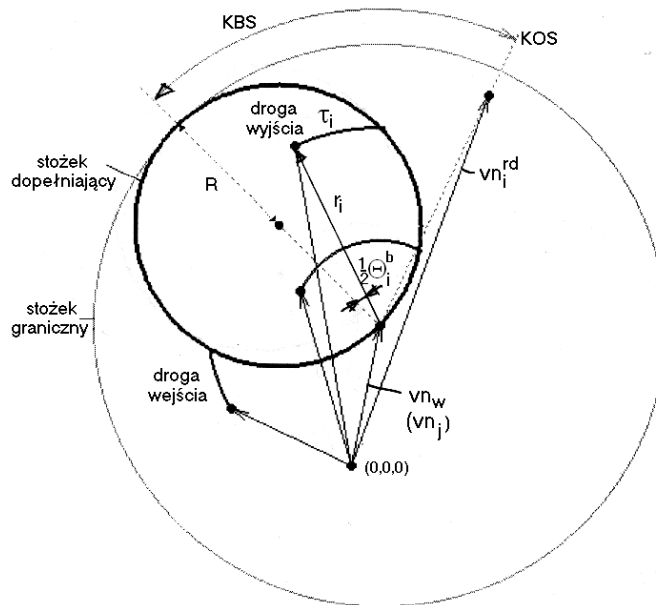
Obrót stożka dopełniającego wokół wektora normalnego ściany

Koncepcja niniejszego algorytmu oznaczanego $FM^{V_{rep}\cap}$, [36], jest oparta na wektorowej reprezentacji wielościanu V_{rep} (w której obiekt jest reprezentowany za pomocą wektorów normalnych ścian zaczepionych w środku układu współrzędnych) i powstała ze spostrzeżenia, że w wielościanie wypukłym nie zachodzi samoprzesłanianie ścian, więc o widoczności decyduje kąt pomiędzy wektorem normalnym a kierunkiem patrzenia. A zatem do wygenerowania reprezentacji widokowej wielościanu powinna wystarczyć jego reprezentacja wektorowa.

Dodając założenie, że warunkiem widoczności ściany jest obecność jej wektora normalnego w stożku dopełniającym można już określić główne kroki tej metody:

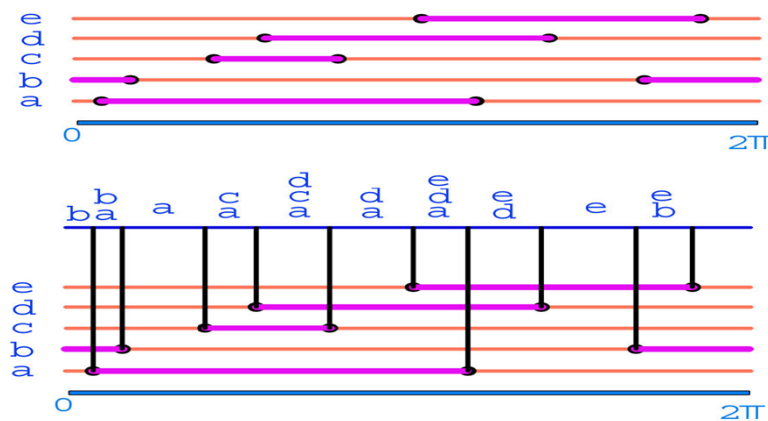
1. Wybór ściany do generowania widoków.
2. Transformacja układu współrzędnych tak aby wektor normalny ściany pokrywał się z wektorem $(1, 0, 0)$ (w celu uproszczenia obliczeń).
3. Pełny (o kąt 2π) obrót stożka dopełniającego (rys. 2.12), którego tworząca pokrywa się z wektorem normalnym ściany rejestrując zdarzenia wizualne (alternatywnie można rozwiązać równania dla jakich wartości parametru, inne wektory normalne leżą na powierzchni stożka dopełniającego) zapisując wartość parametru, dla których wektory innych ścian wchodzą i wychodzą ze stożka.
4. Określenie zestawu widoków z udziałem danej ściany, wyznaczając kolejne przedziały między kolejnymi zdarzeniami wizualnymi i zestaw ścian widocznych w danym przedziale (widok) (rys. 2.13).
5. Powtarzanie powyższych kroków, dla każdej ściany, by otrzymać zestaw widoków z udziałem każdej ściany.
6. Sumowanie widoków z udziałem każdej ściany i usuwanie powtórzeń.

Przyjęte tu założenie, że warunkiem widoczności ściany jest obecność jej wektora normalnego w stożku dopełniającym jest prawdziwe jedynie dla wielościanów regularnych. Niestety w ogólnym przypadku wielościanów wypukłych nie zawsze tak jest. Należy wtedy definiować oddzielnie dla każdej ściany indywidualne stożki dopełniające. Warto zatem przedstawiony sposób postępowania, [37], uzupełnić o poniższe elementy:



Rys. 2.12. Obrót wokół wektora normalnego. Duży okrąg stożka granicznego SG, mały okrąg stożka SD, łuk τ_i drogi końca wektora vn_i , kąt θ oraz promienie: r_i i R skaningu, [70]

- korzystać z indywidualnych stożków dopełniających lub alternatywnie znaleźć punkty przecięcia granic widoczności danej ściany na sferze widokowej⁴⁴ z obszarami widoczności innych ścian;
- rozszerzyć zestaw widoków w każdym z przedziałów 2.13) o widoki bez danej ściany (a nie tylko z udziałem badanej ściany)⁴⁵.



Rys. 2.13. Rejestracja zdarzeń wizualnych podczas omiotu stożkiem dopełniającym, [36]

Powyższa metoda po wspomnianych modyfikacjach generuje kompletny zestaw widoków jedynie dla wielościanów wypukłych. Można wtedy dodatkowo precyzyjnie wyznaczyć obszary jednowidokowe⁴⁶. Szybkość wyznaczania reprezentacji jest tu silnie uzależniona

⁴⁴ Przekięcie płaszczyzny zawierającej daną ścianę ze sferą widokową.

⁴⁵ Korzysta się ze spostrzeżenia, że PW porusza się po granicy obszaru widoczności ściany.

⁴⁶ Co jednak jest tu sprawą wtórną, główny nacisk jest położony na pozyskanie widoków.

od liczby ścian w bryle (bada się przecięcia każdej ściany z każdą a więc ilość przedziałów do zbadania w bryle o n ścianach jest o złożoności $O(n^2)$). Głównymi zaletami tej metody jest generowanie kompletnej reprezentacji widokowej w dość przejrzysty i szybki sposób. Stąd też koncepcja, żeby tę metodę uogólnić na szerszą klasę brył.

Jednym z pierwszych pomysłów (rozwijanym przez autora pracy) było poruszanie się wokół ściany w taki sposób by była ona cały czas widoczna. Zrodziło to koncepcję ograniczenia klasy brył do wielościanów monottonnych co pozwalało na to aby cała ściana ściana mieściła się w stożku dopełniającym. Trajektorja skanowania była wyznaczona przez ruch punktu widokowego (leży on na osi stożka widokowego i dopełniającego) powodowanego przez ruch SD wokół ściany.

Obrót stożka dopełniającego wokół ściany

Metoda generowania widoków, oparta na pełnej reprezentacji naturalnej wielościanu $N_{prep}(W)$, definiowanej jako:

$$N_{prep}(W) = \sum_{i=1}^n \left\{ \vec{n}_i; \sum_{j=1}^{l_i} \vec{k}b_j \right\}, \quad (2.2)$$

(gdzie n oznacza liczbę ścian wielościanu, $k b_j$ to krawędzie boczne ostrosłupów w reprezentacji naturalnej N_{rep} , których w i -tym wielościanie jest l_i), jest uogólnieniem metody poprzedniej, opisaney w [38].

Najistotniejsza różnica w stosunku do algorytmu opartego na reprezentacji wektorowej V_{rep} to odmienny, znacznie bardziej złożony ruch stożkiem skanującym. W ogólności, stożek skanujący wykonuje obroty skanowe nie dookoła prostej pokrywającej się z kierunkiem wektora normalnego ściany, ale wokół samej ściany, wspierając się na kolejnych krawędziach bocznych ostrosłupa ściany widokowej, dopóki nie dotknie z powrotem krawędzi początkowej. Zatem, stożek skanujący wykonuje obrót wokół każdej z tych krawędzi o określony kąt, zależny od kąta wewnętrznego ściany widokowej przy danej krawędzi bocznej i długości przylegających do tego kąta krawędzi ściany widokowej.

Otrzymywane wyniki generowania widoków były zachęcające, jednak zdarzało się, że dla pewnych brył nie wszystkie widoki były generowane. Stąd też w [72, 75] opisano ulepszenia metody polegające na uzupełnianiu zbioru widoków o widok znad środka ściany, włączając do algorytmu omiotu dodatkowe (po omiocie brzegu ściany s_w) ustawienie osi stożka skanującego na środkowej omiecionej ściany s_w , obliczanie widoku i - gdy jest on nowy, dodanie go do bazy widoków. Postępuje się tak dla wszystkich ścian bryły.

Generowanie reprezentacji widokowej ostatnią wersją tego algorytmu, [73], odbywa się następująco:

Niech będzie dana bryła wielościenna o n ścianach. Opnijmy na niej małą sferę o promieniu r , mającą środek w środku geometrycznym bryły. Niech stożek widokowy ma kąt półrozwarcia α . Wówczas stożek skanujący (który jest równy stożkowi dopełniającemu i ma środek w środku bryły) będzie miał kąt półrozwarcia $(\pi/2 - \alpha)$. Stożek ten wytnie

na małej sferze koło o promieniu r^* , przez które będzie widać ścianę widokową i jeszcze pewne inne ściany bryły, zależnie od wzajemnego położenia stożka i ściany widokowej. Nas interesuje, jakie kombinacje ścian bryły będą się pojawiały w stożku, gdy będzie się on obracał dookoła ostrosłupa pewnej ściany, która wybierzemy na ścianę widokową.

Wybermy zatem ścianę widokową s_w wraz z jej ostrosłupem, którego wierzchołek jest w środku bryły razem z wierzchołkiem stożka skanującego. Przedłużmy krawędzie boczne tego ostrosłupa do przecięcia z małą sferą. Obliczenia związane z określeniem kątów obrotu wykonywać będziemy opierając się właśnie na tych śladach krawędzi bocznych ostrosłupa.

Założmy, że ściana s_w ma l krawędzi k_1, \dots, k_l i tyleż samo krawędzi bocznych $kb_1 \div bk_l$ ma jej ostrosłup, rys. 2.14. Oznaczmy kąt wewnętrzny ściany przy wierzchołku i przez γ_i . Założmy też, że początkowe ustawienie stożka skanującego i ściany s_w było takie, że był on styczny do dwóch krawędzi ostrosłupa: l -tej i 1 -szej. Wówczas odległość osi widokowej od każdej z tych krawędzi na powierzchni małej sfery będzie r^* .

Jeśli teraz wykonamy obrót stożka dookoła krawędzi bocznej kb_1 tak, by osiągnął on styczność z krawędzią kb_2 , to kąt tego obrotu będzie wynosił:

$$\sigma_1 = \arccos \frac{k_{l1}}{2r^*} + \arccos \frac{k_{12}}{2r^*} + \gamma_1.$$

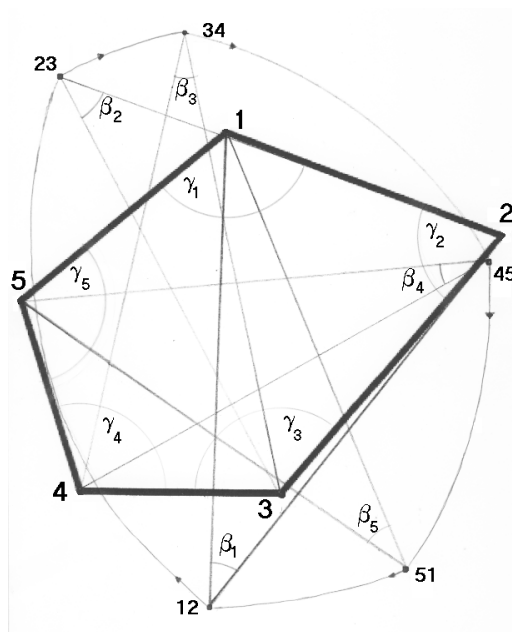
Następny obrót powinien dokonać się wokół krawędzi bocznej kb_2 o kąt

$$\sigma_2 = \arccos \frac{k_{12}}{2r^*} + \arccos \frac{k_{23}}{2r^*} + \gamma_2$$

itd. dopóty, dopóki znowu stożek skanujący nie dotknie krawędzi bocznej kb_1 . Oczywiście, w czasie każdego z tych ruchów będą nas interesowały zdarzenia wizualne, które mogą zaistnieć w wyniku tego ruchu.

Jednym z istotniejszych zagadnień są zasłonięcia jednych ścian bryły przez inne. Zasłonięcia dotyczą konkretnego widoku i są powodowane przez ściany leżące w obszarze widoku i mające wersor normalny na zewnątrz stożka skanującego. Dla każdego widoku można określić fakt występowania zasłonięcia oraz widoczność (pełną lub częściową) poszczególnych ścian widoku. W związku z powyższym formułuje się pewne reguły postępowania, mającego na celu znalezienie wszystkich możliwych widoków:

1. Zasadne jest rozpoczęcie badania widoczności ścian od rozpoznania występowania w obszarze stożka widokowego poszczególnych ścian, czyli od znalezienia konturu oraz wnętrza widoku.
2. Następny etap to badanie, czy w stożku widokowym leżą wersory normalne ścian z wnętrza konturu widoku.
3. Wynik obu tych czynności prowadzi do określenia, które ściany są widoczne, które niewidoczne (tj., znajdujące się w stożku skanującym, ale ich wersor normalny jest na zewnątrz stożka). Są jeszcze ściany zasłonięte lub częściowo zasłonięte: są to ściany leżące w stożku skanującym i mające wersor normalny w stożku skanującym, ale są zasłonięte przez ściany niewidoczne wklęsłości z wersorem normalnym leżącym poza



Rys. 2.14. Skaning otoczenia ściany wielościanu stożkiem SD. Punkty 1÷5 to punkty na małej sferze (opiętej na wielościanie) z liniami bocznych krawędzi ostrosłupa skanowanej ściany. Punkty 12÷51 to punkty na małej sferze znaczone przez oś stożka omiatającego w chwili styczności z sąsiednimi krawędzi ostrosłupa. Kąty γ_i to kąty wewnętrzne omiatanej ściany. Kąty β_i to kąty między promieniami od osi stożka do dwóch kolejnych wierzchołków ściany widokowej (dokładniej - jej śladu na małej sferze), [96]

stożkiem skanującym, graniczące z wypukłą częścią widoku leżącą na drodze promieni widokowych bliżej obserwatora. Definiuje się funkcję cienia F_c zmieniającą się z chwilą zaistnienia zdarzenia wizualnego związanego z zasłanianiem. Występuje ono wtedy, gdy **płaszczyzna cienia** PC zadana przez krawędź konturu wklęsłości ściany niewidocznej zasłoni ściany wklęsłości.

4. To, że zasłonięcie występuje objawia się tym, że wektor normalny pewnej ściany z wnętrza konturu widoku (a więc z wnętrza stożka skanującego) leży poza tym stożkiem. Skaning położenia ścian i wektorów w stożku przebiega wg następującego porządku:
 1. Utwórzmy dla każdej ściany jej **pełną reprezentację wektorową** V_{prep} . Reprezentację V_{prep} stanowi sam wektor normalny danej ściany (czyli element v_i reprezentacji V_{rep}) razem z wektorami krawędzi bocznych ostrosłupa reprezentacji naturalnej N_{rep} , mającego za podstawę rozważaną ścianę.
 2. Skaning zachodzi wokół każdej ze ścian i polega na sprawdzaniu, które składowe reprezentacji wektorowej ściany i dla których ścian znajdują się w stożku skanującym w jego w bieżącym położeniu; ściany, dla których:
 - a) wszystkie składowe znajdują się w stożku skanującym tworzą potencjalnie widoczne ściany widoku (niektóre z nich mogą być zasłonięte przez inne ściany z widoku);
 - b) ściany, dla których wektory krawędziowe są w stożku skanującym, ale wektor normalny jest poza nim, nie są widoczne w widoku, chociaż leżą wewnątrz konturu widoku (więc są składowymi tego widoku);

c) ściany, których wektory krawędziowe są poza stożkiem skanującym nie są widoczne niezależnie od położenia wersora normalnego; są to ściany z przeciwnej strony bryły;

3. W trakcie skaningu poszczególne wektory krawędziowe ścian będą obejmowane przez stożek skanujący, inne będą opuszczane przez niego. Niektóre z tych zdarzeń będą **zdarzeniami wizualnymi**, tj. takimi, które powodują zmianę w widoku. Mogą być różne rodzaje zdarzeń wizualnych i każde z nich będzie powodowało odmienne skutki. W szczególności:

- objęcie stożkiem skanującym ostatniego wektora (z poza stożka) danej ściany spowoduje **potencjalne powiększenie widoku** o tę ścianę (choć może ona pozostać niewidoczna lub stać się zasłoniętą); jeśli jej wersor normalny jest objęty stożkiem, ściana ta staje się potencjalnie widoczna (przestaje być niewidoczna); wówczas należy sprawdzić, czy nie jest w strefie cienia (nie jest zasłonięta przez inną ścianę) i zarejestrować nowy (powiększony) widok;
- objęcie stożkiem skanującym wersora normalnego ściany objętej spowoduje, że ta **ściana staje się potencjalnie widoczna** (jednak może być zasłonięta częściowo lub całkowicie); wówczas - podobnie jak poprzednio, należy sprawdzić, czy nie jest w strefie cienia (nie jest zasłonięta przez inną ścianę) i zarejestrować nowy (powiększony) widok;
- opuszczenie przez stożek skanujący wersora normalnego (całkowicie) objętej ściany spowoduje, że **ściana ta stanie się niewidoczna i może powodować zasłanianie innych ścian widoku leżących we wklęsłości**; wówczas należy rozpoznać, które ściany staną się niewidoczne, określając kontur wklęsłości i zarejestrować nowy widok; zacienienie zacznie się od ścian, które z nią sąsiadują, ale są dalej od osi widoku niż ona;
- jeśli ściana pojawi się wewnątrz konturu widoku, ale jej wersor normalny nie wejdzie równocześnie do stożka skanującego, to też będzie powodować zacienienie;
- opuszczenie przez stożek skanujący jakiegokolwiek wektora krawędziowego ściany objętej spowoduje, że **ściana ta opuści widok**; należy zarejestrować nowy widok;

Gdy w trakcie skaningu zaistnieje jakikolwiek powód rejestracji nowego widoku należy sprawdzić, czy jest to widok z zasłonięciami, a jak takowe występują, to wraz z widokiem należy zarejestrować granicę cienia i wierzchołek, przez który ona przechodzi, zasłoniętą i widoczną część ściany.

