

# PARALEL ARRAY BİLGİSAYARLARDA BLOK DENGELEME İŞLEMİ

Yrd. Doç. Dr. İbrahim BAZ  
Selçuk Üniversitesi  
Müh.-Mim. Fakültesi  
KONYA

## ÖZET

Bu yayında konvensiyonel bilgisayarlardan tasarım, mimari, donanım, yazılım ve işletim sistemi olarak tamamıyla farklılıklar gösteren paralel bilgisayarlar, onların yapıları ve paralel yazılım dilleri hakkında bilgi verilmektedir. Ayrıca havai nirengi probleminin paralel bilgisayarlarda çözümünün ele alınış biçimi anlatılmaktadır.

## 1. GİRİŞ

Geçtiğimiz son 20 yıl içerisinde havai nirengi işlemi ve havai nirengi probleminin çözümü için geliştirilen bilgisayar programlarında gözle görülür gelişmeler ve sonuçlar elde edilmiştir. Uygulamalarda karşılaşılan en büyük güçlüklerden biri çok sayıda bilinmeyen ihtiva eden denklem takımlarının çözümü olarak ortaya çıkmaktadır. Ekonomik olarak bir blokun dengelenebilmesi için bilgisayar kapasitesi ve bilgisayar zamanı hala etkin faktörler olarak karşımıza çıkmaktadır. Daha hızlı bilgisayar performansları daima arzu edilmiş olup, son 10 yıldır bilgisayar teknolojisi ve tasarımında bu yönde gelişmeler görülmüştür. Büyük Ölçekli Entegre Devrelerin kullanılmasıyla birlikte konvensiyonel bilgisayarlardan defalarca daha hızlı ve güçlü vektör/array processorlerin geliştirilmesi sağlanılmıştır. Fakat bu yeni tür processorlerin tasarım, mimari, donanım ve işletim sistemi olarak konvensiyonel bilgisayarlardan tamamıyla farklılıklar göstermesi sebebiyle konvensiyonel bilgisayarlar için geliştirilmiş bulunan bilgisayar programlarının bu processorlerde değerlendirilip, çalıştırılması mümkün olamamaktadır. Bu yazıda, havai nirengi probleminin Paralel Array processorlerde, çözümü için geliştirilen algoritmalar hakkında bilgi verilmektedir. Ayrıca, gerek ışın demetleri, gerekse bağımsız modeller için geliştirilen paralel programların yapısı ve çalışma tarzları izah edilmektedir.

## 2. BİLGİSAYARLARIN SINIFLANDIRILMASI

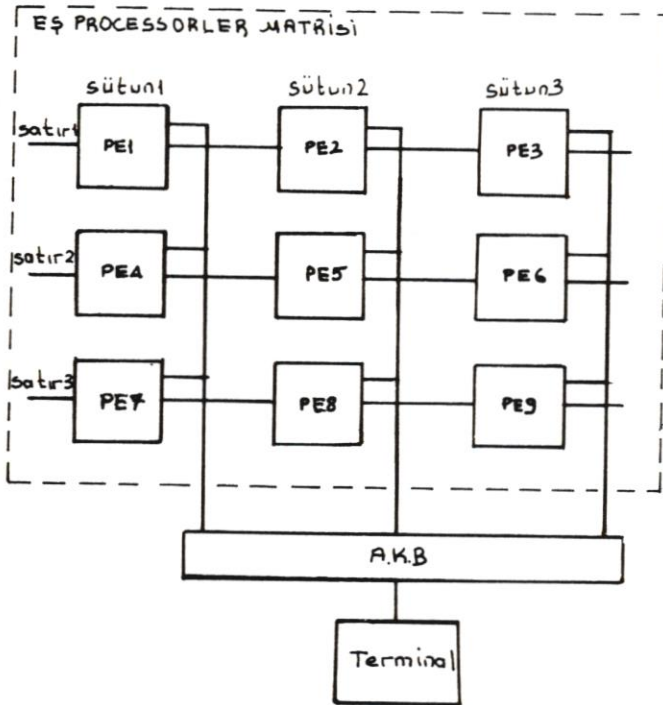
Bilgisayarlar komut ve veri işlenişindeki paralellik durumuna göre sınıflandırılmaktadır.<sup>1,2</sup> Konvensiyonel bir bilgisayarda bir program işlemi görürken birim zamanda sadece bir komut işlem görmektedir. Bu komut ise sadece tek bir veri değerini etki etmektedir. Bilgisayarlar genel olarak şu şekilde sınıflandırılmaktadır.

