Ci son più cose in cielo e in terra, Orazio...

Oggi c'è talmente tanta informazione intorno a noi che è difficile organizzarla e fruirne in modo razionale. Serve un approccio innovativo al problema. Quale? Lo chiediamo a Hanan Samet, in visita al CRS4.

Qualche tempo fa stavo sorseggiando un caffè al bar del CRS4 con un amico. "Quella è una nuova collega, un'esperta di database", osservai, indicando una persona. Sfoggiando il suo consueto humor inglese, il mio amico ribatté: "Un database è una tabella di righe e colonne. *Come si può essere esperti di una tabella?*". Il mio amico aveva in mente un modello *relazionale* di database[1]. È un'idea che risale agli anni '70 del secolo scorso - poco dopo la comparsa dei primi computer "moderni" - la cui importanza risiede, ovviamente, non nella tabella in sè quanto nella possibilità di effettuare operazioni sui contenuti, da quelle elementari di inserimento, lettura, aggiornamento, cancellazione (CRUD) a quelle più sofisticate che includono la ricerca (SCRUD).

La società dei nostri giorni, con i giornali, le TV, le radio, il Web, i social networks, è totalmente intrisa di contenuti informativi. Lo storage digitale di questi contenuti non sembra, per il momento, essere un problema insormontabile per la tecnologia corrente. I database esistono per quello, no? Tra l'altro, all'idea originale di tabella bidimensionale si sono aggiunti modelli multidimensionali più sofisticati[2] e sono anche stati proposti modelli alternativi ottimizzati per compiti specifici. Non parleremo di questo, ma segnaliamo ai non esperti due contributi interessanti, uno a favore degli young guns che si affacciano all'orizzonte, i modelli NoSQL (Not Only SQL)[3], e l'altro in difesa dei modelli "tradizionali"[4]. Questo armamentario tecnologico, ben coadiuvato dalla disponibilità di dispositivi hardware ad alte prestazioni, sta facendo il suo dovere. Dati geografici. Immagini. Cataloghi di proteine. **Il Web**. Sono solo alcuni tra i tanti esempi di strutture dati sempre più frequenti nelle scienze applicate e nella tecnologia di un mondo a complessità crescente come il nostro, rappresentati e manipolati in modo efficiente con database di varia natura.

Ed eccoci arrivati al vero problema: come possiamo estrarre in modo mirato e intelligente l'informazione desiderata da questa infinità[5] di dati? L'approccio corrente, keyword-based, presuppone l'uso di una o più parole chiave (con eventuale esclusione di altre keywords, indicazione del tipo di risorsa cercata, etc). Spesso, però, l'utente non è in grado di specificare le parole chiave esatte perchè ha un'idea solo parziale dell'oggetto della ricerca: in questi casi aiuterebbe un motore di ricerca in grado di funzionare anche "per approssimazione" utilizzando sinonimi o, per dirla in gergo più tecnico, a suo agio anche con ricerche di prossimità oltre che con il keyword matching. Un'altra caratteristica a cui è ormai difficile rinunciare è la granularità: per ovviare alle difficoltà di visualizzazione e navigazione di un mare sterminato d'informazione, è sensato iniziare la propria ricerca selezionando una tra un numero limitato di opzioni generiche offerte dal motore di ricerca, raffinando successivamente la propria query in modo iterativo, quando le opzioni proposte diventeranno più specifiche e delimitate.

Oggi incontreremo qualcuno che propone una soluzione effettiva a queste problematiche: la sua idea è utilizzare gli attributi spaziali (ad esempio le coordinate su una mappa, una spezzata che approssima una strada o il corso di un

Il riferimento nel titolo è così noto che non riteniamo necessario aggiungere dettagli. Approfittiamo invece dell'occasione per ricordare un ex-collega e amico che ci ha da poco lasciati. Questo focus è dedicato a Gianfranco Frau (Cagliari 1974, Cagliari 2015)

[1] In realtà un database relazionale può essere formato da più tabelle interconesse che, assieme al suo sistema di manipolazione (RDBMS), devono sottostare a regole ben precise, codificate in Codd EF, A relational model of data for large shared data banks, Communications of the ACM, **13**, 6:377-387, 1970 e nella letteratura successiva. Invochiamo la benevolenza del lettore e non ci soffermiamo oltre su questi aspetti.

