

# İÇERİĞE DAYALI MELODİ ARAMA

Mehmet TETİK

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz – 2002

**ÖZET****Yüksek Lisans Tezi****İÇERİĞE DAYALI MELODİ ARAMA****Mehmet TETİK****Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı****Danışman: Prof. Dr. Ali GÜNEŞ  
2002, 46 sayfa**

Bu tezde, düzenlenmiş midi dosya topluluğu içerisinde içeriğe dayalı melodi arama yapılmıştır. Müzik parçaları midi dosyaları biçiminde kaydedilmiştir. Aranacak melodi ile aynı nota, süre ve hız bilgisine sahip olmayıp benzer olan melodiler ve melodinin kendisi bulunabilmektedir. Arama için özellik (feature) olarak tam perde farkı (exact pitch difference) kullanılmıştır. Arama motoru olarak Viterbi algoritması kullanılmıştır. Arama yapılacak melodi karakter olarak nota dizisi şeklinde veya örnek kayıtlı midi dosyası şeklinde girilebilir. Bu arama sistemi ve grafik kullanıcı arabirimi MS Visual Basic 6.0 programlama dilinde yazılmıştır. Bulunan sonuçlar uzaklığı ile birlikte kullanıcıya sunulmuştur.

**Anahtar Kelimeler:** İçeriğe dayalı arama, Melodi arama, MİDİ, Viterbi  
Algoritması

**ABSTRACT****Master of Science Thesis****CONTENT BASED MELODY SEACRH****Mehmet TETiK****Anadolu University  
Graduate School of Natural and Applied Sciences  
Computer Engineering Program****Supervisor: Prof.Ali GÜNEŞ  
2002, 46 pages**

In this thesis, content based melody search system is designed for a arranged midi file database. Music is represented in midi file format. Melodies with the different note sequence, duration and speed than search melody and exact melody are listed in search results. Exact pitch difference is used as a feature for search engine. Viterbi Algorithm is used for search engine. Search melody is entered in this system as a string note sequence and a sample midi file. This system and graphical user interface is designed in MS visual Basic programming language. Search results are listed with their matching distance.

**Keywords:** Content Based Search, Melody Search, MIDI, Viterbi Algorithm

# İçindekiler

Sayfa

<b>Özet .....</b>	<b>iii</b>
<b>Abstract .....</b>	<b>iv</b>
<b>İçindekiler.....</b>	<b>v</b>
<b>Şekiller Dizini .....</b>	<b>vii</b>
<b>1 Giriş.....</b>	<b>1</b>
<b>2 İçeriğe Dayalı Arama (İDA).....</b>	<b>4</b>
2.1 Giriş.....	4
2.2 İDA Sistemi .....	9
2.3 İDA Çeşitleri.....	10
2.4 İDA Sorgulama Türleri.....	12
<b>3 İçeriğe Dayalı Müzik Arama.....</b>	<b>13</b>
3.1 Müzik ve Algılanması .....	13
3.2 Melodinin Algılanması .....	18
3.3 İçeriğe Dayalı Müzik Arama Sistemi .....	19
3.3.1 İçeriğe Dayalı Müzik Arama Sistemlerinde Kullanılan Özellikler .....	20
3.3.2 Arama Motoru .....	21
3.4 Sorgulama Çeşitleri .....	22
<b>4 İDMA Melodi Arama Sistemi .....</b>	<b>23</b>
4.1 Giriş.....	23
4.2 İDMA Sisteminin Bölümleri .....	24
4.3 İDMA Sisteminin Oluşturulması.....	25
4.3.1 Veritabanının Oluşturulması.....	25
4.3.2 Arama Motorunun Oluşturulması.....	27
4.4 Programın Çalışması.....	31
4.5 İDMA Programının Yapısı .....	34
4.6 Sonuçlar.....	38
<b>5 Sonuç ve Öneriler .....</b>	<b>39</b>
<b>EKLER.....</b>	<b>40</b>
<b>Kaynaklar.....</b>	<b>47</b>

## Şekiller Dizini

	<i>Sayfa</i>
2.1 Birinci nesil bilgi erişim şekli _____	8
2.2 Yeni nesil bilgi erişim şekli. _____	9
2.3 Bir İDA sisteminin genel blok diyagramı _____	10
2.4 Örnek doku resimleri _____	11
2.5 Eğrilik (Curvature) _____	11
3.1 Müziğin nesne modeli _____	14
3.2 Bir oktavlık not- frekans tablosu _____	16
3.3 Sekiz oktavlık nota-frekans tablosu _____	16
3.4 Nota bilgisinin süre bilgisi ile gösterimi _____	16
3.5 Nota bilgisinin süre bilgi olmadan gösterimi _____	17
3.6 Bir melodiye 2 nota kaydırma yapılması _____	20
4.1 İDMA sisteminin bölümleri _____	25
4.2 İDMA sistemi veritabanının oluşturulması _____	26
4.3 Midi dosyasının içeriği _____	26
4.4 Viterbi algoritmasının kafes sistemi gösterimi _____	28
4.5 İDMA sistemi için viterbi kafes grafiği _____	30
4.6 İDMA programının grafik kullanıcı arabarımı(GKA) _____	31
4.7 İDMA GKA'sında aranacak melodinin girilmesi _____	32
4.8 İDMA GKA'sında arama sonucunun dosya isimi olarak görüldüğü kısım _____	33
4.9 İDMA GKA'sında arama sonucunun melodi içeriği ve uzaklık bilgisi ile görüldüğü _____	33
4.10 İDMA GKA'sında dosya veritabanı işlemlerinin yapıldığı sekme _____	34

# 1 Giriş

Gelişen bilgisayar teknolojisi ile birlikte, kayıtlı bilgi miktarı hayli artmıştır. Özellikle metin dosyalarına göre çok fazla büyüklüğü olan çoklu ortam dosyalarının sayısının artması, bu dosyalardan istenilen dosyaya erişme problemi doğmuştur [1]. Özellikle, bu tür dosyalarda istediğiniz dosyaya erişmek için, eski bilinen anahtar kelime eşleşmeli arama yamanın yeterli olmadığı ortaya çıkmıştır.

Bu çoklu ortam dosyalarındaki ses bilgisi açısından da istenilen dosyaya ulaşmak büyük bir problemdir. Bu dosyalardan biri olan müzik kayıtlarının tutulduğu dosyalarının artması ile bu kayıtlardan aranılana ulaşmayı güçleştirmiştir.

Burada istenilen dosyaya erişimin güçleşmesi, günümüz arama sistemlerinde bir arama yapıldığında dosya sayısının fazla olmasından, aramanın sadece insan eliyle girilen dosya ismi veya meta etiketleri gibi açıklayıcı bilgi etiketlerinin içerisinde karakter eşleşmeli olarak yapılmasından dolayıdır. Bunun yanında istenilen bir dosyaya erişmek anahtar kelime ile ifade edilemiyor olabilir. Bu durumda şu anki sistemler hiç kullanılamaz. Örneğin elimizdeki örnek bir resme benzer resimleri bulmak, içerisinde bas seslerin ağırlıkta olduğu ses kayıtlarını bulmak veya içerisinde yeşil rengin ağırlıkta olduğu resimleri bulmak.

Arama sistemlerinde ki problemleri şöyle isimlendirebiliriz:

1. Arama yapıldığında çok fazla ilgisiz sonuç gelmesi.
2. İstenilen dosyalara erişmek için başka tür sorgulama çeşitlerine izin vermemesi.

Çoklu ortam dosyalarından biri olan müzik kayıtları içinde aynı problemler geçerlidir. İnternet'in gelişmesi ve bilgi depolama teknolojilerin ilerlemesi ile müzik kayıtları da büyük ölçüde artmıştır [2]. Arama sistemlerinin yukarıda iki madde de toplanan problemler müzik kayıtları içinde geçerlidir. Arama sistemleri yine müzik dosyaları içinde istenilen dosyaya erişmekte her türlü sorgulamaya

izin vermez. Örneğin: elimizde bulunan bir müzik dosyasına benzer kayıtları, içinde piyano çalınan kayıtları bulmak veya içerisinde belli bir melodinin geçtiği kayıtları bulmak.

Bu projede geliştirilen iDMA (İçeriğe Dayalı Melodi Arama) sistemi ile elle nota dizisi olarak girilen melodiye ve midi dosyası olarak verilen bir melodiye erişmek mümkün kılınmaktadır. Böylece önceden yapılmış bir besteye erişmek mümkün olmaktadır.

iDMA sisteminde sadece direkt aranılan melodinin birebir aynısının bulunmasının yanında melodinin aynı olduğu kayıtları bulmakta mümkündür. Çünkü müzik sisteminde bir melodi tek bir şekilde çalınmamaktadır. Aynı melodi farklı müzik aleti ile farklı oktavda çalınabilir.

iDMA sisteminde melodinin benzerliğini bulmak için nota geçiş farkları kullanılmıştır. Bunun için önce nota dosya veritabanı oluşturulmuştur. Sonra nota geçiş farkları çıkarılmıştır. Arama için bu farklar kullanılmıştır.

Melodi arama, aranacak melodiyi bir dizi (sequence) olarak düşünebileceğimiz için nota geçiş fark veritabanında dizi aramak problemine dönüşmüştür.

iDMA sisteminde ilk aşama olarak monofonik (monophonic) midi dosyalarıyla çalışılmıştır.

Bu çalışma içerisinde ikinci bölümde genel olarak içeriğe dayalı arama sistemlerinden bahsedilmektedir. Bu tezin konusu olan müzik yani melodinin yanında diğer çoklu ortam dosyaları için içeriğe dayalı arama sistemlerinin özelliklerinden, çeşitlerinden bahsedilmektedir.

Üçüncü bölümde ise içeriğe dayalı melodi arama ile ilgili genel bilgiler verilmektedir.

Dördüncü bölümde ise iDMA sistemi hakkında bilgi verilmektedir. iDMA sisteminin özellikleri, nasıl oluşturulduğu ve çalışma sistemi anlatılmaktadır. Ayrıca geliştirilen programın ekran kesitleri verilmektedir.

Beşinci bölümde ise bu proje ile ilgili düşünceler ve üzerine yapılabilecek geliştirmeler hakkında görüşler verilmiştir.



## 2 İçeriğe Dayalı Arama (İDA)

### 2.1 Giriş

İçeriğe Dayalı Aramayı (İDA) ve niçin ihtiyaç duyulduğunu anlamak için İDA kullanılmadan yapılan arama sistemlerini incelemek iyi olur. Bilgi teknolojilerinin gelişmesi veritabanı sistemleri için hesaplama-yoğunluklu ve bilgi-yoğunluklu arasında bir ayrıma sahne olmuştur. Artan bilgi ile beraber büyük miktarda bilgi içeren veritabanlarına erişen programlara ihtiyaç olmuştur [3]. Veritabanındaki istenilen bilgilere erişmek için bilgiler yapısal bir şekilde olmalıdır. Veritabanındaki istenilen bilgilere erişirken yapısal sorgulamalar kullanılır. Bu yapısal sorgulamalarda girilen anahtar kelimeler veritabanındaki alanlarda karakter katarı ile eşleştirme yapılır. Bilgileri bir bütün olarak bilinen veritabanı sistemlerinde olmayıp dosya topluluğu olarak saklanılan veritabanı sistemlerinde de aynı durum geçerlidir. Arama sisteminde karakter katarı olarak girilen anahtar kelime dosyaların tutulduğu veritabanı içerisinde karakter eşleşmeli olarak aranır. Bu aramada anahtar kelime veritabanındaki dosyalarla ilgili kaydedilmiş bazı dosya bilgileri içerisinde yapılır. Bu bilgiler dosyanın ismi, dosyayı oluşturan kimsenin adı, dosyayı oluşturan kişi tarafından girilen ve dosyayı nitelendiren meta etiketler olabilir. Bu bilgilerden bir çoğu dosyayı oluşturan yada veritabanına giren kişi tarafından girilir.

Bilinen veritabanı sistemlerinde ve dosya topluluğu veritabanlarında istenilen dosyaya erişmek için yapılan sorgulamalar, dosyayı oluşturan yada veritabanına bu bilgileri giren kişiye bağlı olmaktadır. Veritabanında o dosya ile ilgili bilgiler kişiye bağlı olarak eksik yada farklı girilebilmektedir. Örneğin içerisinde bir kuşunda yer aldığı doğa resminde ağaçlar baskın olduğu için bu dosyayı isimlendiren yada onunla ilgili bilgilerin belirtildiği meta etiketleri giren kişiye bağlı olarak, bu resimde ağaçlar baskın olduğu için kuş ile ilgili hiçbir

kelime geçmeyecektir. Bu yüzden kuş kelimesi olarak anahtar kelime seçilen bir sorgulamada bu resme ulaşmak mümkün olmayacaktır. Bu resmede ulaşabilmek için içeriğe dayalı arama yapmak gereklidir.

İnternet'in özellikle WWW ve FTP'nin kullanımının yaygınlaşması ile artan bilgi ile Gigabayt'ları Yönetmek (Managing Gigabytes) diye bir kavramda oluşmuştur [4]. Bu gittikçe artmaya devam eden gigabaytlarca bilgi içerisinde bilgiden içerik-açıklayıcı bilgileri otomatik üreten ve böylece kişiye bağlı kalmadan doğru ve istenilen bir aramak yapma gereği artmıştır.

Arama yapmak için bir dosya için girilebilecek bilgileri sınıflandırırsak [3]:

- **İçerik-Bağımsız Bilgi** (Content-independent metadata): Bu bilgi direkt olarak resim yada video ile ilgili olmayan, fakat bir şekilde onu nitelendiren dolaylı bilgilerdir. Bu bilgilere örnek olarak : yazarın ismi, tarih, dosyanın biçimi(format), yer, sahiplik vb.
  - İçerik ile ilgili bilgiler:
    - **İçerik-Bağımlı Bilgi** (Content-dependent metadata): Bu bilgiler daha çok dosyanın alt ve orta seviye özellikleri (low/intermediate features) ile ilgilidir. Örnek olarak: renk, doku (texture), şekil, uzaysal (spatial) ilişkiler, ve bunların kombinasyonları. Bu bilgiler genelde algılama ile ilgili bilgilerdir. Durağan bir resim için algılanan ilk özellikler renk ve dokudur, bunun yanında video için renk, doku ve hareket özellikleridir.
    - **İçerik-Açıklayıcı Bilgi** (Content-descriptive metadata): Bu bilgiler dosyanın anlamı (semantic) ile ilgili bilgiler. Bir resim için içerisinde bulunan adamın yüz ifadesine göre gülmesi yada ağlaması mutluluk yada üzüntü soyut kelimelerine karşılık gelmeleri örnek verilebilir.

