ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 35 Nr kol. 1402

Katarzyna STĄPOR
Politechnika Śląska, Instytut Informatyki

PRZEGLĄD PRZESTRZENNYCH METOD INDEKSACJI

Streszczenie. W artykule dokonano przeglądu istniejących, przestrzennych metod indeksacji stosowanych w celu realizacji zapytań w przestrzennych bazach danych. Opisane zostały struktury danych i podstawowe algorytmy wyszukiwania. Podano istniejące klasyfikacje tych metod według różnorodnych kryteriów.

THE REVIEW OF SPATIAL ACCESS METHODS

Summary. This article is a survey of spatial access methods used in spatial query processing in database systems. It describes underlying spatial structures of those methods and basic algorithms. The existing classification of spatial indexing methods based on different criteria is given.

1. Wstęp

W związku z intensywnym rozwojem geograficznych systemów informacyjnych (ang. Geographical Information System, GIS) pojawiło się ogromne zapotrzebowanie na tzw. przestrzenne systemy baz danych (ang. Spatial Database System, SDB), umożliwiające efektywne zarządzanie danymi przestrzennymi, czyli stosunkowo prostymi obiektami geometrycznymi, np. zbiorem punktów, linii, wielokątów, posiadających określoną lokalizację w przestrzeni, którą może być np. 2-wymiarowa abstrakcja części powierzchni Ziemi. Każdy obiekt przestrzenny opisany jest poprzez minimalnie jeden **atrybut przestrzenny**, tzn. atrybut opisujący jego geometrię i położenie w przestrzeni. Atrybut taki zawiera 2- lub 3-wymiarowe dane typu punkt, linia, prostokąt, wielokąt, powierzchnia. Przykładowo, dla obiektu las atrybutem przestrzennym może być atrybut granice, stanowiący reprezentację linii brzegowej lasu. Zbiór obiektów przestrzennych posiadających ten sam

zestaw atrybutów definiuje relację przestrzenną. Na przykład relacja lasy z atrybutem granice opisującym jej geometrię. **Zapytania przestrzenne** to zapytania dotyczące właśnie atrybutów przestrzennych obiektów. Zapytania przestrzenne można podzielić na dwie grupy:

a) **zapytania pojedynczego dostępu** (ang. single-scan), wymagające co najwyżej jednego dostępu do obiektu w trakcie wykonania zapytania, charakteryzujące się liniową zależnością czasu wykonania zapytania od ilości obiektów w relacji. Najbardziej znanymi przykładami takich zapytań są: 1) zapytanie **typu punkt** (ang. point query): mając dany punt P w d-wymiarowej przestrzeni znajdź wszystkie d-wymiarowe prostokąty R, zawierające P; 2) zapytanie **typu okno** (ang. window query): mając dany d-wymiarowy prostokąt S znajdź wszystkie d-wymiarowe prostokąty R przecinające (zawierające, należące do) S; 3) zapytanie **typu najbliższego sąsiada** (ang. nearest neighbour).

b) **zapytania wielokrotnego dostępu** (ang. multiplescan), wymagające wielokrotnego dostępu do obiektu w trakcie wykonania zapytania, charakteryzujące się superliniową zależnością czasu wykonania zapytania od ilości obiektów. Najbardziej znanym przykładem takiego zapytania jest **złączenie przestrzenne** (ang. **spatial join**). Operacja złączenia przestrzennego polega na kombinacji sterowanej pewnymi własnościami obiektów z dwu zbiorów. Przykładowo, w przypadku relacji „lasy" i „miasta", gdzie atrybut przestrzenny reprezentuje w każdej z nich brzegi (granice), przykładem złączenia przestrzennego jest zapytanie: „znajdź wszystkie lasy w mieście".

Rekord, stanowiący podstawę modelu relacyjnego w konwencjonalnych bazach danych może być uważany za punkt w przestrzeni wielowymiarowej. Aby wykorzystać podejście stosowane w konwencjonalnych bazach danych do danych przestrzennych, dokonuje się najczęściej parametryzacji w celu redukcji do takiego punktu wielowymiarowego. Realizuje się to poprzez dodanie pola opisującego interesujące własności przestrzenne obiektu do struktury rekordu. Pole to stanowi atrybut przestrzenny. Można wówczas wykorzystać do realizacji zapytań metody stosowane w konwencjonalnych, relacyjnych bazach danych. Problem powstaje jednak przy realizacji niektórych zapytań przestrzennych, w szczególności dotyczących zajmowanego przez obiekt miejsca (lokalizacji) oraz jego sąsiedztwa.

Aby móc efektywnie realizować zapytania przestrzenne, obiekty muszą zostać posortowane zgodnie z atrybutem wyrażającym ich położenie w przestrzeni. **Indeks przestrzenny** lub inaczej, **przestrzenna metoda dostępu** (ang. spatial access method, SAM), to indeks dla zbioru obiektów przestrzennych oparty na atrybutach przestrzennych, tzn opisujących lokalizację obiektu w przestrzeni.

Umożliwiający realizację wymienionych operacji przestrzennych, **przestrzenny system bazy danych** powinien mieć następujące własności:

a) Jest systemem bazy danych, tzn. ma cechy charakterystyczne dla klasycznego systemu bazy danych, a więc: możliwość składowania i manipulowania alfanumerycznymi (nieprzestrzennymi) danymi, zarządzanie współbieżnym dostępem do danych, odtwarzanie po awarii, bezpieczny dostęp do danych.

b) Ma wbudowane w swój model danych i język zapytań tzw. **przestrzenne typy danych** (ang. spatial data type, SDT), np. **POINT, LINE, REGION** wraz z przestrzennymi operacjami na tych typach, np. p INTERSECTS q, AREA(p) > x. Typy te stanowią podstawową abstrakcję struktury obiektów geometrycznych rozmieszczonych w przestrzeni, umożliwiając tym samym jej modelowanie. Operacje z kolei stanowią opis wzajemnych związków między obiektami oraz własności.

c) Ma możliwość efektywnej realizacji zapytań przestrzennych (np. selection, join), co jest równoznaczne z posiadaniem przynajmniej jednego indeksu dla przestrzennych typów danych, tzw. indeksu przestrzennego.