[2] Questi ultimi usano rappresentazioni basate su ipercubi, in cui tutte le informazioni simili sono raggruppate lungo una stessa dimensione, rendendo la ricerca più facile. Il modello multidimensionale rispecchia più fedelmente la struttura intrinseca di un dataset: le relazioni tra i vari dati sono più evidenti e le operazioni di manipolazione degli oggetti sono in generale più performanti, si veda ad esempio Collins J, An assessment of multi-dimensional databases and their use, http://goo.gl/Bf1Ze0, 2003

[3] Oliver AC, Which freaking database should I use?, InfoWorld, http://goo.gl/YYRRJQ, 2012

[4] Sotnikov D, Don't write off relational databases for big data just yet, ReadWrite, http://goo.gl/1sw95, 2013

[5] Per un esempio di database spaziale e delle operazioni che può supportare, si veda Samet H, Alborzi H, Brabec F, Esperança C, Hjaltason GR, Morgan F, and Tanin E, Use of the SAND spatial browser for digital government applications, Communications of the ACM, **46**(1):63-66, 2003

fiume, un area rettangolare che delimita una regione geografica) per indicizzare le informazioni contenute nei database e rendere i motori di ricerca più potenti e flessibili. Certo, se state cercando l'ultimo articolo sulla soluzione delle bidomain equations 3D con volumi finiti (key search words: "bidomain equations" "finite volumes" 3D -monodomain filetype:pdf), probabilmente non vi interesserà specificare se il luogo di pubblicazione è la Nuova Zelanda piuttosto che Milano o Boston, e anche la granularità avrà un peso relativo (non aspettatevi centinaia di migliaia di articoli sull'argomento). Ma per argomenti più generali e più attinenti la nostra vita quotidiana, l'approccio map query può fare la differenza. Anche perchè quasi tutti noi abbiamo in tasca un dispositivo potenzialmente in grado di misurare (e di trasmettere) attributi spaziali legati ai nostri spostamenti: il GPS device dei nostri smartphones[6]. Gli attributi spaziali saranno, verosimilmente, sempre più comuni e sempre più presenti.

Il nostro ospite, Hanan Samet, è un guru indiscusso nel campo delle strutture di dati spaziali multidimensionali e delle tecniche matematiche per il sorting delle informazioni spaziali. Il suo libro "Foundations of multidimensional and metric data structures"[7] è un must per chiunque si occupi di algoritmi di ricerca di dati spaziali in computer graphics, visualizzazione, GIS, image processing, computer vision, games programming e non solo. Hanan, che ha ottenuto il Ph.D. alla Stanford University nel 1975 ed è attualmente Distinguished Professor all'Università del Maryland, oltre ad essere ACM Distinguished Speaker ha ricevuto un numero impressionante di riconoscimenti accademici[8], incluso l'ACM Paris Kanellakis Theory and Practice Award e lo IEEE Computer Society's Wallace McDowell Award. Lo incontriamo grazie ai buoni uffici di Enrico Gobbetti e Fabio Bettio del Visual Computing Lab del CRS4[9].

Hanan Samet al CRS4, luglio 2015 (foto di A. Mameli)

Professor Samet, è un grande piacere incontrarla qui al CRS4. Come mai da queste parti?
Sono qui per tenere un ciclo di tre seminari dell'ACM Distinguished Lecture Series organizzato dal CRS4[10]: è una delleinziative del progetto europeo DIVA[11], guidato da Enrico Gobbetti del CRS4 e Renato Pajarola dell'Università di

[6] Anche altri prodotti dell'elettronica di consumo vanno in questa direzione, si pensi ad esempio alle capacità di geotagging delle ultime fotocamere

[7] by H. Samet, Morgan Kauffman Publishers, 2006, http://goo.gl/Oh2rkS

[8] http://goo.gl/XEviOQ

[9] L'intervista è stata effettuata il 22/7/2015 dal sottoscritto e da Andrea Mameli all'Is Molas Hotel di Pula, che ringraziamo per l'ospitalità. Un grazie sentito anche ad Hanan per l'impegno profuso nella realizzazione di questo focus, incluso l'editing della grammatica e dello stile della versione inglese.

[10] http://goo.gl/ScxIJN

[11] http://diva-itn.ifi.uzh.ch/

Zurigo, focalizzato sulla visualizzazione e analisi di grandi volumi di dati. Inoltre, sono venuto per incontrare Enrico Gobbetti e i ricercatori del Visual Computing Lab del CRS4, e per toccare con mano i risultati del loro lavoro. È la prima volta che visito il vostro centro e sono rimasto impressionato sia dai risultati della ricerca in Visual Computing, di assoluta rilevanza internazionale, che dalla grande varietà di applicazioni pratiche. Oltre al laboratorio, ho visitato, ad esempio, l'installazione al Museo Archeologico Nazionale di Cagliari per l'esplorazione interattiva del complesso statuario di Mont'e Prama *[ne parleremo tra pochissimo, non perdiamoci di vista!]*.