Metne dayalı arama ile içerik bağımsız bilgilerin ve elle girilen bilgilerin içerisinde arama yapılabilir. Elle girilen bilgilerde tabii ki kişiye bağlı olduğu için eksik olacaktır. Bununla birlikte içerik ile bilgiler üzerine bir arama yapmak zaten mümkün olmayacaktır. Burada içerik-bağımlı bilgilerin çıkarıldığı bir veritabanında arama yapılması, yani içeriğe dayalı arama yapılması gerekir. Yukarıdaki sayılan özelliklerin çıkarılması içeriğe dayalı aramanın yapılabilmesi için ilk şarttır. Aramanın yapılacağı veritabanında bu dosyalar ile ilgili bu bilgilerin kaydedildiği ayrı bir veritabanı daha oluşturulmalıdır.

Metne dayalı arama ile içeriğe dayalı aramayı karşılaştıracak olursak:

- Metne dayalı arama için elle girilen anahtar kelimeler hem girilmesi pahalı hem de tam olarak her özelliği nitelendiremez.
- Anahtar kelime dosyanın yada medyanın neye benzediğini ve ne hissettirdiğini anlatamaz.
- Sorgulamayı ifade etmenin zor olduğu durumlarda arama yapmak çok zordur.
- Bir dilde yazılmış anahtar kelimeyi başka bir dile çevirmek her zaman mümkün değildir. Yani başka bir dildeki veritabanına erişmek her zaman mümkün olmayabilir.
- İki anahtar kelime arasındaki benzerliği anlayamazsınız. Çünkü metne dayalı aramada direkt birebir kelime eşleştirme kullanılır.

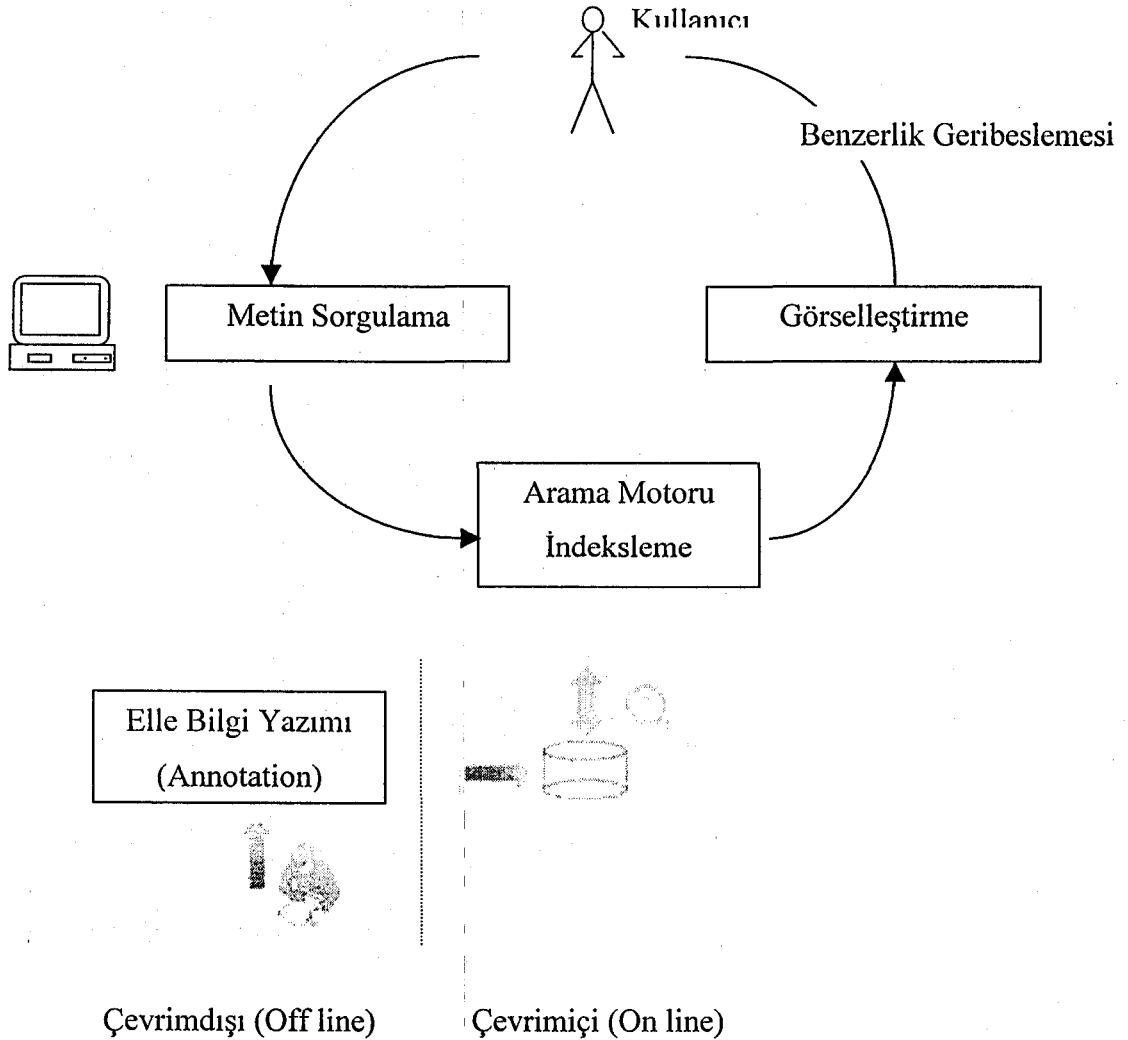
Özetle, örnek üzerinde karşılaştırsak, içerisinde kaplan bulunan resimleri aramak istediğimizde arama sistemlerinde dosya ismi yada meta bilgi olarak girilen bilgiler içerisinde arama yapılacaktır. Fakat bu bilgiler kişiler tarafından elle girildiği için resim içerisinde kaplan şeklinin baskın olmadığı durumlarda dosya isminde yada meta bilgilerde bu kelime geçmediği için kaplan görüntüsü olduğu halde bu resimlere ulaşmak mümkün olmayacaktır. Ama bizim aramak

istediğimiz resim içerisinde kaplan bulunan resimleri içeriğe dayalı arama ile bulmamız mümkündür. Bununla birlikte sorgulama yaparken elimizdeki örnek resme benzer resimleri bulmamızda mümkündür. Bu örneği ses için düşünürsek içeriğe dayalı arama sistemi içerisinde aranılan kelime geçen ses kayıtlarını bulunmasını sağlamaktadır.

Bununla birlikte içeriğe dayalı arama sistemlerinde bulunan sonuçlar yakınlık derecesine göre sıralanır [5]. Yani İDA sisteminde belirlenmiş yakınlık fonksiyonunun değerine göre bir sorgu sonucu getirilir.

Bilgiye erişimin (information retrieval) ilk nesli olarak metne dayalı içerik aramayı gösterebiliriz. Bu sistemde görüntü, ses ve video dosyalarına ancak karakter katarı değişkenleri ile erişilebiliyordu [6,7]. Bu sistem için tipik bir arama örneği: “Topkapı Sarayının 18. yüzyıl tablo görüntüleri” yada “Picasso’nun peyzaj resimleri” .

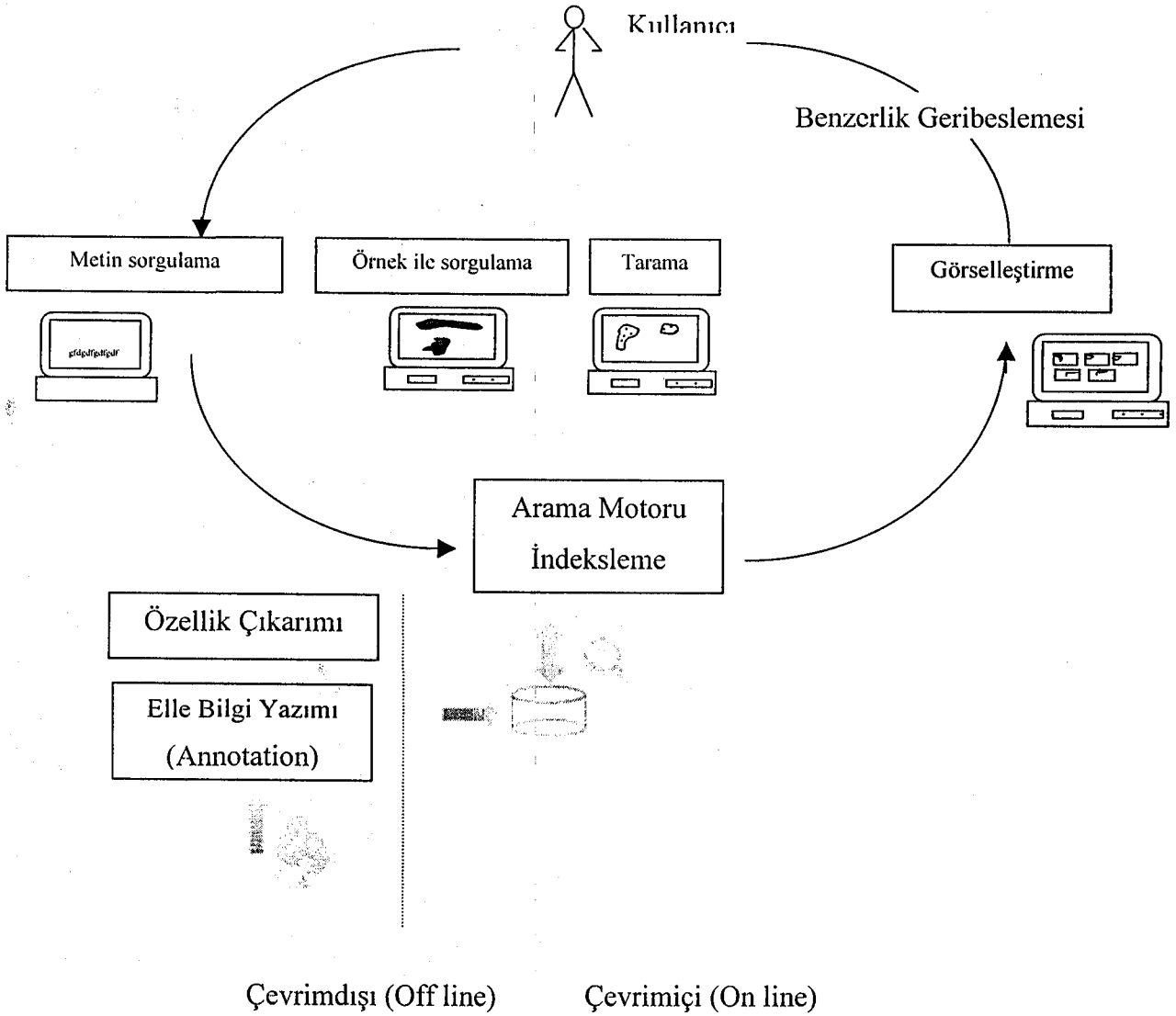
Bu sistemde içerik-bağımsız meta bilgiler alfa-nümerik karakter katarı olarak saklanır. İçerik-bağımlı ve içerik-açıklayıcı bilgiler anahtar kelime yada yazı dizisi (script) ile ifade edilir. Yani örneğin bir resim yada video için manzara doğa resmi, insan, mutlu kelimelerini kullanmak. Erişimde metin veritabanında geleneksel sorgulama dilleri (SQL gibi), yada direkt metin eşleşmesi kullanılır.



**Şekil 2.1 Birinci nesil bilgi erişim şekli**

İlk nesil arama sistemlerinde düşük/orta seviye bilgiler (renk, doku, şekil vb.) erişim için kullanılmaz. Yani bir güzel sanatlar öğrencisinin elindeki bir resme benzer resimleri bulması ya da istediği renk ağırlıklı Picasso'nun tablolarını sorgulama imkanı yoktur. Çünkü bu sistemde bu düşük/orta seviye bilgiler karakter katarı ile ifade edilememektedir.

Yeni nesil bilgi erişim (information retrieval) sistemlerinde ise bu bilgilere de ulaşabilmektedir. Bu bilgilere erişmeyi sağlamak için görüntü işleme, örüntü tanıma, ses analizi ve konuşma tanıma gibi teknolojiler kullanılır.

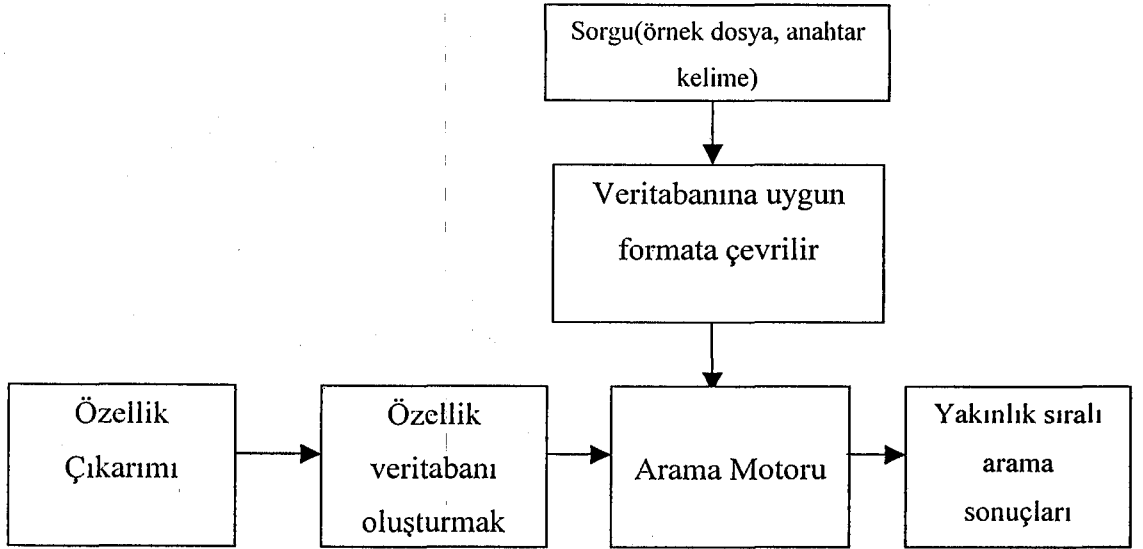


*Şekil 2.2 Yeni nesil bilgi erişim şekli.*

## 2.2 İDA Sistemi

İçeriğe dayalı arama yapabilmenin temel unsuru arama için kullanılacak özelliklerin seçimidir (feature extraction). Seçilen özellikler her sistemde farklı olabilir. Buna bağlı olarak arama motoru gerçekleştirilir. Aranacak sorgu için de aynı özellikler bulunur ve arama motoruna girdi olarak verilir. Arama motorunun

sonucu bulunan eşleşmeler yakınlık sırasına göre listelenir.