1. Birim Zamanda Bir Komut / Bir Veri İşlenen Bilgisayarlar. Bu tür bilgisayarlar konvensiyonel bilgisayarlardır.
2. Birim Zamanda Bir Komut / Çok Veri İşlenen Bilgisayarlar. Bu tür bilgisayarlar array, paralel, paralel array veya vektör bilgisayarları olarak bilinmektedir. Bu bilgisayarlarda çok sayıda aynı donanımlı küçük işlemciler (Processing Elements - PE'lere) bir kontrol biriminden komut gönderilmektedir. Bu komut

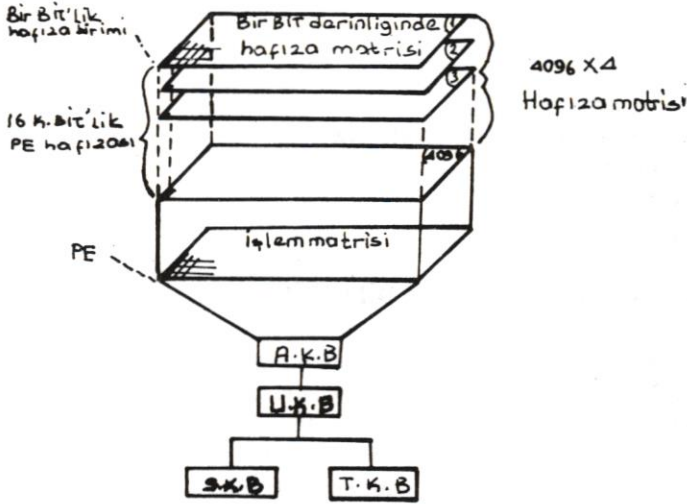
PE'lerin tamamında birden aynı anda işlem görmektedir. Bundan anlaşıldığı gibi bu bilgisayarlarda paralellik veri işlenişinde sağlanmaktadır.

3. Birim Zamanda Çok Komut / Bir Veri İşlenen Bilgisayarlar. Bu tür bilgisayarlar paralelliği veri yerine komut işlenişinde sağlamaktadır. Bu bilgisayarlarda bir veri değeri birden fazla sayıda farklı komutla aynı anda işlem görmektedir.
4. Birim Zamanda Çok Komut / Çok Veri İşlenen Bilgisayarlar. Bu tür bilgisayarlarda paralellik komut ve veri işlenişinin her ikisinde de sağlanabilmektedir. Genellikle birden fazla sayıda aynı türde bilgisayarlar birbirlerine bağlanılarak gerekli donanımı oluşturmaktadır. Böyle bir donanım içerisinde bir program işlem görürken programın birbirlerinden bağımsız olan kısımları aynı anda farklı bilgisayarlarda işlenebilmektedir.

Genellikle paralel bilgisayarlarda komutları yayınlanan bir kontrol birimi ve verilerin saklanıp işlem gördüğü matris elemanları şeklinde dizili PE'lerden ibarettir (Şekil 1). Her bir PE çok küçük ve basit bir bilgisayarın yapısında görülebilecek bir işlem (aritmetik-mantıksal) birimine ve küçük kapasiteli bir hafıza birimine sahiptir. Bu hafıza birimleri PE'lerin hepsi için aynı bit/byte derinliğinde kapasiteye sahiptir. Hafıza düzeninin tasarımı itibarıyla PE'ler bit derinliğinde matrisler oluşturmaktadır (Şekil 2). Bu tür donanıma sahip bilgisayarlara örnek olarak Burroughs ILLIAC IV, Burroughs PEPE ve ICL DAP verilebilir.<sup>3,4,5,6</sup>



Şekil 1. Paralel bilgisayarların yapısı.



Şekil 2. Paralel bilgisayarlarda hafıza düzeni tasarımı.