Lei è un esperto molto conosciuto di strutture di dati metrici e multidimensionali. Perchè questo tipo di dati sono così importanti, specialmente in confronto a quelli usati nei tradizionali database relazionali?
In realtà anche i tradizionali database relazionali possono essere usati per immagazzinare dati multidimensionali[12], poichè una riga di un modello relazionale può evidentemente rappresentare un punto in uno spazio multidimensionale. Alcuni dati multidimensionali, come i dati spaziali (ad eccezione dei punti indicanti una posizione), hanno una peculiarità importante: l'estensione (si pensi ad aggregati di punti come segmenti, regioni, superfici o volumi). Ad esempio, i segmenti rettilinei possono essere trasformati in punti di uno spazio multidimensionale semplicemente annotando i valori delle coordinate dei loro estremi. Però questa trasformazione non preserva la prossimità (si veda il Technical Corner) nel senso che due segmenti vicini nello spazio fisico possono non esserlo nello spazio trasformato, a dimensione maggiore: questo rende inefficienti le operazioni di ricerca perchè lo spazio trasformato è molto più grande dello spazio fisico di partenza. Oggi abbiamo a che fare con i dati "di posizione" e i loro aggregati (ad esempio curve corrispondenti ad un percorso) quotidianamente, come conseguenza dell'uso dei nostri smartphone che contengono dispositivi GPS. La rappresentazione efficace dei dati è cruciale per il buon funzionamento di questi e altri dispositivi. Ovviamente questo solleva questioni di privacy: vogliamo o no che la nostra posizione sia resa nota ad altri, come i fornitori di servizi? Ma questa è un'altra questione che non riguarda la mia visita di questi giorni.

Quali sono oggi le applicazioni più importanti dei database multidimensionali e delle strutture di dati metrici?
Sono senz'altro molto usati in similarity searching, ad esempio per la ricerca del nearest neighbour in machine learning e altre applicazioni dove lo scopo finale è trovare chi/cosa è *simile* all'oggetto di partenza. Queste strutture dati permettono di effettuare un sorting non tradizionale, ampliando il concetto usuale di ordine rispetto ad un indice monodimensionale: è possibile una differenziazione dei dati, i cui attributi vengono mappati da uno spazio ad alta dimensione ad uno spazio 1D tramite mapping lineare. La chiave per differenziare gli oggetti è fare un sorting usando come indice lo spazio che essi occupano. Si fa usando tecniche di bucketing o associando una griglia allo spazio fisico, così come avviene per un'operazione più familiare, l'indicizzazione delle mappe.

Oggi la computer graphics gioca un ruolo fondamentale nelle scienze, nell'ingegneria, nella didattica, nella medicina, nelle produzioni multimediali e altro ancora, e richiede in genere una potenza di calcolo notevole. Quali sono le sfide più impegnative per la computer graphics attuale e perchè le rappresentazioni gerarchiche sono così importanti?
Beh, invariabilmente queste operazioni complesse comportano un'operazione di ricerca, e il modo di accelerare la ricerca è "organizzare" i dati, di solito tramite sorting. In effetti, molti algoritmi di computer graphics devono rispondere alle domande chiave "data una location, quali caratteristiche ci sono?" oppure "data una caratteristica, dove si trova?". Ovviamente non sempre ci rendiamo conto di questo processo, ma è quello che avviene in pratica.

Nel suo libro sono presentati un gran numero di metodi per il sorting di dati spaziali, ad esempio quelli basati su una decomposizione dello spazio e quelli che usano gerarchie di oggetti che possono o no sovrapporsi. È uno zoo molto affollato. Perchè così tanti metodi e come si fa a scegliere quello più adatto alle proprie applicazioni?
In realtà, come amo dire spesso, ci sono due famiglie di metodi, uno basato su una gerarchia di oggetti, l'altro su una gerarchia di "celle": stranamente questo non viene sempre compreso. Inoltre, i problemi nascono spesso in aree separate di applicazioni, ed esperti con formazioni diverse cercano di risolverli in modo diverso. Io cerco di mostrare la somiglianza di metodi apparentemente diversi. Quindi, sì, ci sono molto tecniche ma spesso sono imparentate tra loro.