*Şekil 2.3 Bir İDA sisteminin genel blok diyagramı*

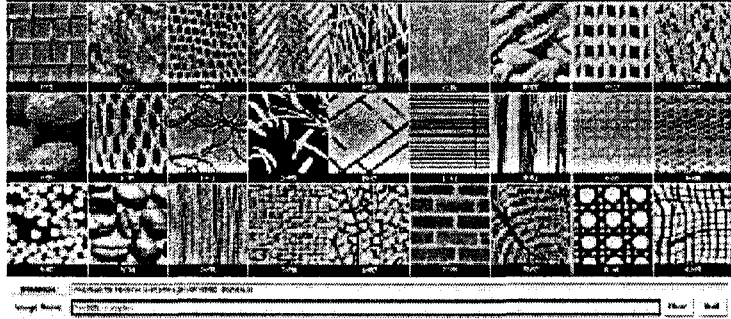
## 2.3 İDA Çeşitleri

İçeriğe dayalı aramanın en kapsamlısı video için yapılan sistemlerdir. Fakat burada video dosyasını arka arkaya konmuş görüntü ve ses birleşimi olarak düşünülebilir. Yani içeriğe dayalı video aramayı içeriğe dayalı görüntü ve ses arama olarak düşünebiliriz. İçeriğe dayalı ses aramayı konuşma, ses (insan sesi dışındaki sesler) ve müzik (melodi) arama olarak ayırabiliriz.

Özellik seçimi görüntü, ses ve müzik için farklıdır. Görüntü için özellikleri ikiye ayırabiliriz: Düşük seviye (Low level) ve yüksek seviye (High level) özellikler olarak [4]. Fakat yüksek seviye bilgiler alt seviye bilgilerden çıkartılır. Yani düşük seviye bilgi içerisinden kullanılmayan, ikinci bir teknik ile yüksek seviye bilgiye erişilebilir. Örneğin bir resim içerisindeki kenar geçişlerinden o resimdeki ağaç, insan, yüz şekli bilgisinin çıkarılması. Düşük seviye özellikler:

- Renk: Gri seviye ve çeşitli renk uzaylarındaki renk bilgisi RGB (Red Green Blue), YUV, HSV...

- Doku (Texture) Resim üzerindeki kontrast, sertlik, renk geçişlerinin oluşturduğu bilgi. Doku örnek resimleri Şekil 2.4'te verilmiştir:

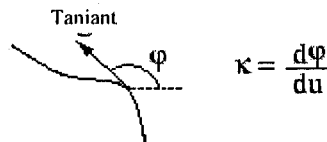


*Şekil 2.4 Örnek doku resimleri*

- **Kenar/yön bilgisi:** Resimdeki kenarların ve yönlerinin çıkarılması. Gabor süzgeçleri yada kenar tespit etme (Edge detection methods) yöntemleri bulunabilir.

Yüksek seviye bilgiler:

- Eğrilik (Curvature-Shape) : En çok kullanılan eğrilik ölçümü  $\kappa$  kenar çizgilerinin tanjant yönünün ark uzunluğu olan  $u$ 'ya göre değişimidir. Şekil 2.5'te gösterilmiştir:



*Şekil 2.5 Eğrilik (Curvature)*

- Şekil (Shape): Bir nesne yada bölge kendisinin sınır çizgileri (boundary contour) ile temsil edilebilir (Chain-code).



- Pozisyon (Position): Bir nesne yada bölgenin uzaysal yeri.

Video için özellikler [4]:

- Hareket (Motion)
- Sahne geçiş ve kesimleri: Bir video çerçeve dizisi içerisinde en önemli çerçeve ilk ve son çerçevedir.
- Düzenleme özellikleri: Solukluk, tarama, yakınlaştırma.

Ses için [8]

- Perde (Pitch): Kısa-zamanlı (short-time) Fourier spektrumun tepe (peak) noktalarıdır.
- Genlik (Amplitude): Ses sinyalinin RMS (Root Mean Square) Karekökleri karesinin ortalaması değeridir.
- Zero-crossing: Sinyalin sıfır seviyesinin kesme sayısıdır.
- Parlaklık (Brightness): Sesin parlaklığı sinyalin merkez frekansı ile gösterilir.

## 2.4 İDA Sorgulama Türleri

İçeriğe dayalı aramanın en büyük özelliği karaktere dayalı sorgulamanın yanında değişik sorgulama tiplerine izin vermesidir. Bu sorgulama türlerini İDA sistemlerimize göre ayrı ayrı incelersek:

Görüntü için: Örnek resme benzer resimleri aramak, kullanıcının çizdiği şekle benzer resimleri aramak.

Ses için : İçerisinde istenilen ses kaydına benzer kayıtları bulmak, mikrofona söylenen kelimenin geçtiği kayıtları bulmak.

Aslında sorgulama türlerinin en kapsamlısı video için olanıdır. Çünkü hem ses hem de görüntü mevcuttur. Örneğin video için şöyle bir sorgulama düşünülebilir: “İçerisinde Atatürk’ün “Türk Gençliği” dediği video kayıtlarını getir.”

Sorgulama tipleri tasarlanan İDA sistemine bağlı olarak değişir.

## 3 İeriĐe Dayalı Mzik Arama

DiĐer oklu ortam dosyaları gibi mzik kayıtlarının artması ve klasik metin eŐleşmeli arama sistemlerinin yetersizliĐi ieriĐe dayalı mzik aramayı gerekli kılmıŐtır. Eski sistemlerde mzik kayıtları araması yapılırken, Őarkı liriklerinin yada mzik zelliklerinin (melodi, tempo, nota...) bulunduĐu veritabanı ierisinde karakter eŐleşmeli arama yapılmaktadır [9]. Bununla birlikte mikrofondan mırıldanılan melodiyi yada kayıtlı bir mziĐi sorgulamak bu tr sistemlerde mmkn deĐildir. Burada ieriĐe dayalı arama sistemleri rol almaktadır.

İeriĐe dayalı mzik aramanın yapılabilmesi iin nce mzik kayıtlarından arama yapılacak zelliklerin ıkarılması gerekir. Zaten bir mzik parası ararken ikincil bilgilerden (Paranın baŐlıĐı, syleyen vb.) ok ieriĐi (melodi) ile ilgili bilgiler parayı niteler [2]. zelliklere karar verildikten sonra arama motorunun gereklenmesi gerekir.

Bu blmde takip eden alt blmlerde nce genel mzik bilgisi verilecektir. Mzik aramada en ok kullanılan dosya tr olan midi dosya tr hakkında bilgi verilecektir. Genel olarak ieriĐe dayalı mzik aramanın zelliklerinden ve mevcut sistemlerden ve zelliklerinden bahsedilecektir.

### 3.1 Mzik ve Algılanması

Mzik ile ilgili arama yaparken mzik kaydının dalga formu (waveform) ile arama yapmak hi anlamlı deĐildir. Bunun yerine dalga formundan ıkarılan performans bilgileri zerine arama yapmak daha uygundur [9]. Bir mzik parası belli bir tempo ierisinde alınan kanallardan meydana gelir. Her kanal melodi ve ritim bilgisi ierir. Melodi ierisinde bir enstrman tarafından alınan nota dizisini ve ritim ile baŐka enstrmanlarda (davul vb.) da almayı saĐlayacak olan vurma sırası bilgisini aktarılmaktadır. Bir nota belli bir perde (pitch), sre (duration), ve hıza (velocity) sahiptir. Perde dinleyicinin sesi algıladıĐı frekanstır. Notanın sresi

ise bir notanın ne kadar süre basıldığını gösterir. Hız ise diğer basılan notalara göre göreceli olarak sesin yüksekliğidir (Piyano için düşünürsek hız piyanonun tuşuna ne kadar hızlı basıldığı anlamına gelmektedir). Bir müzik parçasını nesne modeli ile gösterirsek, Şekil 3.1'deki modeli elde ederiz:

```
Nesne NOTA {
  Perde
  Süre
  Hız
}
```

```
Nesne MELODİ {
  Kanal
  Enstrüman
  Nota dizisi
}
```

```
Nesne MÜZİK {
  Melodi
  Tempo
}
```

### *Şekil 3.1 Müziğin nesne modeli*

Nesne modelinde de görüldüğü gibi müzik çalınan bilgi (melodi) ve bunun sırası diyebileceğimiz tempo bilgisinden oluşur. Melodi bilgisi de enstrümanda çalınan nota, enstrümanın ismi, ve eğer melodi çok-sesli (ileride açıklanacak) ise hangi kanalın kullanıldığı. Nota nesnesi de yukarıda da bahsedildiği gibi perde, süre ve hız bilgisinden oluşur.

Bununla birlikte müziğin dokusu (texture) vardır. Müzikte doku terimi çeşitli seslerin ve melodilerin aynı anda çalınarak oluşturdukları müzik harmanıdır. Doku bakımından müziği ikiye ayırabiliriz: *1- Tek sesli* (monophonic) *2- Çok sesli* (polyphonic). Tek sesli en basit dokudur. Tek sesli müzik sadece tek bir melodiden oluşur. Yani aynı anda sadece bir kanaldan bir nota çalınır. Çok seslide ise aynı anda bir çok melodinin çalınması ile oluşan bir dokudur.

Bu tez içerisinde, melodi dendiği zaman tek sesli (monophonic) müzik kastedilmektedir.

Melodinin içeriği bu şekilde tanımlanmakla beraber, melodi müziğin algılanma özelliğidir [2]. Bazen bir insanı melodi olarak algıladığı başka bir insanın algıladığından farklı olabilir. Melodiyi ilginç ve hatırlanabilir kılan birkaç maddenin kombinasyonudur: melodinin genel şekli- bazen bu melodi çizgisi yada melodik kontur olarak adlandırılır-, ritim ve hareket (ani, dereceli vb.). Kendi sistemimiz olan iDMA'da melodi ile ilgilenirken melodinin perdesi ve tek sesli müzik ile ilgilenilecektir.

Melodiyi oluşturan ana bilgi olan nota bilgisi insan sesinin duyabileceği ve müzik aletlerinin çıkarabileceği ses frekansı aralığı belli bir ölçeğe göre ayrılmıştır. Standart müzik ölçeği Eşit Dağılımlı Ölçek (Equally Tempered Scale) standardına göre yapılmıştır. Bununla birlikte bu ölçek 12-ton kromatik ölçek olarak da adlandırılır. Bu ölçekte oktav 12 eşit aralıklı frekanstan oluşmuştur. Bir oktav içerisinde bir nota frekansı kendisinden önceki notanın frekansının  $2^{(1/12)}$  katıdır (2'nin 12'de bir katı ve yaklaşık olarak 1.059463'e eşittir). Böylece bir notadan sonraki 12. notada o notanın iki katı olmuş olur. Bu ölçek hesaplanırken A440 notası başlangıç notası seçilir ve 440 Hz olarak atanır. Diğerleri bu notadan  $2^{(1/12)}$  formülü ile hesaplanır. Şekil 3.2'de A440 oktavının nota-frekans tablosu verilmiştir.

Müzik	Frekans	Avrupa	
Notaları(ABD)	Denklem	Notasyonu	
A	440	440.00	do
A#	$440 \times 2^{1/12}$	466.16	
B	$440 \times 2^{2/12}$	493.88	re
C	$440 \times 2^{3/12}$	523.25	
C#	$440 \times 2^{4/12}$	554.37	mi
D	$440 \times 2^{5/12}$	587.33	fa
D#	$440 \times 2^{6/12}$	622.25	
E	$440 \times 2^{7/12}$	659.26	sol
F	$440 \times 2^{8/12}$	698.46	
F#	$440 \times 2^{9/12}$	739.99	la

G	440 x 2 <sup>10</sup> /12	783.99	
G#	440 x 2 <sup>11</sup> /12	830.61	si
A2	440 x 2 <sup>12</sup> /12	880.00	do

### Şekil 3.2 Bir oktavlık not- frekans tablosu

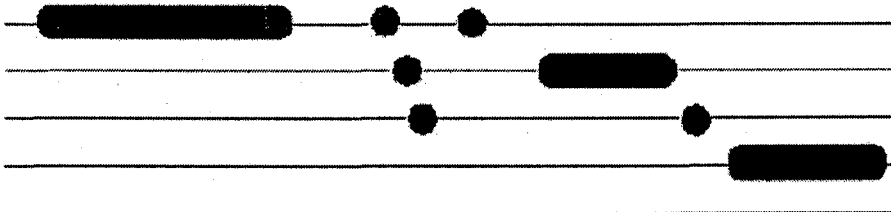
Burada görülen iki tane nota isimlendirmesi vardır. Biri Amerikan Notasyonu, diğeri Avrupa notasyonudur. Amerikan sisteminde Nota gösterimi için A, B, C, D, E, F, G harfleri, Avrupa sisteminde ise Do Re Mi Fa Sol La Si kelimeleri kullanılmıştır.

