



EMO



KTÜ



TÜBİTAK

ÖNSÖZ

Giderek gelenekselleşen Elektrik Mühendisliği Ulusal Kongrelerinin beşincisinde Trabzon'da buluşuyoruz. EMO ile KTÜ Elektrik-Elektronik Mühendisliği Bölümü'nün işbirliği ve İÜBİAK'ın kalkışıyla gerçekleşmekle olan Kongremizin başarılı ve verimli geçmesi umudundayız. kongre sonuçlarından kıvanç duymak istiyoruz.

Kongre'de, büyüme kadar yapılmış çalışmalar ve yayınlanmış duyurulardan da anlaşılacağı gibi, bilinen yöntemlerin yanı sıra gelecek yıllara deneyim aktarabilecek yeni yaklaşımlar uygulanmaya çalışılmıştır. Bildiri özetlerinin değerlendirilmesine katılan uzman sayısının sistematik olarak artırılması, değerlendirme biçiminin dalına da nesnelleştirilmesi, bildiri kitabında yeni yazım ve sunuş biçimlerinin oluşturulması gibi teknik gelişmelerin dışında ilginç olacağı sanılan panellerle güncel sorunların irdelenmesi ve yöresel öğelerle sosyolojik etkinliklerle renk katılması amaçlanmıştır.

Kongrenin hazırlık ve düzenleme çalışmalarında bazı aksaklıklar olmuştur. Öncelikle kongre katarının olması gerekenden dalın geç alınabilmiş olması, özet değerlendirme sürecinin posta trafiğinin çok yoğun olduğu bayram dönemlerine rastlaması hem Yürütme Kurulu'nu hem de Kongre'ye katılmak isteyenleri zor durumda bırakmıştır.

Kongrenin düzenlenmesi sırasında edinilen deneyimler ışığında sorunları çözücü ilkesel önerilerin ortaya konması yararlı olacaktır. Bunları kısaca sıralayabiliriz. Örneğin 6. Kongre'nin ya da kısaca EMUK'95'in nerede ve ne zaman yapılacağını şimdiden kararlaştırmak gerekmektedir. Bundan sonra Konferans olarak adlandırılması daha uygun olacak Kongre için sürekli ya da uzun süre görevli bir 'Ulusal Düzenleme Kurulu'nun oluşturulması ve bu Kurul'un temel ilkesel karar ve yöntemleri üretmesi daha elverişli olacaktır. Kongre'nin yapılacağı konumdaki işleri ise 'Yetel Düzenleme Kurulu' üstlenmelidir. 'Bilimsel Değerlendirme Kurulu'nun da ayrıntılı bir sınıflandırma ve nitelik belirlenmesi ile bir kere oluşturulması, yalnızca gelişen koşullara göre güncelleştirilmesi düşünülebilir.

CMUK, böylesi bir yapılaşma ile daha sağlıklı, zaman planlaması daha verimli bir konferansa dönüşecekli kanısındayız. Örneğin bu durumu bildiri tam metninin de değerlendirme ve etimoloji sürecine gitmeleri olanaklı kılınacak, şu ana kadar ancak Yürütme Kurulu'nun ayıntılı olarak bilincine varabildiği teknik sorunlar ortadan kalkacaktır. Konferansda da içerik ve düzey açısından belirli bir iyileştirme sağlanabilecektir. Bunu en yakında, EMUK'95'de gerçekleşmiş olarak görmek dileyimizdir.

Bilindiği gibi Kongremiz Elektrik, Elektronik-Haberleşme, Kontrol ve Bilgisayar Sistemleri alanlarında bilimsel-teknolojik örgün katkılarının (attırılıp değerlendirilmesi ile araştırma, geliştirme, uygulama ve eğitim süreçlerindeki kişi ve kuruluşların bir birleriyle doğrudan iletişimini saklamayı amaçlamaktadır. Ayrıca sosyal yakınlaşma ve dayanışmaya da

katkıda bulunmaktadır. Ancak Kongre ve onunla birlikte oluşturulan sergi/fuların çok değerli bir 'Mesleki Eğitim ve Geliştirme' aracı olduđu bilincinin kiři ve kurumlarda daha çok yerleşmesi için çaba gösterme geređi de ortaya çıkmaktadır.

Kongrenin gerçekleşmesini sağlayan, hazırlık ve düzenlemeleri üstlenen KTÜ, EMO ve TÜBİTAK'a, oluşturulmuş olan kurulların üyelerine, ayrıca burada adlarını saymakla bitmeyecek kiři ve kamu - özel - akademik nitelikli kuruluşlara, yardım ve katkıları nedeniyle, Kongre'nin yararlı sonuçlarını paylaşacak olan topluluğumuz adına teşekkürlerimizi sunmak isteriz.

Kongremizin başarılı ve verimli bir biçimde gerçekleşmesi, ülkemiz için bilimsel - teknolojik kazanımlar üretmesi dileđiyle Yürütme Kurulu olarak saygılarımızı iletiriz.

Doç. Dr. Güven ÖNBİLGİN
Yürütme Kurulu Başkanı

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

YÜRÜTME KURULU

Güven ÖNBİLGİN (KLU)
Yakup AYDIN (EMO) Sefa AKPINAR (KTU)
Canan TOKER (ODTU) Kaya DOZOKLAR (EMO)
Hasan DINCER (KTU) A.Oğuz SOYSAL (IU)
Abdül İlah SEZGİN (KTU) İrfan SENLİK (EMO)
Kenan SOYKAN (EMO) Y.Nuri SEVGEN (EMO)

DANIŞMA KURULU

Rasim ALDEMİR (BARMEK) Mehmet KESİM (Anadolu U)
Teoman ALPTURK (TMMOB) Macit MUTAF (EMO)
Ahmet ALTINEL (TEK) Erdiñç ÖZKAN (PTT)
İbrahim ATALI (EMO) Kamil SOGUKPINAR (TETSAN)
Malik AVİRAL (ELİMKO) Sedat SİSBOT (METRONİK)
Emir BIRGUN (EMO) Atıf URAL (Kocaeli U.)
Sıtkı CİĞDEM (EMO) İ. Ata YİĞİT (EMO)
R. Can ERKÖK (ABB) Fikret YÜCEL (TELETAS)
Bülent ERTAN (ODTU) Hamit SERBEST (CU)
Uğur ERTAN (BARMEK) Canan TOKER (ODTU)
İsa GÜNGÖR (EMO) Nusret YUKSELER (İTU)
Ersin KAYA (Kaynak) Kemal DZMEHMET (DEU)
Okyay KAYNAK (Boğaziçi U)

SOSYAL ETKİNLİKLER KURULU

Y. Nuri SEVGEN (EMO)
Necla ÇORUH (PTT) Hatice SEZGİN (KTU)
Esen ÖNKİBAR (TEK) Yusuf TANDOGAN (PTT)
Abdullah SEZGİN (KTU) Ömer K. YALCIN (TELSER)

SEKRETERLİK HİZMETLERİ

Necmi İKİNCİ (EMO) Elmas SARI (EMO)

BİLİMSEL DEĞERLENDİRME KURULU

Cevdet ACAR (I TU)
İnci AKKAYA (I TU)
A.Sefa AKPINAR (KTU)
Ayhan ALTINTAS (Bi I .U)
Fuat ANDAY (I TU)
Fahrettin ARSLAN (I U)
Murat ASKAR (ODTÜ)
Abdullah ATALAR (Bi I .U)
Selim AY (YTU)
Umit AYGÖLU (I TU)
Atalay BARKANA (Anadolu U)
Mehmet BAYRAK (Selçuk U)
Atilla BİR (I TU)
Galip CANSEVER (YTU)
Kenan DANIŞMAN (Erciyes U)
Ahmet OERVISOĞLU (I TU)
Hasan D INCER (KTU)
M.Sezai D INCER (Gazi U)
Giinse l DURUSOY (I TU)
Nadia ERDOĞAN (I TU)
Aydan ERKMEN (ODTÜ)
İsmet ERKMEN (ODTÜ)
H.Bülent ERTAN (ODTÜ)
Selçuk GEÇİM (Hacettepe U)
Cem GÖKNAR (I TU)
Remzi GULGUN (YTU)
Filiz GUNES (YTU)
İrfan GÜNEY (Marmara U)
Fikret GÜRGEN (Boğaziçi U)
Fuat GURLEYEN (İTU)
Cemil GURUNLU (KTU)
Nurdan GUZELBEYOĞLU (İTU)
Emre HARMANCI (İTU)
Altuğ İFTAR (Anadolu U)
Kemal İNAN (ODTÜ)
Asım KASAPOĞLU (YTU)
Adnan KAYPMAZ (İTU)
Ahmet H. KAYRAN (İTU)
Mehmet KESİM (Anadolu U)
Erol KOCAOĞLAN (ODTÜ)
Muhammet KÖKSAL (İnönü U)
Hayrettin KÖYMEN (Bil. U)
Hakan KUNTMAN (MU)
Tamer KUTMAN (IfU)
Duran LEBLEBİCİ (I TU)
Kevork MARDIKYAN (İTU)
A.Faik MERGEN (I TU)
Avni MORGUL (Boğaziçi U)
Güven ÖNBİLGİN (KTU)
Bülent ÖRENCİK (İ TU)
Bülent ÖZGÜC (Bi I .U)
A.Bülent ÖZGÜLER (Bi I .U)
Yılmaz ÖZKAN (I TU)
Muzaffer ÖZKAYA (I TU)
Kemal ÖZMEHMET (DEU)
Osman PALAMUTCUOĞLU (İTU)
Erdal PANAYIRCI (İTU)
Halit PASTACI (YTU)
Ahmet RUMELİ (ODTÜ)
Bülent SANKUR (Boğaziçi U)
M.Kemal SARIOĞLU (I J"U)
Müzeyyen SARI TAS (Gazi U)
A.Hamit SERBEST (CU)
Osman SEVAİOĞLU (ODTÜ)
A.Oğuz SOYSAL (IU)
Taner SENGÖR (YTU)
Emin TACER (I TU)
Nesrin TARKAN (İ TU)
Mehmet TOLUN (ODTÜ)
Osman TONYALI (KTU)
Ersin TULUNAY (ODTÜ)
Nejat TUNCAY (I TU)
Atif URAL (Kocaeli U)
Alper URAZ (Hacettepe U)
Gökhan UZGÖREN (IU)
Yiğit UCTUĞ (ODTÜ)
Asaf VAROL (Fırat U)
Sıddık B. YARMAN (IU)
Mümtaz YILMAZ (KTU)
Melek YÜCEL (ODLU)
Nusret YUKSELER (I TU)
Selma YUNCU (Gazi U)

Taflan linre Gündem

Boğaziçi Üniversitesi. Bilgisayar Mühendisliği Bölümü

ÖZET

Bu bildiri, bağıntısal veri tabanlarında giriş ve çıkışların optimal bir şekilde işlenmesini sağlayacak dosya yapılarının otomatik olarak elde edilmesini içeren bir yöntem sunmaktadır. Yöntemde ilk önce giriş ve çıkışlarla ilgili istatistiksel bilgiler kullanılarak bir küme aday dosya yapısı üretilir. Daha sonra bu kümenin en optimal ali kümesi global olarak seçilir. Yöntem sonuca polinom zaman içerisinde ulaşır.

ANAHTAR SÖZCÜKLER

Veritabanı fiziksel tasarımı, optimizasyon; ağırlıklı çizge eşlendirme yöntemi.

1. GİRİŞ

Bağıntısal veritabanları günümüzde yaygın olarak kullanılmaktadır. Özellikle büyük bağıntısal veritabanlarının randımanlı çalışması dosya yapılarının iyi seçilmesine bağlıdır. Bağıntısal veritabanlarının fiziksel ve mantıksal seviyelerinin birbirinden bağımsız olması fiziksel seviyede islediğimiz dosya yapılarını kullanma olasılığını sağlamakta ve sonuç olarak randımanı artıracak dosya yapılarının seçimi problemi karmaşıklaşmaktadır.

Geçmişle dosya yapıları tasarımıyla ilgili önemlilerde araştırma yapıldı. S.T.March [1] yazısında dosya yapılarıyla ilgili çalışmaları özetlemiştir. Bu yazısında kayıtların ayrılması ("segmentation") ve birleştirilmesi ("aggregation") ilgili problemlere çözümler görmekteyiz.

Bununla beraber, kayıtların birleştirilmesiyle ilgili çözüm sadece hiyerarşik veri modelleri için geçerli. Kayıtların ayrılmasıyla ilgili çözümlerde ise ya hiyerarşik metodlar kullanılmış ya da kısmi ayrılma incelenmiştir.

Dosya yapının seçimiyle ilgili diğer önemli bir çalışma da Katz ve Wong [2] tarafından yapılmıştır.

*Bu çalışına Boğaziçi Üniversitesi Araştırma Fonu tarafından 90 HA 0138 nolu proje olarak desteklenmiştir.
ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Bu çalışmalarında Katz ve Wong dosya yapılarının soyut özelliklerini fonksiyonel veri modeli kullanarak modelliyorlar. Daha sonra tamsayı programlama problemi kullanılarak dosyaların kullanım sıklığını optimize eden bir optimizasyon problemi çözülüyor. Bilindiği üzere tamsayı programlama problemi en kötü durum analizinde eksponansiyonel zaman almaktadır.

Bizim çalışmamızda elde ettiğimiz metodoloji özellikle bağıntısal veri modeli için geliştirilmiş olup probleme global bir optimizasyon problemi olarak yaklaşmaktadır. Metodoloji algoritmik olarak elverişli ve randımanlı olup aynı zamanda yaygın olan somut dosya yapılarını göz önünde bulundurur. Genel olarak çalışmamız literatürdeki diğer çalışmalardan aşağıda özellikler açısından farklıdır.

i) Herşeyden önce geliştirilen metodolojiden bağıntısal veri modeline uygulanabilir.

ii) Ayrıca aday olacak dosya yapılarının seçimi gösterilmiş olup bu seçim veritabanının giriş ve çıkış özelliklerine bağlandırılmıştır.

iii) Çok önemli olarak belirtilmesi gerekir ki, global optimizasyon, veritabanına alt toplam giriş ve çıkışların işleme masrafını asgariye indirme kriteri üzerine kurulmuştur. Bunun aksine literatürde görülen dosya yapılarının kullanım sıklığını azamiye ulaştırma kriteri, her zaman veritabanı giriş ve çıkışlarının işleme masrafını asgariye indirecek veri yapılarının seçimini sağlamamaktadır.

iv) Yaygın olarak kullanılan somut dosya yapısı tipleri seçilebilmektedir.

v) Global optimizasyon problemi çizge eşlendirme problemi olarak formüle edilerek elverişli bir şekilde çözülüyor. Bilindiği üzere çizge eşlendirme problemleri polinom zamanda çözüm üretmektedirler. Örneğin Papadimitriou [7] tarafından verilen çizge eşlendirme yöntemi en kötü durumda, n'nü düğüm sayısı olduğu varsayılırsa n^m mertebesi zaman içerisinde çözüm üretmektedir.

Geliştirilen metodlar programlanmış ve denenmiştir. Geliştirilen program giriş olarak dosyalarla ilgili formülleri, veritabanı planı, giriş çıkış şekilleri ve sıklıkları, bağıntı büyüklükleri, gibi verileri almakta ve hangi bağıntıların ne şekilde dosyalanabildiğini gösteren aday dosya yapılarını üretmektedir.

Üretilen adayların en optimal alt kümesi bir çözüme eşlendirme yöntemi kullanılarak elde edilebilmektedir.

2. TEMEL KAVRAMLAR VE TERİMLER

Bağıntı bir veya daha fazla **değer kümesinin** kartezyen çarpımının bir alt kümesidir. Bağıntı tanımı, $S (A_1, A_2, \dots, A_n)$, bağıntı İsmi, S , ve bağıntının isimlendirilmiş değer kümeleri olan **bağıntı niteliklerini**, A_1, A_2, \dots, A_n içerir. S bağıntısının bir **elemanı**, $l(u_1, U_2, \dots, u_n)$, n kısımdan meydana gelir ve her bir kısım, u_j , bir bağıntı niteliği, A_j ile ilişkilidir. Bağıntısal veritabanı ise birbiriyle ilişki bağıntıların tümüdür. **Bağıntı anahtarı** bir bağıntının tanımında adı geçen bağıntı niteliklerinin, en az bir elemanı olan, bir alt kümesidir. Bir anahtar meydana getiren bağıntı niteliklerinin alt kümesi anahtar olamaz. Bir bağıntı için, o bağıntının herhangi iki elemanının, anahtarı meydana getiren bağıntı nitelikleriyle ilgili kısımları aynı değerlere sahip olamaz. Bu yüzden bağıntı anahtarı bir bağıntının elemanlarını özbeöz olarak tanımlar. **Bir kullanım şekli** aşağıda belirtilen özelliklere sahip bir bağıntı elemanları kümesidir: Orijinde bulunan ortalama bağıntı elemanları sayısı, kullanım sıklığı, ilijülüşüm tipi (bağıntı elemanı eklemek, silmek veya sadece okumak) ve içiçerilen bağıntı elemanlarını betimleyen bağıntısal cebir işlemi (çakışma, fark, toplam, seçme, projeksiyon, kartezyen çarpımı ve birleştirme). Her kullanım şekliyle ilişkili bağıntısal cebir işleminin fiziksel bağıntılara uygulanması sonucu ortaya çıkan bağıntı elemanları, o kullanım şeklinin **erimi** yani içerdigi bağıntı elemanları kümesidir. Bir veritabanı ile ilgili kullanım şekillerinin tümü o veritabanının toplam kullanım şeklini gösterir.

Bir **dosya** birbiriyle ilişkili kayıtların ardarda gelecek şekilde fiziksel bilgisayar sayfalarına konmuş şeklidir. Bağıntıların fiziksel olarak bilgisayarda saklanması dosyalarla olur. Her bir bağıntı elemanı bir kayıtla ilişkilidir. Değişken olmayan kayıtlarda genelde her kayıt bir bağıntı elemanını içinde bulundurur. Bu durumda kayıtların her bir alanı bir bağıntı niteliğiyle ilişkilidir.

Bu tasarımda kullandığımız dosya yapıları tipleri **yığıma, sıralama** ve **gruplandırma**dır. Yığıma dosya yapısında kayıtlar boş bulunan yerlere ardarda

220

yerleştirilirler. Her bir bağıntı için bir dosya ve her bir bağıntı elemanı için bir kayıt bulunur. Sıralama dosya yapısı yığıma dosya yapısının bir veya daha fazla bağıntı niteliğine göre sıralanmasıdır. Gruplama dosya yapısında birbiriyle ilişkili birden fazla bağıntı bir dosya içerisinde saklanır. Bu tasarımda biz sadece iki bağıntıyı bir dosya içerisinde saklıyoruz. Her bir kayıt değişken boylu olup, A bağıntısından bir eleman ilişkili olduğu B bağıntısı elemanlarıyla aynı mantıksal kayıt içerisinde saklanır.

Veri yapılarında **genelleştirme**, birden fazla benzer fonksiyonlu veri yapısını bir araya getirip tek bir veri yapısıyla değiştirmektir. Genelleştirme ile ilgili daha fazla bilgi Gündem [3] den elde edilebilir.