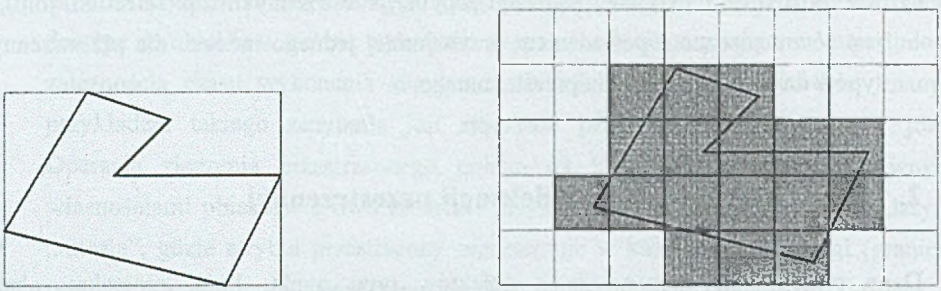
2. Istota i podział metod indeksacji przestrzennej

Dane przestrzenne złożone są z obiektów typu punkt, linia, prostokąt, obszar, powierzchnia, bryła. Aby wykorzystać metody stosowane w konwencjonalnych bazach danych do obiektów przestrzennych, dokonuje się ich przekształcenia do punktu wielowymiarowego (tzw. **punkt reprezentatywny**) w przestrzeni o tym samym [24], mniejszym [18] lub większym wymiarze [9]. Ta analogia nie zawsze jest jednak dobra dla danych przestrzennych. Wymiar przestrzeni jest najczęściej za wysoki. Jednym z rozwiązań tego problemu jest aproksymacja obiektu, tak aby zredukować wymiar punktu reprezentatywnego.

Z kolei stosowane przekształcenia do punktu wielowymiarowego najczęściej nie zachowują najbardziej istotnej z punktu widzenia przetwarzania zapytań przestrzennych **własności bliskości**. Powstaje problem przy realizacji zapytań przestrzennych dotyczących zajmowanego przez obiekt miejsca (lokalizacji) oraz jego sąsiedztwa. W przypadku takich zapytań ich realizacja musi odbywać się na podstawie własności przestrzennych, które nie są zapisane w bazie w sposób jawny. Na przykład w bazie danych o drogach nie są zapisane wszystkie punkty przecięcia dróg, gdyż koszt takiego wstępnego przetwarzania byłby zbyt wysoki. Bardziej opłacalne jest obliczenie ich „w locie" (reprezentacja niejawna), na co

potrzebna jest jednak odpowiednia reprezentacja danych związana z zajmowaną przez nie przestrzenią, tak aby nie trzeba było znać typów zapytań z góry.

Aby można było przetwarzać zapytania przestrzenne dotyczące lokalizacji obiektów przestrzennych efektywnie, dane muszą zostać posortowane według kluczy wyrażających własności przestrzenne obiektu - a więc miejsce zajmowane przez obiekty. Takie techniki indeksacji są nazywane **indeksacjami przestrzennymi**. W przypadku takich metod rerezentacja danych przestrzennych odbywa się na podstawie miejsca zajmowanego przez te obiekty (ang. spatial occupancy). Podstawą takich metod indeksacji przestrzennej jest dekompozycja przestrzeni, w której narysowane są obiekty przestrzenne (np. przestrzeni 2D zawierającej segmenty linii) na obszary zwane komorami (ang.bucket), stąd inna nazwa tych metod - **metody komorowe** (ang. bucket methods).



Rys.1. Aproksymacje obiektu : MBR (po lewej) oraz siatkowa (po prawej)
Fig.1. MBR (left) and grid (right) aproximations of spatial object

W metodach indeksacji przestrzennej dokonuje się przybliżania (aproksymacji) obiektów, tak aby uzyskać jako indeks przestrzenny znacznie prostsze geometrycznie obiekty w stosunku do rzeczywistych obiektów przestrzennych. Aproksymacja umożliwia znaczną oszczędność zapisu złożonych obiektów, a podstawowe charakterystyki obiektu: położenie oraz rozciągłość obiektu wzdłuż wszystkich osi zostają zachowane. Stosowane są dwa rodzaje takiej aproksymacji obiektów:

a) **ciągła**, której przykładem jest minimalny prostokąt ograniczający (ang. Minimum Bounding Rectangle, MBR), tzn. najmniejszy zawierający dany obiekt prostokąt, o osiach równoległych do osi układu współrzędnych. D-wymiarowe obiekty przestrzenne są aproksymowane d-wymiarowym sześcianem. Rys. 1 pokazuje to dla

przykładu przestrzeni 2-wymiarowej: obiekty-wielokąty i odpowiadające im aproksymacje.

b) **siatkowa**, w której przestrzeń jest podzielona za pomocą regularnej siatki na komórki, a obiekt jest reprezentowany poprzez zbiór komórek, które przecina (rys.1).

W literaturze spotyka się wiele różnych klasyfikacji przestrzennych metod dostępu uwzględniających różnorodne kryteria. Jeśli chodzi na przykład o rodzaj zastosowanych do indeksacji struktur danych, metody można podzielić na dwie grupy:

a) metody wykorzystujące **dedykowaną zewnętrzną strukturą danych** (dla punktów, prostokątów)

b) metody **odwzorowujące obiekty przestrzenne w 1-wymiarową przestrzeń** tak, że możliwe jest użycie standardowej struktury indeksu, np. B-drzewa.