Określenie obszaru zacienienia oraz znalezienie ścian zasłoniętych (czy częściowo zasłoniętych) wydaje się być zagadnieniem dość złożonym. W terminach stożka skanującego zasłanianie można określić w sposób następujący: gdy stożek skanujący obejmie (wszystkie) wektory krawędziowe pewnej ściany oprócz jej wersora normalnego, wówczas ściana ta zaczyna zasłanianie ścian (dotychczas potencjalnie widocznych) na obszarze nakrytym przez **płaszczyznę cienia** PC , zadaną przez punkt widokowy PW oraz krawędź konturu wklęsłości ściany cieniującej.

W sytuacji, gdy wersor normalny opuści stożek skanujący, ściana (potencjalnie) widoczna staje się niewidoczna, pojawia się wklęsłość, a kontur wklęsłości powoduje rozleglejsze zasłonięcia, aż do opuszczenia ściany niewidocznej. Gdy zarówno ściana, jak i jej wersor normalny są objęte i następuje fakt opuszczenia ściany, przestaje ona być (potencjalnie) widoczną, bowiem przechodzi poza kontur widoku.

Kolejne zagadnienie, to określenie, jakie dane powinny być sprzężone z widokiem, by był on pełną (na potrzeby identyfikacji) reprezentacją bryły. Wydają się do nich należeć:

- kontur widoku oraz ściany zawarte wewnątrz niego: ich wzajemne sąsiedztwa (aspekt) oraz pełne dane numeryczne nt. współrzędnych wierzchołków, krawędzi i ścian;
- ściany niewidoczne, leżące wewnątrz konturu widoku;
- ściany zasłonięte i częściowo zasłonięte wewnątrz konturu widoku (dokładniej: zarówno część zasłonięta ściany jak i niezasłonięta, granica cienia oraz wierzchołek, przez który ta granica przechodzi);
- kontur wklęsłości i ściany leżące wewnątrz niego.

Wymienione elementy stanowią o treści widoku. Jakakolwiek różnica w tych danych włącznie z wielkością częściowego zasłaniania ściany, powoduje odmienność widoku.

Oto szkic algorytmu generowania widoków uwzględniający te elementy:

ALGORYTM $MS^{N_{rep} \cup}$

1. Utwórzmy reprezentację $V_{p_{rep}}$ dla bryły,
2. Przeprowadźmy skanowanie przestrzeni widokowej bryły dla każdego wersora normalnego v_i z V_{rep} ścian wielościanu kontrolując zdarzenia wizualne wywołane przez v_j oraz n_k .
3. Wybierzmy ścianę n_1 i dokonujmy skanowania wokół tej ściany. Rozważmy początkową sytuację, gdy w widoku jest tylko jedna wklęsłość i jest cała widoczna. Zarejestrowany widok zawiera jedynie ściany widoczne oraz adnotację o tym.
4. Puśćmy teraz w ruch stożek skanujący. Zaobserwować możemy następujące zdarzenia:
 - a) **Jeśli pewna ze ścian wypukłej części bryły pojawi się bądź zniknie** (czyli znajdzie się w tym stożku bądź wyjdzie z niego równocześnie ze swoim wersorem normalnym), traktujemy to jako **zwykłe zdarzenie wizualne** powodujące powstanie nowego widoku, który rejestrujemy z odpowiednią adnotacją w jego metryce.
 - b) **Jeśli wyjdzie ze stożka skanującego tylko ściana, ale pozostanie w nim jej wersor normalny** oznacza to, że widok zmniejszył się o tę ścianę (też zwykłe zdarzenie wizualne ale z odpowiednią adnotacją o wektorze).
 - c) **Natomiast, jeśli stożek skanujący opuści tylko wersor normalny pewnej ściany widoku, a sama ściana w tym stożku pozostanie** (czyli stanie się niewidoczna) oznacza to, że w **stożku skanującym znalazła się wklęsłość**, a zniknięta ściana jest ścianą konturową tej wklęsłości (i kolejne ściany tej wklęsłości będą zasłaniane, bądź też będą stawały się niewidoczne), włączamy algorytm obliczania funkcji cieniującej

dla krawędzi tej ściany. Algorytm śledzi zmiany w widoczności poszczególnych ścian wklęsłości i dokonuje stosownych czynności w następującej kolejności:

- i. zarejestruj nowy widok (bez widoczności danej ściany - chociaż wewnątrz konturu widoku);
 - ii. znajdź elementy tej wklęsłości: ściany i kontur; można to wykonać następująco:
 - A. znajdź ściany widoku sąsiadujące ze ścianą znikniętą (która na pewno jest ścianą wklęsłości) i sprawdzaj ich wzajemne wklęsłości:
 - jeśli pewna sąsiednia ściana jest we wklęsłości ze ścianą znikniętą, to i ona jest we wklęsłej części widoku,
 - jeśli nie jest ze ścianą znikniętą we wklęsłości - nie leży we wklęsłości⁴⁷;
 - B. znalezione ściany są ścianami z wklęsłej części bryły; z każdą z tych ścian postępuj tak, jak ze ścianą znikniętą: sprawdzaj, czy z sąsiadującymi z nią ścianami tworzy wklęsłość; postępuj tak ze wszystkimi ścianami z widoku;
 - C. znajdź **kontur wklęsłości** jako granicę między ścianami tworzącymi wklęsłość i resztą ścian z wnętrza konturu widoku;
 - iii. znajdź **płaszczyznę cienia** PC dla krawędzi konturu wklęsłości będącej krawędzią znikniętej ściany;
 - iv. kontynuując ruch stożkiem skanującym: obserwuj przykrywanie kolejnych wierzchołków ścian wklęsłości przez płaszczyznę PC , przy czym:
 - gdy zostanie zasłonięty jedynie fragment pewnej ściany (przez płaszczyznę PC), wyznacz krawędź cienia (widoczności pozostałej części ściany przechodzącą przez właśnie przykryty wierzchołek) oraz rejestruj to jako zdarzenie wizualne (tzn. zarejestruj widok z mniejszym fragmentem zasłanianej ściany oraz jego metrykę),
 - gdy zostanie zasłonięta cała ściana, to również traktuj to jako zdarzenie wizualne: zarejestruj widok bez tej ściany oraz wypełnij odpowiednio metrykę tego widoku;
 - gdy zniknie z widoku kolejna ściana wklęsłości, wówczas przejdź poziom wyżej, czyli znajdź jej płaszczyznę cieniującą i postępuj tak, jak w przypadku pierwszego zniknięcia;
 - v. postępuj tak dalej, aż wszystkie ściany tej wklęsłości staną się niewidoczne;
- d) Jeśli w stożku skanującym pojawi się (izolowana) niewidoczna ściana (jej krawędź), oznacza to, że jest to początek pojawiania się w widoku wklęsłości, a pojawiona ściana jest jedyną niewidoczną ścianą tej wklęsłości. Pozostałe ściany tej wklęsłości weszły do stożka równocześnie z nią, ale ich wersory normalne weszły wcześniej. Tę ściany są zasłonięte. Wówczas identyfikujemy wklęsłość (tzn. znajdujemy jej ściany i kontur), rejestrujemy widok z tą ścianą (jako niewidoczną), a pozostałymi

⁴⁷ Jest tak dlatego, że przyjęto założenie, iż wklęsłość rozważana oddzielnie od reszty bryły jest wypukła.

zasłoniętymi i włączamy funkcję cieniowania: wyznaczamy płaszczyznę cienia PC dla tej niewidocznej ściany (przechodzącą przez punkt widokowy i jej krawędź konturu wklęsłości. Kontynuując ruch stożkiem skanującym (punktem widokowym) przystajemy, gdy płaszczyzna PC przechodzi przez kolejny wierzchołek ściany wklęsłości traktując to jako zdarzenie wizualne, następnie wyznaczamy granicę cienia (przechodzącą przez wspomniany wierzchołek) i rejestrujemy stosowny widok z właściwym wpisem do jego metryki, po czym kontynuujemy ruch stożkiem. Wyłączenie funkcji cienia nastąpi, gdy niewidoczna płaszczyzna stanie się widoczną (przez co cała wklęsłość stanie się widoczna).

- e) Jeśli w stożku skanującym pojawi się tylko wersor normalny (bez samej ściany), to jest to zapowiedź rychłego pojawienia się pewnej wklęsłości, w której ściana tego wersora występuje. Nie interesujemy się nim (póki w widoku nie pojawi się ściana).
5. Postępuj tak, aż pełny obieg danego skanu zostanie wykonany;
 6. Postępuj tak, aż zostaną wykonane pełne skany dla każdej ściany bryły.

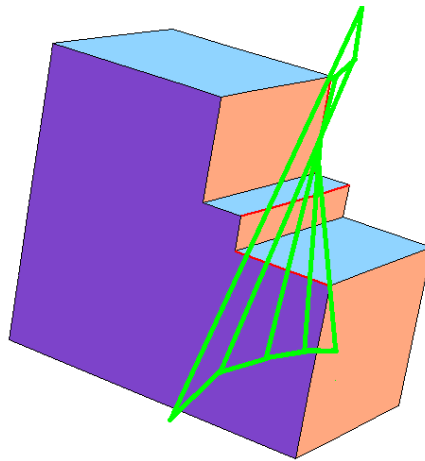
Tak skonstruowany algorytm dawał więcej widoków (w wielu przypadkach komplet bryły). Zdarzało się jednak, że dla pewnych brył nie wszystkie widoki były generowane, stąd uzupełnianie zbioru widoków o widok znad środka ściany. Wynika to oczywiście z faktu, że trajektoria skanowania jedynie przybliża obszar widoczności ściany. Naturalnym krokiem było zatem wyznaczanie widoków w obszarze potencjalnej widoczności ściany z określoną rozdzielczością. Jako sposób poruszania się w tym obszarze wybrano ruch spiralny.

Sposoby generowania reprezentacji widokowej możemy podzielić na **dokładne** (generujące kompletny zestaw widoków) i **przybliżone**, w których widoki są wyznaczone z dokładnością do określonej rozdzielczości. Wybrane przybliżone metody generowania omówione zostaną w następnym rozdziale.

3. Przybliżone metody generowania reprezentacji widokowej

W tym rozdziale zaprezentowane będą algorytmy w których nie ma potrzeby wyznaczenia obszarów jednowidokowych. Podejście takie wynika z faktu, że niezbędny jest widok bryły, a nie obszar, w którym taki widok można uzyskać.

Jak już wcześniej wspomniano, wyznaczenie obszaru widoczności ściany w niewypukłej bryle prowadzi do zbadania zdarzeń typu krawędź-wierzchołek i krawędź-krawędź-krawędź (liczba takich zdarzeń jest rzędu n^3 a liczba przecięć z SWP n^6), co w połączeniu z trudnością wyznaczenia przecięć powierzchni w zdarzeniach kkk (które mogą się pojawić nawet w tak prostej klasie brył jak wielościany monotonne, patrz rys. 3.1 widzimy przykładowe zdarzenia tego typu dla wielościanu monotonnego) z SWP skłania do rezygnacji z koncepcji dokładnego wyznaczenia obszaru widoczności ściany i próbkowania (wyznaczania widoków ze sfery widokowej) na określonej trajektorii skanowania z przyjętą wcześniej rozdzielczością.



Rys. 3.1. Przykład zdarzeń kkk w wielościanie monotonnym

Pozostaje natomiast otwarty problem, jakim torem powinien się poruszać punkt widokowy na sferze, oraz jak najkorzystniej rozmieścić punkty widokowe na tym torze, by wygenerować kompletną¹ reprezentację.

Pierwszą z metod z grupy spiralnego ruchu punktu widokowego po sferze widokowej był sposób opisany w pracy [74], której główne koncepcje zostaną teraz przedstawione.

¹ Z dokładnością do przyjętej rozdzielczości.

3.1. Skanowanie obszaru widoczności ściany torem spiralnym

Główna idea tej metody to - tak, jak w bezpośrednich metodach generacyjnych, wyznaczenie wszystkich widoków z udziałem danej ściany. Przeprowadzenie tej operacji dla wszystkich ścian daje komplet widoków. Komplet widoków z udziałem danej ściany uzyskuje się skanując obszar potencjalnej widoczności ściany. Jako trajektorię skanowania przyjęto ruch spiralny po sferze nad daną ścianą z wymogiem, by punkty widokowe PW na niej były rozłożone możliwie najbardziej równomiernie. Uzyskuje się to przez żądanie, by kąt między kolejnymi zwojami spirali omiotu był stały.

Równanie spirali na sferze, spełniające to założenie, zapisuje się następująco:

$$\begin{cases} x = R \cos \theta \sin \phi \\ y = R \sin \theta \sin \phi \\ z = R \cos \phi, \end{cases} \quad (3.1)$$

a pomiędzy kątami θ i ϕ zachodzi następująca zależność:

$$\theta = k \cdot \phi. \quad (3.2)$$

Kąt $\phi \in [0, \pi]$, a liczba pełnych obrotów wokół osi z^2 jest wyznaczona przez wartość $2k$. W zapisanym równaniu kąt pomiędzy kolejnymi zwojami spirali jest równy $\frac{2\pi}{k}$. Jednakże otrzymywane punkty PW na trajektorii nie są rozłożone równomiernie (zdecydowanie gęściej na „biegunach”, patrz rys. 3.9).

By tego uniknąć stosuje się nieliniowe przekształcenie parametru t w następujący sposób:

$$\begin{cases} \theta = k\sqrt{t} \\ \phi = \sqrt{t}. \end{cases} \quad (3.3)$$

Jeżeli punkt widokowy dotrze do granic potencjalnej widoczności ściany lub równoważnie, tworząca indywidualnego stożka dopełniającego (o kącie rozwarcia 2β) ściany pokryje się z wektorem normalnym, wówczas wykonujemy pełny obrót³ stożkiem dopełniającym wokół normalnej ściany. Biorąc pod uwagę równania 3.1 i 3.3 ruch punkty widokowego torem spiralnym wraz z kończącym skanowanie pełnym obrotem wokół ściany możemy przedstawić następującym zbiorem równań:

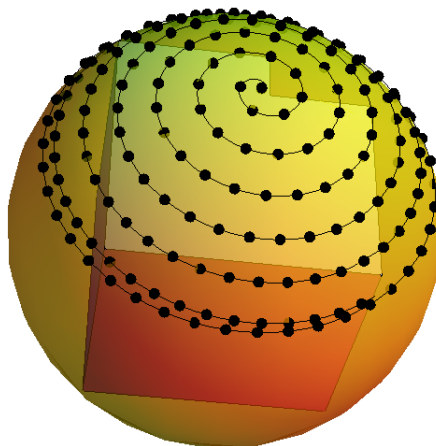
$$\begin{cases} x = \cos k\sqrt{t} \sin \sqrt{t} \\ y = \sin k\sqrt{t} \sin \sqrt{t} \text{ dla } t \in [0, \beta^2] \\ z = \cos \sqrt{t}. \end{cases} \quad (3.4)$$

$$\begin{cases} x = \cos t \sin \beta \\ y = \sin t \sin \beta \text{ dla } t \in (\beta^2, \beta^2 + 2\pi] \\ z = \cos \beta. \end{cases} \quad (3.5)$$

² Równanie jest zapisane dla prostoty jako ruch spiralny wokół osi z .

³ O kąt 2π .

Obszar potencjalnej widoczności ściany (OPWŚ) na sferze widokowej ma postać „czaszy”. Wierzchołkiem tego obszaru jest punkt przecięcia prostej kierunkowej wektora normalnego ściany ze sferą widokową, podstawą zaś jest okrąg⁴ (rys. 3.2).