Eşit Dağılımlı Ölçek'in ( Equally Tempered Scale ) tüm frekans dağılımı EK-B de verilmiştir. En çok kullanılan aralığın frekans karşılıkları Şekil 3.3'te verilmiştir:

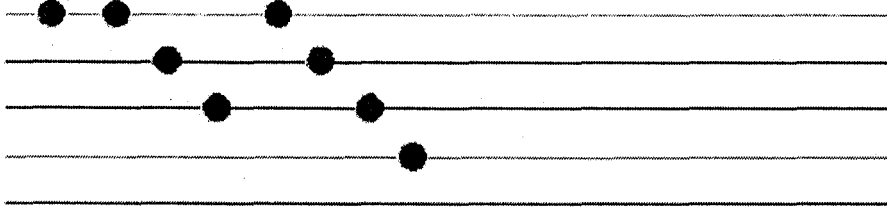
OCT #	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1	32.703	34.648	36.708	38.891	41.203	43.654	46.249	48.999	51.913	55.000	58.270	61.735
2	65.406	69.296	73.416	77.782	82.407	87.307	92.499	97.999	103.831	110.000	116.541	123.470
3	130.811	138.591	146.831	155.561	164.811	174.611	185.000	196.000	207.652	220.000	233.082	246.940
4	261.622	277.182	293.661	311.133	329.633	349.233	369.999	392.000	415.304	440.000	466.164	493.880
5	523.244	554.371	587.331	622.255	659.266	698.467	739.999	783.999	830.611	880.000	932.339	987.770
6	1046.488	1110.871	1174.711	1244.511	1318.533	1396.933	1480.000	1568.000	1661.222	1760.000	1864.719	1975.540
7	2093.022	2221.751	2349.422	2489.022	2637.022	2793.822	2960.000	3136.000	3322.432	3520.000	3729.339	3951.111
8	4186.044	4443.491	4698.844	4978.044	5274.044	5587.759	5919.962	6271.964	6644.870	7040.000	7458.679	7902.111

### Şekil 3.3 Sekiz oktavlık nota-frekans tablosu

Melodi ifade edilirken süre (duration) bilgisi önemlidir. Ama nota hırsızlığı açısından aynı notaları süreleri değiştirilerek yapılan müziklerinde bulunması istenebilir. Bu durumda iDMA sistemi kullanılabilir. Melodi için nota ve süre bilgisinin karşılaştırılması Şekil 3.4'te verilmiştir:



### Şekil 3.4 Nota bilgisinin süre bilgisi ile gösterimi



*Şekil 3.5 Nota bilgisinin süre bilgi olmadan gösterimi*

Şekil 3.4 ve 3.5'te verilen melodi aynı notalara sahiptir ama süre bilgileri ikinci şekilde çıkarılmıştır. Nota bilgisi bakımından bu iki melodi aynıdır.

Müzik sistemlerinde arama yapıldığında veritabanı için en çok kullanılan dosya türü midi dosyasıdır. iDMA sisteminde de midi dosyaları kullanılmıştır. Midi dosyası müzik ile ilgili tüm bilgileri içerir. Ama kötü bir yanı da içerisinde sakladığı bilgilerin dışında başka bir bilgi elde etmek, çıkarmak mümkün değildir. Çünkü, midi dosyası çalınan enstrümanı taklit etmekte ve enstrüman üzerinde yapılan tüm hareketleri kaydetmektedir. Yani bu metot müziğin doğrudan sinyal<sup>1</sup> bilgisinin kaydedilmesinin yerine, müziğin performans bilgilerinin kaydedilmesidir [9]. Bu yüzden müziğin direkt dalga formundan elde edilebilecek bilgilere erişmek mümkün değildir. Yani eğer müzik yapılırken arkada özellikle yapılmış bir düzenli gürültü yada enstrüman kullanılmadan çıkarılmış bir sesi midi ortamında duymak mümkün değildir. Bununla birlikte müziğin içeriği ile ilgili esas teşkil eden tüm bilgiler kaydedildiği için midi dosyası müzik arama sistemlerinde en çok kullanılan dosya türüdür [9].

Midi enstrümanın tüm hareketlerinin kaydedilmesidir. Bu kayıtlar enstrüman ile bilgisayarın yada midi kayıt aletlerinin birleştirilmesi ile yapılmaktadır. Bu enstrümanlardan gelen hareket bilgileri bilgisayara midi ve meta olaylar olarak belirli bir sabit formatta gönderilir. Böylece enstrüman üzerinde yapılan tüm hareketler zamana bağlı olarak kaydedilmiş olur.

<sup>1</sup> Örneğin: Wav, MPEG3 dosyaları

MIDI, İngilizce açılımıyla Musical Instrument Digital Interface, müzikal enstrümanları dijital ortamla bağdaştırmaktır. Yukarıda bahsettiğimiz gibi müzik aletini simüle ederek onu bilgisayar ortamına taşımak için kullanılan standarttır. Bilgisayar ortamında midi dosyaları kullanılarak yani enstrümanlar üzerinde yapılmış tüm hareketler kullanılarak o müzik aletinin çıkardığı ses çıkarılmaktadır. Midi dosyasında kaydedilen bilgilerle ve o çalınan enstrümana ait ses özellikleri bilinerek yapılmaktadır. Genel midi formatı farklı üreticiler için ürettikleri enstrümanların birbiriyle doğru çalışmasını sağlayan bir standart olmuştur [10].

İnternet'te şu an büyük miktarda midi dosya veritabanı bulunmaktadır. Bir haber grubuna gönderilen midi dosyalarını arşivleyen bir site 15.000 midi dosyasına sahiptir [11].

Midi dosya formatının spesifikasyonu Ek-A da verilmiştir.

### 3.2 Melodinin Algılanması

Bir müzik parçasını diğerinden ayırabilmemizi sağlayan nedir? İki müzik parçasını birbirinden ayırmamızı sağlayan, melodidir. Bu sayede insanlar doğuştan gelen bir yetenekle, basitçe mırıldanarak yada ıslık çalarak aynı melodiyi üretilmektedir. Bir müzik parçası dinlendikten sonra insanların aklında kalan o müziğin melodisidir. Bir müziği yıllar sonra sözlerini hatırlamasak da melodisini, tonunu hatırlayabiliriz [2]. Bir önceki bölümde de bahsedildiği gibi melodiyi oluşturan üç ana unsur vardır: 1- Nota Dizisi 2- Kanal 3- Enstrüman. Nota dizisinin perde, süre ve hız bilgilerini de düşünmemiz gerekir. Fakat melodi algılanırken bu üç özelliği aynı olanlar aynı melodidir demek doğru değildir. Melodi algılanırken nota bilgisinden çok notaların sıralanışı, birbiri arkasına gelme düzeni önemlidir. Uitenbogerd (1998) müzik benzerliği ile ilgili olarak aşağıdaki faktörleri sıralamıştır [12]. (Önemliden daha önemsiz doğru sıralanmıştır). Ritim yönü bu sıralamada göz önüne alınmamıştır:

1. Aynı anahtarda kaymanın olması (Exact transposition in the same key)

2. Çok yakın anahtarda kaymanın olması (Exact transposition in a closely related key)
3. Farklı anahtarda kaymanın olması (Exact transposition in a distant key)
4. Aynı kontur ve tone (Same contour and tonality)
5. Aynı kontur ve atonal (Same contour but atonal)

### 3.3 İÇERİĞE DAYALI MÜZİK ARAMA SİSTEMİ

Diğer içeriğe dayalı arama sistemlerinde uygulanan prosedür müzik içinde geçerlidir. Sistem müzik için tasarlandığından dolayı kullanılan özellik (features) ve sorgulama çeşitleri farklılık gösterir. Fakat genel çatı bütün içeriğe dayalı arama sistemlerinde benzerdir.

İçeriğe dayalı müzik arama sistemlerinde de önce istenilen özellik çıkarımının yapılabileceği ham veri formatı seçilir ve bu formattaki dosyalardan bir dosya veritabanı oluşturulur. En çok kullanılan dosya türü müzik için midi ve wav dosya formatıdır. Bu dosyalar kullanılarak arama sistemini motorunun kullanabileceği özellik veritabanı oluşturulur. Arama bu veritabanı içerisinde arama motoru tarafından yapılır.

Arama sisteminde yapılabilecek yada istenilen sorgulama türüne göre kullanılacak özelliklere karar verilir.

Özellik veritabanı oluşturulduktan sonra arama motoru bu özelliklere göre dizayn edilir. İçeriğe dayalı arama sisteminde arama motorunun verdiği sonuç sıralı liste şeklindedir. Tabii eğer bulunan sonuçlar arasında tam eşleşme (exact match) varsa, oda sıfır uzaklıkta olarak sıralı listenin en başında gösterilir.

Sistemin son bölümü olarak da bu arama motoru ile kullanıcı arasında kullanıcı arabirimi oluşturmak vardır. Kullanıcı arama motoruna bu arabirim sayesinde sorgulama yaptırır.



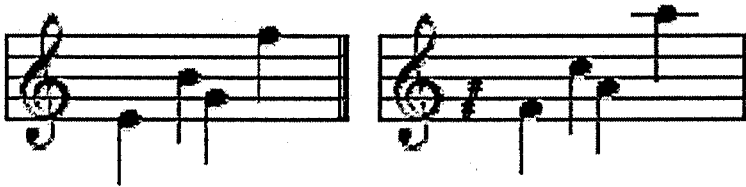
### 3.3.1 İeriĐe Dayalı Mzik Arama Sistemlerinde Kullanılan zellikler

İeriĐe dayalı mzik arama sistemlerinde en ok nota bilgisi kullanılır. Fakat bu nota bilgisi elde edildikten sonra, veritabanında istenilen nota dizisini aramak direkt bir karakter eŐleŐmeli metin arama iŐlemine dnŐmŐ olur. Fakat bir melodiyi sadece bir tek nota dizisi ile ifade etmek mmkn deĐildir. nk bir melodiyi aynı nota dizi kullanılmadan da alınabilir [11].

Bu yzden mzik arama iin nota bilgisi elde edildikten sonra, nota bilgisinden arama yapılabilecek zelliĐin ıkarılması gerekiyor. Bunları Őyle sıralayabiliriz:

1. Melodik Konturu (Melodic Contour): Melodik kontur melodinin notalarının birlerine greceli olarak izdikleri izgidir(kontur).

Bir melodi bir nota (ya da bir oktav) yukarı ya da aŐaĐı kaydırıldığında hala aynı melodidir. Notaya baĐlı arama yapıldığında bu kaydırılmış melodinin bulunması mmkn deĐildir. Melodik kontur bu problemi ortadan kaldırır. AŐaĐıda iki melodi aynıdır ama biri diĐerinin 2 nota kaydırılmış halidir.



*Őekil 3.6 Bir melodiye 2 nota kaydırma yapılması*

Bir notanın perdesinin kendisinden nceki notanın perdesinden yksek olması U (Up) , alak olması D (Down), aynı olması R (Repeat) ile temsil edilir. Bu notasyona gre Őekil 3.6 da ki iki melodinin melodik konturu U D U olarak bulunur. Bylece bu iki melodiden birincisi iin arama yapıldığında konturları aynı olduĐu iin ikinci melodi bulunan sonular arasında yer alacaktır.

Fakat bu sistemde perde farkları (pitch difference) kullanılmadığı için arama bir çok doğru olmayan sonuç verecektir [9]. Bu arama hızlı sonuç vermesi açısından önemlidir.

2. Perde Farkı (Nota Geçiş Farkı): Bu özellik melodik kontura göre daha iyidir. Fakat saklanan bilgi daha fazladır. Çünkü her nota numarası arasındaki fark sayı olarak bulunmak zorundadır. İDMA sisteminde perde farkı kullanılmıştır.
3. Nota Hızı (Velocity): Melodi benzerliği bulunması açısından tek başına yeterli değildir ama yardımcı özellik olarak kullanılır.
4. Süre (Duration): Bu bilgide arama sisteminde istenilen sorgulamaya bağlı olarak yardımcı özellik olarak kullanılır.

### 3.3.2 Arama Motoru

İçeriğe dayalı müzik arama sistemlerinde düzenlenen veritabanı metin tabanlıdır. Arama motoru olarak seçilen algoritma, sistemin özelliklerine bağlı olarak tam eşleşme (exact matching) yada yaklaşık (approximate matching) sonuç verme ile ilgili olarak değişmektedir. Melodi arama için kullanılan metotlar şunlardır: Dinamik programlama algoritmaları, geometrik algoritmalar ve diğer karakter eşleşme (string matching) algoritmaları [2]. Melodi arama nota dizisi arama gibi düşünüldüğünde bir sıra, zaman değişimi olduğu için Gizli Markov Model (Hidden Markov Model) de kullanılmaktadır. Gizli Markov Modelleme bir dizinin elemanlarının istatistiksel olarak arka arkaya gelme olasılığını kullanarak sonuç verir.

Veritabanının büyüklüğüne göre arama motorunun en kısa sürede en iyi sonucu vermelidir. Bu yüzden arama motorları tam eşleşmeli sistemler yerine, yaklaşık karakter eşleşmeli algoritmaları güçlendiren sistemler olarak dizayn edilmektedir.

Bu konuda yapılmış bazı mevcut sistemler:

CATFIND: Honkong Üniversitesi tarafından İnternet müzik arama motoru olarak geliştirilmiştir. Melodik kontur kullanılmıştır. Tüm melodiler perde değişimine bağlı olarak “/”, “\”, “-” karakterleri ile temsil edilmiştir [13].

ECHO: Waseda üniversitesi tarafından geliştirilmiştir. Bu sistemde perde bilgisi ve ritim bilgisi dinamik programlama ile kullanılarak yapılmıştır.

MELDEX (MELody inDEX): Waikato üniversitesi tarafından melodik perde konturu kullanarak yapılmış 100.000 kadar melodi kaydının bulunduğu bir sistemdir. Bu sistemde mırıldanarak sorgulama (query by humming) yapmak mümkündür [13]. Melodiyi temsil etmek için kullanılan melodik perde konturu olarak “U”, “D” ve “R” harfleri kullanılmıştır.

### 3.4 Sorgulama Çeşitleri

Arama sisteminde yapılabilecek sorgulamalar seçilen özelliklere ve arama motorunun tasarımına bağlı olarak değişir. Şu ana kadar yapılmış sistemlerde görülen arama çeşitleri şunlardır:

1. Örnek müzik parçası ile: Eldeki müzik parçasına, midi yada wav olabilir, benzer kayıtların bulunması.
2. Mırıldanarak sorgulama: Bu sorgulama kullanıcının mikrofon kullanılarak melodinin mırıldanması ile yani konuşma yada müzik parçasının sözleri söylenmeden yapılmaktadır.
3. Elle girilen nota dizisi ile: Bu sorgulamada kullanıcı aranılacak melodinin nota dizisini girer.
4. Melodik perde konturu ile. Bu sorgulamada ise kullanıcı tarafından nota dizisi gibi melodik kontur dizisi girilir.