Come sono cambiate le strutture dati dai primi modelli, proposti ormai 40 anni fa? Alcuni, come i quadtrees, vengono ancora usati. Questo la stupisce?
In realtà no. Penso che non si fosse capito quanto validi e generali sono questi metodi. Le problematiche sono cambia-

[12] Sottolineiamo la distinzione tra *database* multidimensionali (contrapposti ai database relazionali, si veda il Technical Corner) e *dati* multidimensionali

te nel tempo e anche le metodologie per affrontarle si sono evolute in parallelo. Ad esempio i video games e i servizi basati sulla localizzazione dell'utente ora sono di uso comune e usano strutture dati come i quadtrees, considerati in passato oggetti esotici buoni solo per chi faceva image processing. Oggi con gli smartphones, la localizzazione è un dato come gli altri. Per gran parte degli eventi sapere dove questi avvengono costituisce un'informazione importante: così la location può essere un modo efficace di indicizzazione degli eventi. Con un device opportuno non ho bisogno di conoscere con esattezza cosa sto cercando: posso indicare un punto su una mappa, ingrandirla, spostarmi, visualizzare l'informazione associata a determinate coordinate spaziali. Ad esempio, se sto cercando un concerto rock a Manhattan e ne trovo uno ad Harlem va bene, perchè la seconda location fa parte della prima: ma anche un concerto rock a Brooklyn probabilmente fa al caso mio, perchè è un evento ragionevolmente vicino, seppure non rispondente alle mie chiavi di ricerca iniziali. Se poi c'è un evento a New York City allora il risultato è molto generico, ma comunque meglio di nulla. È più importante di quanto si potrebbe pensare: se uno usa un motore di ricerca tradizionale e non digita le parole chiave esatte probabilmente non troverà l'evento a cui è interessato: al contrario, la ricerca per location permette una grande flessibilità. Ovviamente questo rende necessario un indexing basato sulla posizione e la disponibilità di una struttura dati pensata ad hoc.

Con i suoi collaboratori avete sviluppato NewsStand[13], un'applicazione per la ricerca di informazioni e notizie tramite location. Ci spiega come funziona?
L'aspetto più importante è che posso ricercare eventi che non conosco nel dettaglio a priori: selezionando una posizione sulla mappa posso diminuire la scala per avere più dettaglio, fare un panning, vedere cosa è accaduto dove, e solo allora fare la mia scelta. Ovviamente si possono fare anche ricerche per parole chiave, relative ad esempio ad un evento preciso, e NewsStand indicherà la location o le locations dove questo evento ha avuto o avrà luogo. Il dettaglio di informazioni ed eventi fornito da NewsStand dipende dalla scala geografica selezionata: un evento locale che riguarda la Sardegna probabilmente non è mostrato quando l'utente visualizza l'intero globo terrestre, ma diventa invece visibile quando la ricerca viene effettuata su scala geografica più delimitata. In altre parole, NewsStand funziona in base ad un principio "granulare" di localizzazione dell'informazione.

L'intuizione alla base di NewsStand è adottare la carta geografica come medium di elezione per la ricerca di contenuti: è un'idea che può essere estesa ad altri tipi di documenti oltre alle notizie (ad esempio voi avete sviluppato TwitterStand, le cui finalità sono evidenti dal nome). Quali sono secondo lei i campi di applicazione più promettenti del futuro prossimo?
Credo che sarà possibile manipolare sempre più informazione tramite location: foto, tweets... in generale ogni oggetto dotato di un'ontologia e di un attributo spaziale potrà essere inserito in un database multidimensionale, esplorabile tramite map query basate su informazione geografica. Un tipo di ricerca molto appealing è quella associata a interrogazioni time-based. Prendiamo ad esempio la diffusione di una malattia come ebola: verosimilmente la comparsa del virus troverà eco nei media locali, e grazie all'informazione contenuta nel database delle notizie sarà possibile ricostruire l'evoluzione spazio-temporale dell'epidemia e la storia della sua propagazione. Un'altra applicazione importante per le aziende è l'identificazione delle aree geografiche dove un determinato brand appare più o meno spesso su stampa, TV, Web, e altri mezzi: questo può aiutare ad esempio a pianificare in modo razionale i propri investimenti pubblicitari, piuttosto che effettuare campagne generalizzate "a pioggia" che possono essere più costose e meno efficaci. Ancora, nelle ambasciate è spesso utile raccogliere dai media locali notizie che riguardano il paese di appartenenza e riportarle al proprio governo centrale. Sono tutti esempi di indagini che da tempo vengono fatte da esperti umani e che NewsStand permetterebbe di accelerare in modo notevole.

Un'idea molto suggestiva, che lei ha affrontato nella sua produzione scientifica[14], è il tentativo di prevedere le caratteristiche di eventi futuri, usando le informazioni del passato recente e del presente. Ci può dire di più su questo argomento?
È una buona domanda. Pensiamo all'esempio precedente della propagazione della malattia: posso cercare di prevedere il pattern futuro di distribuzione dell'epidemia a partire dalla dinamica recente. È un topic molto generale: riguarda sia eventi che sono molto discussi nei media ma anche eventi improvvisi. Le ricadute sono evidenti, sia per gli aspetti di pianificazione delle risorse che per minimizzare i rischi collegati ad eventi naturali potenzialmente catastrofici (ad esempio, dove colpirà un uragano). È una direzione naturale della ricerca che può avere vantaggi