Aday dosya yapısının seçiminde, önce elde bulunan kullanım şekilleri teker teker dosya yapılarına dönüştürülür. Daha sonra genelleştirme uygulanarak aday dosya yapılan elde edilir. Böylece dosya yapıları sadece kullanım şekillerinden elde edilmiş olur. Bu yüzden doğru seçilmiş kullanım şekillerinin elde bulunması çok önemlidir.

3. KULLANIM ŞEKİLLERİNİN DOSYA YAPILARINA DÖNÜŞÜM METODLARI

Aday dosya yapılarının elde edilmesi aşağıda belirtilen iki evreli dönüşümlerle yapılır. Tasarım sonunda her bir bağıntı elemanının bir fiziksel yeri bulunması için elimizde her bağıntıyla ilgili en az bir kullanım şekli bulunması gerekmektedir.

3.1. Birinci Evre Dönüşümler

Elimizde bulunan kullanım şekillerinin, betimlendiği bağıntısal cebir işlemine göre, dosya yapılarına dönüştürülmesi aşağıda belirtildiği gibidir.

B. 1. Seçme (F,R): R bir bağıntı ismi ve F ise $x.op.y$ şeklinde bir formüldür. Formülde x bir sabit, y ise R tanımında bulunan bir bağıntı niteliği ve $op >.=.< v.s.$ gibi bir matematiksel karşılaştırma işlemidir. Seçme işlemi R bağıntısından F'yi doğru kılan bağıntı elemanlarını seçer. Seçme işlemiyle betimlenen kullanım şekilleri sıralama dosya yapısına dönüştürülür. Sıralama F de belirtilen y bağıntı niteliğine göredir.

B.2. Projeksiyon (A,R): R bir bağıntı ismi ve A ise R tanımı içerisinde bir bağıntı niteliğidir. Projeksiyon işlemi R bağıntısı elemanlarının A ile ilgili kısımlarını alır. Sonuç olarak meydana gelen bağıntı elemanları sadece bir tek kısımlıdır. Bu işlemle betimlenen kullanım şekilleri yığıma dosya yapısına dönüştürülür.

B.3. Toplam (R₁,R₂). Fark (R₁,R₂) ve Çakışma (R₁,R₂): R₁'e 1×2 bağıntı isimleridir. Toplam, fark ve çakışma işlemleri küme kuramındaki aynı isimli işlemlere benzerdir.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Yukardaki işlemlerle betimlenmiş bir kullanım şekli biri R_1 ve diğeri R_2 üzerinde iki ayrı sıralama dosya yapısına dönüştürülür. Her iki dosya yapısında da sıralama R_1 ve R_2 nin ortak anahtar bağıntı niteliklerine göre dir.

B.4. Birleştirme (R_1, R_2, G): R_1, R_2 bağıntı isimleri ve G a.op.b şeklinde bir formüldür, a, R, ile ve b, R_2 ile ilgili bağıntı nitelik isimleridir, op $>, =, <$ v.s. gibi bir matematiksel karşılaştırma işlemidir. Sonuç olarak meydana gelen bağıntı elemanları (r_1 R_1 'in nitelik sayısı r_2 de R_2 'nin nitelik sayısı olduğunu varsayarsak) $n+r_2$ kısımlıdır. Sonuçtaki her bağıntı elemanının ilk r_1 kısmı R_1 den geri kalan r_2 kısmı R_2 den bir bağıntı elemanıdır ki bu elemanlar G formülünü doğru kılarlar. Birleştirme işlemiyle betimlenmiş bir kullanım şekli G formülündeki op işlemine göre aşağıda belirtilen dosya yapılarına dönüştürülür.

DEğer op = işlemlyse ve a veya b anahtar ise, kullanım şekil gruplandırma dosya yapısına dönüştürülür. Bu dosya yapısının kayıtları değişken uzunlukla olup her bir mantıksal kayıta (a'nın anahtar olduğunu varsayarsak) bir R_1 elemanı, x, ve bunun yanında x için G'yi doğru kılan R_2 elemanları bulunur.

2)Eğer op = den farklı bir işleme, kullanım şeklinin biri R_1 için diğeri de R_2 için olmak üzere iki ayrı sıralama dosya yapısına dönüştürülür. R_1 üzerindeki dosya yapısı a, R_2 üzerindeki de b bağıntı niteliğine göre sıralandırılır.

B.5. Kartezyen çarpımı (R_1, R_2): R_1 ve R_2 bağıntı isimleridir. Sonuç olarak meydana gelen bağıntı elemanları (r_1 R_1 'in ve r_2 de R_2 nin nitelik sayısı olduğu varsayılırsa) $n+r_2$ kısımlıdır. Sonuç bağıntısının her elemanının ilk r_2 kısmı R_1 'den ve geri kalan r_1 kısmı R_2 'den gelir. Kartezyen çarpımıyla betimlenmiş bir kullanım şeklini biri R_1 ve diğeri R_2 üzerinde olmak üzere iki ayrı yığın dosya yapısına dönüştürürüz.

3.2. İkinci Evre Dönüşümler

Birinci evre dönüşümler sonucu elde ettiğimiz dosya yapılarının bazılarını genelleştirme kavramını kullanarak bir araya getiririz.

E1. Aynı bağıntı üzerinde olan ve aynı bağıntı niteliğine göre sıralandırılmış bütün sıralama dosya yapılarını, genelleştirme kullanarak tek bir sıralama dosya yapısına çeviririz.

E2. Aynı bağıntı üzerinde olan bütün yığın dosya yapılarını o bağıntı üzerinde tek bir yığın dosya yapısına dönüştürürüz.

4. ADAY DOSYA YAPILARININ

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

4.1. Giriş

Kullanım şekillerinden dosya yapılarına dönüşümler sonucunda mantıksal veritabanı modellerimizde her bağıntıyla ilgili olarak yığın, sıralama ve gruplandırma tipi dosya yapılarından en az birini aday olarak elde ederiz. Elde edilen dosya yapılarının bağıntılara göre dağılımını **yerleştirme çizgesi** dediğimiz bir çizgeyle gösterebiliriz. Çizgenin **düğümüleri** veritabanındaki bağıntıları temsil ederler. İki düğüm arasındaki bir **ayrıt** ise İki düğümün temsil ettiği bağıntılar üzerine kurulmuş bir grupta dosya yapısını gösterir. Bir düğümden çıkıp tekrar aynı düğüme dönen bir **özayrıt** ise ya bir yığın ya da sıralama dosya yapısını gösterir. Ayrıtlar üzerindeki ağırlıklar ayrıtın temsil ettiği dosya yapısının masrafını gösterir.

Bir dosya yapısının masrafı İki kısımda incelenebilir: i) Dosya yapısına bağımlı masraf, ii) kullanım şekline bağımlı masraf. Birinci kısımda dosya yapısının meydana getirilme ve daha sonraki bakım masrafları hesaplanır. İkinci kısımda ise dönüşümler sonucu dosya yapısını meydana getiren kullanım şekillerinin, meydana getirdikleri dosya yapısı kullanılarak işlenmeleri sonucu kullanılan bilgisayar sayfa erişim sayısı hesaplanır. Maliyet hesapları bilgisayar sistemine bağımlı olup değişik hassaslıkta parametrik formüller elde edilebilir. Şahin [4]'de örnek bir masraf tablosu çıkarılmıştır.

Örnek olarak Tablo 1'de gösterilen üç bağıntılı bir veritabanı aday dosya yapıları Şekil 1'deki yerleşim çizgesi ile gösterilebilir.

Kabaca, dosya yapıları optimizasyonu problemini, verilen kullanım şekilleri kümesinin toplam işleme masrafını asgariye indiren ve kendi arasında çelişkili olmayan, aday dosya yapılarının ali kümesinin seçimi olarak tanımlıyabiliriz. Bu problemin çözümünü teşkil eden dosya yapıları kümesinde, her bir bağıntı için sadece bir dosya yapısı bulunur. Böylece her bir bağıntı elemanı fiziksel olarak sadece bir yerde saklanır.

Dönüşümler sonucu ortaya çıkan ve aday dosya yapılarını gösteren yerleşim çizgesine aşağıda belirtilen elemeleri yapabiliriz. Bu elemeler en optimal aday alt kümesinin seçimini engellemez ve aynı zamanda aday sayısını azaltıcı bir etkisi vardır.

Elemeler

1) Her hangi İki düğüm arasındaki ayrıtlardan en az ağırlıklı olanı hariç diğer bütün ayrıtları elimine ederiz.

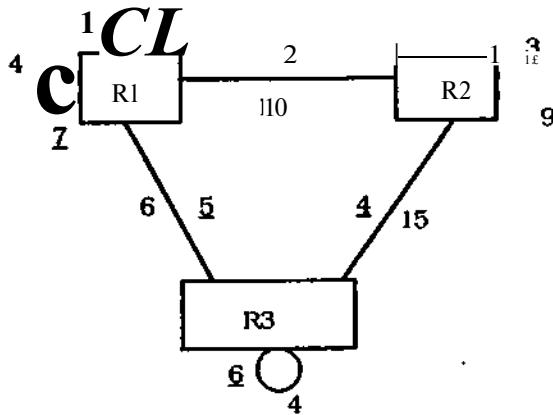
2) Herhangi bir düğüm için eğer birden fazla özayrıt varsa, en az ağırlıklı olan hariç diğer bütün ayrıtları elimine ederiz.

Bundan sonraki kısımlarda yukarıdaki elemelerin yerleşim çizgesine uygulanmış olduğunu varsayalım. Ayrıca yerleşim çizgesinin tek bir çizge olduğunu da varsayalım. Eğer birden fazla çizge varsa, küresel optimizasyon her bir çizgeye değerlerinden bağımsız olarak tek tek uygulanır.

Tablo 1
Aday dosya yapıları

Dosya No	Dosya Tipi	İlişkili Bağlantılar
1	Yığılma	R1
2	Gruplama	R1.R2
3	Sıralama	R2
4	Gruplama	R2. R3
5	Gruplama	R3. R1
6	Yığılma	R3
7	Sıralama	R1

Dosya No	Dosyanın (toplam masrafı)
1	6
2	10
3	9
4	15
5	6
6	4
7	4



Şekil 1. Tablo 1'deki dosya yapılarının yerleştirme çizgesi

4.2. Dosya Yapıları Optimizasyon Problemi

Düğümle v_i (1. n) olarak gösterilen n düğümlü; ayrıtları $e_j, j \in (1, m)$ olarak gösterilen m ayrıtlı; her ayrıt e_j 'nin masrafı $c(e_j)$ olarak gösterilen ve her düğümle ilişkili bir özayrıtı olan bir yerleşim çizgesi verildiğinde, (herhangi bir k için a (k)'nin sadece 0 veya 1 değeri alabildiğini varsayarsak) dosya yapıları optimizasyon problemi

222

$$\sum_{k=1}^m c(e_k) * a(k) \leq I$$

şağıda belirtilen şarta göre asgariye indirilirdir.

Şart: Her düğüm $v_j, j \in (1, n)$ için sadece bir düğüm e_j vardır ki $a(e_j) = 1$ ve herhangi bir e_x için

$$FS(e_j) = (v_z, y, l, x \in (1, n_1) \text{ ve } z, y \in (1, n))$$

olarak tanımlandığında $(v_j) nFS(e_j) * 1 \leq I$ dir.

FS bize bir ayrıtlı ile ilgili dönüşümleri verir. Optimizasyon problemindeki şart her bağlantı elemanının sadece bir dosya yapısında bulunmasını ve aynı zamanda ayrıtlar arasında çelişki olmamasını garanti eder.

Bir veritabanıyla ilişkili yığılma, sıralama ve yerleştirme dosya yapıları için çelişki aşağıda belirtildiği üzere tanımlanır.

Çelişki Tanımı: $a(e_p) = a(e_r) = 1$ olan iki ayrıtlı e_p ve e_r için eğer $FS(e_p) \cap FS(e_r) \neq \emptyset$ ise e_p ve e_r çelişkilidir.

4.3. 0-1 Tamsayı Programlama Çözümü

Belirtilen optimizasyon problemine bir çözüm, problemi 0-1 tamsayı programlama ("Integer programming") problemi olarak formüle etmektir. Dönüm sayısı n ve ayrıtlı sayısı m olan bir yerleşim çizgesiyle ilgili dosya yapıları optimizasyon probleminin tamsayı programlama problemi olarak formülasyonu aşağıda gösterilmiştir.

$a_i = 0$ veya 1 olduğunu, C_j 'nin dosya yapısı i 'nin masrafı olduğunu ve $C_i > 0$ olduğunu varsayarsak

m
 X $C_i a_i$ "1" aşağıdaki kısıtlamaya göre $I = 1$
asgariye indirge.

Kısıtlama: her $k \in (1, n)$ için

X $a_i = 1$, m k düğümüyle ilgili ayrıtları bulunduran kümedir. Kısıtlama her bağlantıyla ilgili olarak bir ayrıtlı olmasını garantiler.

Tamsayı programlama problemini çözmek için Ballas 151'in algoritması kullanılabilir. Ballas algoritması en kötü durumda eksponensiyel zaman almakla birlikte Hu(61) da belirtildiği üzere, ortalama olarak oldukça randımanlı çalışmaktadır.

4.4 Ağırlıklı Çizge Eşlendirme Çözümü
ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Problemin diğeri bir çözümlü ise optimizasyon problemini ağırlıklı çizge eşlendirme problemine dönüştürmektir. Bunun için 4.1 de belirtilen elemelerin uygulandığı bir yerleştirme çizmesini alırız.

Özayrıt olmanın ayrıtları için ağırlıkları maliyetten kazanca çeviririz. Herhangi bir ayrıt e_x için kazanç, o ayrıtla ilişkili düğümlerdeki özayrıtların masraflarının toplamının, e_x 'in masrafından çıkarılmasıdır. Eğer kazanç negatif ise o ayrıtı çizgeden sileriz.

2) Kazançlar hesaplandıktan sonra özayrıtları sileriz.

3) Geri kalan çizgeye azami ağırlıklı eşlendirme problemini tatbik ederiz. Bir çizge için azami ağırlıklı eşlendirme problemi o çizgeyle ilgili hangi ayrıtları seçelim ki ayrıtların ağırlıkları toplamı ulaşılacak en azami değer olsun ve aynı zamanda her düğümlerle ilgili sadece bir ayrıt seçilmiş olsun sorusuyla eşanlamlıdır. Düğüm sayısı n olan bir çizge için n^2 mertebesinde çözüm üreten algoritmalar Papadimitriou [7] ve Ball [8] de mevcuttur.

5. Sonuç

Açıkladığımız algoritma dolaysız olarak bağımlısal veritabanlarına uygulanabilir. Eldeki kullanım şekli sayısı n olduğu varsayılırsa, algoritmanın aday dosya yaralına kısmı kabaca n mertebesinde zaman alır. Optimizasyon kısmı ise bağıntı sayısı m olduğu varsayılırsa, azami ağırlıklı çizge eşlendirme çözümü kullanılarak m^2 mertebesinde zaman alır. Algoritmanın aday üretme kısmı ACTORD ile kullanılarak Şahin [4] tarafından programlanmıştır. Programdaki masraf formülleri bir tabloda toplandıktan ihtiyaca göre değişebilir.

Kaynakça:

1. March, S. T. , Techniques for Structuring Database Records, Computing Surveys, vol.15, no.1, March 1983, pp.45-79.
2. Kalz, R. H. and Vong, E. , Resolving Conflicts In Global Storage Design through Replication. ACM TODS, vol.8, no.1, March 1983, pp. 110-135.
3. Gündem, T. and Ecklund, E. F. , Jr. , Metamorphosis of Views Into Access Structures in Databases; Aggregation and Generalization, Fourth International Symposium on Computer and Information Sciences, izmir, Oct.1989.
4. Şahin, S. "Database Optimization Using Internal Storage Structure". Boğaziçi Üniversitesi, Bilgisayar Mühendisliği bitirme projesi, 1991.

5. Balas. E. An Additive Algorithm for Solving Linear Programs with Zero-One Variables. Operations Research, vol. 1'i, 1905, pp. 517-526.
6. Hu, T. C. and Robinsou, S. M.. **Mathematical Programming**. Academic Press, 1973.
7. Papadimitriou, C. H. and Steiglitz, K. , **Combinatorial Algorithms and Complexity**, Prentice-Hall, Inc. . New Jersey, 1982.
8. Ball, M. O. and Deigs, U., An Analysis of Alternative Strategies for Implementing Matching Algorithms, Networks, vol. 13, 1983, pp.517-549.



Taflan İ. Gündem 1953 Ankara doğumludur. 1975 de B.Ü. den elektrik mühendisliği ve 1978 de elektrik yüksek mühendisliği derecelerini almıştır. İTÜ EHEB'de asistan olarak çalışmıştır. 1985 de A.B.D. de Oregon State Üniversitesinden bilgisayar bilimleri doktora derecesini aldı. A.B.D. de 1.5 yıl yazılım geliştirme şirketinde çalıştı. 1986 da B.Ü. de yardımcı doçent 1990 da doçent oldu. Veritabanı modelleri ve yönetim sistemleri konusunda araştırmaları vardır.

UZAKTAN İZLEME/DENETİME YÖNELİK KİR İLETİŞİM DENETİM BİRİMİNİN TASARIMINDA ÇOK GÖREVLİ ÇALIŞMA YAKLAŞIMI

Doç.Dr. Bülent ÖRENCİK (İTÜ Elektrik-Elektronik Fak. İS(11111111))
Müh. Mustafa BAŞAK (Teletaş Tel. End. T.A.Ş. İstanbul)

Özet

liu tebliğle Teletaş Tel. lind. T.A.Ş. Ar-(!c (iliç h'Jektroniği Bölümünde geliştirilen telefon santralları güç kaynaklarını denetleme izlemeye yönelik bir iletişim denetim biriminin tasarımındaki çok görevli çalışma yaklaşımı tanıtılmıştır. Ayrıca birimin çevresi ile iletişimi için geliştirilmiş olan özgün Mii'RO® protokolünün yapısı açıklanmıştır.

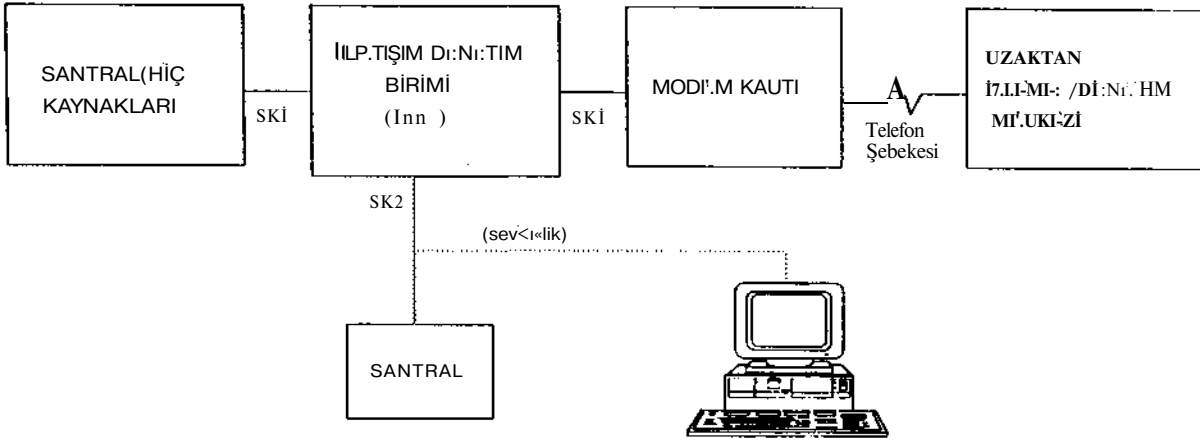
Giriş :

Telefon Santrallarının Güç Kaynakları, akü destekli kesintisiz doğru gerilim üreterek santralları çalıştırarak devamlılığını sağlayan kritik aygıtlardır. Bu aygıtlar, her biri santralları akım gereksiniminin bir bölümünü karşılayan bir dizi modülden oluşur. Son yıllarda güç kaynakları, mikroşleme denetimli sistemler olarak üretilmeye başlanmış ve böylelikle modüllerdeki bozuklukların algılanarak, bunların gereğinde devreden çıkarılması, yedek modüllerin devreye sokulması, operatörlerin sesli ve/veya ışıklı olarak uyarılması kolaylaşmıştır.