## 4 İDMA Melodi Arama Sistemi

### 4.1 Giriş

İDMA (İçeriğe Dayalı Melodi Arama) sistemi midi dosya formatında kaydedilmiş melodiler içerisinde istenilen melodiyi aramayı sağlayan bir sistemdir. Kullanıcı bir melodiyi ifade eden nota dizisini bu sistemde kullanılan veritabanında bulunan midi dosyaları içerisinde melodinin tam olarak aynısını veya benzerini içeren midi dosyalarını arayabilir. Bununla birlikte, metin olarak girilen nota dizisinin yanında bilgisayar ortamında kayıtlı bir melodinin de araması da yapılmaktadır. Bulunan sonuçlar dosya isimleri olarak listelenmektedir.

İDMA sisteminin oluşturulmasını üç bölümde gösterebiliriz:

1. Veritabanının oluşturulması
2. Arama motorunun oluşturulması
3. Kullanıcı arabiriminin oluşturulması

Veritabanı kayıt formatı olarak midi dosyası seçilmiştir. Midi dosyası melodiyi saklamak için geliştirilmiş dosya formatıdır. Böyle bir sistem için en uygun dosya formatıdır. Ama arama yapmak için bu dosyalar istenilen formatta değildir. Bunların geliştirilen arama motoruna uygun olarak istenilen formata çevrilmesi gerekir. Midi dosyası melodi ile ilgili tüm bilgileri içerir. Bu bilgilerden istenilen bilgilerin çıkarılması gerekir. İDMA sistemi için nota geçişlerini gösteren farklar bulunmuştur.

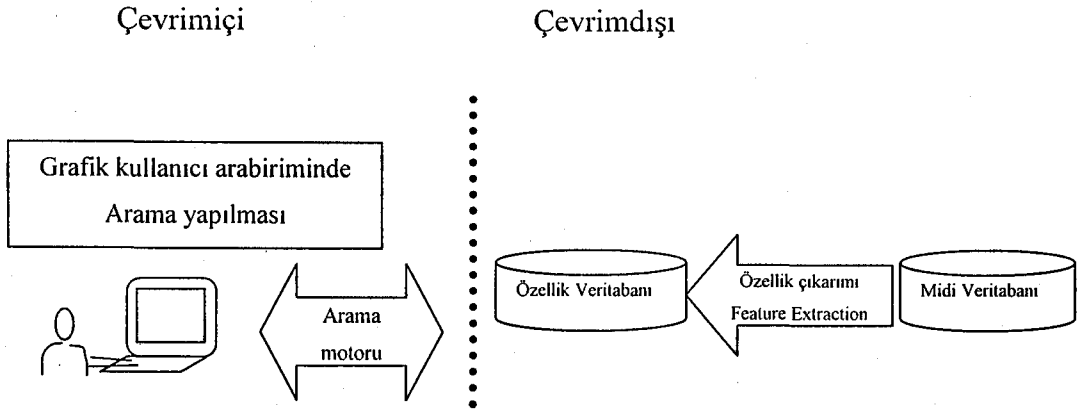
Arama motoru tasarımında bulunan nota geçiş farkları kullanılmıştır. Nota geçiş farklarının dosya ismi olarak verildiği ve geçiş farklarının birleştirilmiş veritabanında geçtiği yerlerin numaralarının bulunduğu bir metin dosyası veritabanı oluşturulmuştur. Esas arama işlemi bu veritabanı üzerinde yapılmıştır. Bununla birlikte birleştirilmiş midi veritabanında dosya boyutlarının tutulduğu

ayrı bir dosya boyut (lookup table) tablosu oluşturulmuştur. Arama algoritması olarak Viterbi arama algoritması kullanılmıştır. Bu algoritma ile aranılan melodi dizisine en yakın maliyetteki (cost) dizinin bulunması için kullanılmıştır. Arama motoru Microsoft Visual Basic 6.0 programlama dilinde yazılmıştır.

Veritabanının istenilen formata getirilmesi ve bu format için tasarlanmış arama motorunun oluşturulmasından sonra kullanıcının sisteme bilgi girişi yapabilmesi ve sonuçları görmesi için düzenlenen grafik kullanıcı arabirimi oluşturulmuştur. Grafik kullanıcı arabirimi Microsoft Visual Basic 6.0 programlama dilinde oluşturulmuştur.

## 4.2 İDMA Sisteminin Bölümleri

İDMA sistemi kullanıcı açısından iki bölüme ayrılabilir: 1-Çevrimdışı (Off line), 2- Çevrimiçi (On line). Çevrimdışı işlemleri kullanıcı görmez. Bunlar programın yöneticisi tarafından yapılan işlemlerdir. Çevrimdışı işlemler veritabanının istenilen formata çevrilmesi içindir. Bu işlemler program ilk defa çalıştırılırken ve veritabanına yeni midi dosyaları eklendiğinde yapılması gerekir. Yani veritabanının güncelleştirilmesi işlemidir. Çevrimiçi işlemler ise aramanın yapılmasıdır. Bu işlemler kullanıcının programı kullandığı sırada (On line) gerçekleştirilir. Çevrimiçi işlem içerisinde kullanıcıdan bilgi alınması yani aranacak melodinin açılması ya da melodi dizisinin girilmesi işlemleri vardır.



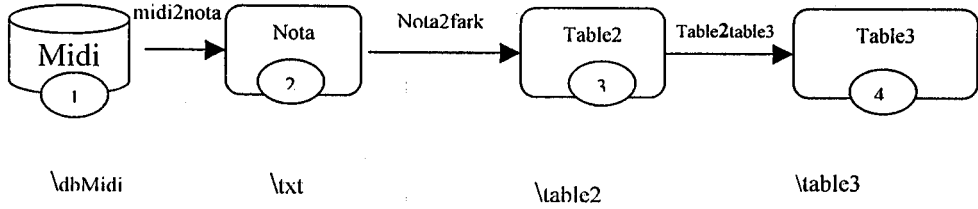
*Şekil 4.1 İDMA sisteminin bölümleri*

### 4.3 İDMA Sisteminin Oluşturulması

İDMA sisteminin oluşturulması üç bölümden meydana gelmektedir.

#### 4.3.1 Veritabanının Oluşturulması

İDMA sistemi için melodi kayıt formatı olan midi dosyaları kullanılmıştır. Midi dosyaları melodi ile ilgili tüm bilgileri içermektedir. İDMA sisteminde kullanılacak bilgilerin alınması ve özelliklerin midilerden çıkarılması gereklidir. İDMA sisteminde nota bilgileri midilerden alınıp, melodi arama için kullanılacak olan nota geçiş farkları çıkarılmıştır. Bu yüzden midi dosyalardan nota veritabanına geçilmiştir. Sonra nota bilgilerinden de nota geçiş farkları veritabanı oluşturulmuştur. Arama sisteminin tasarımına uygun olabilmesi için her midiye ait nota geçiş farkı dosyalarından, birleştirilmiş veritabanı düşünülerek her geçiş için bir dosya atayarak, bu dosyalar içerisinde de o nota geçiş farkının geçtiği sıra numarası kaydedilmiştir.



**Şekil 4.2 İDMA sistemi veritabanının oluşturulması**

Şekil 4.2 de görüldüğü gibi veritabanının oluşturulması dört bölümde gerçekleştirilmektedir. Birinci bölümde tek-sesli (monophonic) midi dosyalarından oluşan veritabanı oluşturulmuştur. Midi dosyası ikilik dosyadır (binary file). İkilik dosyaları onaltılık (hexadecimal) sayı sisteminde gösterilir.

```

000000 4D 54 68 64 00 00 00 06 00 01 00 0B 00 78 4D 54 MThd.....xMT
000010 72 6B 00 00 00 AD 00 FF 59 02 00 00 00 FF 51 03 rk.....Y.....Q.
000020 0B 71 B0 00 FF 58 04 04 02 18 08 CE 60 FF 51 03 .q...X.....\Q.
000030 0B 71 B0 1C FF 51 03 0B 96 C5 1D FF 51 03 0B BC .q...Q.....Q...
000040 CE 1C FF 51 03 0B E3 D4 1D FF 51 03 0C 0B E1 1C ...Q.....Q.....
000050 FF 51 03 0C 35 00 1D FF 51 03 0C 5F 3A 1C FF 51 .Q..S...Q..._...Q
  
```

**Şekil 4.3 Midi dosyasının içeriği**

Midi dosyaları iki bölükten (chunk) oluşmuştur. Birinci bölük Header Chunk diye isimlendirilir. Bu bölükte midi dosyasının kimlik bilgileri ve midi dosyasının formatı kayıtlıdır. İkinci bölük Track Chunk diye isimlendirilir. Bu bölükte ise melodinin kendisini oluşturan midi ve meta mesajları (events) vardır. Midi dosyasının spesifikasyonları Ekler-A da verilmiştir.

Midi mesajları içerisinde nota bilgisi gönderilirken her on oktav içerisindeki her notaya bir sayı verilmiştir. Sıfırıncı oktavın birinci notası (C0) “0” ve onuncu oktavın son notası (G10) da “127” sayısı ile temsil edilir.

iDMA projesinde midi dosyaları \dbMidi klasoru altında tutulmaktadır.

İkinci kısımda midi mesajlarından nota bilgisi alınmıştır. Nota bilgisi alınırken Note On ve Note Off mesajlarından yararlanır. Fakat burada her midi mesajı değişken uzunluk gecikme zamanı (variable length delta time) ile birlikte

gelir(Ek-A). Projede bu dosyalar \txt klasörü altında tutulmaktadır. Bir dosya içeriği aşağıda verilmiştir:

71 72 65 76 65 69 72 76 67 71

Üçüncü bölümde nota bilgisinden notalar arası geçiş farkı alınır. Notalar bir sayı ile ifade edildiği için iki nota arası geçiş iki sayı arasındaki fark ile alınmıştır. Projede bu dosyalar \table2 klasörü altında tutulmaktadır. Bir dosya içeriği aşağıda verilmiştir:

1 -7 11 -11 4 3 4 -9 4

Dördüncü bölümde sıralı olarak bulunan nota geçiş farkları dosyalarından her geçiş farkının veritabanında geçme sırasına göre yeni bir veritabanı daha oluşturulur. Arama motorunda da bu veritabanı kullanılmaktadır. Projede bu dosyalar \table3 klasörü altında tutulmaktadır. Bir dosya içeriği aşağıda verilmiştir:

4 12 15 33 42 44 46 48 51 57 60 71 74 78 148 173 211 229 239  
244 290 300 305 315 325 333 343 351 359 371 459 461 468 491 505  
507 518 521 545 551 553 557 574 577 580 692 696 699 701 703

### 4.3.2 Arama Motorunun Oluşturulması

Arama motoru bölümü kullanıcı açısından çevrimiçi olarak çalışan kısımdır. Bu kısımda kullanıcıdan aranacak melodi alınır. Aranacak melodi iki şekilde girilebilir: 1- Kayıtlı bir midi dosyası olarak 2- Elle metin olarak nota dizisi olarak. Birinci şekilde girilen melodi içerisinde nota bilgisinin çekilmesi ile ikinci şekilde aynı duruma, yani nota dizisi haline getirilir.

Nota dizisi halindeki arama melodisi veritabanına kaydedilen melodilerle aynı işlemlerden geçirilerek arama yapılabilecek hale getirilir. Yani nota geçiş

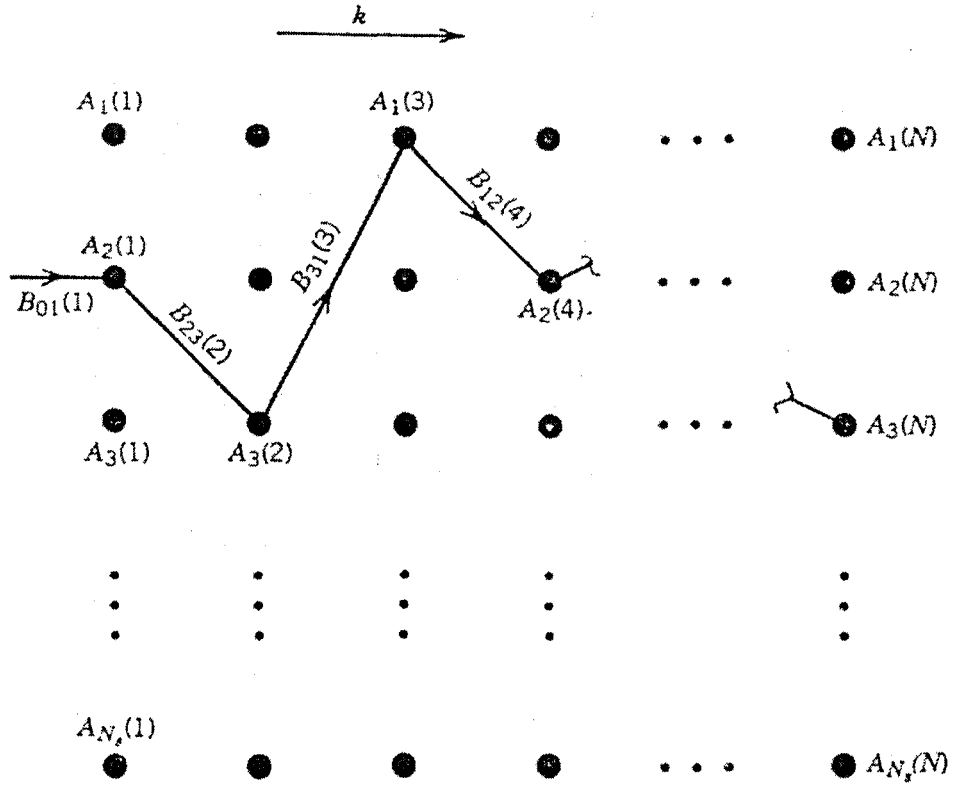


farkı aranacak melodi için de bulunur. Bundan sonra arama motoruna girilen bu nota geçiş farkı dizisini veritabanında bulunması süreci başlar.

Aranacak nota geçiş farkı dizisini veritabanında aramak için Viterbi algoritması kullanılmıştır. Viterbi algoritması Hidden Markov Model ile ses tanımadan, en kısa yol bulmadan, dizi arama (sequence detection) ya bir çok problemin çözümünde kullanılır. Viterbi algoritması en olası durum dizisini (the most likely state sequence) bulmada en etkili yollardan birini sağlar [14].