Rys. 3.2. Przykładowa trajektoria skanowania nad obszarem widoczności ściany opisana równaniami 3.4 i 3.5

OPWŚ jest skanowany w następujących krokach:

1. wyznaczamy punkt startowy, będący przecięciem kierunku wektora normalnego ze sferą widokową;
2. wykonujemy skanowanie torem spiralnym opisane równaniem 3.6, z określonym krokiem skanowania (wyznaczamy kolejne punkty widokowe);
3. dla każdego punktu widokowego ustalamy widok z tego punktu (jeżeli nie ma go w bazie widoków to go dodajemy);
4. gdy docieramy do brzegu obszaru potencjalnej widoczności⁵ ściany, przerywamy ruch spiralny i dokonujemy skanowania, wykonując pełny obrót stożka dopełniającego wokół tworzącej, pokrywającej się z kierunkiem normalnej⁶.

Podczas tego ruchu dla każdego nowo wygenerowanego punktu widokowego ustalamy widok z tego punktu (jeżeli nie ma go w bazie widoków to go dodajemy do zbioru widoków), w ten sposób otrzymujemy komplet widoków z udziałem danej ściany. Powtarzając opisane skanowanie dla każdej ściany uzyskujemy komplet widoków dla wielościanu.

Przeprowadzone testy dla tego algorytmu wykazały generowanie kompletu widoków⁷. Jednak to co było jednym z pomysłów tworzących tę metodę czyli wygenerowania wszystkich widoków z udziałem danej ściany, prowadzi do **głównej wady tej metody**, którą jest **wielokrotne skanowanie tego samego obszaru**. Bowiem, każdy obszar jednovidokowy

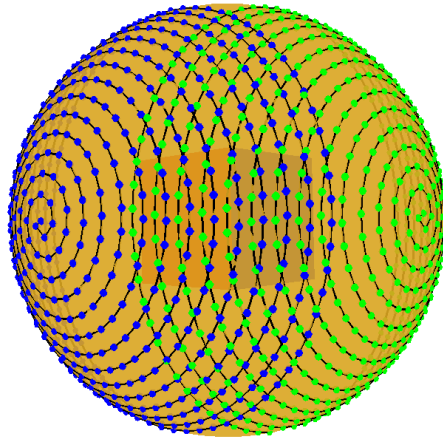
⁴ Przecięcie sfery widokowej z płaszczyzną zawierającą ścianę.

⁵ Tworząca stożka dopełniającego pokrywa się z kierunkiem wektora normalnego ściany.

⁶ Prosta \overrightarrow{OPV} przecina okrąg będący podstawą OPWŚ.

⁷ Z dokładnością do żądanej rozdzielczości.

jest skanowany tyle razy, ile jest potencjalnie widocznych ścian w tym obszarze (patrz rysunek 3.3). Przedstawiona w następnym podrozdziale metoda jest wolna od tej wady.



Rys. 3.3. Wielokrotne skanowanie tego samego obszaru na sferze widokowej

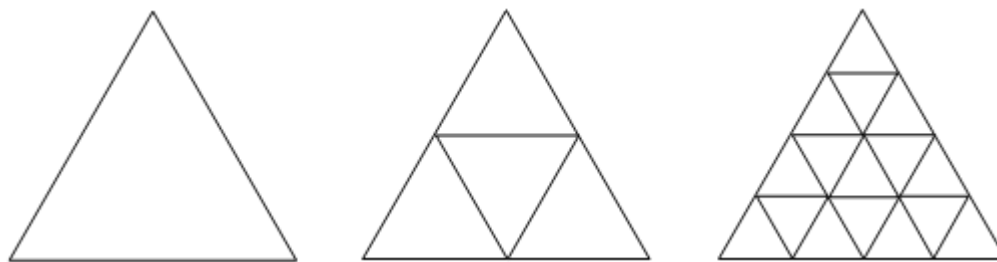
3.2. Metody z wykorzystaniem wielościanu foremnego

Głównym pomysłem w tych sposobach generowania widoków jest jak najbardziej równomierne rozmieszczenie punktów widokowych na całej sferze widokowej. Problem równomiernego rozmieszczenia punktów na sferze jest szeroko opisywany w literaturze i często wykorzystuje się do tego wielościany foremne rozmieszczając punkty PW w ich wierzchołkach. Chen, [14], do generowania *LFD* wykorzystuje dwunastościan foremny, Lindstrom i Turk, [65], również pozyskują widoki z wierzchołków tego wielościanu. Dwudziestościan foremny również znajduje szerokie zastosowanie w równomiernym rozkładzie punktów na sferze ponieważ ma najwięcej ścian (które można dalej dzielić) a w jego ściany mają kształt trójkąta. Przykładowe zastosowania obejmują matematykę, grafikę komputerową i *GIS* (ang. *Geographic Information System*).

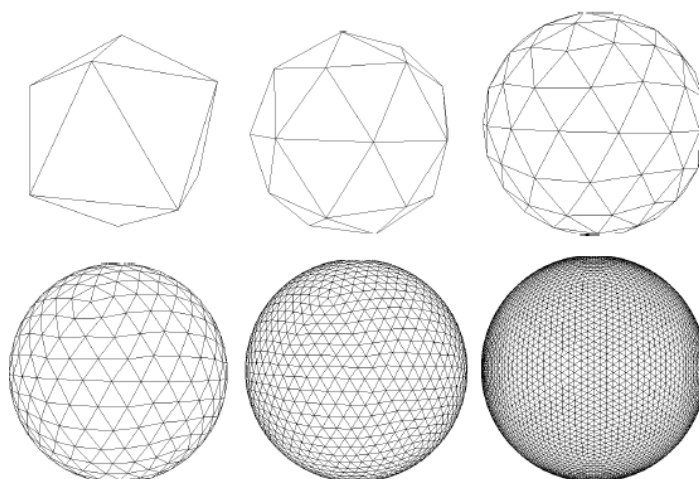
3.2.1. Metoda oparta na podziale dwudziestościanu foremnego

W pierwszym kroku tej metody rozmieszczamy punkty widokowe w wierzchołkach dwudziestościanu foremnego DF wpisanego w sferę widokową. Następnie dzielimy każdą ze ścian na cztery mniejsze trójkąty (patrz rys. 3.4), poprzez generowanie nowych wierzchołków leżących na środkach krawędzi (skalujemy ten wierzchołek tak, aby jego odległość od środka układu współrzędnych wynosiła R). Proces ten powtarzamy, aż liczba tak otrzymanych punktów widokowych zapewni żadaną minimalną odległość między punktami. Spotyka się też modyfikację tego sposobu poprzez podział krawędzi DF na n części zapewniającego odpowiednio gęste rozmieszczenie punktów widokowych.

Na rysunku 3.5 widzimy przybliżanie kształtu sfery przez kolejne iteracje podziału trójkątnej ściany.



Rys. 3.4. Kolejne iteracje podziału ściany dwudziestościanu foremnego



Rys. 3.5. Tworzenie nowych punktów widokowych przez iteracyjny podział DF

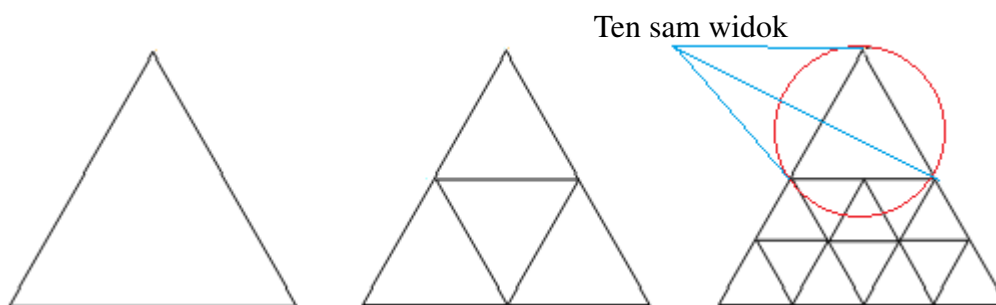
Metoda generowania widoków z wykorzystaniem dwudziestościanu foremnego obejmuje następujące etapy:

1. **Utworzenie zbioru punktów widokowych.**
2. **Pozyskanie widoku z każdego punktu widokowego i dodanie go do zbioru widoków.**
3. **Usunięcie powtarzających się widoków ze zbioru widoków.**

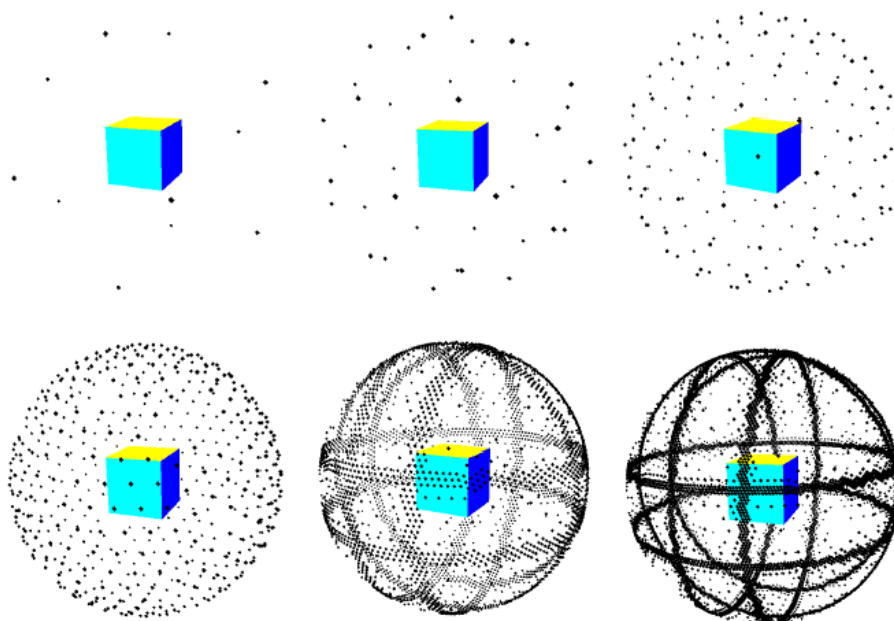
Główną zaletą tej metody jest prostota algorytmu, oraz stosunkowo równomierny rozkład punktów na sferze.

3.2.2. Metoda korzystająca z wypukłości obszaru jednowidokowego

W tej metodzie sposób generowania punktów jest taki sam jak w poprzednio opisanym sposobie. Jednak korzystamy tu ze spostrzeżenia, że obszar jednowidokowy jest obszarem wypukłym. To stwierdzenie jest prawdziwe jeżeli zignorujemy zdarzenia typu kkk jako powodujące mało istotne zdarzenia. Gdy z trzech wierzchołków trójkąta uzyskujemy ten sam widok, nie ma potrzeby dalszego podziału trójkąta, gdyż ponownie uzyskalibyśmy ten sam widok (rys. 3.6). Na rysunku 3.7 widać, jak w kolejnych iteracjach punkty widokowe są generowane tylko w granicach obszarów widokowych.



Rys. 3.6. Podział ścian DW z uwzględnieniem obszarów wypukłych



Rys. 3.7. Zbiór PW w kolejnych iteracjach DW z uwzględnieniem obszarów wypukłych

3.3. Rozmieszczenie punktów na spirali z określoną rozdzielczością

Jest to jedna z najnowszych metod, [91], w której spiralna trajektorja oplata całą sferę widokową⁸. Równocześnie żądamy, by punkty widokowe były rozmieszczone na spirali jak najbardziej równomiernie.

3.3.1. Opis metody

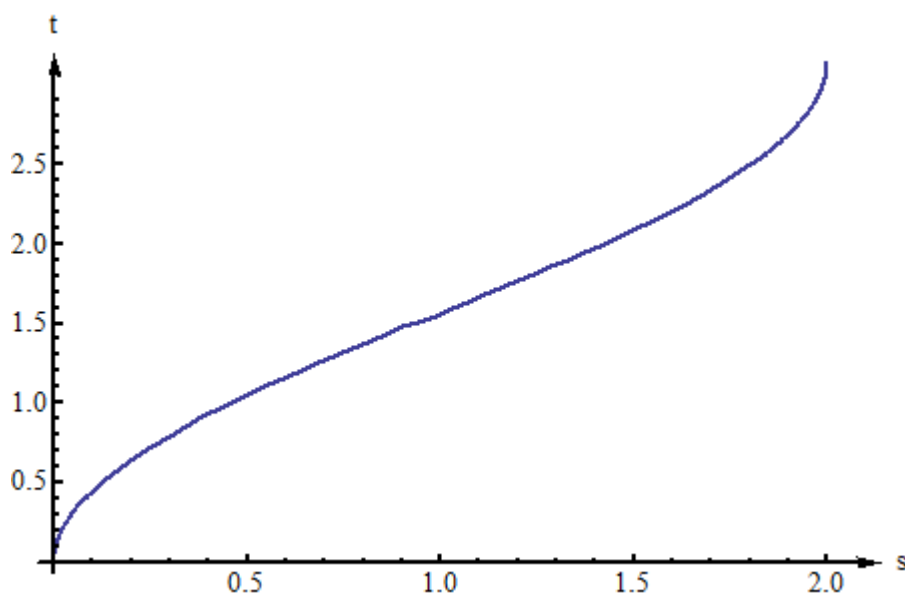
Wykorzystajmy wzór 3.1 na spiralną trajektorję na SWP:

$$\begin{cases} x = R \cos(kt) \sin(t) \\ y = R \sin(kt) \sin(t) \\ z = R \cos(t), \end{cases} \quad (3.6)$$

⁸ A nie tylko obszar potencjalnej widoczności.

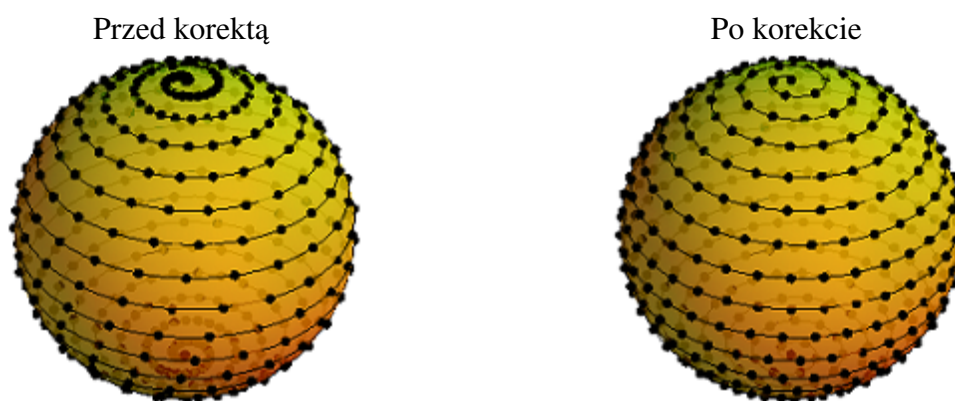
i użyjmy innego wzoru na korektę nierównomierności rozkładu punktów PW na trajektorii omiotu, który spodziewamy się, że będzie korzystniejszy, $f(s) : [0, 2] \rightarrow [0, 2\pi]$ (rys. 3.8):

$$\begin{cases} t = \arccos(1 - s); & \text{dla } s \in [0, 1] \\ t = \pi - \arccos(s - 1); & \text{dla } s \in (1, 2]. \end{cases} \quad (3.7)$$



Rys. 3.8. Funkcja korekty rozkładu punktów na trajektorii

Podstawiając za t we wzorze 3.6 parametr zdefiniowany wzorem 3.7 uzyskujemy zdecydowanie bardziej równomierny rozkład (patrz rys. 3.9).



Rys. 3.9. Spiralny rozkład punktów na sferze widokowej otrzymany za pomocą równania 3.6 i po zastosowaniu funkcji korekty rozkładu 3.7

Żądając, by kąt między kolejnymi zwojami spirali był równy żądanej rozdzielczości skanowania (oznaczonej przez σ), uzyskujemy wzór na k :

$$k = \frac{2\pi}{\sigma}.$$

Biorąc pod uwagę ten wzór oraz uwzględniając fakt, że długość spirali zdefiniowanej wzorem 3.6⁹ jest mniejsza niż $2k + 1$ ¹⁰, wzór na liczbę potrzebnych punktów n na spiralnej trajektorii, do uzyskania żądanej rozdzielności σ , można zapisać w postaci $n = \frac{2\pi + \sigma}{(\sigma)^2}$. Zatem największa odległość między kolejnymi punktami wynosi σ oraz kąt pomiędzy kolejnymi zwojami spirali również jest równy σ . Najbardziej niekorzystne położenie punktu widokowego to w środku kwadratu o boku σ , zatem aby zapewnić minimalną rozdzielczość σ możemy ją pomnożyć przez $\sqrt{2}$ ¹¹. Uwzględniając te oszacowania uzyskujemy następujący **wzór na liczbę punktów widokowych** N w metodzie spiralnego skanowania sfery widokowej (szersze wyprowadzenie tej zależności w rozdziale 3.3.2):

$$N = \frac{4\pi + \sqrt{2}\sigma}{2(\sigma)^2}. \quad (3.8)$$

Widzimy, że liczba punktów widokowych jest odwrotnie proporcjonalna do kwadratu rozdzielczości skanowania. Przykładowy rozkład punktów na trajektorii uzyskanej tą metodą jest przedstawiony na rys. 3.9. Rozdzielczość w tym przypadku nie wynika z cech geometrycznych bryły, a z technicznego parametru urządzenia pozyskującego.

Sposób generowania widoków w tej metodzie można przedstawić w trzech krokach:

1. **Wyznaczenie punktów widokowych** rozmieszczonych na spiralnej trajektorii na SWP dla zadanej rozdzielczości skanowania.
2. **Wygenerowanie widoku z każdego punktu widokowego i dodanie go do zbioru widoków.**
3. **Usunięcie powtarzających się widoków** ze zbioru wygenerowanych widoków.