Większość powstałych klasyfikacji ujmuje tylko część metod indeksacji. Podana poniżej klasyfikacja ujmuje większość istniejących metod w sposób jednolity. Opiera się ona na danych przestrzennych nietransformowanych, tzn. w przestrzeni, w której obiekty zostały narysowane i dokonuje dekompozycji takiej przestrzeni na cztery sposoby dając tym samym cztery grupy metod indeksacji przestrzennej, które zostaną krótko scharakteryzowane.

W **pierwszej grupie** metod wykorzystując aproksymacje MBR obiektów przestrzennych dokonuje się grupowania ich w hierarchie, które zostają następnie zapisane w innej strukturze, np. w B-drzewie. Zachowana zostaje własność bliskości położenia. Przykładem jest np. R-drzewo [5]. Podział przestrzeni nie jest jednak rozłączny, co wymaga nieraz przeszukania całej bazy, jeśli obiekt zawarty jest w kilku aproksymacjach. W **drugiej grupie** metod dokonuje się rozłącznej dekompozycji przestrzeni na komórki (ang. cell) i odwzorowuje je w komory (ang. bucket), przy czym można zastosować różne zasady regularności siatki podziału dla przestrzeni oraz różne sposoby ich agregacji. Obiekt zostaje przy tym zdekomponowany na rozłączne podobiekty, a następnie grupuje się tworzące go podobiekty w strukturę typu B-drzewo. Aproksymacje obiektów (np. MBR) są rozłączne na każdym poziomie struktury. Aby określić obszar pokryty przez jakiś obiekt, należy przeszukać wszystkie komórki, które on zawiera. To samo w przypadku usuwania obiektu. Przykładem jest R+-drzewo [23], cell tree [4]. W **trzeciej grupie** metod dokonuje się rozłącznej dekompozycji przestrzeni na regularne bloki o jednakowym rozmiarze. Metoda nadaje się dla danych rozmieszczonych równomiernie. **Czwartą grupę** charakteryzuje adaptacyjna dekompozycja przestrzeni w zależności od danych. Przykładem może być drzewo czwórkowe (ang. quadtree [22]). Rozmiar bloku jest potęgą liczby 2, a ich pozycje są również ograniczone. Metodę można stosować dla danych o dowolnym rozkładzie. W

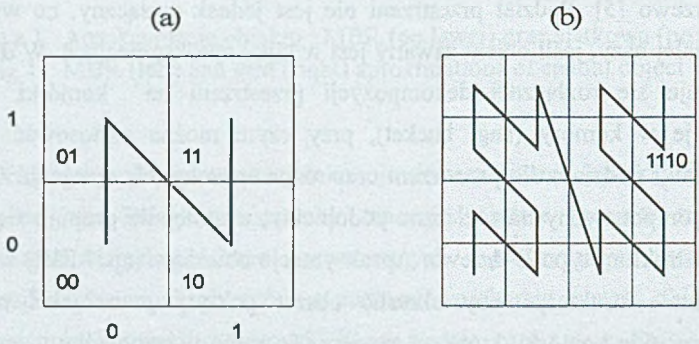
przypadku danych równomiernie rozłożonych degeneruje się ona do jednolitej siatki (grupa 3).

Metody grupy 1 i 2 są zależne od danych, a więc pojawiają się trudności przy wykonywaniu operacji wymagających złożenia różnych operacji i zbiorów. Metody grupy 3 i 4 są niezależne od danych i umożliwiają składanie wielu różnych operacji na różnych zbiorach danych. Najbardziej elastyczna jest metoda 4, gdyż dane przestrzenne najczęściej nie są regularnie rozmieszczone.

3. Budowa indeksu przestrzennego poprzez odwzorowanie obiektów w przestrzeń 1-wymiarową

Zbudowanie takiego indeksu przestrzennego polega na:

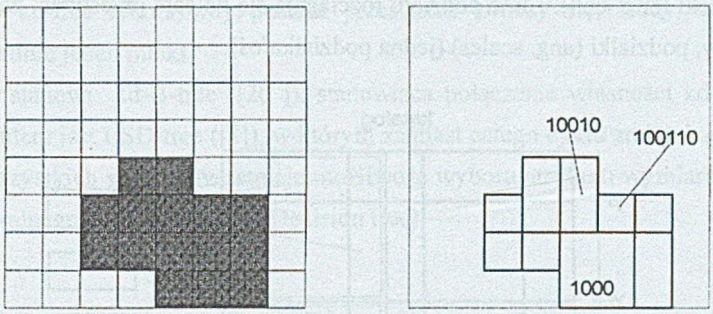
a) znalezieniu takiego liniowego uporządkowania dla komórek siatki, aby komórki leżące blisko siebie w przestrzeni pozostawały również blisko siebie w tym uporządkowaniu;
b) rekursywnej definicji takiego uporządkowania komórek, tak aby możliwe było uzyskanie siatki - podziału przestrzeni poprzez hierarchiczny jej podział.



Rys.2. Z-uporządkowanie komórek w podziale hierarchicznym
Fig.2. Z-order enumeration of cells of a hierarchical partition

Rys.2 pokazuje jedno z bardzo popularnych takich uporządkowań, zwane **przetasowaniem bitów** (ang. bit interleaving), któremu później nadano nazwę **z-uporządkowania** [17]. Na rys. 2a) komórki są z-uporządkowane, zaś na rys. 2b) pokazano kontynuację tego z-porządku na następnym poziomie. Na każdym poziomie komórka uzyskuje indeks będący łańcuchem o długości odpowiadającej poziomowi przynależności:

np. górna prawa komórka na rys. 2a) posiada indeks 11, zaś na rys. 2b) indeks 1110 uzyskiwany jest poprzez złożenie 11 (ojciec, współrzędna x) zajmujący bit 1 i 3 wraz z 10 (współrzędna y), zajmujący bit 2 i 4, co nazywa się przetasowaniem bitów. Tak uzyskany porządek stanowi porządek leksykograficzny łańcuchów bitów.