[13] http://newsstand.umiacs.umd.edu/web/. Si veda anche Samet H, Sankaranarayanan J, Lieberman MD, Adelfio MD, Fruin BC, Lotkowski JM, Panozzo D, Sperling J and Teitler BE, Reading news with maps by exploiting spatial synonyms, Communications of the ACM, **57**(10):64-77, 2014 disponibile a http://tinyurl.com/newsstand-cacm e il video allegato disponibile a http://vimeo.com/106352925

[14] Ho S, Lieberman M, Wang P and Samet H, Mining future spatiotemporal events and their sentiment from online news articles for location-aware recommendation system, Proceedings 1st ACM MobiGIS, 2012, 25–32

tangibili per la collettività.

Lavorerà in questa direzione? Più in generale, come sceglie gli argomenti di ricerca? E com'è il suo rapporto con le aziende?
[*Sorride*] Molto del mio lavoro dipende dal fatto di trovare o meno studenti interessati a lavorare su un particolare argomento. Questo è un problema generale della ricerca: tu puoi avere una buona idea, ma poi devi convincere gli altri, non puoi obbligare nessuno a lavorare ad una cosa a cui non crede: come dico sempre, puoi portare un cavallo all'abbeveratoio ma non puoi costringerlo a bere. Il rapporto con le aziende non è facile: in generale hanno una visione a breve termine di quello che può essere fatto. Non abbiamo molto supporto economico per i nostri studenti da parte delle aziende. Poi abbiamo un "elefante in salotto", qualcosa che non possiamo fare finta di non vedere: la company più famosa al mondo per quello che riguarda i motori di ricerca. Tanti credono che abbia risolto tutti i problemi possibili collegati alla ricerca e alla manipolazione di dati, ma non è così: almeno non per ogni tipo di dati e di argomenti. C'è ancora molto lavoro di ricerca da fare e tanti problemi aperti importanti che andranno affrontati e risolti nei prossimi anni.

— F. Maggio

While “classical” relational databases rely on a 2D traditional row and column structure, multidimensional databases result from using a hypercube representation, where each dimension is associated with an object attribute. As an illustration, consider the example below consisting of the ratings of three leading Italian red wines presented using both a relational (Table 1) and a multidimensional database (Figure 1).

Table 1. Scores of three leading Italian red wines using a relational database

id	*Wine*	*Producer*	*Year*	*A*[a]	*B*[b]	*C*[c]	*D*[d]	*Total*
11	I Sodi di San Niccolò	Castellare di Castellina	2006	99	85	93	90	367
14	Turriga	Argiolas	2006	95	85	95	90	365
21	Barolo Cannubi Boschis	Luciano Sandrone	2006	95	90	93	85	363

[a] Gambero rosso, http://www.gamberorosso.it/vini
[b] L’Espresso, http://goo.gl/UYOQCI
[c] Veronelli, http://www.seminarioveronelli.com/
[d] Luca Maroni, http://lucamaroni.com/

Let us now suppose that we want to add more data to our red wine database, say the ratings of these wines for the years 2007 and 2008. In the case of a relational database, a simple way to do this is to add 3 new rows to Table 1 for each of the two additional years. Alternatively, we can represent the same information using the multidimensional database shown in Figure 1:

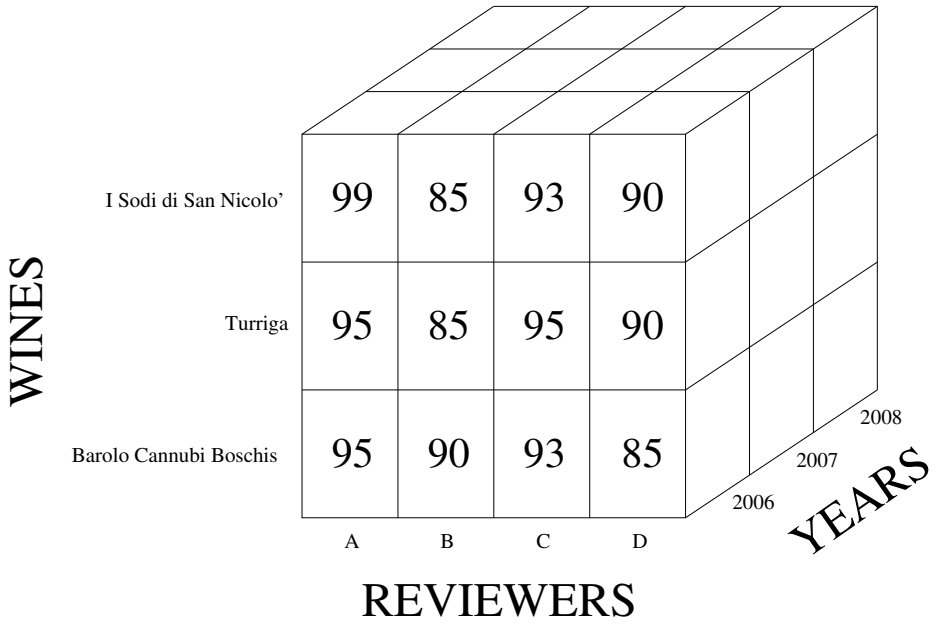

Figure 1. The multidimensional database corresponding to the relational database of Table 1 (with the addition of 2007 and 2008 years)