Ülkemizde son zamanlarda iletişim alanında görülen yoğun yapılanmanın

sonucunda en uzak yerleşim birimlerine kadar hizmet vermek üzere binlerce yeni santral kurulmaktadır. Bu santralların büyük bir kısmında operatör bulundurmaktan ekonomik olmaktan çıkmış, santralları besleyen güç kaynaklarının uzaktan izlenmesi ve denetlenmesi gündeme gelmiştir.

Teletaş Tel. Cıd. T.A.Ş. firmasının Araştırma - Geliştirme bölümünde yürütülen bir çalışma, güç kaynaklarının uzaktan izlenmesi ve denetimini kolaran bir iletişim denetim biriminin (İDB) tasarımı amaçlanmaktadır. Söz konusu birim, gözlediği güç kaynaklarında oluşan alarm durumlarını, bir seri kanalla bağlı olduğu bir modem kartını sürerek telefon şebekesi üzerinden uzaktan izleme/denetim merkezine bildirebilmektedir. Modem merkezden arandığında da, İDB'yi uyarıp gelen isteklerin güç kaynaklarına ulaşmasını sağlayabilmektedir. Güç kaynaklarına erişim, İDB üzerindeki diğer seri kanal uyarılarak, birime bağlı bir kişisel bilgisayar (PC'den veya telefon santralları kendi denetim bilgisayarından da sağlanabilir. Şekil 1'de iletişim biriminin çevresiyle bağlantı yapısı şematik olarak gösterilmiştir.



Şİ-KIL-1 İLETİŞİM DENETİM BİRİMİNİN ÇKVR SİLLİA ("İL AN 11 YAFISI

2. İDB Donanımı :

İDB donanımı 8 bitlik, kullanımı çok yaygın bir mikrodeneleyici (Intel 8032) tabanlı olarak seçilmiştir. Mikrodeneleyici, kendi üzerindeki seri kanal (SK1) yardımıyla, güç kaynaklarının denetimini üstlenen 16 bitlik diğer bir mikrodeneleyici (Intel 80196) ile bağlantı kurabilmektedir. İDB, bunun yanı sıra gelişmiş bir iletişim denetleyici (Advanced Micro Devices 8530)'nin bir kanalı (SK2) üzerinden sandal bilgisayarına ya da bir PC'ye erişebilmektedir. İletişim denetleyicinin diğer kanalı (SK3) üzerinden ise bir modem kartına ulaşılmaktadır. Modem, İDB'den gelen Hayes AT tipi komutlarla kullanılabilir. İDB'nin denetim programı 32k x 8'lik bir EPROM'a (27C256) kazınmıştır. Yığın, iletişim kuyrukları ve ortak veri alanı için gereken yaz - oku bellek 8k x 8'lik bir RAM (6264) ile sağlanmıştır.

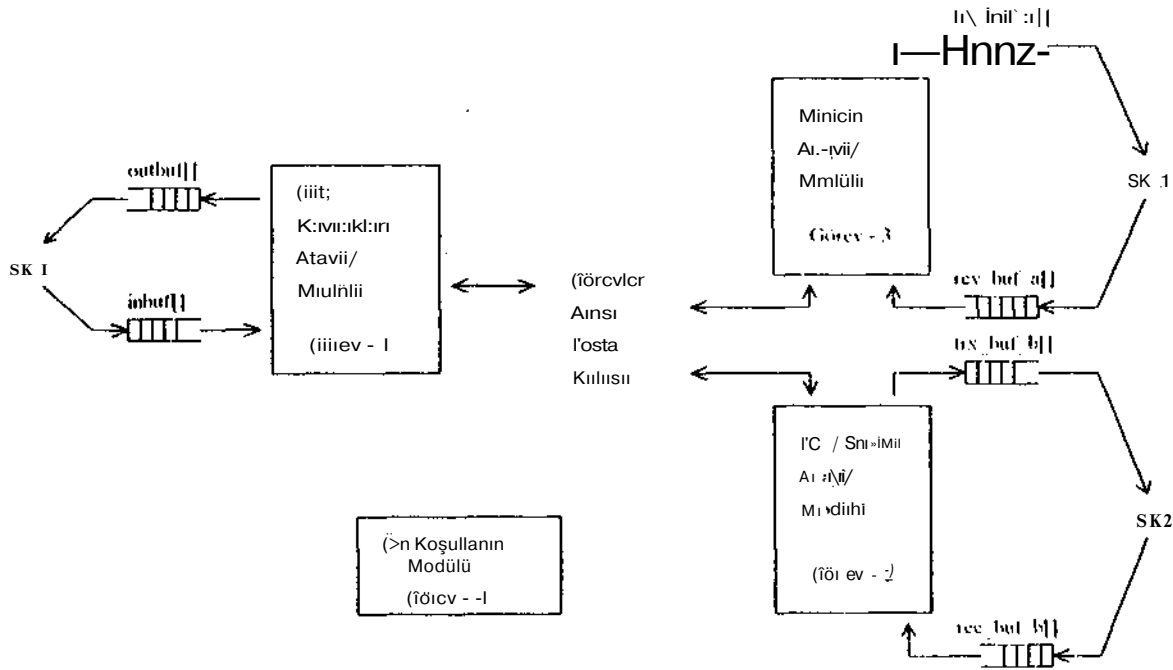
3. Çok Görevli Çalışma Yaklaşımı ve İDB Donanımına Uyarlanması :

Çok görevli çalışma yaklaşımı [1], sistem tasarımını kolaylaştıran, hızlandıran, hataların ayıklanmasını basitleştiren ve tasarımda takım çalışmasını teşvik eden nitelikler içerir. Bu nedenle son yıllarda geliştirilen işlemcilerin bir kısmında çok görevli çalışma ilkelerine ilişkin mekanizmalara rastlanmaktadır (Intel 80x86, Transputer, MDP gibi). Bu mekanizmaların belli başlıları hızlı görev anahtarlama, görevler arası iletişim ilkeleri, iş sıralayıcı, zaman aşımı denetimi gibidir. Öte yandan donanımında bu tür özellikler bulunan ya da bulunmayan işlemciler için çok görevli çalışma ilkelerini içeren araçlar da pazara sürülmeye başlanmıştır. Bu araçların en gelişmişleri gerçek zamanlı işletim sistemleri olarak anılırlar (pSOS-t, MTOS, OS9 gibi). RTX-51 Tiny, Intel 8051 ailesi için geliştirilmiş bu tür araçların en basitlerindedir. Keil

Hlektronik firmasının C5 I Professional Developers Kit isimli yazılım geliştirme ortamının içinde yer almaktadır [2]. Söz konusu araçta, görev yaratma (en çok 16), romıd robıı görev analılarlarına, görevden göreve işaret gönderme, başka bir görevden işaret geldiğinde ya da uzunluğu tanımlanabilir bir süre sonunda tetiklenme gibi ilkeller bulunmaktadır. Görevler arası analılarlarına 10 - 50 ps, iş sıralayıcının bir göreve ayırdığı zaman dilimi 1 - 65 ms arasında olabilmektedir.

İDB'de, özellikleri yukarda kısaca açıklanmış olan RTX-5 I Tiny'nin çok görevli çalışma desteğinden yararlanılmıştır. Şekil 2'de İDB için geliştirilen yazılımın blok gösterilimi verilmiştir.

İDB çalışmaya başladığında (ince Ön



ŞEKİL -2 İDB YAZILIMI BLOK GÖSTERİMİ

Koşullama Modülü (Görev_4) etkinleşir. Bu görev, mikrodenetleyici ve iletişim denetleyici tündeveleinin seri yongalarını ön koşullar ve 6 mesaj kapasiteli görevler arası posta kutusunu temizler. Ardından RTX-51 Tiny'nin os_createtask işlevini kullanarak Güç Kaynakları Arayüz Modülünü (GörevJ), PC/Santral Arayüz Modülünü (Görev_2) ve Modem Arayüz Modülünü (Görev_3) yaratır. Son olarak ta os_deletetask işlevini çağırarak kendini yok eder. Bu andan itibaren iş sıralayıcı yaratılan bu görevleri 5 ms'lik zaman dilimleri süresince romıd robıı düzeninde etkin kılar.

Görev_1 SK1 seri kanalı üzerinden santral güç kaynakları denetim aygıtı ile iletişim kurmak ve sürdürmek ile yükümlüdür. Sonraki bölümde tanıtılacak olan özgün MİPRO

protokolü çerçevesinde, Görev_1'e SKİ üzerinden sorgu (F.NQ) ve bilgi (İINFO) çerçeveleri gelir. Çerçeve karakterlerini inbıf|| kuyruğuna yerleştirmek, iletişim hatalarını belirlemek, geçerli bir İINFO çerçevesi alındığında ilgili bayrağı (mcjcv_nies) birlemek, SKİ kanalına bağlı bir kesme hizmet programıyla (serial()) gerçekleştirilir. Görevi bu bayrağı yoklamak için RTX-51 Tiny'nin os_wait(K_SIG + K_TMO, <zaman aşımı süresi>) işlevini düzenli olarak çağırır. Bu işlev, eğer görevler arası posta kutusunda Görev_1'e herhangi bir mesaj gelmediği takdirde zaman aşımı süresi kadar Görev_1'i bekletmeye yarar. Bu süre programda 20 ms seçilmiştir. Böylelikle Görevi düzenli aralıklarla mc_rcvjnes bayrağını yoklar. Bu bayrağı 1 bulunduğu inbıllj] kuyruğundaki İINFO mesajını görevler arası posta kutusuna taşır ve eğer mesaj modem üzerinden iletilecekse Görev_3'ü, PC/santral bilgisayarına gönderilecekse Görev_2'yi uyarır. Bunun için de RTX-51 Tiny'nin os_send_signal(<Görev_adi>) işlevini kullanır.

Benzer şekilde Görev_1'e yönelik ve Görev_2 ya da Görev_3 tarafından üretilen mesajlar da aynı posta kutusunda bulunur. Bu mesajı üreten görev ayrıca os_send_signal(<Görev_1>) işlevini de yürütür. Bu durumda Görev_1, os_wait(K_SIG + K_TMO, <zaman aşımı süresi>) işlevi ile beklemede

iken uyarılır ve beklemeden çıkar. Bundan sonra Görev_1 posta kutusunu tarayarak kemlisine yönelik mesajı yakalar ve bunu outbul]] kuyruğuna aktarır. serial() kesme hizmet programının bir görevi de outbul]] kuyruğundaki mesajı, ENQ mesajına yanıt olarak karakter karakter SKİ üzerinden yollamaktır.

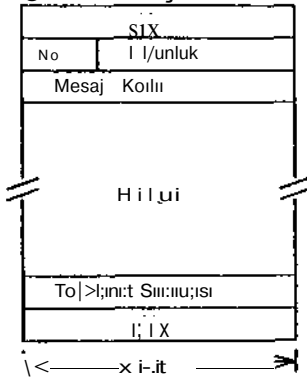
Görev_2 ve Görev_Tün yapıları, Görev_1'inkiyle büyük benzerlik gösterir. Bu görevlerin eriştiği kuyruk yapıları aynı, yalnız isimleri farklıdır. Kuyruklarla seri kanallar arasındaki iletişimi kolaran kesme hizmet programı (inUımc()) da, 8530 iletişim denetleyici lümdevresinin kesme istek çıkışının sürdüğü 8032 dış kesme girişine bağlanmıştır. Ancak yapısı, seial()'inki ile aynıdır.

4. MİPRO İletişim Protokolü :

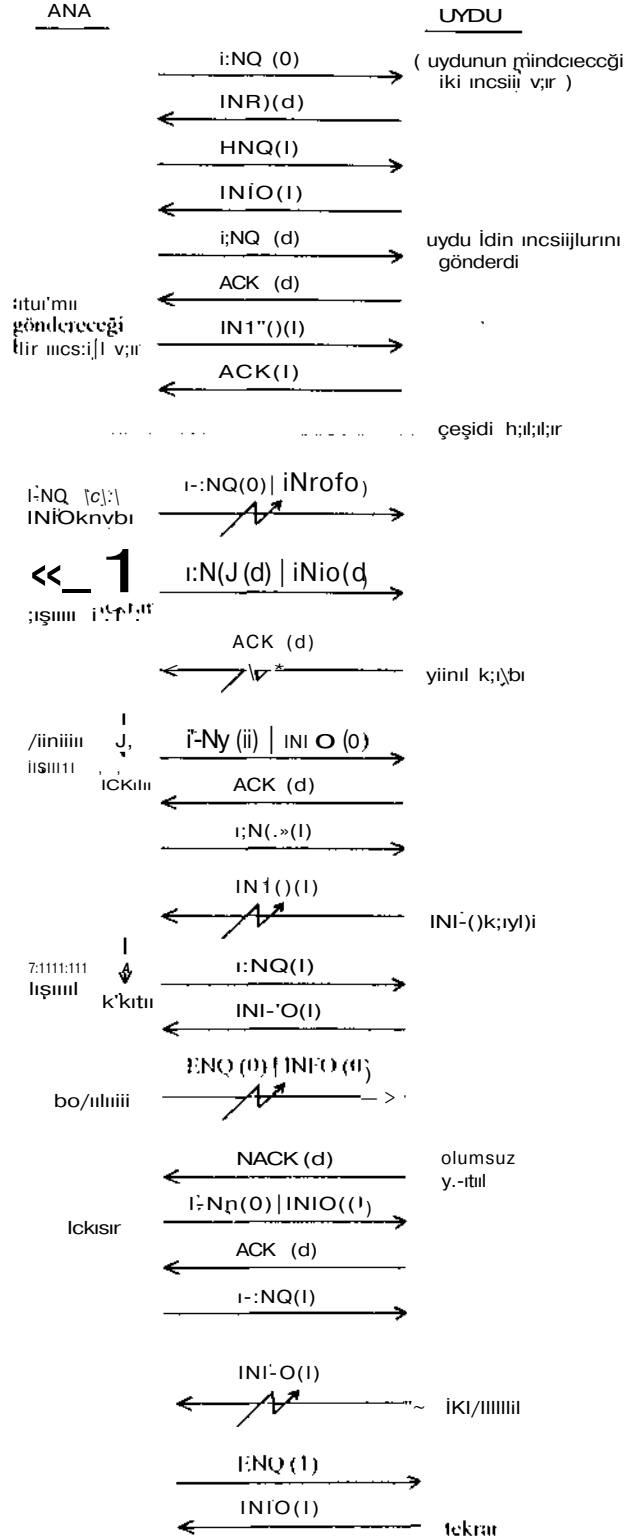
İDB'nin SKİ üzerinden santral güç kaynakları mikrodenetleyicisi ile ve SK2 üzerinden PC/santral bilgisayarı ile veri alış veriş yarı çift yönlü, pencere genişliği 1 olan, karaktere yönelik asenkron bir protokol (Mikroiletişim PROlokolu - MİPRO) ile sağlanmaktadır [3,4]. İletişimde kullanılan çerçeveler modulo 2'ye göre numaralanmakta, çerçeve baş ve sonu özel karakterlerle belirtilmekle (STX, FTX), hata sezme amacıyla çift eşlikli toplama sınaması uygulanmaktadır. Bilgi çerçevelerinin dışında sorgulama amaçlı ve çerçevenin doğru ya da yanlış alındığını gösteren denetim

çerçeveleri de kullanılmaktadır. Bilgi çerçeveleri ile, güç kaynaklarının çeşitli analog ölçü noktalarındaki değerler, bunların günlük ve aylık maksimum, minimum ve ortalama değer istatistikleri, değişik alarmlar, çeşitli sayısal giriş ve çıkış değerleri taşımabilmektedir.

MIPRO'da taraflardan biri ana, diğeri uydu rolünü üstlenir. Ana, uyduya varsa bir bilgi çerçevesi (INFO) göndermeye davet için ENQ çerçevesi gönderir. Ana ayrıca INFO çerçevesi gönderebilir. Uydu INFO çerçevesini doğru aldığını ya da İNQ davetine gönderecek bir yanıtı olmadığını ACK, yanlış bir çerçeve aldığını NACK, elindeki asıl verileri de INFO çerçevesi ile gönderir. MIPRO protokolunda çerçeve yapısı Şekil 3'te, değişik iletişim senaryoları Şekil 4'te gösterilmiştir.



SIX ... ASCII SİMİ t t K-XI k:ıkk-ii
 Ne ... (,V)\-e\o Ne (I I)il
 l1/ıııııık ... Çor v»-''' boyu
 Uılııı ... Fn çok X socki/li
 l oplnı:» sm;nn:ısı l l:lt sc/inc i'-in
 I' IX ... ASCII l'ml MI ' l'cXI k:ıııkfii
 Şİ.Kil. - ' MII'U(I, 'I R, TV:ı YAFISI



Şİ:Kil. -ı MII'KODA Di-Ç.ŞİK İLİ-İ'İM Sİ: NARYOLARI

İDB. güy kaynakları
nikiodenetici icisi ile iletişimde
uydu. IH 'sandal bilgisayar ile
iletişimde ana muindedir. Modem ile
iletişim ise burada konu edilmeyecek
olan İlayes AT protokolü ile
kötartılmaktadır.

5. Sonuç :

Çok görevli yalısına yaklaşımı geyek
zamanlı sistem tasarımına yeni bir
boyut gelirmiş ve sistemlerin tasanın,
geyikleme. doğu ulama sürecini
önemli ölyüde kısallınıştırı. Bu
yalışmada Keil Elektronik İlimasının
C5 I Professional Developers Kil isimli
yazılını geliştirme orlanı kullanılmış,
yazılımın lamamı C dilinde
oluşturulmuşun. (ielişirine orlanı
iyinde yer alan KTX-5I Tiny isimli
geyek /amanlı işletim sistemi
çekirdeği, yok görevli yalısına
yapısının kurulmasını kolaylaştırtıuş ve
tasarımın 5 insan - ay gibi yok kısa bir
sürede tamamlanmasında başlıca elken
olmuştur D

6. kaynaklar :

/I/ Tanenbaum. A.S., Modern
Operaling Systems, Prenlice - Hall
International. 1992

ili C>I Professionals Developers Kil
Technical İnformatiıı. Keil E.leklıouik
(imhII, 199|

/V Hughes, E..Data Communications,
Mc(ııa\v - İ İııı, 19'P.