#### 4.3.2.1 Viterbi algoritması

Viterbi algoritması en olası durum dizinin bulunması için kullanılan algoritmalarından biridir. Kafes (trellis) grafiği ile Viterbi algoritmasını anlamak daha rahat olacaktır:



Şekil 4.4 Viterbi algoritmasının kafes sistemi gösterimi

Bu grafik  $N$  düğümlü zaman indeksli olarak  $k$  ile temsil edilen sütunlardan oluşmuştur. Her sütun içinde  $N_s$  mümkün durum bulunmaktadır. Kafes içerisindeki bir yol belli bir durum dizisine karşılık gelmektedir. Bunlardan bir tanesi Şekil 4.4 da gösterilmiştir. Her düğüm için  $A_i(k)$  ile gösterilen ağırlık (weight); her bir dal (branch) için  $B_{ji}(k)$  ile gösterilen ağırlık atanmıştır. Bir yolun ağırlığı yol içinde bulunan tüm düğüm ve dal ağırlıklarının toplamıdır. Bir yolun ağırlığı şu şekilde gösterilir:

$$\sum_{k=1}^N A_i(k) + B_{ji}(k)$$

Viterbi algoritması kafes içindeki birinci sütundan  $N$ . sütuna çizilen yollardan en yüksek ağırlığa sahip olan yolu seçmek olarak yorumlanabilir. Bu yollar içerisinde en optimalinin seçilmesi şu şekilde gerçekleşir:  $k$  sütunundan bir önceki sütun için ağırlığı hesaplanmış olarak düşünürsek, yani  $k-1$  nci sütunun her düğümü için bir numaralı sütundan itibaren en optimal yolun ağırlığı  $W_j(k-1)$  olarak hesaplanır. Kafes grafiğinde  $k$  nci sütundaki her düğüm  $i$  için optimal yol bulunurken  $k-1$  inci sütundaki düğüm  $j'$  den düğüm  $i$  ye çizilen yollardan aşağıdaki formülle verilen ağırlığı maksimize eden yol seçilir [15]:

$$W_{j'}(k-1) + B_{j'i}(k) + A_i(k)$$

Bu yolu bulabilmek için 11.62 de verilen ağırlığın  $k-1$  nci sütundaki her düğüm için hesaplanarak maksimize eden yol bulunur. Bu işlem sonunda  $k$  nci sütundaki her düğümün ağırlığı şöyle hesaplanır [15].

$$W_i(k) = \max_{j'} \left[ W_{j'}(k-1) + B_{j'i}(k) \right] + A_i(k)$$

Diğer sütunlar içinde aynı işlemler yapılarak her düğüm için ağırlık hesaplanır. Her düğüm için ağırlık hesaplandıktan sonra birinci sütundan  $N$  nci sütuna en optimum yol kolaylıkla bulunur. Bunu yapmak için  $N$  nci sütundaki en

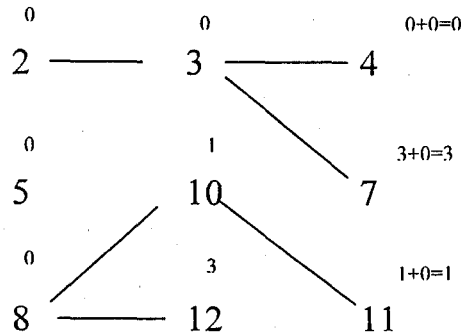
büyük ağırlığa sahip düğüm bulunur ve bu düğümden geriye doğru yol takip edilerek optimal yol bulunmuş olur.

Bu algoritmanın iDMA sisteminde nasıl kullanıldığını yada nasıl uygulandığını anlamak için iDMA sistemindeki durumu anlatmak yerinde olur. iDMA sisteminde aranacak melodinin nota geçiş farkları bulunduktan sonra, her fark elemanı Viterbi kafes grafiğinde bir sütunu oluşturur. Örnekle açıklarsak:

Aranacak melodinin nota dizisi: C2 D2 A1 E2, 24 26 21 28

Aranacak melodinin nota geçiş fark dizisi: 2 -5 7

Veritabanında 2, -5 ve 7 farklarının geçtiği yerleri kafes grafiğine yerleştirirsek:



**Şekil 4.5 iDMA sistemi için viterbi kafes grafiği**

Bu kafes grafiğinden de görüldüğü gibi son sütunun her düğümü için maliyet (cost) hesaplanmıştır. Dallar arasındaki maliyet ise düğümlerin sayı farkının bir eksigidir. En son sütundan bir önceki sütunun tüm düğümlerine çizilen yollardan en az maliyetli olan seçilmiştir. Son sütundaki en düşük maliyetli düğümden geriye gidilerek en az maliyetli yol bulunmuş olur.

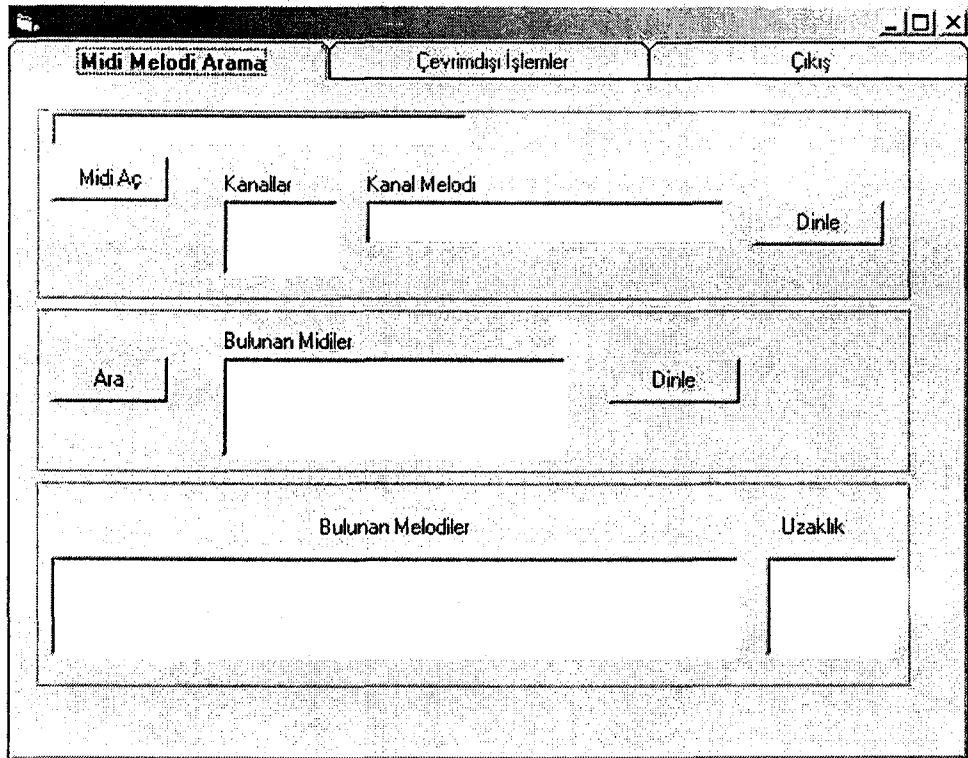
iDMA sistemi için bulunan en kısa yol eğer sıfır maliyetli ise veritabanında melodinin tam eşleşmesi bulunmuş olur. Sıfır maliyetten fazla çıktığında ise melodiye uzaklığı belirlenen eşik değerinden fazla değilse bu bulunan yol yani dizide arama sonucu listesinde gösterilir.

Yani yukarıdaki örneğe göre bulunan 2 3 4 yolu, melodi arama sistemi için dizi diyebiliriz, aranan melodinin veritabanında geçtiği yeri göstermektedir. Veritabanındaki 2. notadan başlayan dört notanın bulunduğu midi bu aranan

melodi ile aynıdır. Çünkü bu yolun maliyeti (uzaklığı) sıfırdır. Yani bu aranan nota dizisi arka arkaya veritabanında geçtiği yer bulunmuştur. Bununla birlikte, 8 10 11 yolu da 1 maliyeti vardır. Yani aranan melodiye benzemekte ama notalar arasında farklı bir nota girmiştir.

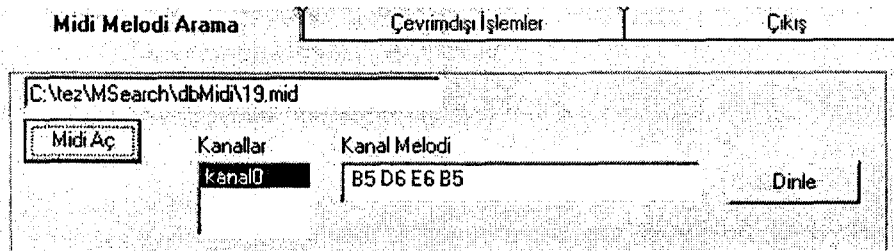
#### 4.4 Programın Çalışması

İDMA sistemi Microsoft Visual Basic 6.0 programlama dilinde yazılan diyaloga dayalı (dialog-based) bir programla gerçekleştirilmiştir. Program kullanıcı yönünden iki kısma ayrılmıştır: Midi Melodi Arama ve Çevrimdışı İşlemler. Program sekme parçalı dizayn edilmiştir. Yukarıda bahsedilen bölümler için bir sekme ayrılmıştır. Program bu iki sekmeye ek olarak programı sonlandırmak için konulmuş üçüncü sekme ile beraber toplam üç sekmeden oluşmuştur. Program ilk çalıştırıldığında Midi Melodi Arama penceresi açık olarak gelir.



Şekil 4.6 İDMA programının grafik kullanıcı arabarimi(GKA)

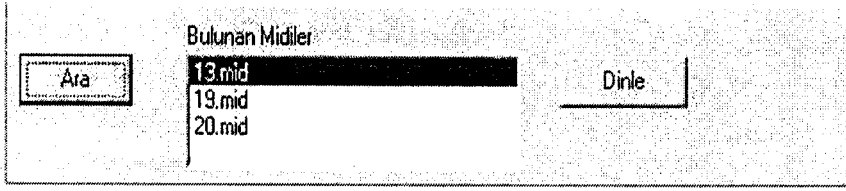
Aranacak melodi midi dosya formatında “Midi Aç” butonuna tıklanarak ve “Kanal Melodi” metin kutusu içerisine Amerikan Nota Notasyonu kullanılarak nota dizisi girilerek olmak üzere 2 şekilde girilebilir. Nota dizisi girilirken iki nota arasına boşluk, “,” , “.”, “-” karakterleri kullanılabilir. Açılan midi dosyası monofonik (monophonic) ise tek kanal olacaktır ve o kanal’ın içeriği “Kanallar” liste kutusunun içerisine yazılır. Eğer polifonik (polyphonic) ise yani aynı anda bir çok kanaldan nota basılıyorsa, bu kanalların hepsi liste kutusuna aktarılır. İDMA sistemine sadece bu kanallardan çalınan melodilerden bir tanesi girilebilir. Seçilen kanalda çalınan melodinin nota karşılıkları “Kanal Melodi” içerisine yazılır.



**Şekil 4.7 İDMA GKA’sında aranacak melodinin girilmesi**

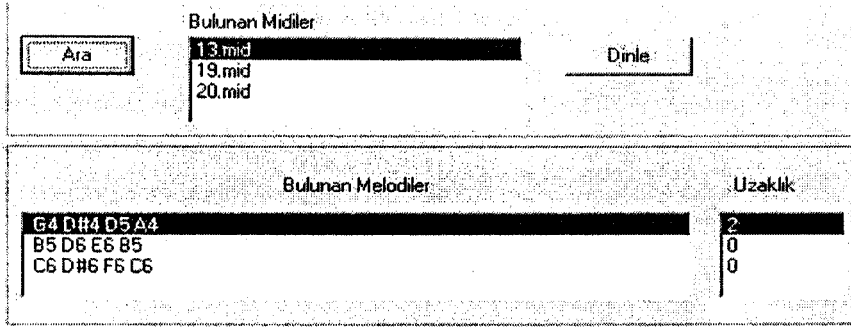
“Dinle” düğmesi ile açılan midi dosyasını program içerisinden dinlemek mümkündür. Midi dosyasını dinlemek için Microsoft MCI (Multimedia Controller Interface) kullanılmıştır. Bu yüzden bu programla beraber aynı anda çalıştırılmış bir MCI’yi kullanan hatalı programa dikkat etmek gerekir. Eğer bu program açıkta ve MCI’ya yapılan bağlantıyı kapatmamış ise dinle düğmesi ile midi dosyasını çalmamız mümkün olmayabilir. Bu yüzden bu programlara dikkat edip mümkünse aynı anda çalıştırılmamalıdır.

Aranacak melodi açıldıktan sonra “Ara” düğmesine basılarak İDMA arama motoru çalıştırılır. Dosya veritabanı içerisinde bulunan midiler sıralanır.



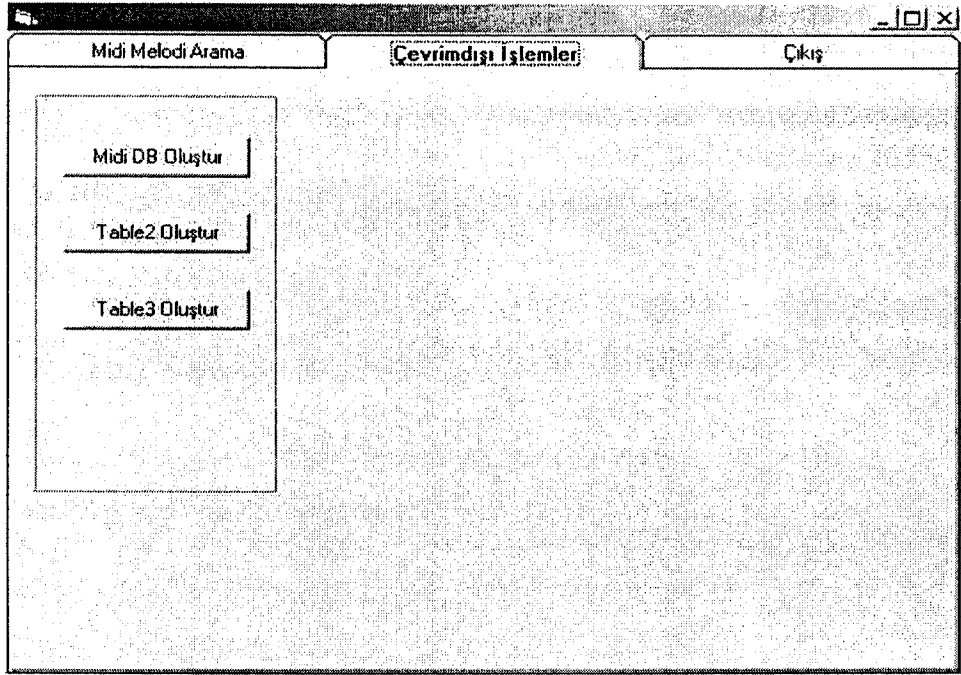
**Şekil 4.8 İDMA GKA'sında arama sonucunun dosya isimi olarak görüldüğü kısım**

Bulunan midiler liste kutusu içerisinde listelenen midilerden istenilen midi fare ile tıklanarak seçilir ve yanında bulunan “Dinle” düğmesi ile çalınabilir. Seçili olan midi dosyasının nota içeriği “Bulunan Melodiler” liste kutusu içerisinde aktif hale getirilir.