Postępując w opisany powyżej sposób uzyskuje się komplet widoków z dokładnością do żądanej rozdzielczości.

3.3.2. Analiza kompletności generowanej reprezentacji

Wyniki przeprowadzonego w poprzednich paragrafach modelowania kształtu spirali skanującej sferę widokową i rozmieszczenia na niej punktów widokowych, zapisujemy w postaci dwóch następujących lematów:

Lemat 3.3.1. *Kąt pomiędzy kolejnymi zwojami spirali jest stały i równy $\frac{2\pi}{k}$.*

Dowód. Niech p_1 będzie dowolnym punktem na trajektorii skanowania

$$p_1 = (\cos(kt_1) \sin(t_1), \sin(kt_1) \sin(t_1), \cos(t_1)), \quad (3.9)$$

⁹ Długość tej spirali dla $R = 1$ jest całką eliptyczną drugiego rodzaju i wynosi $2E(-k^2)$

¹⁰ Przyjęto oszacowanie liniowe prawdziwe dla $k > 3$, w literaturze (np. [79]) można spotkać bardziej korzystne oszacowania, jednak zawierają one funkcje nieliniowe, zwiększające czas generowania punktów i nie zmniejszają istotnie liczby punktów widokowych.

¹¹ W pracy [91] przedstawiono mniej dokładny wzór biorąc pod uwagę tylko odległości pomiędzy punktami widokowymi.

a p_2 będzie położony na kolejnym zwoju spirali (a zatem kt powinno wzrosnąć o wartość 2π ¹², stąd $t_2 = t_1 + \frac{2\pi}{k}$). Zatem współrzędne punktu p_2 możemy zapisać następująco:

$$p_2 = \left(\cos\left(k\left(t_1 + \frac{2\pi}{k}\right)\right) \sin\left(t_1 + \frac{2\pi}{k}\right), \sin\left(k\left(t_1 + \frac{2\pi}{k}\right)\right) \sin\left(t_1 + \frac{2\pi}{k}\right), \cos\left(t_1 + \frac{2\pi}{k}\right) \right) \quad (3.10)$$

Iloczyn skalarny unormowanych wektorów $\vec{p}_1 = p_1 - (0, 0, 0)$ i $\vec{p}_2 = p_2 - (0, 0, 0)$ jest równy:

$$\vec{p}_1 \cdot \vec{p}_2 = \cos 2\pi/k \quad (3.11)$$

A zatem wykazaliśmy, że kąt pomiędzy dwoma punktami położonymi najbliżej siebie na kolejnych zwojach spirali jest stały i wynosi $\frac{2\pi}{k}$. \square

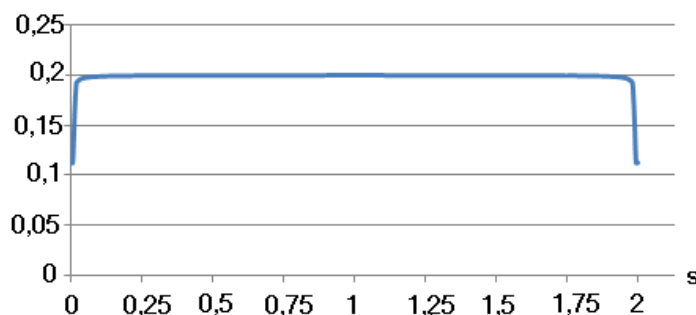
Lemat 3.3.2. *Kąt pomiędzy kolejnymi punktami widokowymi rozmieszczonymi na spiralnej trajektorii nie przekracza wartości σ .*

Dowód. Dla ustalenia uwagi przyjmijmy $R = 1$ (mówimy wówczas wymiennie o odległości geodezyjnej między punktami lub kącie pomiędzy nimi). Wtedy długość spiralnej krzywej (równanie 3.6) wynosi $2E(-k^2)$ i jest ona mniejsza niż $2k + 1$ (dla $k > 3$),

$$L = \int_0^\pi \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2} = 2E(-k^2) < 2k + 1. \quad (3.12)$$

Niech $N = \left\lceil \frac{L}{\sigma} \right\rceil + 1$ i niech odległość między dwoma różnymi punktami na torze spiralnym jest mniejsza lub równa σ . W przypadku, gdy tor spiralny będzie bliski odległości geodezyjnej, kąt między punktami będzie równy tej wartości, dla punktów położonych bliżej punktu startowego i końcowego skanowania wartość tego kąta będzie mniejsza od tej wartości. **Wykazaliśmy zatem, że dla dowolnych sąsiednich punktów położonych dla spiralnej trajektorii kąt pomiędzy nimi jest mniejszy niż σ .** \square

Rysunek 3.10 przedstawia wartości kąta między kolejnymi punktami na trajektorii skanowania dla $\sigma = 0,2$ w zależności od parametru s .



Rys. 3.10. Kąty pomiędzy kolejnymi punktami spiralnej trajektorii

¹² Upraszczamy wyliczanie kąta pomiędzy kolejnymi zwojami spirali, wyliczając odległość pomiędzy punktami położonym na tym samym „południku”.

Założmy, że kąt pomiędzy kolejnymi zwojami spirali powinien wynosić $\sqrt{2}\sigma$. Wtedy $k = \frac{2\pi}{\sqrt{2}\sigma}$ oraz odległość między kolejnymi punktami na trajektorii będzie mniejsza niż $\sqrt{2}\sigma$. Wtedy

$$\frac{L}{\sqrt{2}\sigma} = \frac{2k + 1}{\sqrt{2}\sigma} = \frac{2 \frac{2\pi}{\sqrt{2}\sigma} + 1}{\sqrt{2}\sigma} = \frac{4\pi + \sqrt{2}\sigma}{2\sigma^2} < \left[\frac{4\pi + \sqrt{2}\sigma}{2\sigma^2} \right] + 1, \quad (3.13)$$

stąd wzór 3.8 na liczbę punktów widokowych dla zadanej rozdzielczości skanowania.

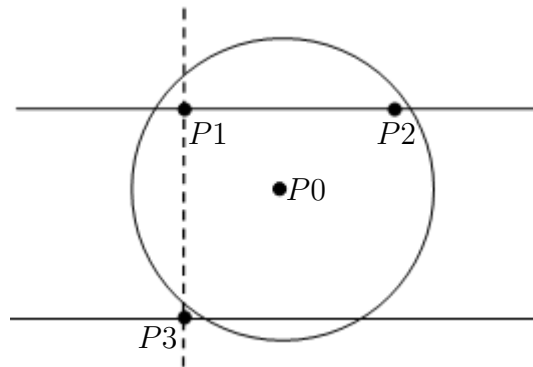
Przy pomocy powyższych lematów udowodnimy następujące twierdzenie:

Twierdzenie 3.3.1. *Każdy widok, w którego obszar widokowy można wpisać stożek o wierzchołku leżącym w środku sfery widokowej i kącie półrozwarcia równym rozdzielczości skanowania σ zostanie wygenerowany metodą spiralnego skanowania.*

Dowód. (nie wprost) Założmy, że istnieje widok niewygenerowany naszą metodą w którego obszar jednowidokowy można wpisać stożek opisany w twierdzeniu. Z każdego punktu widokowego w tym obszarze otrzymujemy ten brakujący widok, w szczególności w obszarze wpisanego stożka. Wykażemy teraz, że w obszarze wpisanego stożka na pewno mieści się punkt widokowy wygenerowany naszą metodą (kąt pomiędzy nim a osią stożka jest mniejszy lub równy σ).

Niech punkt P_0 będzie przecięciem osi stożka ze sferą widokową. Ponieważ kąt pomiędzy kolejnymi zwojami spirali jest stały (lemat 3.3.1) i jest równy $\sqrt{2}\sigma$ a rozwarłość stożka jest równa 2σ , trajektoria skanowania jest odległa od P_0 przynajmniej o mniej niż $\frac{\sqrt{2}\sigma}{2}$. Najbliższy od P_0 punkt widokowy na tej trajektorii oznaczmy przez P_1 . Sąsiedni punkt widokowy na trajektorii skanowania oznaczmy przez P_2 . Niech P_3 będzie położony na kolejnym zwoju spirali (na tym samym „południku” co P_1). Oczywiście kąt pomiędzy P_1 i P_2 również jest mniejszy niż σ (lemat 3.3.2) a kąt między P_1 i P_3 jest mniejszy niż $\sqrt{2}\sigma$ (lemat 3.3.1). Maksymalny kąt pomiędzy P_2 i P_3 wynosi 2σ . Z nierówności trójkąta wynika, że kąt między P_0 a jednym z punktów P_1 , P_2 lub P_3 jest mniejszy lub równy σ .

A zatem P_1 , P_2 lub P_3 leżą w obszarze wpisanego stożka, co oznacza, że ten widok został wygenerowany, a to daje sprzeczność z założeniem. \square



Rys. 3.11. Ilustracja do dowodu kompletności reprezentacji

Wykazaliśmy więc, że każdy losowo wybrany punkt na sferze widokowej jest odległy od wygenerowanego punktu opisywaną metodą co najwyżej o σ , co w praktyce oznacza generowanie reprezentacji z zadaną dokładnością.

3.4. Implementacja algorytmu

Przedstawiony sposób generowania widoków został przetestowany na szerokiej grupie wielościanów monottonnych. Niektóre bryły i uzyskane widoki są dostępne w dodatku „Wyniki”, z kolei szczegółowe algorytmy opisane w tym rozdziale znajdują się w dodatku „Algorytmy”.

Jako żadaną rozdzielczość skanowania przyjęto

$$\sigma = 0,25^\circ (0,00436 \text{radiana})$$

co odpowiada rzeczywistej dokładności urządzeń pozyskujących a kąt rozwarcia stożka widokowego $2\alpha = 45^\circ$ ¹³. Zgodnie z wzorem 3.8 daje to 330187 punktów widokowych rozmieszczonych na spiralnej trajektorii na sferze SWP.

3.4.1. Przygotowanie wielościanu i generowanie zbioru punktów widokowych

Przygotowanie wielościanu do wyznaczania jego widoków, obejmuje następujące kroki:

1. Wyznaczenie środka geometrycznego obiektu i przesunięcie całego obiektu tak, by ten środek był położony w środku układu współrzędnych.
2. Walidacja modelu bryły (sprawdzenie, czy jest to rzeczywiście wielościan monottonny).
3. Wyznaczenie promieni r (sfera opisana) i R (sfera widokowa).

W walidacji wielościanu monottonnego 1 rodzaju bardzo przydaje się spostrzeżenie, że dla każdej jego ściany opisanej równaniem:

$$Ax + By + Cz + D = 0, \tag{3.14}$$

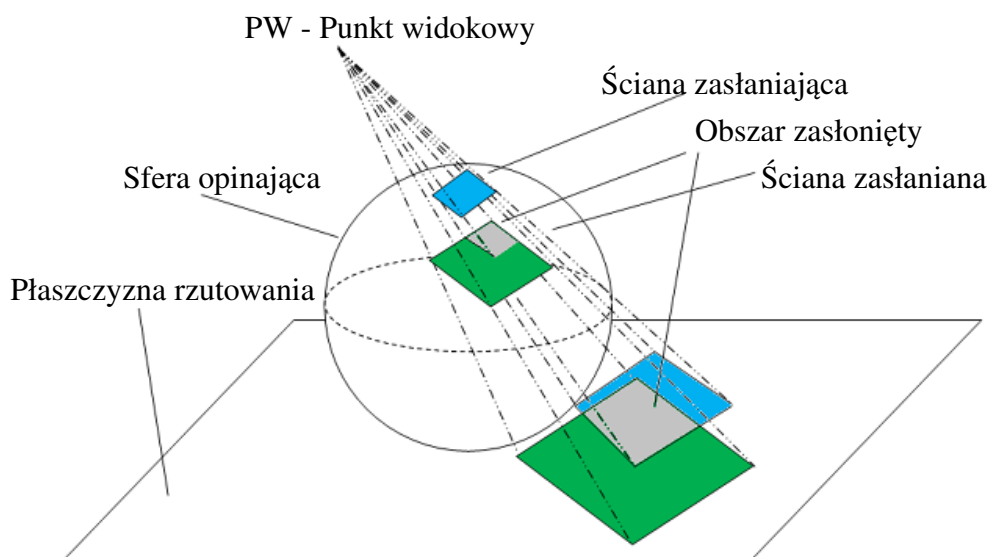
środek układu współrzędnych leży „po wewnętrznej” stronie ściany, a zatem wystarczy sprawdzić czy współczynnik D w równaniu 3.14 jest ujemny. Następnie następuje etap wygenerowania zbioru punktów widokowych na SWP z których będą wyznaczone widoki. Robimy to dla ustalonego promienia R oraz zadanej rozdzielczości σ . Mając punkty widokowe możemy wyznaczać z nich widoki.

3.4.2. Wyznaczanie widoku z punktu widokowego

Do wyznaczenia widoku potrzebujemy wyznaczenia jego widocznych cech dystynktywnych, którymi w naszym przypadku były ściany wielościanu. Widoczność ścian z określonego punktu widokowego w opisywanych algorytmach dla wielościanów monottonnych jest wyznaczana w następujący sposób:

¹³ Przeprowadzono też testy dla innych kątów i rozdzielczości.

1. Wyznaczamy zestaw ścian na wypukłej części bryły, które nie mogą być przesłonięte przez inne ściany wielościanu. W tym celu sprawdzamy, czy są wierzchołki wielościanu leżące po jej zewnętrznej stronie. Brak takich punktów oznacza, że ściana należy do tego zbioru. Jeżeli są takie punkty to dodajemy je do zbioru punktów leżących na zewnątrz określonej ściany. W tym też kroku definiujemy zestaw ścian sąsiadujących z nimi (ściany sąsiadujące nie powodują zasłonięć)¹⁴.
2. Wyznaczamy ściany niewidoczne następująco: gdy punkt widokowy leży na ścianie lub jest po jej wewnętrznej stronie, to ściana jest **niewidoczna**. W przeciwnym wypadku, gdy ściana należy do zbioru ścian leżących na wypukłości bryły, wtedy jest **widoczna**, natomiast gdy nie należy do tego zbioru - jest określana jako **potencjalnie widoczna** i wykonujemy kolejne kroki w celu określenia widoczności ściany.
3. Dla ścian potencjalnie widocznych dokonujemy rzutowania na płaszczyznę prostopadłą do wektora \overrightarrow{OPW} a następnie pozbywamy się jednej z współrzędnych rzutując na jedną z płaszczyzn: OX, OY, OZ (w zależności od największej składowej wektora \overrightarrow{OPW}).
4. Ze zbioru wierzchołków leżących na zewnątrz ściany zostawiamy takie, które mieszczą się w indywidualnym stożku widokowym ściany i których odległość od punkty widokowego jest mniejsza niż największa odległość wierzchołka należącego do testowanej ściany od PW.
5. Następnie wyznaczamy zbiór ścian zawierających te wierzchołki, z których usuwamy ściany odwrócone tyłem oraz sąsiadujące z testowaną ścianą.



Rys. 3.12. Analiza zasłonięć przez rzutowanie na płaszczyznę

6. Ostatnim krokiem jest rzutowanie na płaszczyznę rys. 3.12. W wyniku otrzymujemy zbiór wielokątów na płaszczyźnie i można kolejno odejmować od testowanego wielokąta

¹⁴ Czynności zdefiniowane w tym punkcie możemy wykonać raz dla całej bryły, co jest istotnym czynnikiem optymalizacyjnym wpływającym na czas wykonywania algorytmu

inne wielokąty ze zbioru potencjalnie widocznych ścian. Jeżeli powierzchnia obszaru po tych operacjach jest mniejsza (większa bądź równa) od założonego progu, wtedy ściana jest niewidoczna (widoczna)¹⁵.