Rys. 3. Zbiór z-elementów aproksymujących wartość przestrzennego obiektu
Fig. 3. A set of z-elements approximating a value of spatial object

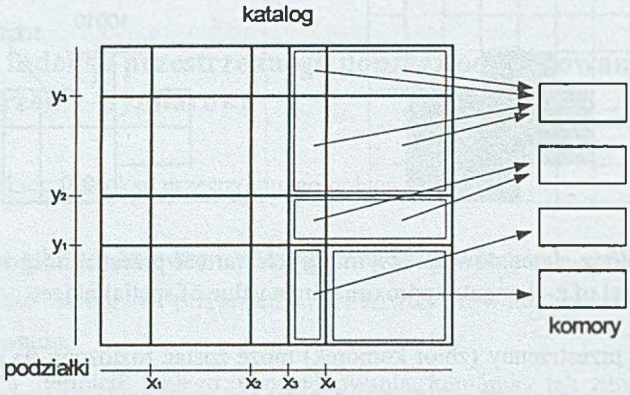
Każdy obiekt przestrzenny (zbiór komórek) może zostać rozłożony na minimalną liczbę komórek na różnych poziomach, używając poziomu możliwie najwyższego. Każda taka komórka zwana z-elementem posiada etykietę stanowiącą łańcuch bitów, zaś cały obiekt jest reprezentowany przez zbiór z-elementów, które mogą służyć następnie jako klucze przestrzenne. Aby zbudować indeks dla zbioru obiektów, należy dokonać leksykograficznego uporządkowania zbioru wszystkich takich kluczy przestrzennych w B-drzewo, co da w efekcie bardziej efektywne przetwarzanie zapytań. Na przykład dla pytania z zakresem (wyrażonego za pomocą prostokąta r reprezentowanego poprzez zbiór swoich z-elementów) przeszukiwana będzie tylko część B-drzewa, tzn. te liście, które zawierają jako prefiks dany z-element, co da w rezultacie zbiór obiektów-kandydatów, które w następnym kroku należy sprawdzić opierając się na ich dokładnej geometrii.

4. Dedykowana struktura indeksu przestrzennego dla punktu wielowymiarowego

Ze względu na reprezentację rekordu w postaci punktu k-wymiarowego struktury danych dla reprezentacji punktów k-wymiarowych zostały opracowane na długo przed pojawieniem się pierwszych przestrzennych systemów baz danych. Indeksy dla zapytań

wieloatrybutowych w tradycyjnych bazach danych mogą zostać wykorzystane jako indeksy w przestrzennej bazie danych. Dwie najbardziej znane takie struktury to grid file [10, 16] oraz kd-tree [1].

Struktura typu **grid file** (rys. 4) dokonuje podziału przestrzeni za pomocą nieregularnej sieci na komórki (ang. cell). Linie podziału rozciągają się na całą przestrzeń, zaś pozycje ich położeń w tzw. podziałki (ang. scales) (jedna podziałka/oś).



Rys. 4. Struktura typu grid file
Fig. 4. Structure of the grid file

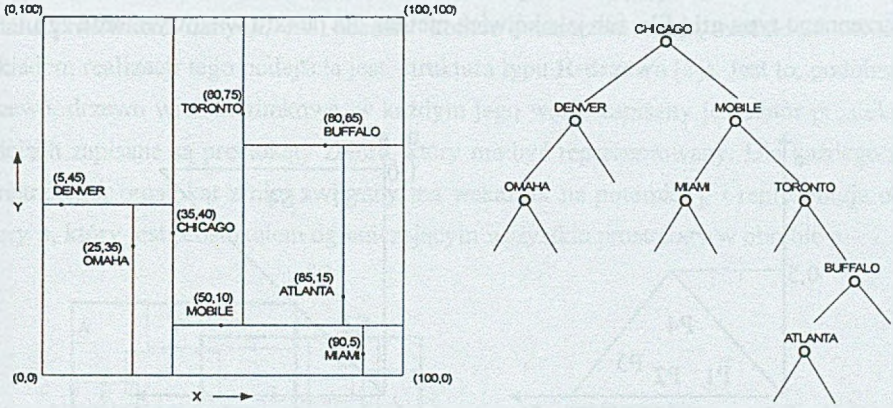
Katalog stanowi k-wymiarową tablicę, której elementy stanowią logiczne wskaźniki do komór (ang. bucket). Każda komórka przestrzeni danych odpowiada jednemu elementowi katalogu, wszystkie punkty leżące w obrębie komórki są zapisane w komorze wskazywanej przez odpowiadającą komórkę katalogu (rys. 4). Wiele komórek może być odwzorowanych w tę samą komorę, w ogólnym przypadku komory składają się z więcej niż jednej komórki. Podziałki są zapisane w pamięci operacyjnej, katalog w zbiorze stron dyskowych.

Aby znaleźć komorę zawierającą jakiś wyróżniony punkt, należy określić za pomocą podziałki adres strony zawierającej wejście w katalogu dla komórki zawierającej ten punkt. Drugi dostęp do dysku powoduje już uzyskanie komory. Pytania z zakresem mogą być realizowane poprzez określenie na podstawie katalogu zbioru komór zawierających komórki przecinane przez prostokąt zapytań i następne badanie punktów w tych komorach.