**Şekil 4.9 İDMA GKA'sında arama sonucunun melodi içeriği ve uzaklık bilgisi ile görüldüğü kısım**

İkinci sekmede ise veritabanı ile ilgili işlemler bulunmaktadır. Bu işlemler kullanıcıya göre kullanıcı çevrimdışı iken yapıldığı için çevrimdışı işlemler olarak adlandırılmıştır. Program ilk çalıştırıldığında midi dosyalarından oluşan veritabanından aramada kullanılacak özellik veritabanının oluşturulması gerekir. Bu işlem uzaktan erişimli istemci-sunucu mimarisinde kullanıcıyı ilgilendirmeyen bir işlemdir. Bu yüzden bu işlemler çevrimdışı işlemler olarak gruplanmıştır. Veritabanı yöneticisi tarafından yeni bir melodi girildiğinde bu işlemler tekrarlanarak veritabanının güncellenmesi gerekir.



*Şekil 4.10 İDMA GKA'sında dosya veritabanı işlemlerinin yapıldığı sekme*

## 4.5 İDMA Programının Yapısı

iDMA programı Visual Basic programlama dilinde yazılmıştır. Visual Basic dilinin özelliği olarak İDMA programı da olay tetiklemeli (event driven) yapıdadır. Toplam ana 6 olay vardır. Bunları çevrimiçi bölümünde 3 ve çevrimdışı bölümünde 3 olmak üzere iki kısma ayırabiliriz.

Çevrimdışı bölümünde 3 adet komut düğmesine bağlı olaylar vardır.

1.NotaDbOluştur\_Click 2.Tablo2oluştur\_Click 3. Tablo3Oluştur\_Click

NotaDbOluştur içerisinde dosya isimleri 1 rakamından başlayarak sıralı atlama yapılmamış \dbMidi klasörünün altındaki midi dosya topluluğundaki midilerin nota bilgilerini içeren txt dosyaları \txt klasörü altında oluşturulur.

```
On Error GoTo HandleErrors
```

```
While Not a = 1
```

```
filenum = FreeFile
```

```
Open app.Path & "\dbMidi\" & CStr(i) & ".mid" For Input As #filenum
```

```

Close #filenum
Call Midi2Nota(app.Path & "\dbMidi\" & CStr(i) & ".mid", app.Path & "\txt\" & CStr(i) & ".txt", kanal())
i = i + 1
Wend

```

Burada  $i = 23$  için 23.mid dosyası açılmakta ve 23.txt dosyasına nota içeriği yazılmaktadır. Eğer 24.mid yoksa On Error GoTo HandleErrors satırı çalıştırılıp HandleErrors satırına atlanmaktadır. HandleErrors satırında da açık bulunan dosya bağlantıları Close komutu ile kapatılıp altyordam sonlandırılır. Açılan midi dosyasının içeriği Midi2Nota yordamı ile yapılmaktadır. Her midi dosyası için bu yordamı çağrılmaktadır.

Midi2Nota yordamı içerisinde midi dosyası açılır ve midi dosyasını oluşturan bölükler BölükBaytlarınıAl fonksiyonu ile alınır. ChunkBytes ve ChunkLength değişkenlerinde bölüğün içeriği ve uzunluğu döndürülür.

```
ErrCod = BölükBaytlarınıAl(ChunkBytes, ChunkLength, InFileNum)
```

Sonra bu bölük içeriği işlenir.

```
Call BolukIsle(ChunkID, ChunkLength, ChunkBytes(), OutFileNum, kanal())
```

Burada işlenecek iki çeşit bölük vardır: HeaderChunk (MThd) ve Trackchunk (MTrk). Nota bilgisi MTrk bölümü içerisinde geçmektedir.(Ek-A)

```
Case "MTRK"
```

```
Posn = 1
```

```
ErrCod = MidiMesajIsle(ChunkBytes, Posn, OutFileNum, kanal())
```

```
End Select
```

MTrk bölümü içeriği ise MidiMesajIsle fonksiyonu ile işlenir. Bu fonksiyon içerisinde bütün midi mesajlarına bakılarak nota bilgisi çekilir ve \txt klasörünün içerisine midi dosyasının ismi ile aynı olan txt dosyası oluşturulur.

```
Case &H90 To &H9F 'Note On
```

```
NoteNum = ChunkBytes(Posn)
```

```
Posn = Posn + 1
```

```
Velocity = ChunkBytes(Posn)
```



```

Posn = Posn + 1
If Not (Velocity = 0) Then
    Print #OutFileNum, NoteNum;

```

Onaltılık sistemde H90-H9F olarak gelen midi mesajları Note On Mesajıdır. Eğer notanın hızı sıfır ise Note On değil Note Off olmuş olur bu yüzden hızın sıfır olmadığı mesajların içerisinde gelen nota numaraları alınmıştır.

Tablo2Oluştur altyordamında ise NotaDbOluştur altyordamının benzeri olarak \txt klasörü altında sıralı dizilmiş txt dosyaları açılıp, \Tablo2 klasörü altına nota geçiş farkları yazılır. Bunun için txt dosyası içindeki birbiri arkasına gelen iki rakamın farkı alınıp yeni txt dosyasına yazılır.

Tablo3oluştur altyordamında ise her \tablo2 klasörü altındaki txt dosyasının içinde geçen nota geçiş farkı için \tablo3 klasörü altında bir dosya oluşturulup nota geçiş farklarının geçtiği sıra numarası yazılır. Bunun yanında her nota geçiş dosyasının içerisinde kaç adet geçiş olduğu LookUpTable adlı dosyaya yazılır. Bu dosya arama sonucunda bulunan dizinin hangi dosya içerisinde geçtiğini bulmak için kullanılır.

Çevrimiçi bölümünde ise 3 adet komut düğmesine bağlı olay vardır: 1- MidiAç\_Click 2- Dinle\_Click 3- Ara\_Click

MidiAç içerisinde kullanıcının aranacak midi dosyasını bilgisayar içerisinde bulup, açması sağlanmaktadır. Bunun için common dialog nesnesi kullanılmıştır. CD1 isimli common dialog nesnesi bilgisayar içerisinden midi dosyasının yeri ve ismini kullanıcıdan kolayca alınmasını sağlamaktadır.

```
InFileNam = DosyaIsmiAl(CD1, "Midi Dosyası Aç", "*.mid", "Open", "*.mid", LastDir, Cncl)
```

DosyaIsmiAl fonksiyonu ile yeri ve ismi alınan midi dosyasının yukarıda anlatılan Midi2Nota yordamı ile nota içeriği alınır. Kullanıcıya kanal sayısı ve kanalda geçen nota dizisi gösterilir.

Dinle\_Click içerisinde ise MCI (Multimedia Controller Interface) kullanılarak midi dosyası çalınmıştır. MCI'yı kullanmak için mciSendString fonksiyonunun deklare edilmesi gereklidir.

```
Private Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal
lpstrCommand As String, ByVal lpstrReturnString As String, ByVal uReturnLength As Long, ByVal
hwndCallback As Long) As Long
```

Aşağıdaki kodda W yerine dosya ismi ve diğer değişkenlere gerekli değerleri vererek midi dosya ile bağlantı kurulur ve sonraki kod ile midi dosyası çalınır.

```
i = mciSendString("open sequencer!" & W & " alias midi", RS, 128, cb)
```

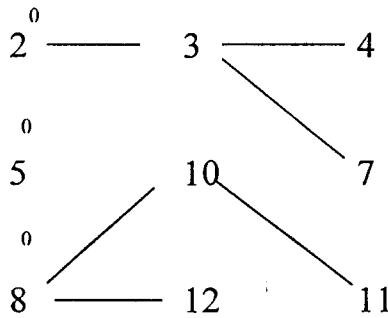
```
i = mciSendString("play midi ", RS, 128, cb)
```

Ara\_Click içerisinde açılan midi dosyasının nota dizisi ya da elle girilen nota dizisi Nota2Num fonksiyonu ile nota numaralarına çevrilir. Yani:

```
C2 D2 A1 E2 aranacak dizi için 24 26 21 28
```

Daha sonra NotaFark fonksiyonu ile nota geçiş farkı çıkarılır. Yukarıdaki örnek için : 2 -5 7'dir.

Bu nota nota geçiş farkı dizisi ViterbiSearch yordamı çağrılarak arama yapılır. Yukarıdaki örnek üzerinde açıklarsak: 2 -5 7 dosyaları açılır, böylece 2 geçişinin -5 geçişinin ve 7 geçişinin olduğu yerler alınmış olur. 2 geçişi 2. 5. ve 8. sırada; -5 geçişi 3. 10. 12. sırada ve 7 geçişi 4. 7. 11. sırada olmuşsa aşağıdaki kafes gösterimi meydana çıkacaktır:



bu kafes grafiğine göre viterbisearch yordamında gidebilecek tüm yolları içeren bir indis değişkeni tanımlanmıştır. Indis ( len(2), len (-5), len(7)) boyutundadır. Yani bu örnek için indis (3,3,3) boyutundadır.

İlk sütünün elmanları ile ikinci sütünün elamanları arası uzaklıklar hesaplanır. Olası olan yollar bulunur.

3 için 2; 10 için 8 ve 12 için 8 alınması yani 2-3, 8-10 ve 8-12 alınmıştır. İkinci sütün ve 4. sütün için 4 için 2-3 yolu; 7 için 2-3 yolu, 11 için 8-10 yolu alınmıştır. Burada hep rakamlar arası fark uzaklık maliyeti olarak kullanılmıştır.

## 4.6 Sonuçlar

iDMA sisteminde midi dosyalarından oluşturulmuş melodi veritabanında nota geçiş farkı özelliği kullanılarak içeriğe dayalı arama yapılmaktadır. Arama algoritması olarak Viterbi algoritması kullanılmıştır.

iDMA sistemi ile aynı melodinin notalarının kaydırılarak çalınmış hali, daha yüksek sesle çalınmış hali ve aynı notaların sürelerinin değiştirilerek çalınması ile elde edilen melodiler bulunabilmektedir. Bununla birlikte, aynı melodinin içerisinde yanlışlıkla araya giren farklı notaların olması durumunda çıkan melodilerde bulunmaktadır. Bu durumda aranan melodiye olan uzaklık bilgisi programda kullanıcıya verilmektedir.

## 5 Sonuç ve Öneriler

İDMA sisteminde doğrudan nota bilgisi özellik olarak kullanıldığı için arama veritabanı büyüdüğüne oldukça vakit almaya başlamaktadır. Bu sorun bu şekilde tam eşleşmenin yani aynı melodinin kesin olarak bulabilen sistemlerde yaşanan bir problemdir. Eğer bu özellikten vazgeçilir mesela melodik perde konturu kullanılırsa sistem daha hızlanacaktır.

iDMA sistemi ile teksesli melodiler içerisinde arama yapılmıştır. Bu sistemin ileriki aşaması olarak çok sesli sistemler üzerinde çalışma yapılabilir. Tabii böyle bir sistemde çok sesliden tek sesli ye geçmek mantıklı olacaktır. Diğer türlü direkt olarak çoksesli müzik içerisinde arama yapmak oldukça zordur.

Müzik arama için düşünülebilecek diğer bir aşama ise makam (structure) üzerine arama yapabilecek bir sistem tasarlanabilir.

Bu çalışma ile içeriğe dayalı arama kavramına girilmiştir. Bu sistemin kullanılma alanı daha fazla olan içeriğe dayalı ses (sound) aramaya geçilebilir. İçeriğe dayalı ses arama, müzik aramaya göre çok farklıdır. Farklılık ses için bulunan özellik ve dolayısıyla arama motorunun farklı olmasından ileri gelir. Fakat iki sisteminde yapısı aynıdır. Bu sistem sayesinde ses arama konusuna, içeriğe dayalı müzik arama geçiş basamağı olarak kullanılabilir.

## EKLER

### EK-A Midi Dosya Formatı Özellikleri

Standart Midi Dosyası (Standart Midi File SMF) enstrümanlar tarafından gönderilen bilgileri kaydedip çalmaya yarayacak şekilde dizayn edilmiş bir dosya türüdür.

Bu dosya biçiminde, standart MIDI mesajları (durum bayt ve data bayt şeklinde) ve ek olarak her mesaj için zaman etiketi (her olayın gerçekleşmesi için ne kadar saat atımı (clock pulse) bekleneceğini gösteren bayt topluluğu) kullanılır. Bu format tempo, çeyrek nota için atım süresi (birim/saniye cinsinden), zaman, anahtar ve iz (track) isimleri kaydetmek mümkündür.

Midi dosya formatında bilgiler bölükler (chunks) içerisinde kaydedilir.

SMF = <başlık\_bölüğü> + <iz\_bölüğü> [+ <iz\_bölüğü> ...]