'M Tanenbaum, A.S., l'ompuler
Net\orks, Prenlice - Hall
International,ı'8'>



Hükn-ı ÖRKİNÜk
1950 İstanbul doğumlu.
1973 ve 1975 yıllarında
İTÜ ' Elektrik
Fakültesinde mühendis
HII ve yüksek mühendis.
198.V yılında İTÜ Fen

Bilimleri İİsililüsünde doktor unvanını
aldı. 1989 yılında Bilgisayar Sistem
Yapısı bilim dalında doyenı oldu. 1971
yılından beri İTÜ'nün- değişik
bölümlerinde görev yaptı. Halen İİİ
Elektrik - Elektronik Fakültesi Kontrol
ve Bilgisayar Mühendisliği Bölümü
öğrelin üyesidir. İlgı ve yalısına
alanları Bilgisayar Mıınası. (İeyek
/amanlı Sistemler, Bilgisayar İletişimi,
Çok İşlemcili Sistemler ve Paralel
Bilgi İşlemedir.



Muştafa UŞAK 19(>6
İstanbul doğumlu. 1987
yıllıla Hacettepe
Üniversitesi Elektronik
Mühendisliği
Bölümünden mezun

oldu. 1987 yılından beri
EN I ES ve II-IT I AŞ İimakknnun Ar-
(ie bölümlerinde ISDN uybirimi,
değişik mikroişleincili sistem
lasaunlarında görev alınıştır. Ayrıca
1990 yılında <)RTAŞ İlimasında
1SPBX proje danışmanlığı yapmıştır.
İlgı alanları Bilgisayar İletişimi ve
Cıcryck /amanlı sistemlerdir.

GRAFİK TABANLI YAZILIM PAKETİ TASARIMI

Selma YÜNCÜ, Selma ÖZAYDIN
Elektrik ve Elektronik Mühendisliği Bölümü, Gazi Üniversitesi
06570 Maltepe ANKARA

İNT

Bu çalışmada, IBM PC türü veya IBM uyumlu kişisel bilgisayarlarda kullanılacak grafik tabanlı bir yazılım paketi geliştirilmiş ve uygun amaya konmuştur. Yazılım paketi oluşturulurken Turbo Pascal 6.0 programlama dili kullanılmıştır.

Yazılım paketi içinde, çeşitli grafiksel şekillerin istenilen renk ve desende çizilmesi ve boyanması sağlanmıştır. Çizilebilen şekiller; çizgi, daire, elips, dörtgen, dilim ve yaydır. Hatalı çizilen kısımların silinebilmesi için silme seçeneği de eklenmiştir.

Aynı yazılım paketinde, grafiksel yazı yazılması için grafik editör kısmı da bulunmaktadır. Bu editör için 5 değişik font ve herbir font için 5 değişik büyüklük vardır. Font ve büyüklüklerin birisi seçildiğinde, grafiksel sayfa üzerinde istenilen bir konumdan itibaren yazı yazılabilmektedir.

Grafiksel pakette hazırlanan çalışmalar, istenilen isim altında bir dosyaya kaydedilebilmekte, dosyadan ekrana okunabilmekte, yazıcıdan çıktısı alınabilmektedir.

i.GİRİŞ

Bilgisayar grafiği, bilgisayar biliminde hızla büyüyen ve en ilgi çekici alanlardan biri olmuştur. Günümüzde kullanılan bilgisayar ekranları genelde grafik ekran özelliğine sahiptirler. Bilgisayar grafik özelliğinden yararlanan işyerleri, endüstri, hükümet, sanat, eğlence, reklamcılık, eğitim, araştırma, öğretim, tıp gibi

birkaç alanı söylemek, bilgisayar dünyasında bu konunun önemini anlatmak bakımından yerinde olacaktır. [1]

Bilgisayar grafiği, bilgisayar yardımıyla resim oluşturma, hareket ettirme, ve görüntülemeyi içerir. Bu özelliklerinden dolayı bilgisayar grafiği son yıllardaki insan ve bilgisayar arasında artan iletişimin en büyük etkenidir. [2]

Bilgisayar Destekli Tasarım, BDE (Computer Aided Design, CAD) ticari çevrelerde bilgisayar grafiği alanında hızla yayılmaktadır, örneğin, entegrelerle oluşturulmuş devre tasarımı bilgisayarda grafiksel resim şeklinde gösterilebilir. Kompleks bir entegre devre tasarımını elle çizmek, bir mühendisin haftalarını alabileceken aynı devre bilgisayarda çok daha kısa sürede ve daha hassas olarak yapılabilir. [2]

Bilgisayarlar, text ortamında kullanılmaya başladıktan sonra bunu destekleyen çok sayıda editörler ve yazılımlar geliştirilmiştir. Grafik ortamdaki çalışmalar, programlama dillerindeki komut eksikliği nedeniyle text ortamındaki kadar fazla olmamıştır. Grafik ekran modunda hazırlanan bilgisayar programları, programın imkanlarını artırdığı gibi, kullanıcının programı daha hızlı ve kolaylıkla uyum sağlamasına da yardımcı olmaktadır.

Bu bildiride, grafiksel şekil çizimi ve yazı yazımı özellikleri üzerinde durulmuş olup, Pascal 6.0 programlama dilinin grafiksel komutları yardımıyla, grafik ekran modunda olması gereken işlemleri yapan ve irdeleyen bir yazılım paketi hazırlanmıştır. [3], [4]

2. YAZILIM PAKETİNİN YAPISI

Yazılım paketinin çalışabilmesi için bilgisayar ekranı VGA tipinde olmalıdır. Programın çalıştırılması ile VGA ekran 640x480 pixel grafik ekran özelliğini alır.

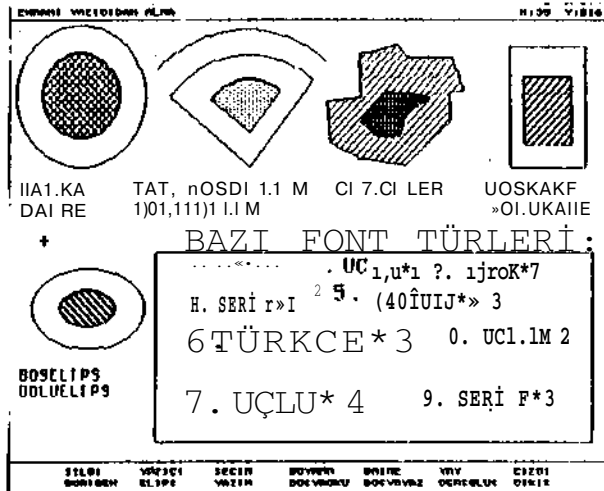
Grafiksel ekran, Ust mesaj penceresi, Çalışma ekranı, Alt mesaj penceresi olmak üzere üç pencereye ayrılır. [5]

Ust mesaj penceresi, seçilen seçenekler hakkında kullanıcıya bilgi veren bir mesaj penceresidir.

Çalışma ekranı, yazılım paketiyle yapılabilecek çeşitli fonksiyonların gerçekleştirilebilmesi için sürekli olarak kullanılan penceredir.

Alt mesaj penceresi, program ilk çalıştırıldığında ana menüyü görüntüler. Daha ilerki adımlarda ana menüdeki seçeneklerin alt seçeneklerini veya seçeneklerden herhangi birisi çalışırken, tuşların işlevlerini gösteren bir mesaj ekranı olarak kullanılır. Yazılım paketi ile hazırlanan örnek bir sayfa çıktısı, şekil 1'de görülmektedir.

Yazılım paketinin çalışmasıyla ekrana gelen ana menü'nün seçenekleri incelenecek olursa, yazılım paketinin yapılabileceği işlevler daha iyi anlaşılabilir.



Şekil 1 örnek bir sayfa çıktısı

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

2. 1. Ekran Ana Menüsü Seçenekleri :

- | | | |
|-------------|-------------|---------|
| i.SİLGİ | 2.YAZICI | 3.SEÇİM |
| 4.BOYAMA | 5.DAİRE | 6.YAY |
| 7.ÇİZGİ | 8.DÖRTGEN | 9.ELİPS |
| 10.YAZIM | 11.DOSYAOKU | |
| 12.DOSYAYAZ | 13.ÇIKIŞ | |

Program çalıştırıldıktan sonra, kursör çalışma ekranında iken ana menüye geçmek için ESC tuşuna basılır. Ana menüden çalışma ekranına dönmek için tekrar ESC tuşuna basılır. Ana menüdeki seçeneklerin alt seçenekleri olan alt menülerden ana menüye dönüş ise F10 tuşu ile gerçekleşir. Alt menülere geçildiğinde, üst mesaj ekranında, geçilen menüye ait ilgili mesaj verilir. Alt mesaj ekranı ise yeni bir alt menü için menü ekranı olarak veya seçilen fonksiyon tuşları ve bu tuşların işlevi hakkında bilgi veren mesaj ekranı olarak kullanılır. Çizimler esnasında, SAG, SOL, YUKARI, AŞAĞI yön tuşlarıyla şeklin büyüüp küçülme, çizimden vazgeçilirse "ESC" tuşu ile çizimden çıkma özelliği sağlanmıştır. Daire, yay, çizgi, dörtgen, elips çizimlerinde; çizim esnasında yön tuşlarıyla şeklin çizim merkezi ayarlanıp, RETURN tuşu ile merkez seçilir ve tekrar yön tuşlarıyla geçici şeklin çizimine başlanır. Çizimden vazgeçilirse, ESC tuşu ile şekil iptal edilebilir. RETURN tuşu ile şekil sabitleşir. İstenirse yeni merkezler seçilip yeni şekiller çizilebilir.

1. SİLGİ : Ekranda yapılan hataların düzeltilmesi için bu seçenek kullanılır. SİLGİ menüsünün alt seçenekleri Ekransil ve llatasil seçenekleridir.

EKRANSİL seçeneği, çalışma ekranının silinmesi için kullanılır.

HATASİL seçeneği, grafik ekranda çalışırken, hatalı yazılan veya hatalı çizilen kısımların silinmesi için kullanılır. Kullanılacak silginin boyutu INSERT tuşuna basarak ayarlanabilir.

bilir.

2. YAZICI : Çalışma ekranının yazıcıdan alınması için kullanılır. Seçildiğinde, çalışma ekranında bulunan herşey yazıcıya aktarılır. Yazıcı ile ilgili bir sorun varsa (yazıcı kapalı ise veya bilgisayara bağlı değilse) uyarı verir.

3.SF.ÇİM: Daire, elips, dörtgen, dilim çizimlerinde, bu şekillerin desenli olarak çizilebilmesi için SEÇİM menüsü kullanılır. SEÇİM menüsünde çizilecek şeklin dış rengi, iç deseni, desen rengi gibi bilgiler belirlenir. Daha sonra, şekil çizimine geçilir. Yukarıda bahsedilen çizimlerine 11. seçenekleri olan DAİRE, 1)0 LU Eİ, İPS, DOLUKARE. DOLUDT-İ.1M seçeneklerinden herhangi birisi seçildiğinde çizilecek şekillerin dış renkleri, iç renkleri ve desenleri SEÇİM menüsünde belirlenen renk ve desene göre kendiliğinden çizilir.

SEÇİM menüsünün alt seçenekleri RENK, DESEN, ÇİZGİ seçenekleridir.

RENK seçeneğinde, alt mesaj penceresinde, 20x18 pixellik kare kutular içinde 16 değişik renk ekrana gelir.

DESEN seçeneğinde, ekrana DESENTURU ve DESENRENGİ olmak üzere iki seçenek daha gelir. DESENTURU'nde, alt mesaj penceresinde 35x18 pixellik kare kutular içinde 12 değişik desen ekrana gelir. DESENRENGİ seçeneği, çizilecek şeklin deseni için renk belirlemede kullanılır. İşletilmesi RENK seçeneği ile aynıdır.

ÇİZGİ seçeneği, çizilecek şekillerin çizgi türünü belirlemede kullanılır. Alt mesaj ekranında 8 değişik çizgi türü bulunur.

İler üç seçenekte de, istenen renk, desen veya açının seçilmesi için ilk kutu üzerinde görüntülenen kursor, SAG.SOE tuşlarıyla ilerletilip' d i g e r kutular üzerine konumlandırılarak, istenen renk, desen veya çizgi belirlendikten sonra RE-232

TURN tuşu ile seçenekten çıkarılır.

4. BOYAMA: Daire, elips, dörtgen, dilim gibi şekillerin, SEÇİM seçeneğinde belirlenen renk ve desenle doğrudan desenli çizilmesi mümkündür. Bunun yanında kapalı olarak çizilen herhangi bir eğrinin içinde sonradan herhangi bir renk veya desenle boyanmak istenebilir. Bu amaçla BOYAMA seçeneği belirlenmiştir. Desen ve desen rengi belirlendikten sonra, şeklin içinde herhangi bir yere gelinerek bu seçenek seçilir ve RETURN tuşuna basılırsa istenen desen ve renkte boyama gerçekleştirilir.

5. DAİHİL : Ekranda istenen büyüklükte daireler çizmek için kullanılır. DAİRE'nin alt seçenekleri HALKA ve DAİRE seçenekleridir. Bunlar, içi boş veya desenli daireler çizmek için kullanılır.

HALKA seçeneği, Seçim menüsünde belirlenen desen ve desen rengini dikkate almadan, içi boş daireler yani halkalar çizmek için kullanılır.

DAİRE seçeneği, Seçim menüsünde belirlenen desen ve desen rengine göre istenildiğinde desenli daireler çizmek için kullanılır.

6. YAY: Yay şeklinde açık eğriler veya dilim şeklinde kapalı eğriler çizmede kullanılır. YAY'ın alt seçenekleri YAY, BOŞDİLİM, DOLUDİLİM seçenekleridir.

YAY seçeneği, yay şeklinde eğriler çizmede kullanılır.

BOŞDİLİM seçeneği, SEÇİM menüsünde belirlenen desen ve desen rengini dikkate almadan, içi boş dilimler çizmek için kullanılır.

DOLUDİLİM seçeneği. SEÇİM menüsünde belirlenen desen ve desen rengine göre istenildiğinde desenli dilimler çizmek için kullanılır.

7. ÇİZGİ : SİLĞİ seçeneğinin alt seçenekleri SABİT, SİLİMEN, YÖNLÜ çizgidir.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

SABİT çizginin özelliği üzerinden tekrar geçildiğinde siinmemesidir. RETURN ile çizilecek konum başı belirlenir. Yön tuşlarıyla istenen yönde çizgi çizilir. RETURN ile çizgi sonu belirlenir.

SİLİNEN çizginin özelliği üzerinden tekrar geçildiğinde silinmesidir. RETURN ile çizilecek konum başı belirlenir. Yön tuşlarıyla istenen yönde çizgi çizilir. RETURN ile çizgi sonu beirlenir.

YÖNLÜ çizgide, geçici çizgi 0-360 derece arasında çevrilip islenen yön ve büyüklük ayarlandıktan sonra çizgi çizildiği için kullanıcıya büyük kolaylık ve avantaj sağlamaktadır.

H. DÖRTGEN : DÖRTGEN'in alt seçenekleri BOŞKARE, ÜOLUKARE seçenekleridir.

BOŞKARE seçeneği, SEÇİM menüsünde belirlenen iç renk ve deseni dikkate almadan ekranda içi boş kare çizmek için kullanılır.

DOLUKARE seçeneği, SECİM menüsünde belirlenen iç renk ve deseni dikkate alarak ekranda içi desenli ve renkli kare çizmek için kullanılır.

9. ELİPS : ELİPS 'in alt seçenekleri DÖŞELİPS ve DOLUELİPS seçenekleridir.

İDÖŞELİPS seçeneği, SEÇİM menüsünde belirlenen iç renk ve deseni dikkate almadan ekranda içi boş elips çizmek için kullanılır. DOLUELİPS seçeneği, SEÇİM menüsünde belirlenen iç renk ve desen dikkate alınarak ekranda içi desenli ve renkli elips çizmek için kullanılır.

10. YAZIM : YAZIM" in alt seçenekleri YAZIYAZ, EONTSEÇ, JIARE-BOŞ seçenekleridir.

YAZIYA.Z seçeneği, ekranda değişik font ve büyüklükte grafiksel yazı yazılmasını sağlar [61. Bunun için geliştirilen yazılımın rahatlıkla kullanılabilerek özelikte bir editör görevi yapmasına özen gösterilmiştir. SAG, SOL ve RETURN tuş-ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

ları kullanılarak ekranda istenen konuma gelinip buradan itibaren yazı yazımına başlanabilir. Yazım Fontu FONTSEÇ menüsünde belirlenen fonttur. Yazı yazarken, DEL, INS, BACK, Yön tuşları fonksiyonel görevlerini yerine getirirler. RETURN tuşu satır başı yapmayı, HOME satır başına, ENİ) satır sonuna gitmeyi sağlar. Yazılan yazının rengi, SEÇİM menüsünde belirlenen dış renkle aynıdır. ESC tuşu ile yazını menüsünden çıkıp ana menuye döner.

FONTSEÇ seçeneği, grafiksel yazım için gerekli fontun belirlenmesinde; kullanılır. Normal, Uçlü, Ufak, Şerif, Gotik adlarında beş değişik font kullanılmış ve bu fontlar için de beş değişik büyüklük ayarlanmıştır. EONTSEÇ seçeneğinde, ekranda bütün fontlar sırasıyla görünür. SAG, SOL tuşlarıyla bir fonttan diğerine geçerek istenen font belirlendikten sonra RETURN tuşuna basılır. Font seçiminden vazgeçilmek isteniyorsa ESC ile çıkılır.

Belirli fontlarla yazı I;111 yazılarda harfler arası aralıklar değişik olurlar, örneğin bir türlü fontla yazılan yazıda aralık sıkışık veya lüzumundan fazla geniş olabilir. Bu aralıkları ayarlamak için HAREGEN seçeneği kullanılır. Bu seçenek kullanıldığında 7, +7 pixel arasında sayılar verilir. Eksi sayılar verildiğinde, harfler arası aralık yazılan miktarda pixel sıkıştırılmakta, artı sayılar verildiğinde de aynı miktarda aralıklar pixel açılmaktadır.

11. DOSY AOKU: Grafiksel olarak bu programla hazırlanan bir dosyanın diskten ekrana okunması içindir. Bu seçenekte okunacak dosya adının sorulduğu bir mesaj ekrana gelir. Burada dosya adı girildiğinde, o dosya diskte aranır. Bulunursa çalışma ekranına getirilir. Bulunamazsa. bu .111namadığına dair uyarı verilerek seçenektен çıkılır.