Struktura typu **kd-tree**, gdzie k oznacza rozmiar przestrzeni, stanowi strukturę typu drzewo binarne, którego każdy węzeł wewnętrzny zawiera klucz jednego z k-wymiarów. Klucz korzenia (poziom 0) dzieli przestrzeń ze względu na atrybut wymiaru 0, klucze w

węzłach stanowiących jego potomków (poziom 1) dzielą dwie podprzestrzenie ze względu na atrybut wymiaru 1, i tak dalej, aż do wymiaru k-1, po czym następuje nowy cykl kolejno po atrybutach reprezentujących wszystkie wymiary: 0 - (k-1). Rys. 5 pokazuje kd-tree dla przypadku 2-wymiarowego, wówczas porównuje się wartości współrzędnej x w korzeniu i parzystych węzłach, zaś wartości współrzędnej y w węzłach nieparzystych. W przypadku oryginalnego kd-tree rekursywny podział przestrzeni kończy się, kiedy każda komórka zawiera dokładnie jeden punkt.

Odmianę stanowi kd-B-tree ([20]), stanowiąca połączenie własności kd-tree i B-tree. Innym wariantem jest LSD-tree ([8]), w którym zamiast całego cyklu zmian badania wartości atrybutów wszystkich wymiarów istnieje możliwość wyboru atrybutu wymiaru na podstawie kryterium lokalnego (ang. Local Split Decision tree).



Rys. 5. Struktura typu k-d-tree
Fig. 5. Structure of k-d-tree

5. Dedykowana struktura indeksu przestrzennego dla prostokąta

5.1. Metody budowy indeksu

Zarządzanie strukturami danych dla prostokąta jest trudniejsze niż dla punktów, gdyż w ogólnym przypadku prostokąty nie mieszczą się w obrębie jednej komórki podziału przestrzeni danych, lecz przecinają granice podziału.

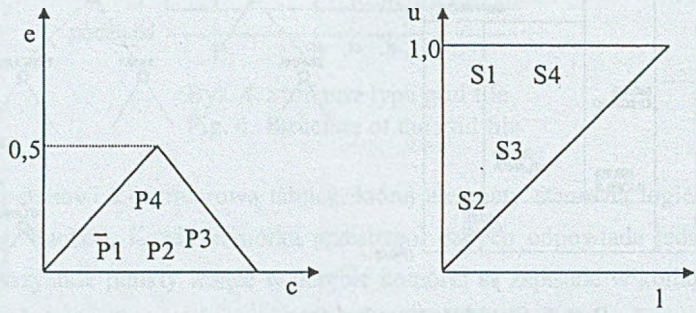
Zasadniczo wyróżnia się trzy podejścia do budowy indeksu przestrzennego opartego na aproksymacjach obiektów przestrzennych za pomocą minimalnych prostokątów ograniczających (MBR):

a) podejście oparte na transformacji danych,

b) podejście z nakładającymi się obszarami,

c) podejście z obcinaniem.

Podejście oparte na **transformacji danych** (ang. transformation approach) polega na reprezentacji minimalnych prostokątów ograniczających obiekty przestrzenne poprzez punkty w przestrzeni o wyższym wymiarze, np. 2-wymiarowego prostokąta o osiach równoległych do osi układu współrzędnych za pomocą punktu 4-wymiarowego (**reprezentacja środkowa**): (c1, c2, e1, e2), gdzie (c1, c2) stanowią współrzędne środka prostokąta, zaś (e1, e2) odległości środka prostokąta od jego boków. Do takiego przekształconego zbioru prostokątów w postaci punktów 4-wymiarowych można zastosować np. strukturę indeksu przestrzennego typu grid file lub jakąkolwiek metodę dla punktu wielowymiarowego.



Rys. 6. Reprezentacja środkowa oraz brzegowa 1-wymiarowego prostokąta
Fig. 6 Central and corner representation of 1-dimensional rectangle

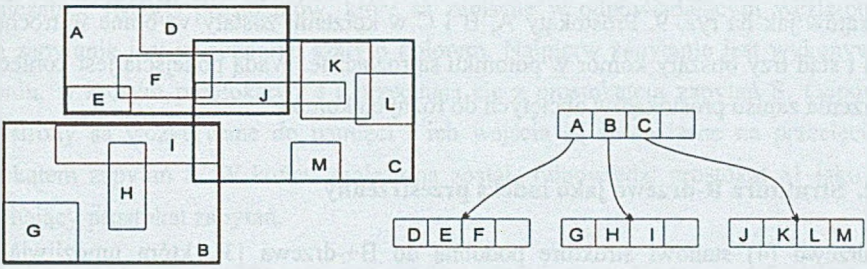
Innym sposobem reprezentacji jest reprezentacja 2-wymiarowego prostokąta poprzez jego dolny lewy (l1, l2) oraz prawy górny (u1, u2) wierzchołek (**reprezentacja brzegowa**). Sposób reprezentacji wpływa na wydajność przestrzennej metody dostępu. Podstawową zaletą reprezentacji środkowej jest oddzielenie reprezentacji położenia (środka) od reprezentacji rozciągłości. Ponadto dla środka prostokąta odległość do wszystkich punktów prostokąta jest zminimalizowana.

W celu ilustracji rozpatrzmy 1-wymiarowe prostokąty (segmenty). Segment ten może być reprezentowany w postaci środkowej (c, e), lub brzegowej (l, u). Większość przestrzennych metod dostępu dla punktu wielowymiarowego zakłada kwadratową przestrzeń <0, 1>. Wydajność realizacji zapytania maleje w przypadku niewykorzystania przestrzeni przez dane,

co zachodzi np. dla reprezentacji brzegowej 1-wymiarowych prostokątów. Dane występują tylko powyżej przekątnej (gdyż zawsze l<u), a jeśli przedziały są małe, to blisko przekątnej. Zaprojektowana w tym celu struktura LSD ma możliwość adaptacji do takiej pochyłej dystrybucji danych. Z kolei w przypadku reprezentacji środkowej, dane są zlokalizowane w przestrzeni w kształcie trójkąta (rys. 6), którego wykorzystanie najczęście redukuje się do podzbioru - trapezu. Podstawowy problem polega na odpowiedniej redukcji przestrzeni martwej, aby zwiększyć efektywność realizacji zapytania.