From now on, we no longer dwell on the difference between relational and multidimensional databases (often refered to as models) as either one or the other model can be used for representing spatial data - the actual object of our interest -, by simply setting the appropriate attributes. For instance, the spatial attributes for a line segment representing a portion of a road or a river, consist of 2 pairs of (x, y) coordinate values[15] of their endpoints, resulting in either 4 columns in a table or a point in a 4-dimensional real space. Independently of the model adopted, the representation of spatial data has a well-known problem: indexes commonly used for query and

[15] We assume a flat Earth approximation

look up in non spatial databases are not well-suited for manipulation and search of spatial datasets. In fact, while traditional indexes do a great job when simple retrieval of data is needed, they have serious drawbacks with spatial queries: one of the most important being that they do not preserve proximity[16]. For example, consider the situation depicted in Figure 2, where segments AB and CD are part of the representation of a river R and a road S, respectively, and one wants to query if S is near R or, even more importantly, if S crosses R.

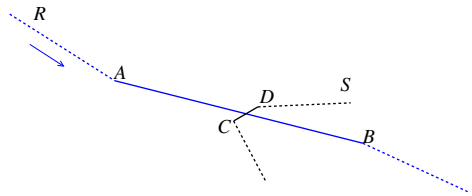

Figure 2. The proximity violation for a pair of segments represented by the coordinate values of the two endpoints

Actually, given a simple representation of a segment AB in terms of the coordinate values of its endpoints, the metric distance in a 4-dimensional space between the segments is clearly an unreliable indicator of the proximity of the two spatial objects. This representation shows a major issue in practical queries with spatial data which is that it only identifies the space spanned by the data objects (e.g., all of the points that make up the line segment), information which is not feasible to provide explicitly. For example, it is not practical for the database to store the names of all of the roads that pass through each point in the underlying space, or equivalently to store the name of every road that crosses every river. Such considerations motivated researchers to introduce new index types, complying with proximity preservation, and often based on suitable mathematical structures like trees to speed up access to the underlying data. The obvious idea is to separate the non-spatial components, which can be handled in a standard way, from the spatial components which deserve special indexing treatment. A comprehensive introduction and analysis of the different spatial index types may be found in the previously mentioned book of Hanan Samet. Here we limit ourselves to summarizing the main features of the most important families of such methods, without going into too much detail. The basic idea is to decompose the space to which the data belong into regions called *buckets*. There are several ways to do that.

- **Object hierarchies with minimum bounding rectangles (MBRs)**
  In this case we aggregate the group of objects into collections of distinct subgroups of a finite size which are recursively aggregated so the the result is a tree. The fact that the objects can be arbitrarily-shaped means that operations such as point lookup (i.e., if a point is contained in an object) or object intersection detection can be complex and thus we simplify such tests by associating a minimum bounding box (e.g., rectangle, sphere, etc.) with each object or object aggregate as now the operation involves one or two known shapes. This approach is primarily used with rectangle minimum bounding boxes (known as *MBR*s), and here we restrict ourselves to the two-dimensional case. The result is a non-disjoint decomposition of the underlying space as the MBRs can overlap. This means that the area spanned by a single object (e.g. a line segment) may be included in multiple MBRs, even if each object is associated with just one MBR. The R-tree and the R*-tree are examples of data structures that belong to the object hierarchy family. An example of the space decomposition induced by an R-tree for a piecewise linear curve is shown in Figure 3.

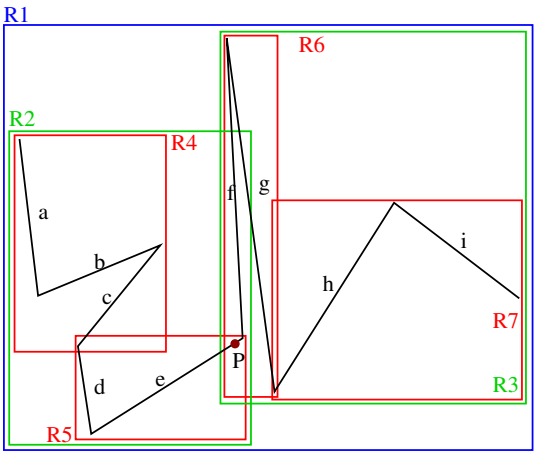

Figure 3. A piecewise linear curve represented by an object hierarchy with minimum bounding rectangles (MBRs). Note the introduction of some empty space in order to enable the illustration of the boundaries of the MBRs which are actually coincident as is the case, for example, for the right boundaries of R3 and R7 as well as for the right and the left boundaries of R6 and R7, respectively. Adapted from [16].