İZOTROP VE KOMPOZİT MAKİNA ELEMANLARINDA DÜZLEM GERİLME ANALİZİ İÇİN BİR BİLGİSAYAR PROGRAMI

M.TOPÇU, S.TAŞGETİREN

Pamukkale Üniversitesi, Mühendislik Fakültesi, DENİZLİ

Ö/et

Bütün doğrultularında özellikleri aynı olan izotrop malzemelerle dizayn ve üretime alışmış mühendis ve imalatçılar için doğrultulara göre özellikleri değişiklik gösteren kompozit malzemenin yapılacak dizayn farklılık arz etmektedir. Bu bildiride, izotrop ve kompozit malzemelerden imal edilen makina elemanlarında gerilme analizi genel hatlarıyla gözden geçirilmekte ve iki boyutlu gerilme analizi probleminin bilgisayara uyarlanması esas alınan Sonlu Elemanlar Yöntemi tanımlanarak bu yöntemle gerilme problemini çözen bilgisayar programının yapısı anlatılmaktadır. Programın hazırlanmasında sonlu eleman problemini en iyi tanımlayabilen izoparametrik elemanlar kullanılmıştır. Lineer denklem sistemlerinin çözümünü doğru ve kısa zamanda en az bellek kullanarak yapılmasını sağlayacak bir program geliştirilmiştir. Programın işleyişini göstermek için basit bir örnek ile çözüm analitik çözümle karşılaştırılmalı olarak verilmiştir.

1. Giriş

Belirli özgül özelliklerinin çok yüksek olması sebebiyle kompozit malzeme kullanımı hızla artmaktadır. Pek çok alanda da geleneksel malzemelerin yerini almaya başlamışlardır. Başlangıçta hafiflikleri sebebiyle uzay sanayiinde kullanımı ile dikkat çekmişlerse de günümüzde gıda sektöründen otomotiv sektörüne, inşaat sektöründen denizcilik sektörüne kadar hemen her alanda kullanıma imkanları araştırılmaya başlanmıştır.

İzotropik malzemenin her alanda yalnızca bir değerle ifade edilebilen özelliklerinin, kompozit malzeme kullanımı durumunda birden çok değerle ifade edilmesi gerekmektedir. En basit durumda, tek yönde takviye edilmiş ortotropik yapıya sahip bir levhada takviye doğrultusunda ve buna dik doğrultuda olmak üzere iki çekme mukavemeti, iki elastisite modülü, iki ısı genleşme katsayısı ve ısı iletim katsayısı gibi değerlerin bilinmesi ve tasarımı göz önünde bulundurulması gerekmektedir. Üstelik bu değerler doğrultulara göre

büyük farklılıklar göstermektedir [1,7].

Bu durum, çok değişik şekillerde takviye edilebilen kompozitler gözönüne alındığında ortaya çıkacak zorlukları ve dikkate alınması gereken parametrelerin çokluğu sebebiyle kompozit malzemenin yapılacak makina elemanının dizaynında optimizasyonu önemini ortaya çıkarmaktadır. Bütün bunlarla beraber bilgisayar teknolojisinde sağlanan gelişmelerle anizotropik elastisite sahasındaki ilerlemeler bir ölçüde bu sahadaki dizayn zorluklarını ortadan kaldıracak seviyelere gelmiştir.

2. (Düzlem) Şekil Değiştirme İlişkisi

Düzlem gerilme durumu için izotrop malzemelerde gerilme şekil değiştirme ilişkisi

$$\{a\} = [D] \{e\} \quad (1)$$

şeklinde yazılabilir [2]. Burada $\{a\}$ düzlem gerilme bileşenlerini, $\{e\}$ düzlem şekil değiştirme bileşenlerini $[D]$ ise (3×3) boyutlarında olup elastik sabitler matrisini ifade etmektedir. İzotrop malzemeler için malzemenin E elastisite modülü ve ν poisson oranı biliniyorsa hemen elastik sabitler matrisi oluşturulabilir. Oysa yalnızca bir yönde takviye edilmiş bir ortotrop kompozit malzeme için en az dört sabitin bilinmesi gerekmektedir. Takviye şekli ile takviye doğrultusunun değişmesi bir de malzemelerin birleşme hacim oranlarının değişkenliği dikkate alınırse kompozit levhalarda gerilme problemi iyice karmaşıklaşmaktadır. Ayrıca kompozit makina elemanının birden çok katmandan oluşması gerekirse bu katmanların kalınlık, malzeme cinsi ve her birinin farklı şekillerde takviye edilme durumu göz önünde bulundurulması hesap yapılması gerekmektedir. Anizotrop malzemelerde en genel durum için gerilme şekil değiştirme ilişkisi

$$\{\sigma_{ij}\} = [C_{ijkl}] \{\epsilon_{kl}\} \quad i,j,k,l=1,2,3 \quad (2)$$

şeklinde ifade edilebilir [3]. Burada $[C_{ijkl}]$ elastik

sabitler matrisidir. Bu malzemedeki simetri durumları ve takviye şekilleri dikkate alındığında ortotop malzemeler için 81 değişken 9 sabite indirgenir. İki eksene göre simetriklik gösteren özel orlrop malzeme için elastik sabitler matrisinin dört bağımsız değişkeninin bilinmesi durumunda bu matris tanımlanabilir. Bu sabitler deneylerle elde edilebileceği gibi kompozit malzemeyi meydana getiren matriks ve fiberlerin özellikleri biliniyorsa birleşmedeki hacim oranlarından hareketle her iki dogiulludaki elastisite modülleri ve poisson oranları bulunarak elastik sabitler matrisi hesaplanabilir. Geliştirilen programda her iki durum da dikkate alınmış olup her durum için kompozitlere ait elastik sabitler matrisi bulunabilmektedir.

3. Sonlu Elemanlar Metodu

Mühendislik problemlerinde genellikle çalışma şartları, yüklenme durumu, yapı ve deney halaları kesin olarak bilinemez. Bu nedenle deney sonuçları ile aynı değerlerin elde edilebildiği yaklaşık çözümleme metodları kolaylık, ucuzluk gibi özellikleri sebebiyle tercih edilmektedir. Bunu çok hızlı işlem yapan bilgisayarların da etkilediği açıktır. Yaklaşık çözüm metodlarından biri olan "Sonlu Elemanlar Metodu" özellikle son yıllarda büyük gelişmeler göstererek kesin çözüme yakın neticeler veren ve ilgililerce geçerli bir çözüm tekniği olduğu kabul edilen nümerik bir methodur. Bu methodla bir çok bilim dalında değişik problemler çözülmektedir [4,5].

Sürekli bir ortamda alan değişkenleri (gerilme, yer değiştirme, basınç, sıcaklık, vs.) sonsuz sayıda farklı değere sahiptirler. Diğer taraftan sürekli bir ortamın belirli bir bölgesinin de aynı şekilde sürekli ortamı özelliği gösterdiği bilinmektedir. Bu alı bölgede alan değişkenlerinin değişimi sonlu sayıda bilinmeyen olan bir fonksiyon ile tanımlanabilir. Bilinmeyen sayısının az ya da çok olmasına göre seçilen fonksiyon lineer veya yüksek mertebeden olabilir. Sürekli ortamın alt bölgeleri de aynı karakteristik özelliği gösteren bölgeler olduğundan, bu bölgelere ait alan denklem takımları birleştirildiğinde bütün sistemi ifade eden lineer denklem takımı elde edilir. Denklem takımının çözümü ile sürekli ortamdaki alan değişkenleri sayısal olarak elde edilebilir.

Sonlu elemanlar metodunun temel prensibi, öncelikle bir elemana ait sistem özelliklerini içeren denklemlerin çıkartılıp daha sonra tüm sistemi temsil edecek şekilde eleman denklemlerini birleştirilerek sisteme ait lineer denklem takımının

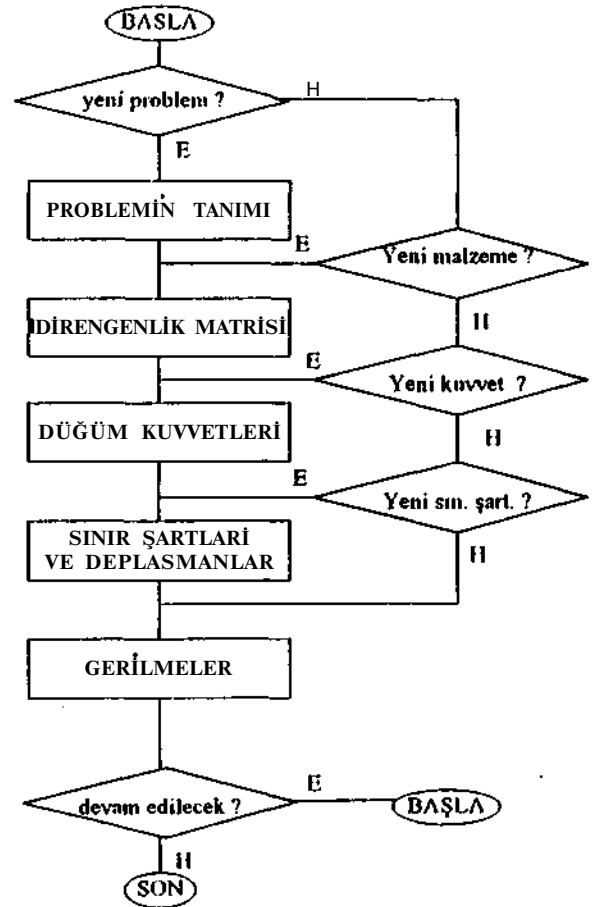
elde edilmesidir. Bir elemana ait denklemlerin elde edilmesi değişik şekillerde yapılabilir.

Sonlu elemanlar metodu ile problem çözümünde hangi yaklaşım kullanılırsa kullanılsın işlem şuası şöyledir;

- Cismin sonlu elemanlara bölünmesi
- Eleman direngenlik matrisinin teşkili
- Sistem direngenlik matrisinin oluşturulması
- Sisteme etki eden kuvvetlerin bulunması
- Sınır şartlarının belirlenmesi
- Sistem denklemlerinin çözümü

4. Programın Yapısı

Düzlem gerilme problemini çözebilmek için yukarıda belirtilen işlemlerin tamamını yapabilen birbiriyle irtibatlı beş ayrı program yapılmıştır. Sırasıyla detayları ve çalışma prensipleri anlatılacak olan programın akış diyagramı şekli' de verilmiştir



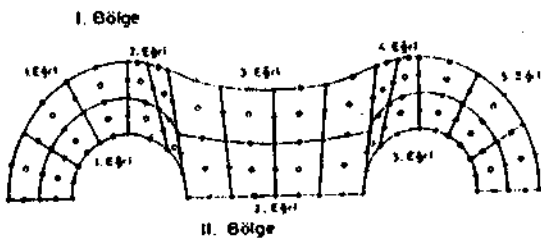
Şekil 1. Program Akış Diyagramı

4.1 Cismin Sonlu Elemanlara Bölünmesi PRG1

Çözüm bölgesinin geometrik yapısı belirlenerek bu geometrik yapıya ai uygun gelecek elemanlar seçilmelidir. Seçilmi elemanların çözüm bölgesini temsil etmeleri oranında elde edilecek neticeler gerçek çözüme yaklaşmış olacaktır [5]. Sonlu elemanlar metodunda iki boyutlu problemlerde genelde düğüm sayılan, seçilen interpolasyon fonksiyonunun derecesine bağlı olarak farklılık arz edebilen üçgen yada dörtgen elemanlar kullanılmaktadır. Bu programda eğri yüzeyleri daha iyi temsil ettiği için izoparametrik dörtgen elemanlar tercih edilmiştir. Bir kenardaki düğüm sayısı değişken alınmış olup kullanıcı istediğini tercih edebilmektedir.

Sonlu elemanlar metodunda cismin elemanlara ayrılması için öncelikle çözüm bölgesinin tanımlanması gerekir. Çünkü problemin özelliğine göre çözüm bölgesinin tamamının veya belli bir kısmının sonlu elemanlara bölünmesi gerekebilir. Bu, şeklin geometrisine, dış kuvvetlerin simetrik olup olmamasına ve malzeme özelliklerine bağlıdır. Sonlu elemanlara ayırma işlemi ve problemin giriş bilgilerinin hazırlanması eleman sayısı artıca hem zaman aha olmakta hem de hata yapma oranı artmaktadır. Bu yüzden elemanlara bölme işlemi kısa sürede hatasız yapabilen bir program yazılmıştır. Elemanlara ayrılan cisim ekranda çizilerek hata kontrolü yapma imkanı vermektedir. Şekil 2'de izotrop ve simetrik yüklenmiş bir zincir baklasının sonlu elemanlar modeli verilmiştir.

Program kullanıcıdan problemin sınırlarını, her parçanın kaç elemana ayrılacağını sormakta ve elemanlara bölme işlemi otomatik olarak yapmaktadır. Problemin sınırlarını doğru, daire parçası ve parabol olarak tanımlamak mümkündür. Bununla beraber her bölgedeki eleman büyüklüklerini farklı olarak tanımlama imkanı vermektedir. Sistem, işlenen kenar düğüm



Şekil 2. Zincir baklasının SEM modeli

sayısına sahip izoparametrik dörtgen elemanlara bölünerek eleman düğüm numaraları ile düğüm koordinatları hesaplanır [6,9].

Bu program düğüm koordinatlarını ve düğüm numaralarını içeren iki dosya ile problemin ileri aşamalarında lazım olacak enine eleman, sayısı, boyuna eleman sayısı ve kenar düğüm sayısı ile problemin adını vb. bilgileri bir dosyada saklayarak diğer programlarda kullanılmasını sağlamaktadır.

4.2 Direngenlik Matrisinin Hesabı PRG2

Eleman direngenliğinin bulunması, elemana etki edici dış etkenler ile alan değişkenleri arasında bir ilişki kurmak anlamına gelmektedir. Mesela, bir elastisite problemde elemana etki eden dış kuvvet ile yer değiştirmeler arasındaki ilişki bir lineer denklem takımı ile karakterize edilir [8]. Sonuçta elde edilen lineer denklem takımı,

$$[K]\{U\} = \{P\} \quad (3)$$

şeklinde ifade edilir. Burada $\{U\}$ düğüm yer değiştirmelerini, $\{P\}$ düğüm dış kuvvetlerini ifade eden sütun matrisidir. $[K]$ ise elemanın geometrik ve elastik özelliklerinden elde edilen Direngenlik Matrisidir. Eleman direngenliğinin elde edilmesinde en çok minimum enerji prensibi ve virtüel işler prensibi kullanılmaktadır [8].

Minimum enerji prensibine göre, eleman içerisinde birim hacimdeki iç enerji ile dış kuvvetlerin yaptığı iş arasındaki farkın minimum olması gereklidir. Elemanın iç enerjisi,

$$dH = \int_V \frac{1}{2} \{e\} \cdot \{a\} \cdot dv \quad (4)$$

denklemi ile bulunur. Dış kuvvetlerin işi ise;

$$dW = \{U\} \cdot \{P\} \cdot ds \quad (5)$$

şeklindeir. Burada $\{P\}$ yüzey kuvvetlerini, $\{U\}$ da yer değiştirmeleri ifade etmektedir. Bu iki ifadeden hareketle izoparametrik elemanlar için eleman direngenlik matrisi

$$[K] = \int_V [B]^T [D] \cdot [B] \cdot \det J \cdot drds \quad (6)$$

olarak elde edilir. Burada $[B]$ yerdeğişürmeleri şekil değiştirmeye bağlayan geometrik sabitler matrisi olup şekil fonksiyonlarının türevlerinden ibarettir. $[D]$ daha önceki bölümde anlatılan elastik sabitler

matrisi, detJ lokal koordinatları global koordinatlara dönüştüren jakobien matrisinin determinantıdır. dr ve ds ise eleman lokal koordinatlarını ifade etmektedir.

PRG1 tarafından hesaplanmış olan eleman düğüm numaraları ve düğüm koordinatlarını ilgili dosyadan okuyarak daha sonra malzeme özelliklerini kullanıcıdan istemektedir. Bu daha önceden programa tanıtılmış izotrop ve kompozit malzemelerden birini seçme imkanı verdiği gibi yeni bir kompozit malzeme tanımlama imkanı da sunmaktadır. Her açıdaki tek ve çapraz takviye edilmiş olma durumlarına göre eleman direngenlik matrislerini hesaplar. Daha sonra eleman matrisleri üst üste toplanarak sistem matrisini oluşturur. Burada sistem direngenlik matrisinin boyutları normal olarak düğüm sayısının (DS), iki katı (2DS.2DS) olması gerekmektedir. Bu durumda matrisin boyutları büyüdüğü için problemde yarı bant genişliği (YBG) otomatik olarak hesaplanıp direngenlik matrisi (2DS,YBG+1) şeklinde oluşturulmaktadır. Bu şekilde hem hafıza hemde zaman tasarrufu sağlanmaktadır. Aşağıda bu işlemi yapan program parçası verilmiştir. Önce yan bant genişliği D (Eleman sayısı, Eleman düğüm sayısı) şeklinde oluşturulan düğüm matrisi kullanılarak tesbit edilir. Daha sonra buna göre elde edilen eleman matrisleri band halde sistem matrisine yerleştirilir.

```

YBG = 0
DO 100 N=1,ES
EK = NN + 1
EB = 0
DO 2J = 1,EDS
IF (EK.LD.D(N, J)) EK = D(N, J)
2 IF (EB.LD.D(N, J)) EB = D(N, J)
C = 2*(EB-EK + 1)
100 IF(YBG.LD.C)YBG = C

DO 100N=1,ES
DO 1 II = 1,EDS
NRT=2*(D(N, II)- 1)
DO 1 rr = 1,2
NR = NRT + II
I = 2 * (II-1)+IT
DO 1 JJ=1,EDS
NCT=2*(D(N,JJ)- 1)
DO 1 JT= 1,2
J = 2*(JJ- 1) + JT
NC = NCT + JT-NR+ I
IF (NC.LE. 0) GOTO I
S(NR, NC) = S(NR, NC) + SE(I, J)
1 CONTINUE
100 CONTINUE

```

4.3 Şişleme Etki Eden Kuvvetlerin Hesabı PRG3

238

Herhangi bir yüzeye etkiyen yayılı dış kuvvetler, iç ve dış basınç, etki ettikleri yüzeyde bulunan düğümlere

$$P = \int_s N.P.ds \quad (7)$$

şeklinde dağıtılır. Sisteme etkiyen kuvvet eğer tekil kuvvetse doğrudan düğüm kuvveti olarak alınır. Dış kuvvetler problemin bulunduğu eksen takımına göre bileşenlerine ayrılır. Örneğin iki boyutlu bir problemde kuvvetler x-y düzlemine

$$P_x = P \cdot \cos \theta, \quad P_y = -P \cdot \sin \theta \quad (8)$$

şeklinde dağılır.

PRG3 Programı ile sisteme etki eden yayılı yükler düğümlere, tekil yükleri ise etki ettiği yönde kuvvet matrisine doğrudan yerleştirilmiştir. Program sırasıyla, sisteme kaç tip kuvvet etki ettiğini, etki eden kuvvetin nasıl bir kuvvet olduğunu, hangi elemanlara etki ettiğini sormakta, yayılı yük ve basınç durumlarında ilgili düğümleri kendisi tesbit ederek o düğümlere ait koordinatları PRG1 de elde ettiğimiz dosyadan okuyup (7) bağıntısındaki integrali ilgili düğümler için hesaplayıp elde ettiği değerleri kuvvet dosyasında saklamaktadır. Sisteme etki eden birden çok kuvvet var ise bunların süper pozisyonunu da yapmaktadır.