Zaprojektowana w tym celu struktura LSD ma możliwość adaptacji do takiej pochyłej dystrybucji danych. Z kolei w przypadku reprezentacji środkowej, dane są zlokalizowane w przestrzeni w kształcie trójkąta (rys. 6), którego wykorzystanie najczęście redukuje się do podzbioru - trapezu. Podstawowy problem polega na odpowiedniej redukcji przestrzeni martwej, aby zwiększyć efektywność realizacji zapytania.

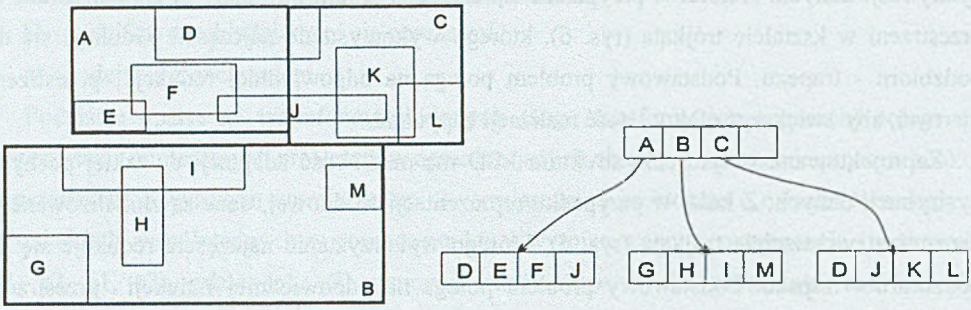
W podejściu **z nakładającymi się obszarami** (ang. overlapping regions) nie dokonuje się podziału przestrzeni danych, zaś obszary komór przestrzeni mogą zachodzić na siebie. Przykładem realizacji tego podejścia jest struktura typu **R-drzewo** [5]). Jest to, podobnie jak B-drzewo, drzewo wielokierunkowe, w każdym jego węźle zapisany jest zbiór prostokątów. W liściach zapisane są prostokąty zbioru, który ma być reprezentowany. Dla każdego węzła wewnętrznego, prostokąt z nim związany ma wskaźnik na potomka p i reprezentuje obszar komory p, który jest prostokątem ograniczającym wszystkie prostokąty w obrębie p.



Rys. 7. Zbiór prostokątów oraz jego reprezentacja w postaci R-drzewa
Fig. 7. A set of rectangles and its R-tree representation

Na rys. 7 pokazany został przykład R-drzewa, korzeń zawiera prostokąt A, który jest prostokątem ograniczającym prostokąty D, E i F zapisane w potomku związanym z węzłem A. Prostokąty mogą zachodzić na siebie. Prostokąt może przecinać wiele obszarów komór, ale będzie reprezentowany tylko w jednym z nich. Problemem staje się jednak wyszukiwanie,

które musi odbywać się wieloma ścieżkami, w przypadkach gdy interesujący obszar leży na przecięciu się dwóch obszarów potomków.



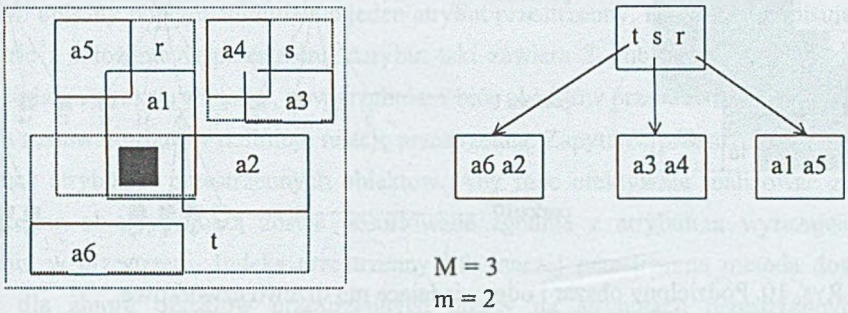
Rys. 8. Zbiór prostokątów oraz jego reprezentacja w postaci R+-drzewa
Fig. 8. A set of rectangles and its R+-tree representation

Podejście z **obcinaniem** (ang. clipping), w którym dokonuje się rozłącznego podziału przestrzeni danych. Jeśli prostokąt przecina granice podziału, to zostaje przycięty na kilka części i reprezentowany za pomocą wszystkich komórek, które przecina. Przykładem może być struktura typu **R+-drzewo** [23], w którym unika się nakładania się obszarów związanych z komorami lub wewnętrznymi węzłami tego samego poziomu poprzez obcinanie prostokątów, jeśli jest to konieczne. Na rys. 8 pokazano R+-drzewo dla tego samego zbioru prostokątów jak na rys. 9. Prostokąty A, B i C w korzeniu zostały wybrane w trochę inny sposób i stąd trzy obszary komór w potomku są rozłączne. Wadą podejścia jest konieczność powtórzenia zapisu prostokątów obciętych do różnych komór.

5.2. Struktura R-drzewo jako indeks przestrzenny

R-drzewo [4] stanowi strukturę podobną do B+-drzewa [3], która umożliwia zapis wielowymiarowych prostokątów stanowiących aproksymacje obiektów (Mbr). Minimalne prostokąty ograniczające obiekty przestrzenne są zapisane w liściach R-drzewa, natomiast węzły pośrednie powstają poprzez zgrupowanie węzłów na niższym poziomie. Rys.4 pokazuje przykład zbioru danych zorganizowanego w R-drzewo. Węzeł pośredni R-drzewa zawiera elementy, z których każde ma postać: (p, mbr), gdzie p stanowi adres węzła stanowiącego węzeł-potomek danego węzła w tym drzewie, mbr zaś stanowi prostokąt będący minimalnym prostokątem ograniczającym dla wszystkich prostokątów stanowiących wejścia w węźle-potomku wskazywanym przez p. Węzły stanowiące liście drzewa mają

wejścia o postaci: (oid, mbr), gdzie oid wskazuje na rekord w bazie danych opisujący dany obiekt przestrzenny, zaś mbr stanowi jego minimalny prostokąt ograniczający.