Figure 4 is the tree structure corresponding to the R-tree of Figure 3. Notice the presence of overlapping MBRs in Figure 3. This means that when, for example, we want to determine which line segment contains a particular point, we may have to search the entire underlying space as the point may lie in several MBRs even though the line segment object on which it lies is associated with only one MBR. In other words, we must examine all of the MBRs in which the point lies. For example, the line segment containing point P in Figure 3 is included in MBRs R2, R3, R5,and R6, while it actually only belongs to R5.

[16] Samet H, A sorting approach to indexing spatial data, International Journal on Shape Modeling, **14**(1):15–37, 2008

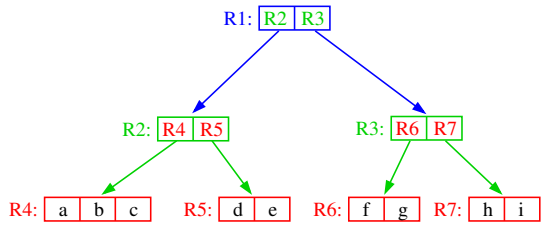

Figure 4. The R-tree associated with the MBR object hierarchy of Figure 3. Adapted from [16]

- **Decomposition of the embedding space**
  In this case, we focus on methods that decompose the space in which the objects are embedded (and hence also the objects) into disjoint non-overlapping cells. There are many variants of such methods. One such method starts with an object hierarchy that employs MBRs such as the R-tree discussed in the previous section and then decomposes the MBRs into disjoint cells so that they span the space that contains the objects. The R^+-tree and the cell tree are examples of it. As objects are split into several parts (i.e., sub-objects), such methods generally take up more space than the method that uses an object hierarchy with the complete MBR. However, this enables the more efficient performance of many operations including those that find the object that contains a particular point, as there is no need to traverse the entire object collection. Figure 5 is the disjoint cell analog of the piecewise linear curve of Figure 3.

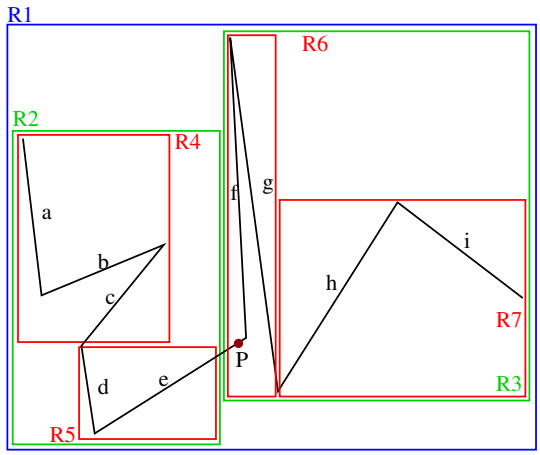

Figure 5. A piecewise linear curve represented by disjoint cells. Note the introduction of some empty space in order to enable the illustration of the boundaries of the MBRs which are actually coincident as is the case, for example, for the right boundaries of R3 and R7 as well as for the right and the left boundaries of R6 and R7, respectively. Adapted from [16].

Figure 6 is the tree structure corresponding to the R^+-tree of Figure 5. Comparing Figure 6 with Figure 4 we see that some of the objects appear more than once in Figure 6 while this is never the case in Figure 4.

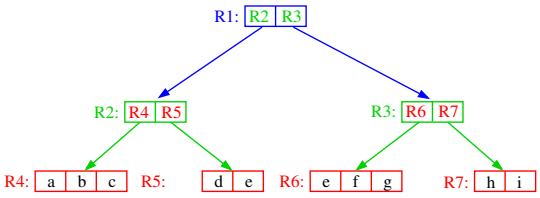

Figure 6. The R^+-tree associated with the disjoint cells model of Figure 5. Adapted from [16].

The quadtree family of representations is a very different example of the methods that decompose the embedding space. In this case the embedding space is usually decomposed recursively into 4 congruent cells (assuming without loss of generality that the underlying space is two-dimensional) until some property of the relationship between the objects and cells is satisfied. For example, one stopping condition in the case of two-dimensional region data is that each cell is a member of at most just one region. In this case, the quadtree is an alternative to the bitmap representation as shown in Figure 7.