4.4 Sınır Şartlarının Belirlenmesi ve Sistem Denklemlerinin Çözümü PRG4

Problemin özelliğine göre hangi düğümlerin hangi yönde serbest yada tutulu olduğunun belirlenip lineer denklem sisteminin çözümünde bunların göz önüne alınması gerekir. Yer değiştirme vektörü oluşturulurken tutulu düğümlere karşılık gelen deplasmanlar sıfır olarak ele alınır. Bilinen deplasmanlar var ise bunlar da doğrudan ilgili düğüme atanabilir. Şekil 2,a'da aksenal simetrik bir cismin 1/4'lük kısmında sınır şartları tarif edilmiştir. Her ne kadar problem geometrik olarak aksenal simetrik ise de kompozit malzemelerde takviye doğrultusunun 0 ve 90° haricindeki takviye durumlarında problem aksenal simetrik olarak ele alınamaz. Bu durumda sınır şartları değişmektedir, çünkü yapı anizotrop duruma dönüşmüştür.

Simetrik problemler için sınır şartları doğrudan program tarafından belirlenebilmektedir. Simetrik ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

olmayan problemler ile bilinen deplasmanlar söz konusu ise bu durumda ilgili düğümlere ait sınır şartlarının ekrandan girilmesi gerekmektedir.

PRG4 Programı yukarıda elde edilen rijitlik matrisi kuvvet vektörlerini okuyarak sınır şartlarını da dikkate alarak Gauss Eliminasyon yöntemi ile $[K]\{U\}=\{P\}$ denklemindeki $\{U\}$ yer değiştirme vektörünü hesaplamaktadır.

Buradan hesaplanan yer değiştirmeler kullanılarak,

$$\{a\}=[D][B]\{U\} \quad (9)$$

denklemlerle gerilmeler hesaplanır.

4.5 Gerilmelerin Hesabı PRG5

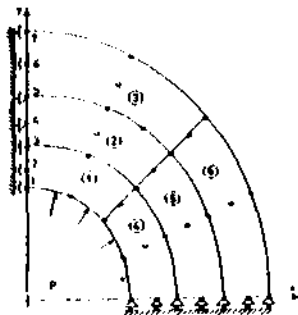
PRG5 programı $\{a\}=[D][B]\{U\}$ eşitliğinden her düğümde ya da elemanın istenen noktalarında gerilme hesabı yapmaktadır.

Her program ayrı ayrı ve bir ana metü yardımıyla çalıştırılmaktadır. Böylece aynı geometrik şekle sahip program için malzeme özelliklerinin veya takviye doğrultusunun değişmesi halinde yeniden PRG1 programını çalıştırmaya gerek kalmamaktadır. Aynı şekilde geometri ve malzeme özellikleri aynı fakat yükleme şekilleri farklı problemler için PRG1 ve PRG2 programlarının yeniden çalıştırılmasına gerek bulunmamaktadır. Aynı durum diğer programlar için de geçerlidir. Bu şekilde programa kullanım kolaylığı ve işlerlik kazandırılmaya çalışılmıştır.

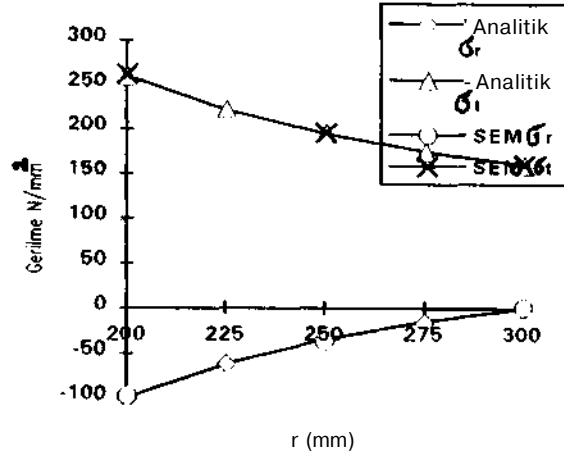
5. Örnek Çözüm

Yapılan programların işlerliğini göstermek için aşağıda boyutları verilen içten basınca maruz kalın cidarlı silindir problemi ele alınmış ve elde edilen sonuçları analitik çözümlerle karşılaştırmıştır. İçten basınca maruz silindir, SEM modeli ve elde edilen gerilmeler Şekil 3'te verilmiştir.

$r_1=200$ mm
 $r_2=300$ mm
 $p=100$ MPa



ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ



Şekil 3. İçten basınca maruz silindirde teğetsel ve radyal gerilmeler

5. Kaynakça

1. Hashin Z. "Analysis of Composite Materials, Journal of Applied Mechanics, September 1983, s.481, 1983
2. Timoshenko, S. P., Goodier, J.N., Çev: Kayan, İ., Şuhubi, E., "Elastisite Teorisi" An kitabevi, 1969, İstanbul
3. Lekhnitskii, S. G , "Theory of Elasticity of an Anisotropic Body", Mir Publishers, 1981, Moscow
4. Tirupathi R. C, Aslıok, D. B , "Introduction to Finite Element in Engineering", Prentice Hall ine. 1991
5. Topçu, M., "The Elasto-Plastic Stress Analysis on the Transmission Roller Chain Link Plates", Research Reports, DEÜ, Institute of Science and Engineering, FBE/MAK-91 -AR-248, İzmir, 1993
6. Topçu, M., "Transmisyon Zincirleri Yan Plakalarında Elasto-Plastik Gerilme Analizi" Doktora Tezi, DEÜ, Fen Bilimleri Enstitüsü, 1991, İzmir
7. Kam T. Y., Lai M. D. "Multilevel Optimal Design of Laminated Composite Plate Structures", Computer and Structures, 31(2) s. 179, 1989
8. Batle, K.J., "Finite Element Procedures in Engineering Analysis", Prentice-Hall Ing., 1982
9. Zienkiewicz, O.C., Philips, D.B., "An Automatic Generation Scheme for Plane and Curved

Yazarlar Hakkında Bilgi



Mızaftcı TOPÇU, 1959 yılında Afyon-
Sındık'da doğdu. 1982 yılında DEÜ-üenizli
Mühendislik Fakültesi Makina Bölümünde
Lisansını. 1986 yılında Selçuk Üniversitesi'nde
Yüksek Lisansını. 1991 yılında DEÜ-Fen
Bilimleri Enstitüsü'nde Doktorasını tamamladı.
1984 yılında Denizli Mühendislik Fakültesinde
Araştırma Görevlisi olarak başladığı akademik
görevine, 1987 yılında Öğretim Görevlisi olarak,
1991 yılında Yrd. Doç. Dr. oldu. Halen PAÜ
Mühendislik Fakültesi Makina Bölümünde
çalışmaktadır.



Süleyman TAŞGETİREN, 1965 yılında
Kahramanmaraş'ta doğdu. 1987 yılında DEÜ-
Denizli Mühendislik Fakültesi'nden mezun
olduktan sonra bir süre bir firmada çalışmıştır.
1988 yılından beri aynı fakültede Araştırma
Görevlisi olarak çalışmakta olup DEÜ- Fen
Bilimleri Enstitüsünde 1992 yılında yüksek
lisansını tamamlamış ve halen aynı enstitüde
doktora öğrenimine devam etmektedir.

V.42bis Veri Sıkıştırma Algoritmasının Yazılımla Gerçeklenmesi ve Karmaşıklık Analizi

*Rifat ÇÖLKESEN, Osman ALİEFENDİOĞLU, Bülent ÖRENCİK, A. Emre HARMANCI

İ.T.Ü. Elektrik- Elektronik Fakültesi 80626 İstanbul

"Çukurova Üniversitesi M.M.F. Elektrik-Elektronik Müh. Bölümü 01330 Adana

Özet

Bu bildiriye, verilerin saklanması ve iletiminde kullanılması giderek yaygınlaşan sıkıştırma algoritmalarının en güncellerinden biri olan CCITT'nin önerdiği V.42bis sıkıştırma algoritmasının başarımı ve yazılımla gerçekleştirilmesi incelenmiştir. Algoritmanın yazılımla gerçekleştirilmesi için birkaç yöntem önerilmiş ve bunların karmaşıklık analizleri yapılmıştır. Böylece uygulamalar için, en uygun yöntemin seçiminin bu yöntemlerin alan ve zaman karmaşıklığından giderek yapılabilmesi sağlanmıştır.

1. Giriş

Veri sıkıştırma, bilgi iletimi ve bilgi saklama maliyetini düşüren bir olgudur; uzak veri tabanlarına erişimde ve uzaktan verilen bilgi işlem hizmetlerinde veri aktarım hızını önemli ölçüde arttırmaktadır.

Veri sıkıştırma, verilerin içerdikleri artıklıkları (redundancy) uygun şekilde kodlayarak, bilgi kaybı olmaksızın veri yoğunluğunu artırma ilkesine dayanır. Bu amaçla geliştirilen algoritmalar "Statik" ve "Dinamik" olarak iki grup altında toplanır.

Statik algoritmalar, verilerin ne tür artıklık içerdiğinin (istatistiksel özelliklerinin) önceden bilinmesini gerektirir [1]. Bu nedenle, sıkıştırılacak veriye göre bu algoritmalarından uygun olanı seçip, sıkıştırmanın gerçekleştirilmesi gerekir. Sıkıştırılacak verilerin istatistiksel özelliği, yer aldıkları metin içinde değişim gösteriyorsa bu tür metinler, statik algoritmayla verimli olarak sıkıştırılmaz ve hatta bazı

durumlarda sıkıştırılmış veri miktarı sıkıştırılmamda göre artabilir.

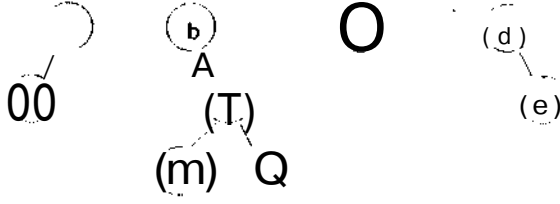
Dinamik algoritmalar ise verilerin metin içinde değişen istatistiksel özellikleri sıkıştırmanın gerçekleştirilmesi sırasında sezilir ve sıkıştırma işlemi temel olan parametreler sürekli güncellenir. Böylece sıkıştırma, işlem süresince verideki istatistiksel değişmeye göre optimize edilir ve kullanıcıya verinin özellikleri ve tipinden (metin, görüntü...) bağımsız bir yöntem sağlanır. Bu algoritmaların çoğu Lempel-Ziv' in gerçekleştirdiği sıkıştırma algoritmasına dayanır [2].

2. V.42bis Veri Sıkıştırma Algoritması

Son yıllarda, bilgisayarların telefon hatları üzerinden haberleşmesini sağlayan modem cihazlarında, özellikle görünür veri aktarım hızını artıran sıkıştırma algoritmaları kullanılmaktadır. Bu algoritmalarından biri, CCITT (International Telegraph and Telephone Consultative Committee) tarafından 1990 yılında modemlerde kullanılmak üzere uyarlanan ve LZW [3] algoritmasına (Lempel-Ziv algoritmasının bir uyarlaması) dayanan V.42bis dinamik sıkıştırma algoritmasıdır.

V.42bis sıkıştırma algoritması değişik uzunlukta karakter dizilerine sabit uzunlukta kodlar atanması ilkesine dayanır. Bu amaçla, kod atanmış karakter katarlarının saklandığı ve sıkıştırma işlemi süresince güncellenen iki sözlük kullanılır. Bu sözlüklerden birisi karakter katarlarının kodlanmasında, diğeri ise kodlanmış verilerin çözülmesinde kullanılır. Her sözlük karakter katarlarının ağaç yapılarıyla gösterildiği bir ağaç kümesinden

(ormandan) oluşur. Şekil 1 buna yalın bir örnektir.



Şekil 1. Bir sözlüğün ağaç yapısını ile gösterilimi

Her bir ağaç kökü alfabe'deki bir karaktere karşılık gelmektedir. 8 bitlik kod kullanan ASCII alfabeti için sözlükte 256 adet kök, dolayısı ile 256 ağaç bulunur. Bir düğüm, kendisine bağlı bir düğüm yoksa yaprak, kendisinin bağlı olduğu bir düğüm yoksa kök olarak adlandırılır. Ağaçların her biri belli bir karakter ile başlayan katarlar kümesini temsil eder. Her düğüm bu küme içinde bir katar karşılık gelir; bu katarı o düğüme kökten erişen yolda bulunan düğümlere atanmış sıralı karakterler oluşturur. Her düğüm, ayrıca, kendisine iliştilmiş katarla atanmış bir kod sözcüğü de taşır.

Başlangıçta sözlük yalnızca köklerden oluşur ve her köke temsil ettiği karakterin ASCII kod tablosundaki sayısal değerinin 3 fazlası kod olarak atanır. 0, 1, 2 sayıları kontrol amacıyla kullanılmaktadır. Daha sonra katar eşleştirme prosedürü ile sözlük oluşturulmaya başlanır.

Katar eşleştirme prosedürü, incelenen karakter katarının alt katarlar halinde sözlükte olup olmadığını aramak için kullanılır. Bu prosedüre göre bir "w" katar değişkeni seçilir ve değişken başlangıçta boş katar değerini alır. Buna göre:

- İlk adımda eşleştirilecek alt karakter katarının ilk karakteri 'K' "w" değişkenine atanır,
- Sözlükte "w" 'nin taşıdığı katar bulunursa, alt karakter katarının bir sonraki 'K'

karakteri okunur "w" değişkenine "wK" yeni değeri atanır ve b) adımı tekrar edilir, c) "wK" katar sözlükte bulunamaz ise, 'K' karakteri "wK" katarından ayrılır ve bu ayrılan 'K' karakterine 'uyuşma göstermeyen karakter' adı verilir; "w" değişkeninde kalan katar üzerinde eşleşme gerçekleşmiştir.

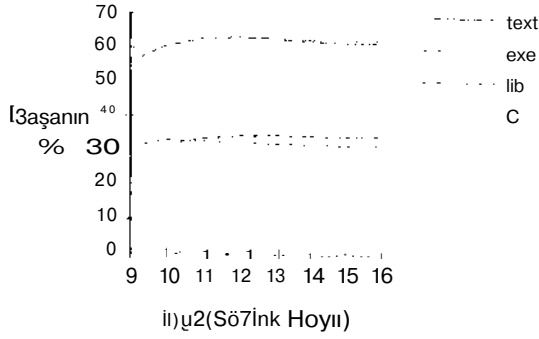
Sözlüğe yeni bir sözcük eklenmesi var olan bir sözcüğe (ilgili düğümün temsil ettiği alt karakter katarına sözcük adı da verilir) yeni bir karakter eklenmesi ile yapılır. Bu yeni karakter yukarıda anlatılan prosedür sonucunda oluşan uyuşma göstermeyen karakterdir. Bir sonraki katar eşleşme prosedürüne ise bu karakterin kök olduğu ağaç ile başlanır.

Yeni oluşturulan bir katarla kod verme işlemi C1 olarak adlandırılan sayacın içeriği atanarak yapılır. C1 sayacına başlangıçta 259 değeri atanır (3 adet kontrol kod sözcüğü + 8 bitlik ASCII kod alfabetinin karakterleri). Sözlük dolana kadar yeni katarlara kod atanması C1 sayacının değerinin birer birer artırılması ile gerçekleşir [4].

Sözlük dolduğunda sık kullanılmayan kodlara ilişkin sözcükler iptal edilir ve bu kodlar yeni sözcüklere atanır. Böylece sözlük yeni gelecek katarlara göre uyarlanır. Her kod atamasından sonra C1' in içeriği aşağıdaki yeni kod elde etme prosedürüne göre güncellenir. Bu prosedürde:

- C1 sayacı bir artırılır;
- Eğer C1 sayacı sözlük boyunu geçerse (örneğin 1024 gözlü bir sözlük için C1 1024 değerini alırsa) C1' e 259 başlangıç değeri yüklenir;
- Yeni C1' in değerini taşıyan düğüm eğer yaprak düğüm değilse a) adımına gidilir;
- Yaprak düğüm ise bu düğüm iptal edilir, C1 kodu yeni oluşturulacak sözcüğe atanır.

V.42bis algoritmasının değişik tipdeki dosyalar üzerinde başarımı Şekil 2' de görüldüğü gibidir [5]. Şekil, algoritmadaki sözlük boyunun değişik tipten veriler üzerindeki başarımına etkisini göstermektedir.



Şekil 2. V.42bis' in değişik tip dosyalar üzerindeki başarımı

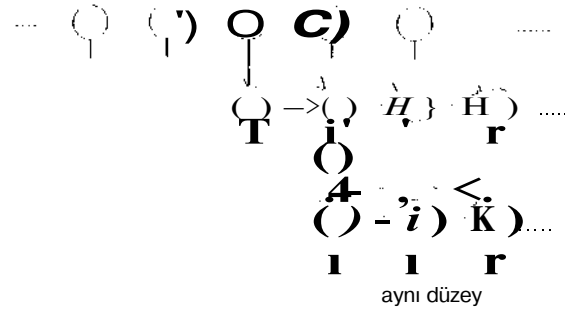
3. V.42bis Algoritmasının Karmaşıklık Analizi

V.42bis veri sıkıştırma algoritması içeriğiyle adreslenebilen bellek (CAM) içermeyen bir sistemde yazılımla gerçekleştirilmek istenirse, kullanılacak birkaç yöntem vardır. Uygulamada hangisinin en uygun yöntem olacağı, bir kod üretmek için gerekli zamana ve algoritmanın gereksinim duyacağı bellek alanına göre belirlenir. Bu amaçla yöntemlerin zaman ve alan karmaşıklığının kullanılması önerilmektedir. İzleyen bölümde, algoritma gerçekleştirirken bağlantılı liste yapılarından yararlanılan dört değişik yöntem önerilmiş ve bunların karmaşık hesapları ayrıntısı ile verilmiştir. Bu hesaplarda T: "haslı" tablosunun boyu, m:alfabe boyu |L|, n: sözlük boyu, d: ağacın derinliği olarak alınmıştır.

a) *Doğrusal bağlantılı liste kullanılması:* Algoritma en basit olarak Şekil 3' deki gibi bağlantılı liste kullanılarak gerçekleştirilir. Bu durumda listenin her bir düğümü, taşıdığı karakter, kendine atanacak kod, bir alt seviyeyi işaret eden bir işaretçi ve aynı seviyedeki bir düğümü işaret eden diğer bir işaretçi içerir. Buna göre bir düğümün veri yapısı aşağıdaki gibidir.

```
struct nodel (
    unsigned char kr;
    unsigned int kod;
    struct nodel -*alt, *yan;
    1;
```

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ



Şekil 3. Doğrusal bağlantılı liste kullanılan yapı

Bir katarın ilk karakteri ağacın köküne denk geldiğinden sözlükte katar arama ikinci karakterden başlar. Her düzeyde de m çevrim gerekebileceğinden arama maliyeti (d-1).m' dir. Algoritmada sözlük boyu sınırlı olacağından ve büyük boyutlu dosyalarda, sözlük hızla dolacağından genelde düğüm koparına işlemi her ekleme için gerekecektir. Dolayısıyla ekleme maliyetinin yanısıra düğüm koparına maliyeti (n-m) de söz konusudur. Bu yapı uyarınca bir kod elde etmenin zaman maliyeti,

$$(d-1).m_1(n-m), O(d.m_1(n-m))$$

olur. Alan maliyeti ise sözlük boyuyla bir düğüm uzunluğunun çarpımı kadardır. 32 bir adres yolu olan bir sistem için alan maliyeti,

$$\cdot // \text{sizeof}(\text{struct nodel}) - n \cdot 11$$

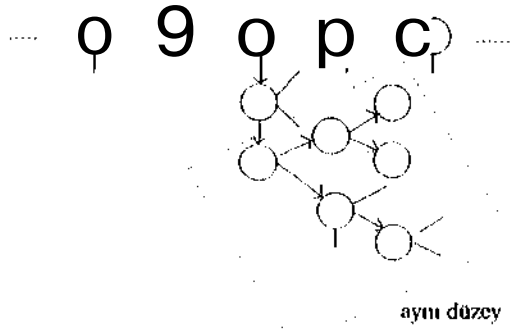
olur.

b) *İkili Ağaç Kullanılması:* bir önceki ağaç yapısında hepsinin aynı köke bağlı olduğunu gösteren yatay doğrusal bağlantılı listenin ikili ağaç biçimine getirilmesiyle (Şekil 4) katar arama maliyeti azaltılır; ikili arama yapılabileceğinden her düzey için gerekli olan m çevrim, logm çevrime düşer.