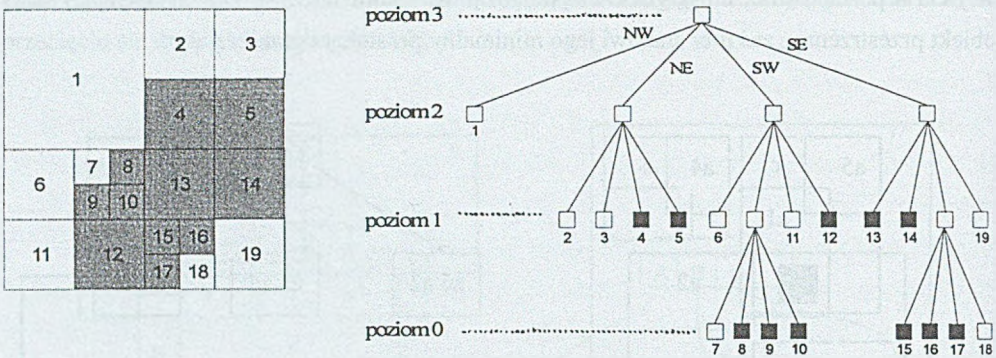


Rys. 9. Przykład R-drzewa z zaznaczonym oknem zapytania
Fig. 9. An example of R-tree with a query window

Realizacja zapytania typu okno (z oknem S) rozpoczyna się w korzeniu drzewa, gdzie obliczane są wszystkie wejścia, które przecinają się z prostokątem S. Dla tych wejść pobierane są do pamięci operacyjnej węzły potomne i realizacja zapytania (jeśli węzeł nie jest liściem) przebiega dalej jak dla korzenia. Rys. 4 pokazuje R-drzewo składające się z trzech stron danych i jednej strony katalogu (jeden węzeł drzewa odpowiada jednej stronie w pamięci zewnętrznej). Prostokąty strony katalogu stanowią minimalne prostokąty ograniczające (MBR) prostokątów, które są zapisane w odpowiadającym węźle-potomku. Okno zapytania jest wyróżnione szarym kolorem. Najpierw zapytanie jest wykonywane na korzeniu, w którym prostokąty t i r przecinają się z prostokątem zapytań S. Odpowiednie dwie strony są wczytywane do pamięci i ich wejścia są sprawdzane na przecięcie się z prostokątem zapytań S. W końcu znaleziona zostaje odpowiedź: prostokąt a1 jako jedyny przecinający prostokąt zapytań.

6. Drzewo czwórkowe

Struktura zwana drzewem czwórkowym (ang. quadtree) jest hierarchiczną strukturą danych, opartą na rekursywnej dekompozycji przestrzeni, w której zapisane są obiekty. Jest blisko związana ze strukturą typu grid file. Drzewa te mogą reprezentować dane typu punkt, linia, obszar, powierzchnia, objętość. Istnieje bardzo wiele odmian drzew czwórkowych. Najbardziej popularne to **„region quadtree"**, w którym dokonuje się podziału przestrzeni danych na prostokąty, których długość boku jest potęgą 2 ([22]).



Rys. 10. Podzielony obszar i odpowiadające mu drzewo czwórkowe
Fig. 10. A region with its maximal blocks and corresponding quadtree

Klasyfikacji drzew czwórkowych można dokonać opierając się na typach danych, jakie reprezentują, regule rządzącej procesem dekompozycji oraz rozdzielczości. Dekompozycja przestrzeni danych może być regularna (podział na równe części na każdym poziomie) lub sterowana poprzez dane wejściowe. Rozdzielczość (tzn. ilość poziomów dekompozycji) może również być stała lub sterowana wejściem.

Najbardziej popularne wśród drzew czwórkowych, **region quadtree**, opiera się na sukcesywnym podziale tablicy obrazu na 4 równe ćwiartki. Jeśli taka ćwiartka nie składa się całkowicie z 0 lub 1 (nie jest całkowicie czarna lub biała, albo inaczej, całkowicie zawarta w obszarze lub rozłączna z nim), to jest dzielona dalej na 4 równe ćwiartki, dopóki wszystkie bloki całkowicie są białe (0) lub czarne (1). Rys.10 ilustruje proces podziału oraz odpowiadające mu drzewo czwórkowe. Korzeń reprezentuje cały obraz, każdy potomek danego węzła reprezentuje ćwiartkę (znakowany NE, NW, SW, SE) obszaru reprezentowanego przez ten węzeł. Liście odpowiadają blokom, dla których dalszy podział nie jest już konieczny. Jest to struktura o zmiennej rozdzielczości. Innym sposobem jest wykorzystanie dekompozycji nieregularnej, co daje mniejszą zajętość pamięci, jednakże określenie optymalnych punktów podziału jest dosyć złożone obliczeniowo.

7. Podsumowanie

Każdy obiekt przestrzenny, czyli obiekt posiadający ustaloną lokalizację w przestrzeni musi być opisany poprzez minimalnie jeden atrybut przestrzenny, tzn. atrybut opisujący jego geometrię i położenie w przestrzeni. Atrybut taki zawiera 2- lub 3-wymiarowe dane typu punkt, linia, prostokąt, wielokąt, powierzchnia. Zbiór obiektów przestrzennych posiadających ten sam zestaw atrybutów definiuje relację przestrzenną. Zapytania przestrzenne to zapytania dotyczące atrybutów przestrzennych obiektów. Aby móc efektywnie realizować zapytania przestrzenne obiekty muszą zostać posortowane zgodnie z atrybutem wyrażającym ich położenie w przestrzeni. Indeks przestrzenny lub inaczej przestrzenna metoda dostępu to indeks dla zbioru obiektów przestrzennych oparty na atrybutach przestrzennych, tzn. opisujących lokalizację obiektu w przestrzeni.