### 3. PARALEL BİLGİSAYARLARDA YAZILIM

Paralel bilgisayarların donanım yapısı konvansiyonel bilgisayarlardan büyük ölçüde farklılıklar göstermektedir. BASIC, FORTRAN, COBOL, PASCAL, vb. gibi yüksek seviyeli programlama dilleri ile yazılan programlar paralel bilgisayarların donanım yapısına ve çalışma sistemine uymamaktadır. Bu nedenle, yazılım şekli itibariyle yüksek seviyeli programlama dillerine benzeyen paralel programlama dilleri geliştirilmiştir.<sup>4,7,8</sup> Bu programlama dillerinin en önemli özellikleri skalar, vektör veya matris birimleri ile işlem yapabilmeleridir. Matris birimleri yapı itibarıyla bilgisayar donanımını oluşturan PE'lerin düzenine uyum sağlamaktadır. Paralel programlamada PE'ler birbirlerinden bağımsız fakat aynı anda işlem görebilmektedir. Gerekliğinde komşu PE'ler arasında aynı yönde ve aynı anda veri alış verişi sağlanabilmektedir. Paralel programlamada bir vektör veya matrisin elemanlarının değerleri diğer bir vektör veya matrisin elemanlarına  $A = B$  gibi basit bir ifade ile atanabilmektedir. Vektör veya matrisler yalnız veya setler halinde oluşturulabilmektedir. Bu setlerden istenildiği takdirde bir vektör veya matris seçilebilmektedir. Ayrıca mantıksal vektör veya matris ifadeleri kullanılarak bir vektör veya matrisin elemanlarından istenilenler seçilebilmektedir. Aşağıda paralel programlama dille-

rinden birisi olan DAP FORTRAN ile yazılmış olan A ve B gibi iki matrisin cebirsel çarpımını ( $C = A \times B$ ) örnek olarak verilmiştir.

REAL\*4 MATRIX FUNCTION MATMUL (A,B,M)

C

C Bu FUNCTION A ve B gibi 64x64 elemanlı iki matrisin

C çarpımını verir.

C

```

REAL*4 A(,),B(,),C(,)
C=0.
DO 1 K= 1,M
1 C=C+MATC(A(,K))*MATR(B(K,))
MATMUL=C
RETURN
END

```

#### 4. PARALEL PROGRAMLAMA DİLLERİNDE ARİTMETİK İŞLEMLER

Paralel programlamada vektör veya matris birimleri (PE'ler) bir hesap elemanı değildir. Öncelikle bu birimler birer hafıza elemanıdır. Bütün PE'ler sadece kendi veya ge-rekli adresleme yapıldığında komşu PE'lerin, hafızalarındaki veri değerlerine aynı anda ulaşabilmektedir. İki vektör veya matrisin toplamı veya farkları lineer cebirde olduğu gi-bi  $A = B + C$  veya  $A = B - C$  olarak ifade edilmektedir. Örnek olarak bir paralel bilgisa-yarın  $3 \times 3$ 'lük matris şeklinde 9 adet PE'den ibaret olduğunu varsayarsak,  $A=B+C$  ifade-sinin işlemi şu şekilde sonuçlanmaktadır.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} b_{11} + c_{11} & b_{12} + c_{12} & b_{13} + c_{13} \\ b_{21} + c_{21} & b_{22} + c_{22} & b_{23} + c_{23} \\ b_{31} + c_{31} & b_{32} + c_{32} & b_{33} + c_{33} \end{bmatrix}$$

İki vektör veya matrisin çarpımı ise bilinen matris çarpım kurallarına benzeme-ktedir. Burada sadece aynı indisli matris elemanları birbirleri ile çarpılarak işlem ya-pılmaktadır. Vektör veya matrislerin çarpımı  $A = B \times C$  şeklinde ifade edilebilmektedir. Sonuç olarak da;

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} b_{11} \times c_{11} & b_{12} \times c_{12} & b_{13} \times c_{13} \\ b_{21} \times c_{21} & b_{22} \times c_{22} & b_{23} \times c_{23} \\ b_{31} \times c_{31} & b_{32} \times c_{32} & b_{33} \times c_{33} \end{bmatrix} \quad \text{elde edilmek-} \\ \text{dir.}$$

Bilinen kurallara göre iki vektör veya matris birbirleri ile bölünemez iken, paralel programlamada iki vektör veya matrisin elemanları birbirlerine bölünebilmektedir. Böl-me işlemi  $A = B/C$  olarak ifadelendirilmekte ve işlem sonunda;