Bir kod üretmek için zaman maliyeti,

$$(d-1) \log m_1(n-m), O(d. \log m_1(n-m))$$

olur.



Şekil 4. İkili Ağaç Kullanılan Yapı

Alan maliyeti ise yine sözlük boyuyla bir düğüm uzunluğunun çarpımından hesaplanır. Ancak bir düğüm için gerekli yapının uzunluğuna bir işaretçi daha eklendiği için alan maliyeti büyür. Bir düğüm için gerekli yapı, aşağıdaki gibi olup

```
struct node2 {
    unsigned char kr;
    unsigned int kod;
    struct node2 *alt, *sag, *sol;
};
```

32 bit adres yolu olan bir sistemde alan maliyeti,

$$/; \text{sizeof(struct node2)}^t \cdot 15$$
 ofur.

Görüldüğü gibi birinci yöntemde zaman maliyetinin solunda bulunan terim \log mertebesinde azalmıştır, ancak alan maliyeti yapıya bir işaretçi eklenmesinden dolayı artmıştır.

Böyle bir ağaç yapısı kurulduğunda, hesaplanan zaman maliyetindeki ifadenin sol yanındaki terim sağ yanındaki terime göre ihmal edilecek kadar küçülmüştür. Yani sağ terim maliyette daha baskındır. Örneğin, CCITT'nin önerdiği optimum değerler ve ASCII karakterler için ($d=6, m=256, n=2048$),

$$(d-1) \log m \setminus (t-m) \\ - (6-1) \cdot \log 256 \setminus (2048-256) \\ - 40 + 1792$$

olarak hesaplanır. Düğüm koparma maliyeti aramaya göre oldukça fazladır.

c) *Yapraklar ağacı kullanılması:* düğüm koparma maliyetini azaltacak olan bu yöntemde, ikinci yöntem ek olarak yalnızca yaprakların bulunduğu bir veri yapısı daha kullanılır. Bu veri yapısı doğrusal liste ya da ikili ağaç biçiminde olabilir. Zaman maliyetini azaltması açısından ikili ağaç olması daha uygundur; ancak bu seçim alan maliyetini artıracaktır.

İkili ağaç kullanımında, sözlük dolu iken bir ekleme yapılmak gerektiğinde, koparma işleminde uygun düğümün bulunması için arama gerekmez, yapraklar ağacında C1' e en yakın üst kodu taşıyan yaprak seçilir. Bu durumda sözlükten düğüm koparma maliyeti $(n-m)$ 'den küçük olur. Ancak, yapraklar ağacının kurulması ve yaprak durumundan çıkarılmanın yapraklar ağacından çıkarılması zaman maliyetini artıracaktır. Bu işlemin karmaşıklığı aşağıda incelenmiştir:

1° Yeni düğüm eklenirken, eklemenin yapılacağı düğümün yaprak olup olmadığına bakılmalıdır. Eğer yapraksa, bu düğüm yapraklar ağacından çıkarılmalıdır. Maliyeti, $\log(n-m)$ ' dir.

2° Sözlükten, yeni ekleme için bir düğüm koparıldığında, onun bağlı olduğu eski yerin yaprak durumuna gelip gelmediği kontrol edilmelidir. Gelmişse, o düğüm yapraklar ağacına eklenmelidir. Maliyeti, $\log(n-m)$ ' dir.

3° Sözlüğe her yeni sözcük eklemeye eklenen düğüm, sözlükte yeni bir yaprak düğüm olacağından, bu düğümün yapraklar ağacına da eklenmesi gerekir. Sözlük dolduğunda yapraklar ağacından kod ataması için seçilen düğüm aynı kodla başka bir yaprağa atanacağından yapraklar ağacında yeri değişmeyecektir. Buna göre sözlük dolana kadar yapraklar ağacına ekleme, sözlük dolduktan sonra da C1' e atanacak kodu yapraklar ağacında arama maliyeti $\log(n-m)$ olur. Rastgele seçilmiş bir metin sıkıştırılırken oluşmuş yapraklar ağacında sıkıştırma sürecinde dengenin saklı kaldığı [6]' da gösterilmiştir. Bu nedenle sözlük dolduktan sonra yapraklar ağacında düğüm çıkarma ya da

yeni düğüm ekleme sırasında bir dengeleme işlemine gerek görülmemiştir. Bu incelemenin ışığında algoritmanın zaman maliyeti,

$$(d-1) \cdot \log m + 3 \cdot \log(n-m), O(d \cdot \log m \cdot \log(n-m))$$

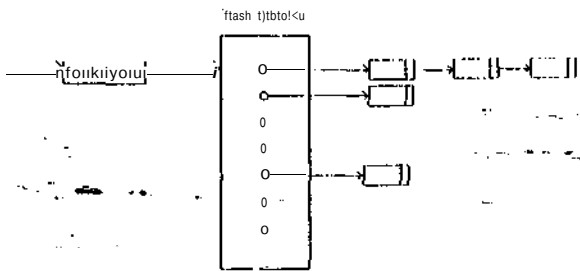
olur. Görüldüğü gibi büyük değerlerde olan m ve (n-m)'ler logaritmik duruma gelir. Bu yöntemin alan maliyeti yapı içi ve yapılar arası bağlar nedeni ile oldukça artar; ikinci yöntemin alan maliyetine n · sizeof(struct leafnode) eklemesi yapılarak hesaplanır. Yapraklar ağacındaki bir düğümün yapısı şudur:

```
struct leafnode (
    unsigned int kod;
    struct leafnode *sol, *sag;
    struct node *p;
};
```

Buna göre 32 bit adres yolu olan bir sistem için, yöntemin alan maliyeti

```
// sizeof(struct node2)+
(n-m) · sizeof(struct leafnode) ≈
n · 15 + (n-256) · 14 = n · 29 - 3584
olur.
```

d) Dinamik "hashing" kullanmak: Sıkıştırma algoritması yazılımla gerçeklerinken kullanılabilen bir yöntem de dinamik "hashing" yöntemidir. Bu yöntem, uygun tablo boyu ve "haslı" fonksiyonu seçilirse, en iyi zaman maliyeti verir; ancak alan maliyeti yükselir. Böyle bir yapı için bir örnek Şekil 5' de verilmiştir.



Şekil 5. Dinamik Hashing Kullanılan Yöntemin Yapısı

Bu yöntemde düğüm yapısı şöyledir:

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

```
struct a {
    unsigned int kod;
    void *q;
    struct a *arka;
};
```

Yöntemde incelenen katare, "hash" fonksiyonu uygulandığında 0 ile (T-1) aralığında bir tamsayı elde edilir (T: "haslı" tablosu boyu). Sonra bu sayı indis olarak kullanılır ve tablonun ilgili gözüne erişilir. İlgili gözde, çatışmaya uğramış katarların bulunduğu bağlantılı listenin kökü alınarak arama yapılır. Arama sonucu olumlu ise sıkıştırılacak diziden yeni bir karakter alınıp uyuşan katare eklenir ve "hash" işlemi tekrarlanır. Bu yöntemde de yapraklar ağacı kurulmalıdır. Bunlara göre zaman ve alan maliyetleri, homojen dağılım sağlayan bir "hash" fonksiyonu varsayımı ile şudur:

$$\text{zaman: } (d-1) \cdot d \cdot n / T + \log(n-m) \cdot d \cdot n / T$$

$$\text{alan: } T \cdot \text{sizeof}(s\text{iretçi}) + (n-m) [\text{sizeof}(s\text{truct } a) + (d-1)/2 \cdot \text{sizeof}(s\text{truct } leafnode)]$$

5. Sonuç

V.42bis algoritması yazılımla gerçekleştirilmek istendiğinde önerilen yöntemlerden biri kullanılabilir. Tablo 1 'de her 4 yöntem için alan ve zaman karmaşıklıkları verilmiştir.

Tablo 1.

Yöntemlerin alan ve zaman karmaşıklıkları (adres yolu:32 bit, tamsayı: 16 bit, d:6, m:256, n:2048, T:2048)

YönAKarmaşıklığı	Zaman	Alan
1	$(d-1) \cdot m + (n-m)$ =1280+1792	11n
2	$(d-1) \cdot \log m + (n-m)$ =40+1792	15n
3	$(d-1) \cdot \log m + 3 \cdot \log(n-m)$ =40+33	29n -3584
4	$(d-1) \cdot d \cdot n / T + (\log(n-m) + d \cdot n / T)$ =30+17	27.5n + 1152

Birinci yöntemde, sözlüğü oluşturan ağaçlar doğrusal bağlantılı liste kullanılarak gerçekleştirilmiştir; ikincisinde, her bir ağacın herhangi bir düzeyindeki düğümün akagaçlarının ikili ağac biçiminde düzenlenmiş.

üçüncüsünde ise ikincisine ek olarak sözlükteki ağaçların yaprak düğümlerinin listesi ayrı bir ikili ağaç olarak tutulmuştur. Dördüncü yöntemde dinamik "hashing" kullanılmaktadır. Tablodaki karmaşıklık ifadesinden de görüleceği gibi yöntemlerin en hızlı olanı dördüncüsü olup, yöntem sıralamasında geriye doğru gidildikçe hızla birlikte bellek gereksinimi de azalmaktadır. Uygulamada hangi yöntemin seçileceği kabul edilebilir hız ve tahsis edilebilecek bellek alanına göre oluşturulan maliyet bağıntılarından gidilerek belirlenebilir.

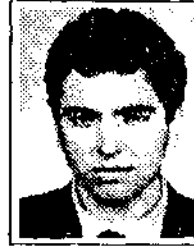
6. Kaynakça

- [1] IJeld, C. Data Compression, John Wiley & Sons, 1988
- [2] Ziv, J., and Lempel, A. A universal algorithm for sequential data compression. IEEE Trans. Inf Theory IT-23 (May 1977), 337-343
- [3] T. A. Welch. A technique for high-performance data compression. I (il il I Computers i 7, pp. 8-19, 1984.
- [4] CCITT; V.42bis Data compression standart, Drall, Sepi., 1989.
- [5] O. Aliefendioğlu, R. Çölkesen, E. Harmancı, B. Örencik. V.42bis Compression Performance. International Symposium On Computer and Information Science VII, 1992
- [6] Hibbard, Thomas N. Some Combinatorial Properties of Certain Trees With Applications to Searching and Sorting. J. ACM 9, 1 Jan 1962, p 13-28.



Rifat ÇÖLKESEN

1962 yılında Adana' da doğdu. 1986 * yılında K.T.Ü.' den elektronik müh., 1990' da İ.T.Ü.' den Bilgisayar Y. Müh. unvanlarını aldı. Şu anda Ç.Ü. Elektik-Elektronik Müh. Bölümü kadrosunda araştırma gör. olup doktora tez çalışmasını tamamlamak üzere İ.T.Ü. ' de görevlendirilmiştir. İlgili ve çalıştığı alanları Mikroİşlemci Mimarileri. Paralel Bilgi İşleme. Lojik Tasarımı ve Gerçek Zamanda Koni roldür. IREE üyesidir.



Osman A. İEFENDİÖÜL U

1968 yılında İstanbul' da doğdu. 1989 ve 1992 yıllarında İ.T.Ü. Elektrik-Elektronik Fak. Elektronik ve İlah. Müh. Bölümünden mühendis ve yüksek müh. unvanlarını aldı. 1991 yılında İ.T.Ü. Elektrik-Elektronik Fak. Kontrol ve Bilgisayar Müh. Bölümü' ne araştırma görevlisi olarak girdi ve halen aynı bölümde bu görevi ve doktora çalışmasını sürdürmektedir. İlgili ve çalıştığı alanları Vci Sıkıştırma. Haberleşme Sistemleri ve Bilgisayar Ağlarıdır. E.M.O. ve IEEE üyesidir.



Hil/ent ÖRENCİK

1950 İstanbul doğumlu. 1971 ve 1975 yıllarında İ.T.Ü. Elektrik Fakültesinde müh. ve yüksek mühendis. 1983 yılında İ.T.Ü. Fen Bilimleri Enstitüsünde doktor unvanını aldı. 1989 yılında Bilgisayar Sistem Yapısı bilimi dalında doçent oldu. 1974 yılından beri İ.T.Ü.' nün değişik bölümlerinde görev yaptı. Halen İ.T.Ü. Elektrik-Elektronik Fakültesi Kontrol ve Bilgisayar Mühendisliği Bölümü öğretim üyesidir. İlgili ve çalıştığı alanları Bilgisayar Mimarisi. (içerik Zamanlı Sistemler. Bilgisayar İletişimi. Çok İşlemcili Sistemler ve Paralel Bilgi İşlemedir. IEKE üyesidir.



A. Emre HARMANCI

1944 yılında İstanbul' da doğdu. 1966 yılında Fransa' daki Insitit National des Sciences Appliquées de Lyon' dan Elektronik Müh. unvanını. 1967 ve 1970 yıllarında Université de Paris' den kuantum elektronisinde D.E.A. uygulamalı fizik dalında d* doktor;] derecesini aldı. 1967 ve 1970 yılları arasında Fransa' da Centre d'Etudes de Saclay-Paris' de araştırma mühendisi olarak. 1971 ve 1977 yılları arasında ise Çekmece Nükleer Araştırma Merkezi' nde araştırma proje yöneticisi olarak çalıştı. 1977 yılından beri İ.T.Ü.' Elektrik-Elektronik Fgk. Kontrol ve Bilgisayar Mühendisliği Bölümü'nde öğretim üyesi olarak görevini sürdürmektedir. İlgili ve çalıştığı alanları Lojik Tasarımı. Bilgisayar Mimarisi ve Bilgisayar Ağlarıdır. IEEE üyesidir.

YIRTI LABİLİR ZAR MODELİYLE RESİM ONARIMI

Tolga Acar ve Muhillin Gökmen
Bilgisayar Bilimleri Ana Bilim Dalı
Elektrik-Elektronik Fakültesi
İstanbul Teknik Üniversitesi
80626 Maslak - İstanbul

Özet

Bu çalışmada., resmin bulanıklaşma nedeninin bilinmesi halinde, bulanık resmin netleştirilerek onarılması problemi ele alınmaktadır. Odaklama hatalarından kaynaklanan bulanıklaşmanın ve bunun üzerine eklenen gürültünün ortadan kaldırılarak, net ve gürültüsüz resmin elde edilmesi amaçlanmıştır. Düzgünleştirme (*regularization*) yaklaşımı yardımıyla yüzey kurarak netleşmiş resmin elde edilmesine çalışılmıştır. Düzgünleştirmede, yırlılabilir zar (*weak membership*) modelinden ortaya çıkan dışbükey enerji fonksiyonelinin, aşamalı dışbükeylik (*Graduated Non-Convexity*) algoritmasıyla çözümü gerçekleştirilmiştir [1, 2]. Böylece resimdeki süreksizlik noktalarının korunmasının yanısıra, bulanıklaşmış olan süreksizliklerin de ortaya çıkarılması sağlanmaya çalışılmıştır. Yöntem, ortalama, alan filtreyle ve Gauss filtresiyle bulanıklaştırılan ve üzerine gürültü eklenen yapay ve gerçek resimler üzerinde uygulanmıştır.

1 Giriş

Resim onarımında temel amaç, bozulmuş bir resimden resmin orijinalinin elde edilmesidir. Bozulma genellikle kamera veya nesnelerin hareketinden, mercek sisteminin iyi odaklanmamasından ve atmosferik olayların neden olduğu bulanıklıktan oluşmaktadır. Bulanıklığa ek olarak, resim alma sırasında kullanılan kayıt ortamından (film zerreciklerinden oluşan gürültü), verinin sayısallaştırıcılarından veya ölçüm hatalarından ortaya, çıkan gürültü de resimde bozulmaya, neden olmaktadır. Gürültülü ve bulanık resim g , orijinal resim f , n gürültü ve D de bu-

lanıklığa karşı düşmek üzere,

$$g = Df + n$$

şeklinde ifade edilebilmektedir [1]. Gürültü raslantısal bir süreç olduğundan, pratik olarak orijinal resimdeki özelliklerin ancak bir kısmı elde edilebilir. Bu yüzden orijinal resim f 'in aynısını elde etmek genellikle olanaksızdır. Resim onarımında, orijinal resim f 'ye olabildiğince yakın olan \hat{f} resminin elde edilmesine çalışılır. Bir onarma yönteminin gerçekleştirilmesi gereken noktalar şu şekilde belirlenebilir:

- Netleştirme,
- Gürültüyü bastırma,
- Süreksizlikleri koruma.,
- Dalgalanmayı azaltma..

Bu amaçla, geliştirilen doğrusal görüntü onarma algoritmaları benek değerlerinde hızlı değişimlerin olduğu bölgelerde dalgalanmaya (*ringing effect*) neden olurlar. Orijinal resimde olmayan bu şekildeki dalgalanmalar istenmeyen bir sonuçtur ve bunları azaltmak için bir takım çalışmalar yapılmıştır [3, 4]. Bu çalışmada netleştirme işlemi, daha önce yüzey kurma amacıyla, geliştirilmiş yırtılabilir zar modelinde gerekli değişiklikler yapılarak gerçekleştirilmeye çalışılmaktadır. Bu modeldeki değişimle, bir yandan netleştirme sağlanırken, diğer yandan çizgi işlevi yardımıyla, süreksizlikler korunmakta ve böylece netleştirme probleminde karşılaşılan dalgalanma sorunu da çözülmeye çalışılmaktadır.

2 Yırtılabilir Zar İle Yüzey Kurma

Yüzey kurma problemine çözüm olarak sunulan yırtılabilir zar modelinde, resim parça parça, düzlenmiş bir fonksiyon yardımıyla model-lenerek yeniden oluşturulmaktadır [1], Yeniden oluşturulan resimde, giriş resmine yakınlığın yanı sıra süreksizlik noktalarının korunmasıyla birlikte gürültü etkisinin de azaltılması istenmektedir.