3.4.3. Otrzymane rezultaty

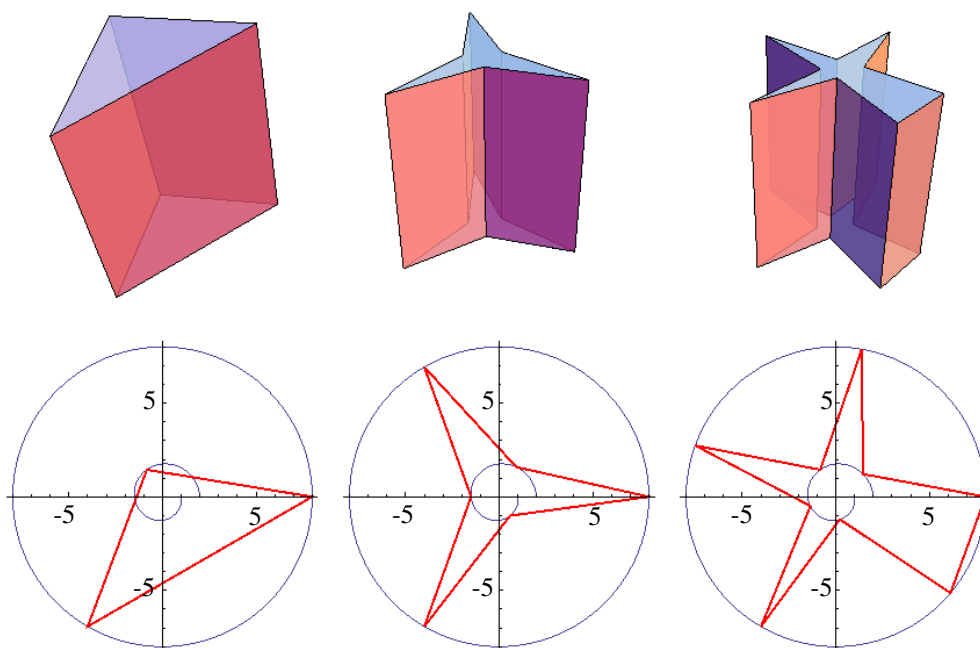
Uzyskane widoki dla kilku wybranych brył znajdują się w dodatku „Wyniki”. Do badania, jak zmienia się czas wykonywania algorytmu oraz liczba uzyskanych widoków, wykorzystano bryłę testową (graniastosłup) o zmiennej liczbie ścian, którego podstawy są konstruowane w następujący sposób:

$$\begin{cases} x = r_p * \cos(t) \\ y = r_p * \sin(t) \\ z = -8, 0 \text{ (} 8, 0 \text{ w przypadku drugiej podstawy graniastosłupa).} \end{cases} \quad (3.15)$$

Wierzchołki podstawy leżą na krzywej opisanej równaniem 3.15. Aby zapewnić podział okręgu na n części, parametr t zmienia się z krokiem $\frac{2\pi}{n}$. Dodatkowo, by szerokości ścian bocznych graniastosłupa były różne, wierzchołki podstawy oparto na promieniu r_p , który zmienia się w następujący sposób:

$$\begin{cases} r_p = 2 - \frac{t}{2\pi} \text{ dla nieparzystego indeksu wierzchołka,} \\ r_p = 8 \text{ dla parzystego indeksu wierzchołka.} \end{cases} \quad (3.16)$$

Podstawy oraz przykładowe graniastosłupy pokazano na rys. 3.13.



Rys. 3.13. Bryły testowe i ich podstawy o 3, 6 i 9 krawędziach.

¹⁵ W kolejnych krokach możemy się posługiwać już wyznaczonym wielokątem.

3.5. Ocena metody

Integralną częścią oceny algorytmu jest analiza jego złożoności obliczeniowej.

3.5.1. Złożoność obliczeniowa metody

Złożoność obliczeniowa metody wyznaczenia widoków wielościanu zależy od liczby ścian w wielościanie (oznacmy ją przez n) oraz liczby punktów widokowych na sferze (przyjmijmy oznaczenie N) i można ją opisać następującą zależnością:

$$Z = k_1 + N * k_2 + k_3, \quad (3.17)$$

gdzie k_1 jest kosztem wyznaczenia N punktów widokowych (zależność liniowa) (krok 1), k_2 jest kosztem uzyskania widoku z punktu widokowego, a k_3 - kosztem usunięcia powtarzających się widoków w bazie. Koszt wyznaczenia widoczności ściany w wielościanie jest liniowy, a zatem koszt wyznaczenia ścian widocznych wielościanu z punktu widokowego ma złożoność rzędu n^2 . Kolejnym krokiem jest usunięcie powtarzających się widoków ze zbioru N -elementowego, co można zrobić kosztem $n \cdot N \log N$ operacji¹⁶. Zatem złożoność obliczeniową wspomnianej metody można zapisać wzorem:

$$Z = O(N + N \cdot n^2 + n \cdot N \log N). \quad (3.18)$$

Jeżeli nie ma się wpływu na rozdzielczość skanowania i jest ona stała, a koszt algorytmu zależy tylko od kształtu bryły, wtedy $N = const$ oraz $N \log N = const$ i uzyskuje się oszacowanie:

$$Z = O(n^2), \quad (3.19)$$

czyli problem ma kwadratową złożoność. W ten sposób **wykazaliśmy pierwszą tezę rozprawy**.

Jak widać, głównym krokiem wpływającym na złożoność tej metody jest sposób wyznaczania powierzchni widocznych. Głównym krokiem optymalizacyjnym w tym zakresie jest jak najbardziej efektywne wyznaczanie powierzchni widocznych i niektóre zastosowane rozwiązania opisane zostały w poprzednim podrozdziale.

Innym rozwiązaniem jest sprawdzanie, czy istnieje już taki widok w zbiorze wygenerowanych widoków, a w przypadku negatywnej odpowiedzi dodanie go do tego zbioru. To jednak może być opłacalne w przypadku małej liczby widoków, co się wiąże z małą liczbą ścian w bryle. Jak już wcześniej wspomniano, zdarzenia wizualne są generowane przez zdarzenia kkk , a wierzchołki obszarów jednowidokowych byłyby generowane przez dwa takie zdarzenia. Zatem **liczba wszystkich widoków** w przypadku

¹⁶ Wynika to z faktu, że etykiety widoków (o długości n) można posortować, co, jak wiadomo, można zrobić w czasie $O(n \cdot N \log N)$, a następnie w sposób liniowy usunąć powtarzające się elementy.

wielościątów monottonnych w modelu sfery widokowej z rzutowaniem perspektywicznym **daje następujące oszacowanie:**

$$\overline{W_{rep}} = O(n^6). \quad (3.20)$$

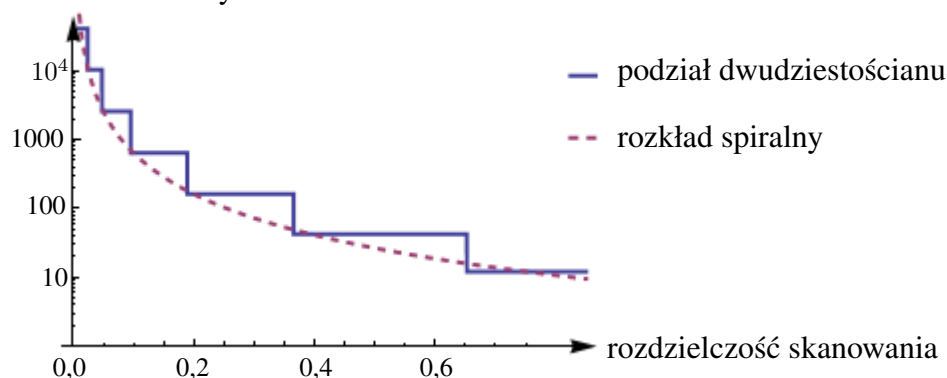
W praktyce liczba widoków nie rośnie tak szybko, jak w oszacowaniu wg równania 3.20 (w przypadku wielościątów monottonnych zdarzenie *kkk* nie są zbyt częste) i przy stosunkowo małej liczbie ścian, rzeczywiście przeszukiwanie zbioru widoków, przed dodaniem nowego, może dać krótszy czas wykonywania algorytmu, ale nie wpływa to na ogólność przyjętego sposobu postępowania.

3.5.2. Wady i zalety proponowanego rozwiązania

Główne zalety opisanego sposobu generowania widoków to:

1. **brak nakładających się obszarów** skanowania (występujących w metodzie ze spiralnym omiotem obszaru nad ścianą), co prowadzi do wyznaczenia znacząco mniejszej liczby punktów widokowych; w konsekwencji ten algorytm jest szybszy od poprzedniego,
2. na ogół otrzymuje się **mniej punktów widokowych** (np. rys. 3.14)¹⁷,
3. zbiór punktów widokowych jest **uporządkowany** (wygoda w budowie innych struktur¹⁸),
4. zbiór punktów widokowych jest rozłożony bardziej **równomiernie**¹⁹.

liczba punktów widokowych



Rys. 3.14. Porównanie liczby punktów widokowych w zależności od żądanej rozdzielczości dla metod: z podziałem 20-ścianu foremnego i równomiernej spirali

Do wad rozwiązania można zaliczyć:

1. możliwość pominięcia niektórych widoków o małej średnicy obszaru widokowego,
2. liczba wygenerowanych widoków zależy od punktu startowego trajektorii.

¹⁷ Jest to szczególnie widoczne w porównaniu z algorytmem z podziałem dwudziestościanu foremnego.

W tym przypadku liczba punktów widokowych może być nawet ok. 2 razy mniejsza.

¹⁸ Np. do generowania grafów aspektowych.

¹⁹ W porównaniu z algorytmem z podziałem dwudziestościanu foremnego.

Powierzchnia obszarów widokowych na SWP niewygenerowanych widoków jest jednak stosunkowo mała, zatem niewielkie jest prawdopodobieństwo otrzymania takiego widoku w procesie pozyskiwania. Postępowanie przeciwne²⁰ prowadziłyby do dużej liczby widoków (patrz zależność 3.20), jak też znacznie wydłużyłoby czas generowania widoków. W praktyce liczba widoków w bazie może być dalej redukowana, np. przez wprowadzenie prawdopodobieństwa uzyskania jakiegoś widoku.

Kolejny rozdział jest poświęcony przykładowym praktycznym zastosowaniem widoków.

²⁰ Generowanie pełnego zbioru widoków przez matematycznie dokładne wyliczenie wierzchołków obszarów jednowidokowych.

4. Przykład zastosowania modeli widokowych

Założenia do koncepcji wizualnego kanału informacyjnego dla robota na bazie 3W modeli widokowych obiektów zostały przedstawione w [71]. Tutaj zostanie przedstawiony przykład wykorzystania reprezentacji widokowej wielościanu do identyfikacji z danymi przechowywanymi w mapach zasięgu.

4.1. Przykład SIW z wykorzystaniem reprezentacji widokowej

Zależnie od zasady działania systemu rozpoznającego można widoki przechowywać w takich strukturach, by proces pasowania danych przebiegał w sposób najbardziej efektywny.

4.1.1. Składowe reprezentacji widokowej

Opisywane w poprzednim rozdziale metody generowania reprezentacji widokowej generowały widoki na podstawie reprezentacji B_{rep} . **Widok** zaś był **reprezentowany przez indeksy ścian widocznych**¹. Takie ogólne podejście można rozwijać, dopasowując strukturę widoku do konkretnych zastosowań. Przykładem takich modyfikacji może być upraszczanie liczby widoków poprzez **usuwanie podobnych struktur geometrycznych**. Na przykład sześciąt daje 26 różnych widoków (biorąc pod uwagę indeks ścian), a tylko 3 widoki z użyciem struktur geometrycznych (ściana, 2 ściany, 3 ściany). Zaletą takiego podejścia jest oczywiście zmniejszenie liczby widoków, ale możemy tracić w ten sposób kontrolę nad informacją czy liczba widoków jest wystarczająca do identyfikacji obiektu.

Innym przykładem jest **wzbogacenie widoku o pewne informacje geometryczne**. Na przykład do informacji o ścianie możemy dodać długości jej krawędzi i kąty pomiędzy nimi. Do widoku - jego centroid i wzajemne położenie wektorów normalnych ścian w widoku. Oprócz cech geometrycznych można w zależności od potrzeby systemu rozpoznającego dodać informację o kolorze lub teksturze ściany.

Widok możemy także wzbogacać o informację, które **ściany** są **całkowicie widoczne (zbiór ŚCW)**, a które są **częściowo widoczne (ŚCzW)** w danym widoku. Jest to szczególnie użyteczne jeżeli jedynymi elementami dystynktywnymi są ściany (nie kontrolujemy widoczności wierzchołków i krawędzi).

¹ Jako cechę wyróżnioną obiektu przyjęto ścianę.

W opisywanej przykładowej metodzie pasowania w widoku rozróżniamy ściany $\acute{S}CW$ i $\acute{S}CzW$ oraz przechowujemy cechy geometryczne ścian i widoku (centroid i suma wektorów normalnych ścian $\acute{S}CW$).

4.1.2. Opis metody

Metoda ta (opisana w pracy [92]) wykorzystuje W_{rep} wielościanu jako narzędzie do identyfikacji z danymi pozyskanymi z map zasięgu. Główne kroki tej metody to:

1. Pozyskiwanie danych.
2. Wyodrębnienie cech charakterystycznych.
 - Wyszukanie obszarów wielokątowych płaskich (ścian).
 - Obliczenie długości krawędzi i kątów między wierzchołkami.
3. Pasowanie ze ścianami zdefiniowanymi w bazie danych.
4. Wybór odpowiedniego widoku(ów) z bazy danych.
5. Trójwymiarowe pasowanie pozyskanych danych z widokami.
 - Wyznaczenie środka geometrycznego zbioru punktów pozyskanych ze sceny, odnoszących się do wybranych ścian.
 - Przesunięcie tych punktów do środka układu współrzędnych.
 - Wyznaczenie średniego wektora z wektorów normalnych wszystkich całkowicie widocznych ścian.
 - Obrót wyznaczonego wektora (a wraz z nim wszystkich punktów pozyskanych ze sceny) tak, aby pokrywał się z wektorem $(0, 0, 1)$.

Ponadto dla każdego widoku z bazy danych, będącego kandydatem na pasujący widok pozyskanych danych:

 - Obrót chmury punktów (ChP) dookoła osi z .
 - Wyznaczenie $h(ChP, W)$ - skierowanej odległości Hausdorffa (opisana w rozdziale 4.2.1) między punktami a widokiem.
 - Wyznaczenie $h(W, ChP)$ - skierowanej odległości Hausdorffa pomiędzy widokiem a punktami.

Pozyskiwanie danych

Przyjmuje się, że narzędziem pozyskiwania danych są skanery zasięgu, z których można otrzymać odległości do uporządkowanego zbioru punktów z otoczenia skanera. Dane te są transformowane do sferycznego, a następnie kartezjańskiego, układu współrzędnych.

Wyodrębnienie cech charakterystycznych

Ponieważ celem jest rozpoznawanie wielościanów, wystarczy skupić się na wyszukiwaniu powierzchni płaskich. Na przykład Jiang i Bunke w [49] podają sposób na segmentację powierzchni płaskich, który ze względu na szybkość i prostotę znajduje szerokie zastosowanie w robotyce (np. [53, 77]).

Pasowanie ze ścianami z bazy danych

Dla danego płaskiego obszaru wyznaczamy długość krawędzi i kąty między nimi. Następnie wyszukujemy w bazie danych ścianę o takich geometrycznych własnościach. Tabela ze ścianami ma dodatkową informację o wielościanach, do których należą. Baza danych z tabelą widoków wielościanu (wzbogaconą o informacje, czy ściana jest całkowicie czy częściowo widoczna), jest tworzona wcześniej (przed procesem pozyskiwania danych).

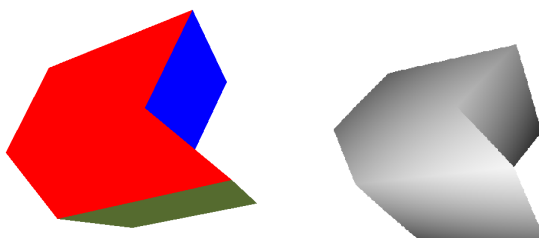
Wybór widoków z bazy danych

Ze zbioru widoków danego wielościanu wybieramy te zawierające tę ścianę jako całkowicie widoczną. Jeżeli w widoku są określone inne ściany ze zbioru ŚCW to również dokonujemy pasowania. Pozostałe płaskie obszary ze sceny są kandydatami na ściany częściowo widoczne w danym widoku (jeżeli takie występują).

Trójwymiarowe pasowanie pozyskanych danych z widokami

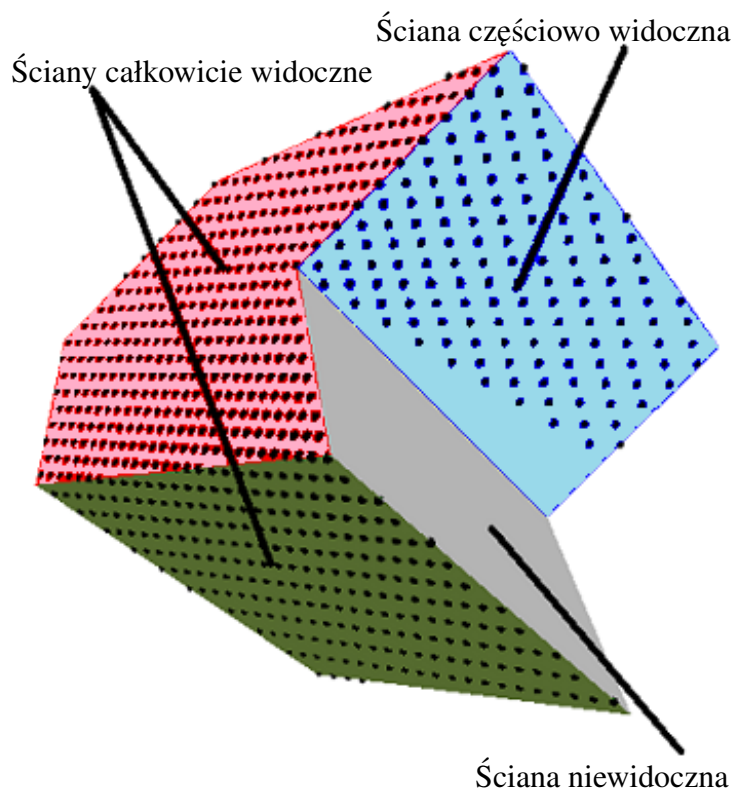
Oznaczmy zbiór punktów pozyskanych ze sceny przez ChP a widok z V_{rep} , pasujący do ścian z wyodrębnionych danych przez W . Wyznaczamy centroid punktów z mapy zasięgu, które sklasyfikowaliśmy jako należące do ŚCW. Tworzymy pewien lokalny układ współrzędnych, w których wyznaczony centroid leży w środku układu współrzędnych i pokrywa się z centroidem ŚCW widoka. Następnie obracamy ChP oraz W w taki sposób, by suma wektorów ścian ŚCW pokrywała się z wektorem $(0, 0, 1)$. Następnie za pomocą technik ICP (zdefiniowanych w rozdziale 4.2.2) szukamy takiego przekształcenia sztywnego (obrotu wokół osi z), aż odległość $(h(ChP, W))$ między ChP i W będzie minimalna. Jeśli $h(ChP, W) \neq 0$ oznacza to, że ChP nie jest podzbiorem W i kontynuujemy technikę ICP. Odległość $h(V, ChP)$ powinna wynosić 0 (w praktyce mniejsza od jakiegoś progu) dla ścian całkowicie widocznych i być większa od zera dla wszystkich ścian.