W pracy dokonano przeglądu istniejących przestrzennych metod dostępu wybierając najbardziej charakterystycznych „przedstawicieli" z każdej grupy istniejących metod.

W literaturze spotyka się wiele różnych klasyfikacji przestrzennych metod dostępu uwzględniających różnorodne kryteria. Jeśli chodzi o rodzaj zastosowanych do indeksacji struktur danych metody można podzielić na dwie grupy: metody wykorzystujące dedykowaną strukturę danych oraz metody odwzorowujące obiekty przestrzenne w 1-wymiarową przestrzeń i wykorzystujące znaną strukturę indeksu dla danych nieprzestrzennych, np. B-tree. Większość znanych klasyfikacji nie ujmuje jednakże wszystkich metod dostępu. Podana w pracy klasyfikacja ujmuje większość istniejących metod w sposób jednolity. Dokonuje ona podziału metod na cztery grupy na podstawie danych nietransformowanych, tzn. w przestrzeni, w której obiekty zostały narysowane.

LITERATURA

1. Bentley J.L.: Multidimensional binary search trees used for associative searching. CACM, 18(9), September, 1975.
2. Faloutsos C., Roseman S.: Fractals for secondary key retrieval. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March, 1989.
3. Greene D.: An implementation and performance analysis of spatial data access methods. Proc. 5th Int. Conf. on Data Engineering, Los Angeles 1989.
4. Gunther O.: The cell tree: an index for geometric data. Memorandum No. UCB/ERL M86/89, Univ.of California, May, 1986.

5. Guttman A.: R-trees: a dynamic index structure for spatial searching. ACM SIGMOD, June, 1984.
6. Güting R.H.: An introduction to Spatial Database Systems. Special issue of VLDB, Journal, No 4, October, 1994.
7. Haas L., Freytag I., Lockmann G.: Extensible query processing in Starburst. Proc. ACM SIGMOD, 1989.
8. Henrich L., Six W.: The LSD-tree spatial access to multidimensional point- and non-point objects. Proc. 15th Int. Conf. on VLDB, Amsterdam, 1989.
9. Hinrichs K., Nievergelt J.: The grid file: a data structure to support proximity queries on spatial objects. Int. Workshop on Graph Theoretic Concepts in Computer Science, Linz, Austria, 1983.
10. Hinrichs K.: The grid file systems: Implementation and case studies of applications. Phd Thesis, ETH Zurrich, 1985.
11. Jagadish H.V.: Spatial search with polyhedra. IEEE Int. Conf. on Data Engineering, February, 1990.
12. Seeger B., Kriegel H.: Techniques for design and implementation of efficient spatial access methods. Proc. of 4 int. Conf. of VLDB, 1988, Los Angeles.
13. Lomet D., Salzberg B.: The hb-tree: a multiattribute indexing method with good guaranteed performance. ACM TODS, 15(4) December, 1990.
14. Morton G.: A computer oriented geodetic database and a new technique in file sequencing, IBM Ottawa, Canada, 1966.
15. Morehouse S.: The architecture of Arc/Info. Proc. Auto-Carto 9, Baltimore, 1989.
16. Nievergelt J., Hinterberger H., Sevcik K.: The grid file: an adaptable symmetric multikey file structure. ACM Trans. on Database Systems, 1984, (9).
17. Orenstein J.: Spatial query processing in an object-oriented database system. ACM SIGMOD, May, 1986.
18. Orenstein J., Merett T.: A class of data structures for associative searching. Proc. of the 3rd ACM SIGACT SIGMOD Symp. on Principles of Database Systems, Waterloo, Canada, April, 1984.
19. Orenstein J., Manola F.: PROBE spatial data modeling and query processing in an image database application. IEEE Tran. on Software Engineering, 14/1988.
20. Robinson J.T.: The k-d-b-tree: a search structure for large multidimensional dynamic indexes. ACM SIGMOD, 1981.
21. Roussopoulos N., Leifker D.: Direct spatial search on pictorial databases using packed r-trees. ACM SIGMOD, May, 1985.

22. Samet H.: The Quad-tree and related hierarchical data structures. ACM Computing Surveys, 1984, 16(2).
23. Sellis T., Roussopoulos N., Faloutsos C.: The R^+-tree: a dynamic index for multi-dimensional objects. Proceedings of the 13th VLDB Conference, September, 1987.
24. Jagadish H.V: On indexing line segments. Proc. 16th Int. Conf. VLDB, 1990, 614-625.

Recenzent: Dr inż. Andrzej Krupiczka

Wpłynęło do redakcji: 20 listopada 1997 r.

Abstract

This article is a survey of spatial access methods used in spatial query processing in database systems. It describes underlying spatial structures of those methods and the basic algorithms for data retrieving. The existing classification of spatial indexing based on diffrent criteria is also given. Three basic techniques for design and implementation of efficient spatial access methods: clipping, overlapping regions and transformation are described.

Based on the sort of data structure methods are divided into to groups: methods with dedicated external data structure for points (grid file (rys.4), kd-tree(rys.5)) and rectangles (R-tree(rys.7); methods mapping multidimensional objects into 1-dimensional space with standard indeks structures (B-tree). The other classification based on spatial occupancy divides methods into four groups: methods using minimum bounding rectangles as approximation of objects and classical index structures like B-tree (R-tree), methods based on disjoint decomposition of space using diffrent approximation (R+-tree (rys.8)), methods based on regular and irregular decomposition of object space into cells (quadtree (rys.10)).