Modellemede kullanılan zara yırtılabilir özelliğinin katılmasıyla, bir boyuttaki hali aşağıdaki gibi olan üç bölümlü enerji fonksiyonu elde edilmektedir [1, 2]:

$$E = \underbrace{\sum_i (f_i - d_i)^2}_D + \lambda^2 \underbrace{\sum_i (f_i - f_{i-1})^2 (1 - l_i)}_S + \alpha \underbrace{\sum_i l_i}_P \quad (1)$$

Harada d giriş resmini, f de oluşturulan resmi göstermektedir. D terimi, oluşturulan resmin giriş resmine yakınlığını ifade etmektedir. S teriminde resmin sürekliliğinin sağlanmasına çalışılmaktadır. Mantıksal l çizgi işlev (*line process*) değişkenleri zarın yırtılabilir olduğu yerlerde 1 değerini, sürekliliğin sağlanması gereken yerlerde ise 0 değerini alarak enerji fonksiyonundaki S ve P terimleri arasındaki dengeyi kurmaktadır. Buna göre l terimi, bütün çizgi işlevlerinin 1 değerini alarak enerjinin S teriminden gelecek kısmının kaldırılmasını önlemekte ve zarın yırtılabilir olduğu yerlerdeki enerji artımını belirlemektedir. Böylelikle S terimindeki birinci türevin minimizasyonu ile süreklilik sağlanmaya çalışılmakta, buna karşılık sürekliliğin sağlanmasının fazla enerji ortaya çıkaracağı yerlerde zarın yırtılmasına izin verilerek, P teriminden gelen enerjinin S 'den gelene oranla daha az olduğu yerlerde süreksizlik koşulu bozulmaktadır. Fonksiyonun parça parça sürekliliği, zarın yırtılabilir olmasıyla sağlanmaktadır.

Yukarıdaki (2) enerji fonksiyoneli, düğüm değişkenleri f ve çizgi işlevleri l fonksiyonudur. Yırtılabilir zar modelinde, çizgi işlevleri f cinsinden ifade edilmekte ve minimizasyon işlemi yalnızca, f 'ler üzerinden yapılmaktadır. Çizgi işlevlerinin kaldırılması amacıyla, E enerji

fonksiyoneli bir boyutta, şu şekilde ifade edilmektedir [1]:

$$F = f + T(f) \ll A(f, f_i - l),$$

$$T(f, l) = \sum_{i,j} \ll \ll n_{i,j} \ll [A^2(f) (1 - l_{i,j}) + \alpha],$$

$$g_{\alpha, \lambda}(t) = \begin{cases} \lambda^2 t^2 & |t| < r \\ \alpha - c(|t| - r)^2/2 & r \leq |t| < r \\ \alpha & |t| \geq r \end{cases} \quad (2)$$

$c = c'/p$, $r^2 = \ll (2/c + 1/A^2)$, $q = r/(A^2 r)$ ile ifade edilmektedir [1]. Çizgi işlevleri şu şekilde tekrar elde edilebilmektedir:

$$l = \begin{cases} 1, & |f| > y/ZI \\ 0, & \text{diğer hallerde.} \end{cases} \quad (3)$$

Sonuç olarak elimizde, iki boyutta aşağıdaki gibi gösterilebilen dışbükey bir enerji fonksiyoneli oluşmaktadır.

$$F = \sum_{i,j} (f_{i,j} - d_{i,j})^2 + \sum_{i,j} g_{\alpha, \lambda}(f_{i,j} - f_{i-1,j}) + C' \ll A(f, l - f_{i,j} + l) \quad (4)$$

Bu durumda, problem, (5) denklemini minimize eden f fonksiyonunu bulmaktır. Ancak söz konusu fonksiyonelin içbükeylik özelliği olmadığı için, sıradan bir optimizasyon algoritması bu fonksiyonelin bütünsel minimumunu bulamaz. Çünkü $N \times N$ boyutlu bir görüntü için bir boyutta 2^N , iki boyutta $2^{N \times N}$ tane f çizgi işlevi, (5) denklemini minimum yapabilir, bu minimumlardan sadece birisi bütünsel minimumdur. Onu bulmak amacıyla tüm f değişkenlerini denemek ve her birisinde en küçük enerjiyi oluşturan f 'i almak çözüm olsa da, bunun uygulanabilir olmadığı açıktır.

Aşamalı dışbükeylik (GNC) yöntemi [1], dışbükey fonksiyonun içbükey yaklaşımdan başlanarak, bir parametreye bağli olarak, git-tikçe dışbükeyleşen fonksiyon üzerinde minimum aranmasıyla gerçekleştirilmektedir. CNC algoritmasında bütünsel minimumu aranacak içbükey olmayan E fonksiyonuna, yeni bir $f^{(p)}$ ($1 \geq p \geq 0$) fonksiyon dizisiyle yaklaşılmaktadır. $F^{(p)}$ fonksiyon dizisinin ilk elemanı $E^{(0)} = E^*$ içbükey bir fonksiyon, ve dizinin son elemanı $F^{(p)}$ ise f 'nin kendisidir. Çözüm, $E^{(p)}$ ile

başlayarak $f^{(0)}$ 'a kadar, E^{\wedge} dizisinde $f^{(n)}$ 'in azalan değerleri için bulunan ara çözümlerin, dizideki bir sonraki fonksiyonun çözümünde kullanılmasıyla elde edilmektedir. İler p değerine ilişkin minimizasyonda SOR (*Successive Over-Relaxation*) algoritması kullanılmıştır [1]. Buna göre enerji fonksiyonundan çıkartılan ayrık zamandaki Euler-Lagrange denklemleri kullanılarak, $(n-1)$. iterasyondaki $f_{i,j}^{(n)}$ değeri şu şekilde hesaplanmaktadır:

$$f_{i,j}^{(n+1)} = f_{i,j}^{(n)} - \omega [2(f_{i,j}^{(n)} - d_{i,j}) + g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i-1,j}^{(n)}) + g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i,j-1}^{(n)}) + g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i+1,j}^{(n)}) + \dots (y^{(n)} - f^{(n)})] \frac{1}{2 + 8A^2} \quad (5)$$

burada LJ SOR'in gevşeme parametresi, $g'_{\alpha,\lambda}$ komşu etkileşim fonksiyonudur [1]:

$$g'_{\alpha,\lambda}(t) = \begin{cases} 2A^2 t, & |t| < q \\ -\langle -(\lambda - r) \rangle \text{sign}(t), & q < |t| < T \\ 0, & |t| \geq r. \end{cases}$$

3 Yırtılabilir Zar Modeli Kullanılarak Netleştirme

Bu çalışmada bulanıklaşmış resimlerin onarımını, yırtılabilir zar modelinin kullanılmasıyla gerçekleştirilmektedir. Denklem (5)'de yüzey kurma amacıyla geliştirilmiş model, bulanık resimlerin onarımında kullanılmak üzere değiştirilmiştir. İki tür dalgalanma gözlenmektedir [A]:

1. Resmin sınırlarına, yakın bölgelerdeki dalgalanmalar,
2. Denk değerlerinin süreksizliklerinin yakınlarındaki dalgalanmalar.

Bu modelde kullanılan çizgi işlevleri yardımıyla dalgalanmanın olmadığı çözümler elde edilmeye çalışılmıştır. Bu çalışmada içinde, bulanıklığa neden olan b operatörünün bilindiği, orijinal görüntünün bilinen f ile bozulduğu kabul edilmiştir. Resmin bilinen bir lineer b bulanıklaştırma operatörüyle bozulması halinde, (5)'deki enerji fonksiyonunda f terimi yerine bulanıklaşmış resmi ifade eden $f * b$ terimi gelecektir. Onarım sonucunda elde edilecek resmin,

(5) denkleminin ilk teriminden görüleceği gibi, bulanıklıkla uğruşmuş orijinal resme yakın olması istenmektedir. Böylece bu modelde, onarılarak elde edilen, resmin b ile bozulmuşunun, bulanık giriş resmine yakın olması gerekliliğinden yararlanılmıştır.

Giriş verisine yakınlık, (5) denklemindeki enerji fonksiyonunun minimumunun bulunmasıyla sağlanmakla birlikte, aynı fonksiyonun ikinci terimiyle, A ölçek parametresine bağlı olarak resmin her yerinde düzleme de yapılmaktadır. Süreksizliklerin olduğu yerlerde düzlemenin olmaması için, süreksizlik noktalarında çizgi işlevi [1] tanımlanmakta, ve bu durumda, enerji fonksiyonu şu hali almaktadır:

$$F = \sum_{i,j} (f_{i,j} - d_{i,j})^2 + \sum_{i,j} g_{\alpha,\lambda}(f_{i,j} - f_{i-1,j}) + \sum_{i,j} g_{\alpha,\lambda}(f_{i,j} - f_{i,j+1}) \quad (6)$$

İttirada f , bulanıklaştırma işlevinin yerden bağımsız nokta dağılım fonksiyonudur (*joint-invariant point-spread function - PSF*). Yukarıdaki fonksiyonun birinci terimi, oluşturulan resmin bulanıklığının ($f * b$) giriş resmine yakın olmasını sağlamaktadır. Bu terim ile resimdeki yavaş değişimler, odaklanarak hızlı değişimler şeklinde açığa çıkarılmaktadır.

$g_{\alpha,\lambda}$ fonksiyonlarından oluşan diğer terimlerde A parametresine bağlı olarak resmin süreksizlik dışındaki bölgeleri seçilerek, bu yerlerde düzleme yapılmaktadır. Birinci terimle geri kalan terimler arasındaki çelişki, enerji fonksiyonunun, süreksizliklerin oluşturulmasını ayırmanın korunması ve bulanıklığın giderilmesiyle sağlanmasıyla birlikte görüntünün sürekli olmasını da sağlayan terimler yoluyla dengesini kurmaktadır.

Minimumu bulunacak (7) enerji fonksiyonunun çözümünde aşamalı dışbükeylik (GNC) algoritması kullanılmaktadır. GNC'nin her bir f değeri için yapılması gereken minimizasyon işleminde SOR kullanılmaktadır. Bu durumda (6) iteratif denklemleri b fonksiyonunun katılımıyla şu şekilde gelmektedir:

$$f_{i,j}^{(n+1)} = f_{i,j}^{(n)} - \omega [\frac{\partial}{\partial f_{i,j}} (f_{i,j} * b - d_{i,j})^2 + g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i-1,j}^{(n)}) + g'_{\alpha,\lambda}(f_{i,j}^{(n)} - f_{i,j+1}^{(n)}) + \dots (f^{(n)} - f^{(n)})] + \langle A (f_{i,j}^{(n)} - f_{i,j+1}^{(n)}) \rangle \frac{1}{2 + 8A^2} \quad (7)$$

g'_a, X komşu etkileşim fonksiyonu (O)'dakine benzer şekilde tanımlanmaktadır [1].

Denklem (8) içindeki $\frac{\partial}{\partial f_{i,j}} \langle d^* l \rangle \sim d_{i,j}^2$ teriminden, bulanıklaştırma sisteminin nokta dağılım fonksiyonu $f_{i,j}$ 'nin boyutu kadar $f_{i,j}$ teriminin türevi gelmektedir. Örneğin 3x3 boyutlu ortalama alan

$$b = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (8)$$

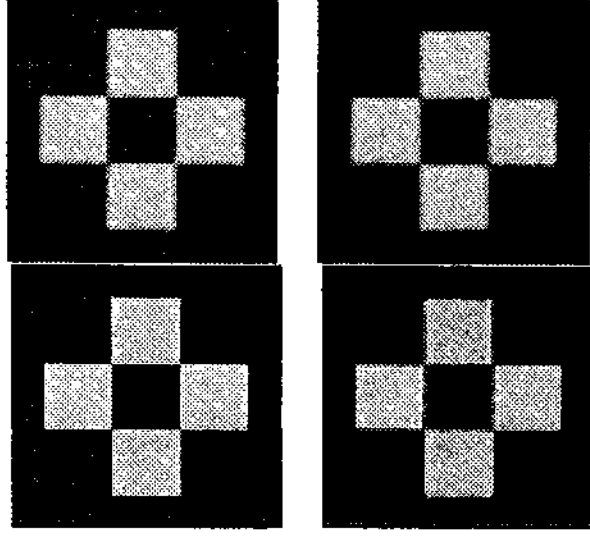
filtresi için sadece bu terime ilişkin denklem şu şekilde olacaktır:

$$\begin{aligned} \frac{\partial}{\partial f_{i,j}} \{ & (f_{i-1,j-1} + f_{i-1,j} + f_{i-1,j+1} + \\ & f_{i,j-1} + f_{i,j} + f_{i,j+1} + \\ & f_{i+1,j-1} + f_{i+1,j} + f_{i+1,j+1}) - \langle h_{i,j} \rangle^2 = \\ & 2 \left\{ \frac{1}{9} [f_{i,j} + f_{i,j+1} + f_{i,j+2} + \right. \\ & f_{i+1,j} + f_{i+1,j+1} + f_{i+1,j+2} + \\ & f_{i+2,j} + f_{i+2,j+1} + f_{i+2,j+2}] + \\ & \dots \dots \dots \\ & \dots \dots \dots \\ & \left. \frac{1}{9} [f_{i-2,j-2} + f_{i-2,j-1} + f_{i-2,j} + \right. \\ & f_{i-1,j-2} + f_{i-1,j-1} + f_{i-1,j} + \\ & f_{i,j-2} + f_{i,j-1} + f_{i,j} + \\ & f_{i+1,j-2} + f_{i+1,j-1} + f_{i+1,j} + \\ & \left. - d_{i,j} \right\} \end{aligned} \quad (9)$$

Değişik bulanıklaştırma (filtreleri için, / terimlerinin sayısı ve katsayıları değişik olacaktır. Gerçeklemede, filtre boyutuna ve içeriğine göre uygun şekilde türev alan ve (10) denkleminin değerini hesaplayan bir fonksiyon yazılmış, herhangi bir bulanıklaştırma. filtresi için GNCT1 zar yöntemiyle yüzey kuran program gerçekleştirilmiştir.

4 Deneysel Sonuçlar ve Yorumlar

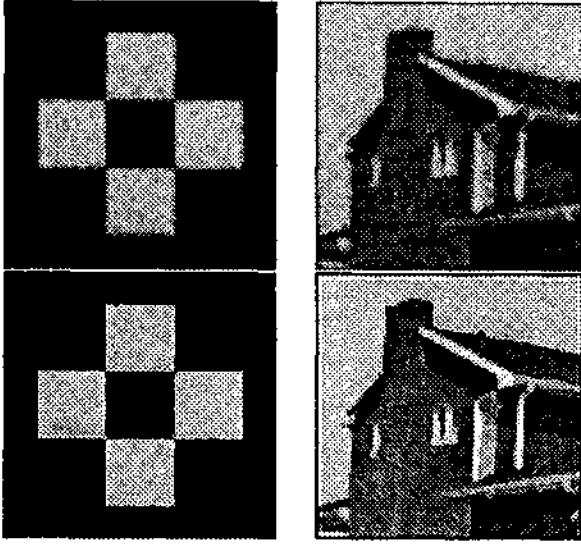
Şekil 1'in üst sırasında, denklem (8)'de verilen 3x3 boyutlu ortalama filtresinin iki ve üç defa 128x128 boyutlu dama tahtası resimlerine uygulanmasıyla elde edilen bulanık resimler görülmektedir. Alt sırada ise, açıklanan yöntemle elde edilen onarılmış resimler gösterilmektedir. Uu çözümlerin elde edilmesinde kullanılan tipik parametre değerleri



Şekil 1: 3x3 büyüklüğünde ortalama filtresinin iki (sol üst) ve üç (sağ üst) defa uygulanmasıyla elde edilmiş bulanık dama tahtası resimleri ve bunlara ilişkin onarılmış resimler (alt sıra).



Şekil 2: 3x3 büyüklüğünde ortalama, filtresinin bir (sol üst) ve iki (sağ üst) defa uygulanmasıyla elde edilmiş bulanık ev resimleri ve bunlara ilişkin onarılmış resimler (alt sıra).



Şekil 3: Bulanıklaştırılmış ve gürültü eklenmiş (SNR=25 dB) dama tahtası (sol üst) ve ev (sağ üst.) resimleri ve bunlara ilişkin onarılmış, resimler (alt. sıra).

Tablo 1: Bulanıklaştırılmış ve gürültü eklenmiş (SNR=25 dB) dama tahtası ve ev resimlerine ilişkin normalleştirilmiş ortalama kare hata (NMSE,) değerleri.

SNR (dB)	NMSE	
	Dama tahtası	Ev
20	107	797
25	17	814
30	13	861
40	6	775
50	3	773
∞	0	773

$A = 1$ (i ve o = KiOO'dür. Şekil 2'de, ayın ortalama, liltresimni bir ve iki defa uygulanması ile oluşturulan bulanık ev resimlerinden elde edilen benzer sonuçlar gösterilmektedir. Şekil 3 ise, bulanıklaştırılmış dama tahtası ve zar resimlerine, sıfır ortalama, değerli, Gauss gürültüsü (SNR=25 dB) eklenerek oluşturulmuş bozuk resimleri ve onlardan elde edilen onarılmış resimleri göstermektedir. Tablo 1'de, değişik SNR değerli, bulanık ve gürültülü dama tahtası ve ev resimlerinin onarılması sonucu elde edilen normalleştirilmiş ortalama kareler hatası (normalized mean square error-NMSE) verilmektedir.

Bu sonuçların elde edilmesi sırasında, A ve a parametre değerlerinin, SOR iterasyonlarının yakınsamasında, oldukça belirleyici oldukları görülmüştür. Örneğin dama tahtası resmi için A parametresinin değerindeki 0.1 Tık bir değişim, iterasyon sayısının 50'den 260'a çıkmasına neden olmaktadır. A parametresinin 1'den küçük değerleri için ise, iterasyonlar sırasında μ 'nin yeni değerlerinin hesaplanması sırasında benek değerlerinin iç bir küme üzerindeki düşümleri (practical convergence) alınmaktadır. Bu p izdüşümü, benek değerleri 8 bitle ifade edilen bir resim için

$$p_{i,j} = \min(1, \frac{1}{\sigma^2} (f_{i,j} - \mu)^2)$$

şeklinde oluşturulmaktadır. Bu izdüşüm, SOR iterasyonları sırasında, özellikle A 'nın küçük değerleri için, benek değerlerinin normalde alamayacakları çok büyük değerlere ulaşmalarını engellemektedir.

Şekil 1,2,3 ve tablo I incelendiğinde, yntılabilir zar modeliyle netleştirme yönteminin, bulanık resimleri başarıyla netleştirdiği görülmektedir. Görüldüğü gibi, yntılabilir zar modelinde yapılan değişikliklerle, bulanık resimler netleştirilmekte, gürültü bastırılmakta, resimdeki süreksizlikler korunabilmekte ve doğrusal onarma yöntemlerinde görülen dalgalanma etkileri resim içinde kaldırılmaktadır. Bu yaklaşımda, süreksizliklerin korunması ve dalgalanmanın azaltılması, zarın yntılabilmesini sağlayan çizgi işlevi dolayısıyla gerçekleşmektedir.

Bulanıklaşmış resimlerin üzerine eklenen sıfır ortalama (Gauss gürültüsünün varyans hesabı şu şekilde yapılmıştır:

$$SNR_{dB} = 10 \log_{10} \left[\frac{\frac{1}{N^2} \sum_{i,j} (f_{i,j} - \mu_f)^2}{\sigma_n^2} \right]$$