Przykładowy widok wielościanu oraz dane z map zasięgu (dla lepszego zilustrowania pokazane w skali szarości) odpowiadające temu widokowi pokazano na rys. 4.1.



Rys. 4.1. Widok wielościanu i mapa zasięgu w skali szarości dla tego widoku

Przykładowe pasowanie punktów z mapy zasięgu z widokiem jest pokazane na rys. 4.2.



Rys. 4.2. Pasowanie punktów z mapy zasięgu z widokiem.

4.1.3. Ocena metody

Przedstawiona metoda jest przykładem pasowania geometrycznego punktów pozyskanych z mapy głębi z reprezentacją widokową. Oczywiście zakres zastosowań W_{rep} jest dużo szerszy, można wykorzystywać widoki do innych systemów z pasowaniem geometrycznym lub przy ich pomocy wyznaczać inne struktury (np. deskryptory statystyczne) pomocne w rozpoznawaniu obiektów 3W.

Identyfikacja obiektu 3W na podstawie jednego wyglądu obiektu uzyskanego z SWP nie zawsze jest możliwa². Można postawić pytanie: ile widoków jest potrzebne, żeby jednoznacznie zidentyfikować wielościan monotony? Problem ten można też sformułować też jako: **jaka jest minimalna liczba pasowań (MLP)** pozyskanych danych wystarczająca do poprawnej identyfikacji wielościanu monotony.

Zauważmy, że w każdym widoku wielościanu monotony co najmniej jedna ściana jest widoczna w całości (w szczególnym przypadku jest to tylko ta jedna ściana). Widok zawierający pojedynczą całkowicie widoczną ścianę wystarcza do identyfikacji ściany, ale nie niesie żadnej informacji o jej położeniu względem innych ścian. Zatem ten widok należy na ogół uzupełnić o widoki zawierające inne ściany.

Zbiór takich widoków w pełni pokrywający powierzchnię wielościanu można skonstruować w następujący sposób:

² Ale można stwierdzić, że to nie jest obiekt którego szukamy.

1. Dla każdej ściany wybieramy widoki, w których jest ona całkowicie widoczna.
 2. Dodatkowo żądamy, by te widoki zawierały ściany sąsiednie.
 3. Dokonujemy pasowania pozyskanego widoku z widokiem z bazy danych spełniającego podane założenia i jeżeli jest takie dopasowanie, to powtarzamy proces dla innych ścian.
- Zatem MLP jest na pewno mniejsza lub równa od sumy liczby sąsiadów dla każdej ściany. Wynika stąd, że MLP możemy oszacować w następujący sposób:

$$MLP = O(n^2). \quad (4.1)$$

W ten sposób została wykazana druga teza rozprawy. Postępując w opisany powyżej sposób otrzymujemy pełne pokrycie badanego wielościanu przez zbiór widoków zapewniający identyfikację obiektu.

Jak widać reprezentacje widokowe można wykorzystywać razem z istniejącymi technikami pasowania geometrycznego.

4.2. Metody pasowania obiektów 3W

Zadanie pasowania, które polega na porównywaniu pozyskanych danych z modelami z repozytorium, opiera się głównie na znalezieniu najlepszej miary podobieństwa tych obiektów. Wtedy baza danych jest przeszukiwana pod kątem znalezienia najlepiej dopasowanego modelu, a zwracane zapytanie może zostać poddane dalszej weryfikacji. Do tej pory zaproponowano wiele miar służących do tego celu.

4.2.1. Metryki używane w pasowaniu

Jedną z najstarszych i powszechnie używanych jest metryka euklidesowa (stosowana np. w algorytmie ICP [9]), korelacja (do porównania deskryptorów opartych na histogramach, opisanych w drugim rozdziale), odległość Minkowskiego (np. do porównania dystrybuanty kształtu) i inne (przegląd miar podobieństwa kształtów możemy znaleźć w pracy [106]). Metryką, która wydaje się być odpowiednia do porównywania kształtów jest metryka Hausdorffa, definiowana jako wartość maksymalna skierowanych odległości Hausdorffa.

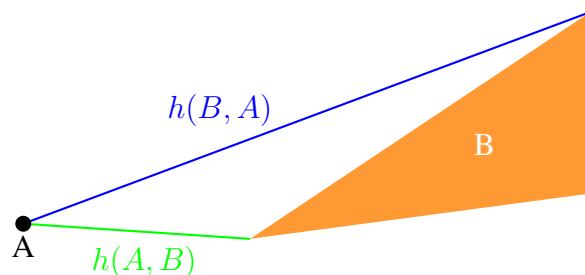
$$H(A, B) = \max(h(A, B), h(B, A)), \quad (4.2)$$

gdzie

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (4.3)$$

jest skierowaną odległością Hausdorffa. Wielkość $h(A, B)$ nie jest symetryczna, co wynika wprost z jej definicji (patrz rys. 4.3). Miara Hausdorffa ($H(A, B)$) spełnia już warunek symetryczności. Metryka ta jest szeroko wykorzystywana do określania podobieństwa między kształtami, np. Huttenlocher i Kedem, [44], omawiają technikę obliczania odległości Hausdorffa pomiędzy zbiorem punktów. Barton w [5] podaje sposób na efektywną technikę

obliczania dokładnej odległości Hausdorffa pomiędzy wielościanami reprezentowanymi przez trójkątną³ siatkę wykorzystując oś środkową obiektu.



Rys. 4.3. Odległość Hausdorffa

Wadą tej metryki jest duża wrażliwość na położenie nawet pojedynczych punktów (fałszywe dane możemy pozyskać na skutek zaszumienia), stąd też czasem używa się zmodyfikowanej odległości Hausdorffa (pomijającej część punktów).

4.2.2. Wybrane metody pasowania geometrycznego

Jako przykład popularnych metod pasowania geometrycznego brył możemy wymienić mieszanie geometryczne oraz ICP.

Mieszanie geometryczne

Metoda ta, zaproponowana przez Lamdana i Wolfson w [63] (ang. *geometric hashing*) działa w dwóch krokach. W pierwszym przygotowawczym kroku przygotowywana jest baza obiektów wraz z opisem ich niezmienniczych cech geometrycznych (np. ściana może być reprezentowana przez zbiór punktów). Punkty są opisywane w sposób maksymalnie niezależny od wyboru układu współrzędnych, np. opisane względem każdej niewspółliniowej trójki punktów (baza). Takie współrzędne stanowią klucz w tablicy mieszającej a wartością jest para (model,baza). Przekształcenia sztywne bryły nie zmieniają tych wartości. W fazie rozpoznawania, mając wyodrębnione cechy (punkty), wybieramy pewną bazę, wyliczamy współrzędne punktów w tej bazie i przeszukujemy tablicę mieszającą. Każdy punkt „głosuje” na pewną parę (model,baza).

Przykłady wykorzystania tej techniki w systemach rozpoznawania są w [97, 69].

ICP

Jednym z najpowszechniej stosowanych algorytmów pasowania jest jest algorytm **iteracyjnego najbliższego punktu ICP** (ang. *Iterative Closest Point*) przedstawiony w [9, 17]⁴, a później modyfikowany w wielu innych pracach. Algorytm który przedstawili Besl

³ Aby wyznaczyć odległość Hausdorffa punktu od trójkąta wystarczy tylko sprawdzić odległości punktu od jego wierzchołków.

⁴ Podobne rozwiązanie, powstałe niezależnie.

i McKay w [9] polegał na tym, że w dwóch zbiorach punktów najpierw trzeba było znaleźć odpowiadające sobie punkty (szukano najbliższego punktu z drugiego zbioru), następnie szuka się takiego przekształcenia sztywnego (obrotu R i przesunięcia t , by funkcja kosztu (4.4) była jak najmniejsza

$$f = \sum |Rp_i + t - q_i|^2. \quad (4.4)$$

Znajdowanie najbliższego punktu jest najbardziej złożonym obliczeniowo krokiem w tym algorytmie. Początkowo ten krok sprowadzał się do znalezienia punktu z drugiej siatki o najmniejszej metryce euklidesowej. Chen i Medoni minimalizują funkcję kosztu używając odległości punktu od najbliższej stycznej płaszczyzny w drugiej siatce punktów. Turk i Levoy w [104], by przyspieszyć działanie algorytmu do wyliczania najbliższej odległości używają równomiernie rozłożonej siatki punktów na obiekcie (przybliżającej obiekt) pasując ją z danymi z map zasięgu.

Rusinkiewicz i Levoy w [90] zaproponowali szybką i efektywną wersję algorytmu ICP składającego się z 6 kroków:

1. wybór zbioru punktów do pasowania (na jednej lub dwóch siatkach);
2. szukanie par punktów z obu siatek;
3. znajdowanie miary pomiędzy dopasowanymi punktami;
4. odrzucanie par punktów zbyt oddalonych od siebie;
5. wyznaczenie funkcji kosztu dla bieżącego dopasowania;
6. minimalizacja błędu przy użyciu np. metody najmniejszych kwadratów.

Główne wady metod pasowania opartych na ICP to:

- wysoki koszt obliczeniowy dla skomplikowanych modeli (np. duża liczba wierzchołków);
- szybka początkowa zbieżność lecz mniejsza w późniejszych iteracjach.

Opisane metody są tylko wybranymi metodami pasowania geometrycznego (przegląd innych technik pasowania obiektów 3D można znaleźć n.p. w [46] (2000 r.), [82] (2004 r.) i [101] (2008 r.).

5. Podsumowanie

Niniejsza praca przedstawia omówienie metod generowania reprezentacji widokowej. Ważnymi powodami, dla których zrezygnowano z wyznaczania dokładnej reprezentacji tytułowej klasy brył, były:

- duża liczba potencjalnych obszarów jednowidokowych na sferze widokowej $O(n^6)$;
- trudności z opisem powierzchni przecięcia powierzchni zdarzeń typu kkk (krzywa drugiego stopnia) z SWP;
- generowanie widoków mało istotnych dla systemów rozpoznających (małe prawdopodobieństwo uzyskania widoku, często mniejsze niż dokładność urządzenia pozyskującego) a zwiększających liczbę widoków.

W pracy przedstawiono następujące sposoby generowania reprezentacji widokowej dla wielościanów monottonnych :

- obrót stożka dopełniającego wokół ściany (pomijanie niektórych widoków, brak kontroli nad dokładnością generowanej reprezentacji);
- skanowanie obszaru nad ścianą torem spiralnym (nakładające się obszary skanowania);
- podział ścian wielościanów wielościanów regularnych (metody iteracyjne, niedające jednak w pełni równomiernego rozkładu);
- skanowanie ze spiralnym rozmieszczeniem punktów widokowych na sferze widokowej.

Jako najlepszy sposób generowania W_{rep} proponowany jest autorski sposób wyznaczania jej z punktów widokowych rozłożonych na sferze widokowej z rzutowaniem perspektywicznym w sposób spiralny z zadaną dokładnością. Metoda ta jest w większości wolna od wad innych sposobów wytwarzania reprezentacji widokowej, dodatkowo zapewnia możliwość uporządkowania widoków. Główną wadą tej metody jest większa ilość punktów widokowych wokół początku i końca trajektorii skanowania. W czwartym rozdziale przedstawiono pewien przykład na wykorzystanie reprezentacji widokowej do pasowania z danymi z map zasięgu. W ten sposób **zrealizowano cele pracy**.

Udowodniono również **tezy rozprawy** o możliwości wytworzenia $3W$ przybliżonej reprezentacji widokowej wielościanu monottonnego z jego reprezentacji powłokowej B_{rep} za pomocą algorytmu o złożoności $O(n^2)$ oraz o minimalnej liczbie pasowań danych i modelu wielościanu monottonnego w oparciu o dane z mapy głębi i $3W$ reprezentację widokową, która wynosi $O(n^2)$.

Rozwiązania przedstawione w trzecim i czwartym rozdziale rozprawy obejmują w większości **oryginalne rozwiązania proponowane przez autora rozprawy**.

W szczególności **wkładem autora pracy** w badania nad reprezentacją widokową wielościanów monottonnych **są między innymi:**

1. Implementacja i testowanie algorytmów obrotu stożkiem dopełniającym wokół ściany.
2. Implementacja i testowanie algorytmów związanych z podziałem dwudziestościanu foremego.
3. Opracowanie sposobu spiralnego rozkładu punktów na sferze widokowej (nad ścianą wielościanu i na całej sferze) i porównanie ich z innymi metodami.
4. Wykazanie kompletności generowanej reprezentacji (z zadaną dokładnością) dla metod spiralnych.
5. Opracowanie przykładowego sposobu na wykorzystanie reprezentacji widokowej do pasowania z danymi pozyskanymi z map głębi.

Wyniki przedstawione w tej pracy **rozszerzają klasę brył dla których potrafi się wyznaczać jej reprezentację widokową** (wcześniej potrafiono to robić dla wielościanów wypukłych i dopiero podejmowane były pierwsze próby dla szerszej klasy brył). Przedstawione w pracy metody przybliżone generowania reprezentacji widokowej są na tyle uniwersalne, że po niewielkiej modyfikacji (dotyczy to głównie sposobu wyznaczania powierzchni widocznych) mogą być z powodzeniem stosowane dla brył innych rodzajów. Szczególnie dotyczy to ogólnej klasy wielościanów, ale można je również wykorzystywać dla innych obiektów.

Jako główne kierunki dalszych badań w tym zakresie widzę opracowanie sposobów generowania reprezentacji widokowej obiektów o krzywoliniowej powierzchni, a także (a nawet przede wszystkim) nad praktycznymi zastosowaniami reprezentacji widokowej w zadaniach identyfikacyjnych i rozpoznających (szczególnie w robotyce).

Bibliografia

- [1] M. Ankerst, G. Kastenmuller, H. P. Kriegel, T. Seidl. 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. R. Gutting, D. Papadias, F. Lochovsky, redaktorzy, *Advances in Spatial Databases*, wolumen 1651 serii *Lecture Notes in Computer Science*, strony 207–226. Springer Berlin / Heidelberg, 1999. [cytowanie na str. 15]
- [2] J. B. Arie. Probabilistic Models of Observed Features and Aspects with Application to Weighted Aspect Graphs. *Pattern Recogn. Lett.*, 11:421–427, June 1990. [cytowanie na str. 20]
- [3] F. Arman, J. K. Aggarwal. Model-Based Object Recognition in Dense-Range Images - A Review. *ACM Comput. Surv.*, 25(1):5–43, 1993. [cytowanie na str. 9]
- [4] D. Avis, G. T. Toussaint. An Efficient Algorithm for Decomposing a Polygon into Star-shaped Polygons. *Pattern Recognition*, 13(6):395–398, 1981. [cytowanie na str. 6]
- [5] M. Barton, I. Hanniel, G. Elber, M. Kim. Precise Hausdorff Distance Computation Between Polygonal Meshes. *Computer Aided Geometric Design*, 27(8):580 – 591, 2010. *Advances in Applied Geometry*. [cytowanie na str. 65]
- [6] B. G. Baumgart. Winged Edge Polyhedron Representation. Raport instytutowy, Stanford University, Stanford, CA, USA, 1972. [cytowanie na str. 12]
- [7] B. G. Baumgart. *Geometric Modeling for Computer Vision*. Praca doktorska, Stanford University, Stanford, CA, USA, 1974. AAI7506806. [cytowanie na str. 12]
- [8] G. Bellaire. Feature-Based Computation of Hierarchical Aspect-Graphs. *MG&V*, 2:105–122, 1993. [cytowanie na str. 21]
- [9] P. J. Besl, N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. [cytowanie na str. 65, 66, 67]
- [10] M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Trans. Circuits Syst. Video Techn.*, 11(6):716–719, 2001. [cytowanie na str. 13]
- [11] P. Bose, M. J. Kreveld, G. T. Toussaint. Filling Polyhedral Molds. *Computer-Aided Design*, 30(4):245–254, 1998. [cytowanie na str. 6]
- [12] I. C. Braid. *Designing with Volumes*. Praca doktorska, CANTAB Press, Cambridge Computer Laboratory, University of Cambridge, 1974. [cytowanie na str. 12]
- [13] R. A. Brooks. Symbolic Reasoning among 3D Models and 2D Images. *Artificial Intelligence*, 17:285–34, 1981. [cytowanie na str. 12]
- [14] D. Chen, X. Tian, Y. Shen, M. Ouhyoung. On Visual Similarity Based 3D Model Retrieval. *Comput. Graph. Forum*, 22(3):223–232, 2003. [cytowanie na str. 18, 48]
- [15] H. Chen, B. Bhanu. 3D Free-Form Object Recognition in Range Images Using Local Surface Patches. *Pattern Recogn. Lett.*, 28:1252–1262, July 2007. [cytowanie na str. 17]
- [16] H. H. Chen, T. S. Huang. A Survey of Construction and Manipulation of Octrees. *Computer Vision, Graphics, and Image Processing*, 43(3):409 – 431, 1988. [cytowanie na str. 11]