Midi dosyası başlık bölüğü ve iz bölüğü olarak iki çeşit bölük vardır. Bir midi dosyasında bir çok bölük olabilir. Her bölüğün boyutu, yani kaç bayttan oluştuğu, farklıdır. Her bölük 4 karakter kimlik (ID) ile başlar. Bu ID bölüğün tipini belirtir. Sonraki 4 bayt (32 bit) bölüğün boyutunu (bayt sayısını) belirtir<sup>1</sup>. Bütün bölükler bu iki 4 baytlık bölüm ile başlamalıdır. Bu ilk bölüme bölük başlığı (chunk header) denir.

başlık\_bölüğü = "MThd" + <başlık\_boyutu> + <format> + <n> + <bölüm>

Örnek bir başlık (onaltılık sistemde gösterilmiştir):

4D 54 68 64 00 00 00 06

İlk 4 bayt ascii olarak bölük ID'si olan **MThd** (yani ascii olarak 4D 'M', 54 'T', 68 'h', ve 64 'd'). İkinci 4 bayt ise 6 baytlık bilginin daha geleceğini söylüyor.

Bir dosyanın midi dosyası olup olmadığını ilk 4 baytı okuyup Mthd olup olmadığını kontrol edilmedi.

<sup>1</sup> Dosya boyutu bu ilk 8 baytı içermez.

Bu MThd başlığından sonra 6 baytlık bilginin geleceğini gösteren 4 bayt gelir. Bu 6 bayt şunlardır:

İlk iki bayt midi formatını yada tipini gösterir. Üç değişik midi formatı vardır: 0, 1, 2 rakamları ile temsil edilir. 0 formatı 16 kanal üzerinde çalınabilen tek iz midi dosyasıdır. 1 numaralı format bir ve birden fazla iz içerir ve her iz aynı anda çalınır. 2 numaralı format ise 1 numaralı format gibi bir ve birden fazla iz içerir ve her iz aynı anda başlamak zorunda değildir.

Sonraki 2 bayt kaç tane iz kaydedildiğini gösterir. Format 0 için bu bilgi 1 dir. Diğer iki format için birden farklı olabilir.

En son iki bayt ise çeyrek nota için kaç saat atımı (Pulses Per Quarter Note PPQN ) kullanıldığını gösterir. Örneğin: 96 ppqn

00 60

olarak gösterilir. 1/ppqn ise bölüm (division) olarak adlandırılır.

Aşağıda bütün bir MThd bölümü örneği verilmiştir:

4D 54 68 64	MThd ID
00 00 00 06	MThd bölümünün uzunluğu her midi için 6'dır
00 01	Format tipi 1
00 02	2 adet iz (track)
E7 28	Gecikme süresi (Delta-time) 1/PPQN bir milisaniye.

MThd bölümünün arkasından MTrk bölümü gelir. MTrk bölümü bir iz için bütün midi bilgilerini içerir. MThd bölümünde geçen iz sayısı bilgisi kadar MThd bölümü bulunmalıdır.

MTrk bölümü de aynı MThd gibi 'MTrk' kimliği ile başlar. Sonra da iz'in boyut bilgisi gelir. Her iz için farklı olabilir.

iz\_bölümü = "MTrk" + <uzunluk> + <iz\_olayı> [+ <iz\_olayı> ...]

İz olayı ise en son ki midi olayından sonra gelme (gecikme) süresi (delta-time) ve 3 tane midi olayından birinden oluşur.

iz\_olayı = <delta\_time> + <midi\_olayı> | <meta\_olayı> | <sysex\_olayı>

Delta-time: Değişken uzunluk (variable length) olarak en son ki olaydan sonra gecikme (elapsed time) zamanı.

Midi\_olayı: Midi kanal mesajları, örneğin Note on, Note off gibi.

Meta\_olayı: SMF meta mesajları.

SYSEX: SMF sistem exclusive olayları.

Midi olayları: Midi olayları isteğe bağlı konut baytıdan sonra sıfır, bir veya iki baytlık parametre baytı ile devam eder. Koşma (running) modunda komut göz ardı edilir en son ki komut geçerli sayılır. Komut baytının ilt biti her zaman 1 ve parametre baytının ilk biti her zaman 0 olmalıdır. Bu komut bilgisinde son dört bit ile bu bilginin hangi midi kanalına gönderileceği bildirilir. Toplam 6 adet komut vardır:

Hex	İkilik	Parametre	Tanımlama	Açıklama
8n	1000 cccc	nn vv	Note Off	Bir notanın çalınmasını bitirir nn=Nota No vv=Hız
9n	1001 cccc	nn vv	Note On	Bir notanın çalınmasını başlatır nn=Nota No vv=Hız
An	1010 cccc	nn vv	KeyAfter-touch	nn=Nota No vv=Hız
Bn	1011 cccc	rr vv	Control Change	rr= Kontrol vv= Yeni parametre
Cn	1100 cccc	pp	Patch Change	Kanala atanan enstrümanı değiştir Pp= Enstrüman
Dn	1101 cccc	vv	Channel After-touch	Vv=Hız
En	1110 cccc	nn nn	Pitch Change	Kanalın perdesini (pitch) değiştirir. x'2000' değişiklik yapmaz. Parametre lsb,msb formatında kaydedilir: x'2000', 00 20 olarak.

**Meta olayları:** Meta olayları direkt olarak midi çalıcılara çalınması için gönderilmez, fakat müzik bilgisinin tamamlanmasını sağlayan başlık, oluşturan2ın isimi, zaman bilgileri ve lirik bilgilerini içerir. Bütün meta olayları x 'FF' ile başlar, bir baytlık olay türünü gösteren bilgi ve ek bilgi varsa onun uzunluğunu gösteren bir bayt ile devam eder. Sonrada uzunluğu verilen ek bilgi kadar ,varsa

eğer, bu bilgiler gelir. Eğer olay ek bilgi içermiyorsa, yani ek data uzunluğu 0 ise, bu meta olayı 3 bayt uzunluğundadır. Meta olaylarının listesi aşağıdadır:

<i>Olay</i>	<i>Uzunluğu</i>	<i>Bilgi</i>	<i>Tanımlama</i>	<i>Açıklama</i>
FF 00	02	nnnn	Sequence Numarası	Format 2 midi dosyalarında kullanılan şarkının bir bölümünün sırasını belirtir. İsteğe bağlı bir olaydır ve format 0 ve 1 de kullanılmaz.
FF 01	nn	text	Text	Metin olarak herhangi bir bilgi, mesela şarkıcının isimi gibi.
FF 02	nn	text	Kopya Hakkı	Kopya hakkı ile ilgili bilgi ilk izin başında verilmesi daha uygundur.
FF 03	nn	text	İz ismi	İzin ismini belirtir
FF 04	nn	text	Enstrüman İsmi	İzin hangi enstrümanın çalacağını belirtir.
FF 05	nn	text	Lirik	Şarkının sözleri girilir
FF 06	nn	text	İşaretleyici	Şarkının bir noktasına isimlendirir
FF 07	nn	text	İşaret Noktası	Perdeler açılıyor gibi olayları açıklar
FF 20	01	cc	Kanal Öneki	Midi ve meta olaylarında kullanılan kanal ismi
FF 2F	00		İz Bitimi	İzin sonunda kullanılan ve izin bitimini gösteren zorunlu bir olaydır.
FF 51	03	ttttt	Tempo ayarla	Çeyrek nota için kaç adet microsaniye olduğunu gösterir

Delta zamanı (Delta-time) ve Değişken Uzunluk (Variable Length):

Midi içerisinde kaydedilen her olay (midi ve meta olayları) değişken uzunluktadır, yani her olayın gerçekleşme zamanı bir önceki olaydan sonra olma zamanı (delta-time) değişken uzunluktadır (variable length). Yani delta zamanı kısa ise bir bayt ile uzun ise daha fazla bayt ile ifade edilmelidir. Ama bu süre bilinmediği için sabit uzunlukta delta zamanı ayırmak, gereksiz bilgi saklamak ve dosyanın boyutunun artması demektir.

Her baytın birinci biti son bayt olup olmadığını gösterir. Eğer 0 ise son ve 1 ise son bayt olmadığını gösterir. Gösterilen değeri bulmak için ise her baytın 1-7 (0 alınmıyor) bitleri alınır ve yan yana birleştirilir. Örneğin:

Onluk sistemde 120 onaltılık sistemde x'78' olarak gösterilir. Bu değer 7 bit ile gösterilebildiği için tek bayt yeterlidir. Böylece x'78' ya da b'01111000' ile

gösterilir. Başka örneklerde aşağıda verilmiştir:



Sayı (hex)	Değişken Uzunluk gösterimi (hex)
00000000	00
00000040	40
0000007F	7F
00000080	81 00
00002000	C0 00
00003FFF	FF 7F
00004000	81 80 00
00100000	C0 80 00
001FFFFFF	FF FF 7F
00200000	81 80 80 00
08000000	C0 80 80 00
0FFFFFFF	FF FF FF 7F

## EK-B

### Nota Numarası- Frekans Eşleşmesi Şeması

Nota	Frekans	Nota	Frekans	Nota	Frekans	
C	0	8.1757989156	12	16.3515978313	24	32.7031956626
Db	1	8.6619572180	13	17.3239144361	25	34.6478288721
D	2	9.1770239974	14	18.3540479948	26	36.7080959897
Eb	3	9.7227182413	15	19.4454364826	27	38.8908729653
E	4	10.3008611535	16	20.6017223071	28	41.2034446141
F	5	10.9133822323	17	21.8267644646	29	43.6535289291
Gb	6	11.5623257097	18	23.1246514195	30	46.2493028390
G	7	12.2498573744	19	24.4997147489	31	48.9994294977
Ab	8	12.9782717994	20	25.9565435987	32	51.9130871975
A	9	13.7500000000	21	27.5000000000	33	55.0000000000
Bb	10	14.5676175474	22	29.1352350949	34	58.2704701898
B	11	15.4338531643	23	30.8677063285	35	61.7354126570
C	36	65.4063913251	48	130.8127826503	60	261.6255653006
Db	37	69.2956577442	49	138.5913154884	61	277.1826309769
D	38	73.4161919794	50	146.8323839587	62	293.6647679174
Eb	39	77.7817459305	51	155.5634918610	63	311.1269837221
E	40	82.4068892282	52	164.8137784564	64	329.6275569129
F	41	87.3070578583	53	174.6141157165	65	349.2282314330
Gb	42	92.4986056779	54	184.9972113558	66	369.9944227116
G	43	97.9988589954	55	195.9977179909	67	391.9954359817
Ab	44	103.8261743950	56	207.6523487900	68	415.3046975799
A	45	110.0000000000	57	220.0000000000	69	440.0000000000
Bb	46	116.5409403795	58	233.0818807590	70	466.1637615181
B	47	123.4708253140	59	246.9416506281	71	493.8833012561
C	72	523.2511306012	84	1046.5022612024	96	2093.0045224048
Db	73	554.3652619537	85	1108.7305239075	97	2217.4610478150
D	74	587.3295358348	86	1174.6590716696	98	2349.3181433393
Eb	75	622.2539674442	87	1244.5079348883	99	2489.0158697766
E	76	659.2551138257	88	1318.5102276515	100	2637.0204553030
F	77	698.4564628660	89	1396.9129257320	101	2793.8258514640
Gb	78	739.9888454233	90	1479.9776908465	102	2959.9553816931
G	79	783.9908719635	91	1567.9817439270	103	3135.9634878540
Ab	80	830.6093951599	92	1661.2187903198	104	3322.4375806396
A	81	880.0000000000	93	1760.0000000000	105	3520.0000000000
Bb	82	932.3275230362	94	1864.6550460724	106	3729.3100921447
B	83	987.7666025122	95	1975.5332050245	107	3951.0664100490
C	108	4186.0090448096	120	8372.0180896192		
Db	109	4434.9220956300	121	8869.8441912599		
D	110	4698.6362866785	122	9397.2725733570		
Eb	111	4978.0317395533	123	9956.0634791066		
E	112	5274.0409106059	124	10548.0818212118		
F	113	5587.6517029281	125	11175.3034058561		
Gb	114	5919.9107633862	126	11839.8215267723		

G 115 6271.9269757080 127 12543.8539514160  
Ab 116 6644.8751612791  
A 117 7040.0000000000  
Bb 118 7458.6201842894  
B 119 7902.1328200980

**NOT:** Frekanslar Hertz cinsindedir.

## Kaynaklar

1. Shauble, P., Content-Based Information Retrieval from Large Text and Audio Databases, Kluwer Academic Publishers, Massachusetts, USA (1997).
2. Wei, C., Melody Retrieval On The Web, Master Tezi, Massachusetts Institute of Tech., USA, 10-22 (2001).
3. Delbimbo, A., Visual Information Retrieval, Morgan Kaufmann Publishers, California , USA (1999).
4. Yoshitaka, A. ve Ichikawa T., A Survey on Content-Based Retrieval for Multimedia Databases, IEEE Transactions on Knowledge and Data Engineering, 11,(1): 81-93 (1999).
5. Petkovic, D., Trends in Image and Video Search. IBM Almaden Research, Stanford, USA (1997).
6. Chock, M., Cardenas A.F. ve Klinger, A., Database Structure and Manipulation Capabilities of A Picture Database Management System, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6, 4:484-492 (1984).
7. Joseph, T. ve Cardenas, A., PICQUERY: A High Level Query Language for Pictorial Database Management, IEEE Transactions on Software Engineering, 14, 630-638 (1988).
8. Stephen, W.S. ve Lynn D.W., Indexing the Content of Multimedia Documents, ACM SIGIR'99 Post-Conference Workshop, 19 Agustos, (1999).
9. McDonagh, J. ve Smeaton, A.F., Multimedia Information Retrieval: MIDI as a format for Content Based Retrieval of Audio, National University of Singapore Press, 1-3, Aralık (1997).
10. Raschke, P., Exploring MIDI, School of Music of Northwestern University Press, USA (2000).

11. Alexandra. L. ve Uitdenbogerd, J.Z., Manipulation of Music For Melody Matching, ACM Multimedia 98 - Electronic Proceedings, USA (1998).
12. Uitdenbogerd, A.L.ve Zobel, J., Manipulation of music for melody matching, Proc. ACM International Conference on Multimedia, USA (1998).
13. Blackburn, S.G., Content Based Retrieval and Navigation of Music Using Melodic Pitch Contours, Doktora Tezi, Univeristy of Southampton, USA, (2000).
14. Hull, J.J. ve Srihari, S.N., Experiments in Text Recognition with Binary n-Gram and Viterbi Algorithms, IEEE. Trans. on PAMI 4, 5, CA, USA (1982).
15. Therrien, C.W., Decision Estimation and Classification, John Wiley & Sons, New York, USA (1989).