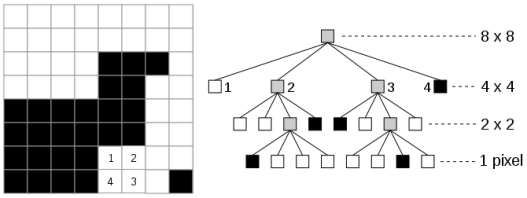

Figure 7. A 2D bitmap (left) and its quadtree representation (right). Numbers 1,2,3,4 indicate indexing order of children. Attribution: Wojciech Muła, via Wikimedia Commons.

More generally, quadtrees can be used to represent various types of spatial data. One common use is to represent a collection of point data (e.g. cities on a map) where the cell is decomposed if it has more than k objects where the cell is like a bucket and k is the bucket capacity. Figure 8 is an example of the decomposition of the underlying space induced by such a quadtree for a collection of European cities with a bucket capacity of 1. This variant of the quadtree is known as a *PR quadtree* where P and R denote point and region, respectively. In this case, the leaf nodes are either empty or contain just one point object along with the coordinate values of the point (not shown in the figure).

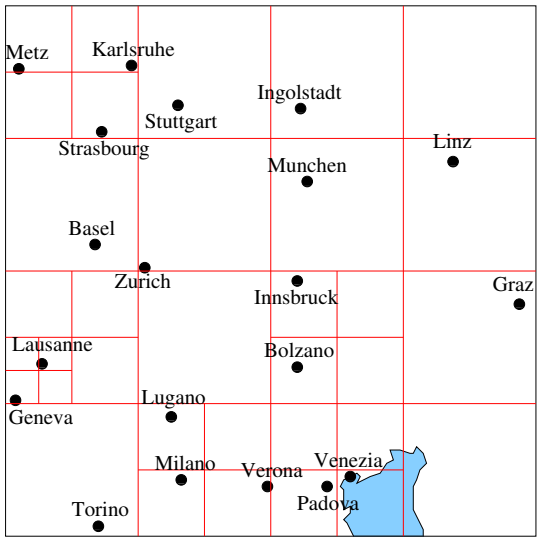

Figure 8. Map of a few European cities subdivided following a quadtree decomposition rule with a bucket capacity of 1. Note that object in close proximity (e.g. Geneva and Lausanne) may result in a finer decomposition of the underlying space and hence a deeper tree structure. See[17].

Figure 9 is the tree representation of the quadtree structure for the collection of European cities in Figure 8. The leaf nodes are represented by either white square (empty nodes) or black ones (nodes containing data). A major drawback of this implementation is that points in close proximity may result in deep trees (e.g., see the cities of Lausanne and Geneva in this example). This problem can be overcome by setting the bucket capacity to a value $c > 1$.

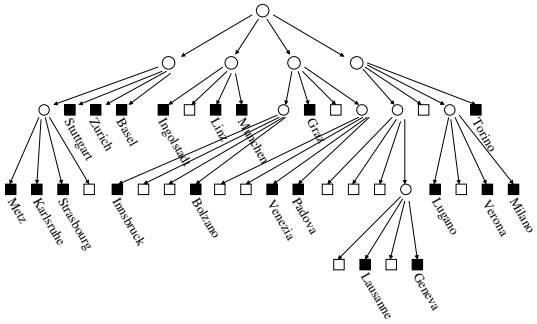

Figure 9. The tree representation of the quadtree associated with the European cities dataset. The numbering order of the subtrees is the same as in Figure 7.

Quadtrees are especially well-suited for location-based queries which is the case that we are given a location and that we want to know what features or objects are associated with it.

- There are considerably many more methods for indexing spatial data that space constraints prevent us from reviewing. Among them is the pyramid, whose internal nodes contain a summary of the information in the nodes below them, which allows us to skip nodes with no relevant information when performing a search query. The pyramid performs well with feature-based queries which correspond to the case that we are given a feature or an object and we seek to know which locations are associated with it (also known as *spatial data mining*)[18].
  Other interesting methods include octrees, partition fieldtrees, hierarchical decompositions by triangles or rectangles, etc. For a detailed review of these techniques we refer to the already mentioned book of Hanan Samet [7] and the associated spatial data structure applets [19].

[17] Samet H, Sorting in space: multidimensional, spatial, and metric data structures for computer graphics applications, in *ACM SIGGRAPH 2008 classes*, 2008, ACM New York, **90**, 1-106

[18] Aref WG, Samet H, Efficient processing of window queries in the pyramid data structure, in Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Nashville, TN, 1990, 265-272

[19] See http://donar.umiacs.umd.edu/quadtree/index.html