- [17] Y. Chen, G. Medioni. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.*, 10(3):145–155, 1992. [cytowanie na str. 66]
- [18] R. S. Choraś. *Komputerowa wizja. Metody interpretacji i identyfikacji obiektów*. Akademicka Oficyna Wydawnicza EXIT, 2004. [cytowanie na str. 10]
- [19] C. S. Chua, R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision*, 25:63–85, 1997. [cytowanie na str. 16]
- [20] D. Cohen-Or, Y. L. Chrysanthou, C. T. Silva, F. Durand. A Survey of Visibility for Walkthrough Applications. *IEEE Transactions on Visualization and Computer Graphics*, wolumen 9, strony 412–431, Lip. 2003. [cytowanie na str. 22]
- [21] B. Cyganek. *An Introduction to 3D Computer Vision Techniques and Algorithms*. John Wiley & Sons, 2007. [cytowanie na str. 10]
- [22] C. M. Cyr, B. B. Kimia. 3D Object Recognition Using Shape Similarity-Based Aspect Graph. *IEEE International Conference on Computer Vision ICCV'01*, 1:254–261, 2001. [cytowanie na str. 22]
- [23] M. Dąbkowska, W. S. Mokrzycki. Generowanie widoków 3W obiektów do systemu wizualnej identyfikacji. *Prace IPI PAN*, number 809. Lipiec 1996. [cytowanie na str. 4, 23, 24, 25, 79]
- [24] M. Dąbkowska, W. S. Mokrzycki. A Multiview Model of Convex Polyhedra. *MG&V*, 6(4):419–450, 1997. [cytowanie na str. 4, 23]
- [25] M. Dąbkowska, W. S. Mokrzycki. Generowanie widoków wielościanów wypukłych z bezpośrednim wyznaczaniem O1-W. *Prace Instytutu Podstaw Informatyki Polskiej Akademii Nauk*, 861:3–22, Lipiec 1998. [cytowanie na str. 27, 79]
- [26] M. Dąbkowska, W. S. Mokrzycki. Wytwarzanie widokowej reprezentacji obiektów wg koncepcji sfery widokowej z rzutowaniem perspektywicznym. *Materiały I Krajowej Konferencji Naukowo-Szkoleniowej Komputerowe Systemy Rozpoznawania KOSYR'99*, strony 293–301, Trzebieszowice, 05 1999. Wyd. Politechnika Wroclawska. [cytowanie na str. 23, 26, 79]
- [27] M. De Berg, O. Cheong, M. van Kreveld. *Computational Geometry: Algorithms and Applications*. Springer, 2008. [cytowanie na str. 6]
- [28] J. Denis Dion, D. Laurendeau, R. Bergevin. Generalized Cylinders Extraction in a Range Image. *3D Digital Imaging and Modeling, International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 0:141, 1997. [cytowanie na str. 12]
- [29] F. Durand, G. Drettakis, C. Puech. The 3D Visibility Complex: A New Approach to the Problems of Accurate Visibility. *Rendering Techniques*, strony 245–256, 1996. [cytowanie na str. 22]
- [30] F. Durand, G. Drettakis, C. Puech. The 3D Visibility Complex. *ACM Trans. Graph.*, 21:176–206, April 2002. [cytowanie na str. 22]
- [31] D. W. Eggert, K. W. Bowyer, C. R. Dyer, H. I. Christensen, D. B. Goldgof. The scale space aspect graph. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:1114–1130, November 1993. [cytowanie na str. 20]
- [32] M. F. S. Farias, J. M. de Carvalho. MultiView Technique for 3D Polyhedral Object Recognition Using Surface Representation. *Revista SBA: Controle & Automacao*, 10(2):107–117, 1999. [cytowanie na str. 14]

- [33] O. Faugeras, J. Mundy, N. Ahuja, C. Dyer, A. Pentland, R. Jain, K. Ikeuchi. Why Aspect Graphs Are Not (Yet) Practical for Computer Vision. *CVGIP Image Understanding Workshop*, wolumen 55(1), strony 212–218, March 1992. [cytowanie na str. 21]
- [34] D. Frejlichowski, P. Nuzkiewicz. Przegląd podstawowych metod reprezentacji kształtów 3d. *Studia i Materiały Informatyki Stosowanej*, 2:15–22, 2010. [cytowanie na str. 18]
- [35] A. Frome, D. Huber, R. Kolluri, T. Bulow, J. Malik. Recognizing Objects in Range Data Using Regional Point Descriptors. T. Pajdla, J. Matas, redaktorzy, *Computer Vision - ECCV 2004*, wolumen 3023 serii *Lecture Notes in Computer Science*, strony 224–237. Springer Berlin / Heidelberg, 2004. [cytowanie na str. 17]
- [36] M. Frydler, W. S. Mokrzycki. New, Fast Algorithm of 3D Multiview Polyhedron Representation Generation on View Sphere with Perspective. *Proc. Advance Computer Systems*, 2004. [cytowanie na str. 35, 36, 79]
- [37] M. Frydler, W. S. Mokrzycki. New Results in 3D Views of Polyhedron Generation on View Sphere with Perspective Projection. K. Saeed, J. Peja, redaktorzy, *Information Processing and Security Systems*, strony 77–86. Springer US, 2005. [cytowanie na str. 35]
- [38] M. Frydler, W. S. Mokrzycki. Scanning Faces Surroundings - New Concept in 3D Exact Multiviews of Nonconvex Polyhedron Generation. M. Kurzynski, E. Puchala, M. Wozniak, A. Zolnierek, redaktorzy, *Computer Recognition Systems*, wolumen 30 serii *Advances in Intelligent and Soft Computing*, strony 379–386. Springer Berlin / Heidelberg, 2005. [cytowanie na str. 37]
- [39] Z. Gigus, J. Canny, R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:542–551, June 1991. [cytowanie na str. 21]
- [40] Z. Gigus, J. Malik. Computing the Aspect Graph for Line Drawings of Polyhedral Objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(2):113–122, 1990. [cytowanie na str. 7, 21]
- [41] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, USA, 1990. [cytowanie na str. 9]
- [42] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, strony 203–212, New York, NY, USA, 2001. ACM. [cytowanie na str. 17]
- [43] B. K. P. Horn. Extended Gaussian Images. *Proceedings of the IEEE*, 72(2):1671–1686, 1984. [cytowanie na str. 14]
- [44] D. P. Huttenlocher, K. Kedem. Computing The Minimum Hausdorff distance For Point Sets Under translation. strony 340–349, 1990. [cytowanie na str. 65]
- [45] K. Ikeuchi, M. Hebert. Spherical Representations: from EGI to SAI. Raport instytutowy CMU-CS-95-197, Computer Science Department, Pittsburgh, PA, October 1995. [cytowanie na str. 15]
- [46] A. K. Jain, C. Dorai. 3D object recognition: Representation and Matching. *Statistics and Computing*, 10:167–182, 2000. 10.1023/A:1008998410728. [cytowanie na str. 67]
- [47] R. Jain, R. Kasturi, B. G. Schunck. *Machine Vision*. McGraw-Hill, Inc., New York, NY, USA, 1995. [cytowanie na str. 9]

- [48] A. Jaklič, A. Leonardis, F. Solina. *Segmentation and Recovery of Superquadrics: Computational Imaging and Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 2000. [cytowanie na str. 12]
- [49] X. Jiang, H. Bunke. Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping. *Machine Vision and Applications*, 7:115–122, 1994. 10.1007/BF01215806. [cytowanie na str. 62]
- [50] A. E. Johnson, M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999. [cytowanie na str. 16]
- [51] S. B. Kang, K. Ikeuchi. The Complex EGI: A New Representation for 3-D Pose Determination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:707–721, July 1993. [cytowanie na str. 14]
- [52] M. M. Kazhdan, B. Chazelle, D. P. Dobkin, T. A. Funkhouser, S. Rusinkiewicz. A Reflective Symmetry Descriptor for 3D Models. *Algorithmica*, 38(1):201–225, 2003. [cytowanie na str. 14, 15]
- [53] J. Kjellander, M. Rahayem. Planar Segmentation of Data from a Laser Profile Scanner Mounted on an Industrial Robot. *The International Journal of Advanced Manufacturing Technology*, 45:181–190, 2009. [cytowanie na str. 62]
- [54] J. Koenderink, A. van Doorn. The Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, 32(4):211–216, 1979. [cytowanie na str. 18]
- [55] J. J. Koenderink. *Solid Shape*. Cambridge, MA, USA, 1990. [cytowanie na str. 14]
- [56] M. Kowalczyk. *Dwuipółwymiarowe modele wielościanów wypukłych do identyfikacji wizualnej*. Praca doktorska, Instytut Podstaw Informatyki, Polska Akademia Nauk, Warszawa, 2003. [cytowanie na str. 23, 32]
- [57] M. Kowalczyk, W. Mokrzycki. Concept of 3D View Model of Non-Convex Polyhedra Generation Using View Sphere Partitioning into Single-View Areas. M. Kurzynski, E. Puchala, M. Wozniak, A. Zolnierok, redaktorzy, *Computer Recognition Systems*, wolumen 30 serii *Advances in Soft Computing*, strony 411–418. Springer Berlin / Heidelberg, 2005. [cytowanie na str. 33, 34, 79]
- [58] M. Kowalczyk, W. S. Mokrzycki. Generation of 3D Views of Polyhedra Using View Sphere with Perspective and Complementary Cone. Raport instytutowy 969, IPI PAN, 2003. [cytowanie na str. 28, 79]
- [59] M. Kowalczyk, W. S. Mokrzycki. Methods for Generating 3D Exact Views of Convex Polyhedra for Visual Identification: Part I: View Sphere with Perspective Projection and Iterative Methods. *Machine Graphics & Vision*, 12(4):409–433, 2003. [cytowanie na str. 28]
- [60] M. Kowalczyk, W. S. Mokrzycki. Methods for Generating 3D Exact Views of Convex Polyhedra for Visual Identification: Part II: Non-Iterative Methods, Implementation and Test Results. *Machine Graphics & Vision*, 12(4):435–452, 2003. [cytowanie na str. 28, 29, 31, 79]
- [61] M. Kowalczyk, W. S. Mokrzycki. Obtaining Complete 2.5D View Representation of Polyhedra Using Concept of Seedling Single-View Area. *Comput. Vis. Image Underst.*, 91(3):280–301, 2003. [cytowanie na str. 30]
- [62] J. Krivic, F. Solina. Superquadric-Based Object Recognition. W. Skarbek, redaktor, *Computer Analysis of Images and Patterns*, wolumen 2124 serii *Lecture Notes in Computer Science*, strony 134–141. Springer Berlin / Heidelberg, 2001. [cytowanie na str. 12]

- [63] Y. Lamdan, H. J. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. *ICCV88*, strony 238–249, 1988. [cytowanie na str. 66]
- [64] J. Li, E. Li, Y. Chen, L. Xu, Y. Zhang. Bundled Depth-Map Merging for Multi-View Stereo. *CVPR*, strony 2769–2776, 2010. [cytowanie na str. 10]
- [65] P. Lindstrom, G. Turk. Image-Driven Simplification. *ACM Trans. Graph.*, 19:204–241, July 2000. [cytowanie na str. 48]
- [66] M. Liu, Y.-s. Liu, K. Ramani. Computing Global Visibility Maps for Regions on the Boundaries of Polyhedra using Minkowski Sums. *Comput. Aided Des.*, 41:668–680, September 2009. [cytowanie na str. 22]
- [67] M. Liu, K. Ramani. Computing an Exact Spherical Visibility Map for Meshed Polyhedra. *Proceedings of the 2007 ACM symposium on Solid and physical modeling, SPM '07*, strony 367–372, New York, NY, USA, 2007. ACM. [cytowanie na str. 22]
- [68] D. Marr. *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco, 1982. [cytowanie na str. 17]
- [69] A. S. Mian, M. Bennamoun, R. Owens. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1584–1601, October 2006. [cytowanie na str. 66]
- [70] W. S. Mokrzycki. 3w modele widokowe niewypukłych wielościanów na sferze widokowej z perspektywą. *Prace IPI PAN*, number 926. 2003. [cytowanie na str. 36]
- [71] W. S. Mokrzycki. Wizualny kanał informacyjny dla robota na bazie 3W modeli widokowych obiektów. *Studia Informatica*, 2(4):137–155, 2004. Wydawnictwo Akademii Podlaskiej. [cytowanie na str. 61]
- [72] W. S. Mokrzycki, A. Salamończyk. Generating 3D Multiview Exact Polyhedron Representation by Scanning Faces Surroundings. *Machine Graphics & Vision*, 15(3):525–536, 2006. [cytowanie na str. 37]
- [73] W. S. Mokrzycki, A. Salamończyk. New Results in Generation of Accurate 3D Multiview Representation of Monotonous Polyhedrons. *Computer Recognition Systems 2*, wolumen 45 serii *Advances in Soft Computing*, strony 100–107. Springer Verlag, 2007. [cytowanie na str. 37]
- [74] W. S. Mokrzycki, A. Salamończyk. Spiral Scanning of Faces to Obtain Complete 3D View Representation of Monotonous Polyhedra. *Computer Recognition Systems 2*, wolumen 45 serii *Advances in Soft Computing*, strony 171–178. Springer Verlag, 2007. [cytowanie na str. 45]
- [75] W. S. Mokrzycki, A. Salamończyk. Using Shaded Views in 3D Multiview Representation of Monotonous Polyhedron. *CISIM '07: Proceedings of the 6th International Conference on Computer Information Systems and Industrial Management Applications*, strony 328–336, Washington, DC, USA, 2007. IEEE Computer Society. [cytowanie na str. 37]
- [76] J. Mundy. Object Recognition in the Geometric Era: A Retrospective. J. Ponce, M. Hebert, C. Schmid, A. Zisserman, redaktorzy, *Toward Category-Level Object Recognition*, wolumen 4170 serii *Lecture Notes in Computer Science*, strony 3–28. Springer Berlin / Heidelberg, 2006. [cytowanie na str. 19, 79]
- [77] E. Natonek. Fast Range Image Segmentation for Servicing Robots. *ICRA*, strony 406–411, 1998. [cytowanie na str. 62]

- [78] D. Nehab, P. Shilane. Stratified Point Sampling of 3D Models. *Eurographics Symposium on Point-Based Graphics*, strony 49–56, Czerw. 2004. [cytowanie na str. 10]
- [79] E. Neuman. Inequalities and Bounds for Generalized Complete Elliptic Integrals. *Journal of Mathematical Analysis and Applications*, 373(1):203 – 213, 2011. [cytowanie na str. 52]
- [80] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Matching 3D Models with Shape Distributions. *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, strony 154–166, Washington, DC, USA, 2001. IEEE Computer Society. [cytowanie na str. 15]
- [81] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Shape Distributions. *ACM Trans. Graph.*, 21:807–832, October 2002. [cytowanie na str. 15]
- [82] M. K. Osman, M. Y. Mashor, M. R. Arshad. A Review on 3D Object Representation and Recognition. *Representations*, 2004. [cytowanie na str. 67]
- [83] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko. Function Representation in Geometric Modeling: Concepts, Implementation and Applications. *The Visual Computer*, 11(8):429–446, October 1995. [cytowanie na str. 13]
- [84] H. Plantinga, C. Dyer. The Asp: A Continuous Viewer-Centered Representation for 3D Object Recognition. *ICCV87*, strony 626–630, 1987. [cytowanie na str. 20]
- [85] H. Plantinga, C. R. Dyer. Visibility, Occlusion, and the Aspect Graph. *Int. J. Comput. Vision*, 5(2):137–160, 1990. [cytowanie na str. 7]
- [86] M. Pocchiola, G. Vegter. The Visibility Complex. *Proceedings of the ninth annual symposium on Computational geometry, SCG '93*, strony 328–337, New York, NY, USA, 1993. ACM. [cytowanie na str. 22]
- [87] A. A. Requicha. Mathematical Models of Rigid Solid Objects. Raport instytutowy, Production Automation Project / TM-28, University of Rochester, 1977. [cytowanie na str. 11]
- [88] A. Rosenfeld. Recognizing Unexpected Objects: A Proposed Approach. *Int. J. Pattern Recogn. Artif. Intell.*, 1:71–84, April 1987. [cytowanie na str. 18]
- [89] S. Ruiz-Correa, L. G. Shapiro, M. Melia. A New Signature-Based Method for Efficient 3-D Object Recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:769, 2001. [cytowanie na str. 17]
- [90] S. Rusinkiewicz, M. Levoy. Efficient Variants of the ICP Algorithm. *Proceedings of The Third International Conference on 3-D Digital Imaging and Modeling*, strony 145–152. IEEE Computer Society, 2001. [cytowanie na str. 67]
- [91] A. Salamończyk, W. Mokrzycki. Generation of View Representation from View Points on Spiral Trajectory. R. Choras, redaktor, *Image Processing and Communications Challenges 2*, wolumen 84 serii *Advances in Soft Computing*, strony 67–74. Springer Berlin / Heidelberg, 2010. [cytowanie na str. 50, 52]
- [92] A. Salamonczyk, W. Mokrzycki. New Method of Polyhedron Views and Range Images Matching Using Hausdorff Distance. R. Choraś, A. Zabłudowski, redaktorzy, *Problemy Współczesnej Nauki, Teoria i Zastosowania. Informatyka.*, strony 106–114. Exit, 2009. [cytowanie na str. 62]

- [93] R. D. Schiftenbauer. A Survey of Aspect Graphs. Raport instytutowy TR-CIS-2001-01, Department of Computer and Information Science, Polytechnic University Brooklyn, New York, 2001. [cytowanie na str. 22]
- [94] I. Shimshoni, J. Ponce. Finite-Resolution Aspect Graphs of Polyhedral Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:315–327, 1997. [cytowanie na str. 20, 21]
- [95] H.-Y. Shum, M. Hebert, K. Ikeuchi. On 3D Shape Similarity. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '96)*, strony 526 – 531, June 1996. [cytowanie na str. 15]
- [96] B. Siemiątkowska, (praca zbiorowa). *Reprezentacja otoczenia robota mobilnego*. Exit, Planowane wydanie 2011. [cytowanie na str. 39]
- [97] F. Stein, G. Medioni. Structural Indexing: Efficient 3-D Object Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:125–145, February 1992. [cytowanie na str. 66]
- [98] J. H. Stewman, K. W. Bowyer. Direct Construction of the Perspective Projection Aspect Graph of Convex Polyhedra. *Comput. Vision Graph. Image Process.*, 51:20–37, July 1990. [cytowanie na str. 21]
- [99] I. Stroud. *Boundary Representation Modelling Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. [cytowanie na str. 12]
- [100] H. Sundar, D. Silver, N. Gagvani, S. Dickinson. Skeleton Based Shape Matching and Retrieval. *Proceedings of the Shape Modeling International 2003*, strony 130–142, 290, Washington, DC, USA, 2003. IEEE Computer Society. [cytowanie na str. 17]
- [101] J. W. H. Tangelder, R. C. Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods. *Multimedia Tools and Applications*, wolumen 39(3), strony 441–447. Springer Netherlands, 2008. [cytowanie na str. 67]
- [102] M. J. Tarr, D. J. Kriegman. What Defines a View? *Vision Research*, 41(15):1981–2004, 2001. [cytowanie na str. 19, 21, 22, 79]
- [103] G. Toussaint. Movable Separability of Sets. *Computational Geometry*, strony 335–375. North-Holland, 1985. [cytowanie na str. 6]
- [104] G. Turk, M. Levoy. Zippered Polygon Meshes from range images. *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, strony 311–318, New York, NY, USA, 1994. ACM. [cytowanie na str. 67]
- [105] S. Ullman, R. Basri. Recognition by Linear Combination of Models. Raport instytutowy, Massachusetts Institute of Technology, Cambridge, MA, USA, 1989. [cytowanie na str. 22]
- [106] R. C. Veltkamp. Shape Matching: Similarity Measures and Algorithms. *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, strony 188–197, Washington, DC, USA, 2001. IEEE Computer Society. [cytowanie na str. 65]
- [107] Z. Xu, M. Suk. Representation and Reconstruction of Polygons and Polyhedra using Hierarchical Extended Gaussian Images. *Annals of Mathematics and Artificial Intelligence*, 13:377–399, 1995. [cytowanie na str. 15]

Spis rysunków

1.1	Sfera widokowa z perspektywą	4
1.2	Sfera podzielona na obszary jednowidokowe	5
1.3	Monotonność 1-rodzaju	6
2.1	Graf aspektowy z wyróżnionymi dwoma aspektami, [76]	19
2.2	Widoki wielokąta w 2W przestrzeni z rzutowaniem środkowym, [102]	19
2.3	Sfera widokowa - płaszczyzny widoczności i widoki	21
2.4	Koncepcja algorytmów iteracyjnych, [23]	25
2.5	Wyznaczanie obszaru jednowidokowego przez wianki ścian, [26]	26
2.6	Wyznaczanie obszaru jednowidokowego przez szpilki wianka widocznego i niewidocznego, [26]	26
2.7	Koncepcja płaszczyzny połowiącej, [25]	27
2.8	Ruch PW w kierunku najbliższego zdarzenia wizualnego, [58]	28
2.9	Przekraczanie granic obszaru jednowidokowego, [60]	29
2.10	Szukanie pierwszej i dalszych krawędzi obszaru zarodkowego, [60]	31
2.11	Ilustracja do wyznaczania cieniowania wklęsłości, [57]	34
2.12	Obrót wokół wektora normalnego	36
2.13	Rejestracja zdarzeń wizualnych podczas omiotu stożkiem dopełniającym, [36]	36
2.14	Skaning otoczenia ściany wielościanu stożkiem SD	39
3.1	Przykład zdarzeń kkk w wielościanie monotonnym	45
3.2	Przykładowa trajektoria skanowania	47
3.3	Wielokrotne skanowanie tego samego obszaru na sferze widokowej	48
3.4	Kolejne iteracje podziału ściany dwudziestościanu foremego	49
3.5	Tworzenie nowych punktów widokowych przez iteracyjny podział DF	49
3.6	Podział ścian DW z uwzględnieniem obszarów wypukłych	50
3.7	Zbiór PW w kolejnych iteracjach DW z uwzględnieniem obszarów wypukłych	50
3.8	Funkcja korekty rozkładu punktów na trajektorii	51
3.9	Spiralny rozkład punktów na sferze widokowej	51
3.10	Kąty pomiędzy kolejnymi punktami spiralnej trajektorii	53
3.11	Ilustracja do dowodu kompletności reprezentacji	54
3.12	Analiza zasłoneń przez rzutowanie na płaszczyznę	56
3.13	Bryły testowe i ich podstawy o 3, 6 i 9 krawędziach.	57
3.14	Porównanie liczby punktów widokowych	59
4.1	Widok wielościanu i mapa zasięgu w skali szarości dla tego widoku	63
4.2	Pasowanie punktów z mapy zasięgu z widokiem.	64
4.3	Odległość Hausdorffa	66

Wykaz ważniejszych oznaczeń

- α - połowa kąta rozwarcia stożka widokowego
- β - połowa kąta rozwarcia stożka dopełniającego
- σ - minimalna odległość do najbliższego punktu na sferze widokowej
- kkk - zdarzenia typu krawędź-krawędź-krawędź
- kw - zdarzenia typu krawędź-wierzchołek
- B_{rep} - powłokowa reprezentacja wielościanu
- N_{rep} - naturalna (ostroslupowa) reprezentacja wielościanu
- Np_{rep} - naturalna pełna reprezentacja wielościanu
- V_{rep} - reprezentacja wektorowa wielościanu

Wykaz ważniejszych skrótów

- MLP - minimalna liczba pasowań wystarczająca do identyfikacji
- OPWŚ - obszar potencjalnej widoczności ściany
- PW - punkt widokowy
- RG - rejestr graniczny
- SD - stożek dopełniający
- SW - stożek widokowy
- SWP - sfera widokowa z perspektywą
- SIW - system identyfikacji wizualnej
- ŚCW - ściana całkowicie widoczna
- ŚCzW - ściana częściowo widoczna

Dodatki

A. Algorytmy

Algorytm: punktyWidokoweNaSpiralnejTrajektorii(R, σ)

Dane: R -promień sfery widokowej, σ -żądana rozdzielczość

Wynik: *PunktyWidokowe* -zbiór punktów rozmieszczonych na sferze widokowej

$$k \leftarrow \frac{2\pi}{\sqrt{2}\sigma};$$

$$N \leftarrow \frac{4\pi + \sqrt{2}\sigma}{2(\sigma)^2};$$

$$\text{krok} \leftarrow \frac{2}{N};$$

for ($s \leftarrow 0; s \leq 2; s \leftarrow s + \text{krok}$) **do**

if $s \leq 1$ **then**

$t \leftarrow \arccos(1 - s);$

else

$t \leftarrow \pi - \arccos(s - 1);$

end

$\text{punkt}.x \leftarrow R \cos(kt) \sin(t);$

$\text{punkt}.y \leftarrow R \sin(kt) \sin(t);$

$\text{punkt}.z \leftarrow R \cos(t);$

 dodaj *punkt* do zbioru *PunktyWidokowe*;

end

return *PunktyWidokowe*;

Algorytm: punktyWidokoweDlaMetodySpiralnej(R, σ)

Dane: R -promień sfery widokowej, σ -żądana rozdzielczość

Wynik: *Widoki* -zbiór widoków uzyskanych ze sfery widokowej

$\text{PunktyWidokowe} \leftarrow \text{punktyWidokoweNaSpiralnejTrajektorii}(R, \sigma);$

foreach *punkt* **w** **zbiorze** *PunktyWidokowe* **do**

$\text{widok} \leftarrow \text{widokZPunktuWidokowego}(\text{punkt});$

if $\text{widok} \notin \text{Widoki}$ **then**

 dodaj *widok* do zbioru *Widoki*;

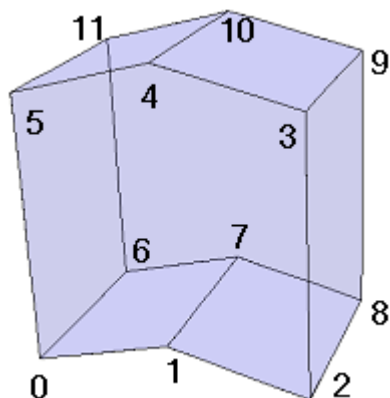
end

end

return *Widoki*;

B. Wyniki

Wielościan o 8 ścianach (46 widoków)



Wierzchołki:

Nr	Współrzędne
0	(-4.0, -4.0, 3.0)
1	(0.0, -3.0, 3.0)
2	(4.0, -4.0, 3.0)
3	(4.0, 4.0, 3.0)
4	(0.0, 5.0, 3.0)
5	(-4.0, 4.0, 3.0)
6	(-4.0, -4.0, -3.0)
7	(0.0, -3.0, -3.0)
8	(4.0, -4.0, -3.0)
9	(4.0, 4.0, -3.0)
10	(0.0, 5.0, -3.0)
11	(-4.0, 4.0, -3.0)

Ściany:

Nr	Wierzchołki
0	1, 0, 6, 7
1	2, 1, 7, 8
2	3, 2, 8, 9
3	4, 3, 9, 10
4	5, 4, 10, 11
5	0, 5, 11, 6
6	1, 2, 3, 4, 5, 0
7	6, 11, 10, 9, 8, 7

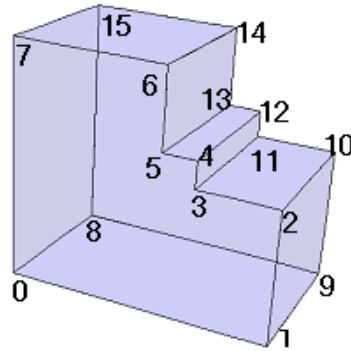
Widoki:

Nr	Ściany
0	6
1	1 6
2	0 6
3	0 1 6
4	5 6
5	2 6
6	1 5 6
7	0 2 6
8	3 6
9	4 6
10	3 4 6
11	0 1 2 6
12	2 3 6
13	4 5 6
14	0 1 5 6

Nr	Ściany
15	2 3 4 6
16	3 4 5 6
17	0 1 2
18	0 2
19	2
20	2 3
21	2 3 4
22	3 4
23	3 4 5
24	4 5
25	5
26	1 5
27	0 1 5
28	0 1
29	3 4 7

Nr	Ściany
30	3 4 5 7
31	4 5 7
32	5 7
33	1 5 7
34	0 1 5 7
35	0 1 7
36	0 1 2 7
37	0 2 7
38	2 7
39	2 3 7
40	2 3 4 7
41	4 7
42	3 7
43	7
44	1 7

Nr	Ściany
45	0 7

Wielościan monotony o 10 ścianach (66 widoków)

Wierzchołki:

Nr	Współrzędne
0	(-5.0, -4.0, 3.0)
1	(5.0, -4.0, 3.0)
2	(5.0, 1.0, 3.0)
3	(2.0, 1.0, 3.0)
4	(2.0, 2.0, 3.0)
5	(0.75, 2.0, 3.0)
6	(0.75, 5.0, 3.0)
7	(-5.0, 5.0, 3.0)
8	(-5.0, -4.0, -3.0)
9	(5.0, -4.0, -3.0)
10	(5.0, 1.0, -3.0)
11	(2.0, 1.0, -3.0)
12	(2.0, 2.0, -3.0)
13	(0.75, 2.0, -3.0)
14	(0.75, 5.0, -3.0)
15	(-5.0, 5.0, -3.0)

Ściany:

Nr	Wierzchołki
0	0,1,2,3,4,5,6,7
1	2,1,9,10
2	3,2,10,11
3	4,3,11,12
4	5,4,12,13
5	6,5,13,14
6	7,6,14,15
7	0,7,15,8
8	1,0,8,9
9	15,14,13,12,11,10,9

Widoki:

Nr	Ściany
0	0
1	0 5
2	0 2
3	0 2 5
4	0 2 4
5	0 3 5
6	0 2 4 5
7	0 2 3 5
8	0 2 3 4 5
9	0 8
10	0 5 8
11	0 3 5 8
12	0 1 3 5

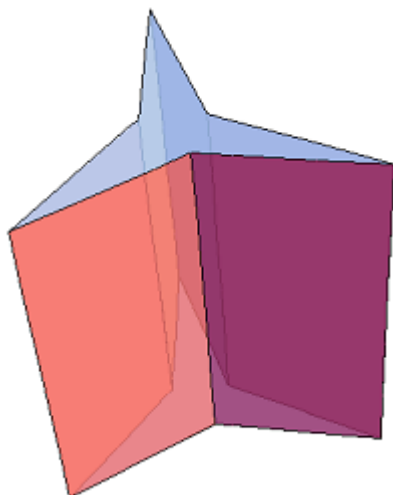
Nr	Ściany
13	0 2 4 6
14	0 7
15	0 2 4 5 6
16	0 2 7
17	0 1 2 3 5
18	0 2 4 7
19	0 1 2 3 4 5
20	0 2 3 4 5 6
21	0 7 8
22	0 1 3 5 8
23	0 2 4 6 7
24	0 1 2 3 4 5 6
25	2 3 4 5 6

Nr	Ściany
26	2 4 5 6
27	2 4 6
28	2 4 6 7
29	2 4 7
30	2 7
31	7
32	7 8
33	8
34	5 8
35	3 5 8
36	1 3 5 8
37	1 3 5
38	1 2 3 5

Nr	Ściany
39	1 2 3 4 5
40	1 2 3 4 5 6
41	1 3 5 8 9
42	1 3 5 9
43	1 2 3 5 9
44	1 2 3 4 5 9
45	1 2 3 4 5 6 9
46	2 3 4 5 6 9
47	2 4 5 6 9
48	2 4 6 9
49	2 4 6 7 9
50	2 4 7 9
51	2 7 9

Nr	Ściany
52	7 9
53	7 8 9
54	8 9
55	5 8 9
56	3 5 8 9
57	2 3 4 5 9
58	2 4 9
59	9
60	3 5 9
61	2 4 5 9
62	2 9
63	2 3 5 9
64	5 9
65	2 5 9

Graniastosłup testowy o 8 ścianach (56 widoków)



Wierzchołki:

Nr	Współrzędne
0	(8.0, 0.0, 8.0)
1	(1.0, 1.7320508, 8.0)
2	(-4.0, 6.928203, 8.0)
3	(-2.0, 0.0, 8.0)
4	(-4.0, -6.928203, 8.0)
5	(1.0, -1.7320508, 8.0)
6	(8.0, 0.0, -8.0)
7	(1.0, 1.7320508, -8.0)
8	(-4.0, 6.928203, -8.0)
9	(-2.0, 0.0, -8.0)
10	(-4.0, -6.928203, -8.0)
11	(1.0, -1.7320508, -8.0)

Ściany:

Nr	Wierzchołki
0	0, 1, 7, 6
1	1, 2, 8, 7
2	2, 3, 9, 8
3	3, 4, 10, 9
4	4, 5, 11, 10
5	5, 0, 6, 11
6	5, 4, 3, 2, 1, 0
7	6, 7, 8, 9, 10, 11

Widoki:

Nr	Ściany
0	6
1	4 6
2	1 6
3	0 6
4	0 3 6
5	3 6
6	2 6
7	2 5 6
8	5 6
9	1 4 6
10	2 3 6
11	4 5 6
12	0 1 6
13	0 1 3 6
14	0 1 2 3 6

Nr	Ściany
15	0 2 3 6
16	2 3 5 6
17	2 3 4 5 6
18	2 4 5 6
19	1 4 5 6
20	0 1 4 5 6
21	0 1 4 6
22	2 3 4 5
23	2 4 5
24	4 5
25	1 4 5
26	0 1 4 5
27	0 1 4
28	0 1
29	0 1 3

Nr	Ściany
30	0 1 2 3
31	0 2 3
32	2 3
33	2 3 5
34	1 4 5 7
35	0 1 4 5 7
36	0 1 4 7
37	0 1 7
38	0 1 3 7
39	0 1 2 3 7
40	0 2 3 7
41	2 3 7
42	2 3 5 7
43	2 3 4 5 7
44	2 4 5 7

Nr	Ściany
45	4 5 7
46	1 4 7
47	0 3 7
48	2 5 7
49	3 7
50	4 7
51	1 7
52	0 7
53	2 7
54	5 7
55	7