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} b_{11}/c_{11} & b_{12}/c_{12} & b_{13}/c_{13} \\ b_{21}/c_{21} & b_{22}/c_{22} & b_{23}/c_{23} \\ b_{31}/c_{31} & b_{32}/c_{32} & b_{33}/c_{33} \end{bmatrix} \quad \text{elde edilmektedir.}$$

#### 5. HAVAI NİRENGİ PROBLEMİNİN PARALEL BİLGİSAYARLARDA ÇÖZÜMÜ

Havai nirengi uygulamalarında veri değerleri genellikle bir komparatör veya bir üniversal çift resim değerlendirme aletinde resim veya modellerin ölçülmesi yoluyla el-de edilmektedir.<sup>9</sup> Bunu takiben de perspektif izdüşüm esaslarına dayanan matematiksel

modeller kullanılarak en küçük kareler yöntemine göre blok dengeleme işlemi yapılmaktadır. Işın demetleri ve bağımsız modeller blok dengeleme yöntemlerinde hesaplamalarda takip edilecek işlem sırası kabaca şu şekilde özetlenebilir.

1. Her bir resim için yaklaşık kamera parametrelerinin hesabı (Işın demetleri yöntemi).
2. Her bir resim veya model noktası için hata denklemlerinin yazılması.
3. Normal denklemlerin oluşturulması.
4. Normal denklemlerin çözümü.
5. Nokta koordinatlarının ve kamera parametrelerinin dengeleme öncesinde yaklaşık değerlerine düzeltmelerin getirilmesi (Işın demetleri yöntemi).
6. Model noktaları koordinat değerlerinin arazi koordinatlarına dönüştürülmesi (bağımsız modeller yöntemi).
7. Yeni bir iterasyon.

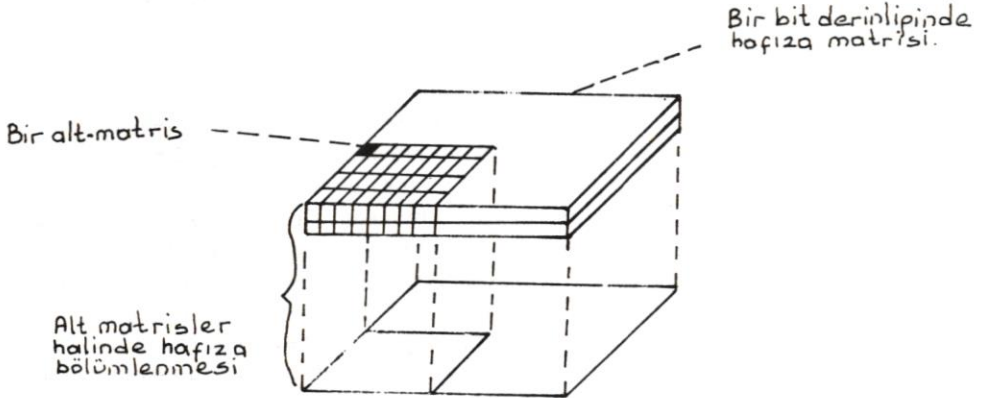
Görüldüğü gibi her bir adım resim, model veya nokta sayısına bağlı olarak aynı türden işlemlerin defalarca tekrarlanmasından ibarettir. Daha önce tanımlandığı gibi, bir komutun aynı tür donanıma sahip çok sayıda PE'nin farklı ve birbirlerinden bağımsız veri değerlerine aynı anda etki edebilmesi ile paralellik sağlanabilmektedir. Buradan havai nirengi probleminin çözüm şeklinin paralel işlem yapabilme özelliklerine uygun olduğu anlaşılmaktadır.

Genellikle paralel bir bilgisayardaki PE'lerin yapısı ve bunların toplam adedi sabittir. Birim zamanda tek bir komut ile değerlendirilmek istenen verilerin sayısı da PE'lerin toplam sayısını geçemez. Bu nedenle, blok dengelemede veri değerlerini temsil etmek üzere seçilecek birimlerin (bir nokta, bir resim, bir model, bir kolon veya bir blok olabilir) toplam sayısı PE'lerin sayısını geçmemelidir. Ayrıca bir bloku oluşturan birimlerin yapısı PE'lerin matris halindeki yapısına uymak zorundadır. Genellikle blok dengeleme yöntemlerinde kabul edilen birim bir resim veya bir model olmaktadır. Paralel bilgisayarlar için geliştirilen ışın demetleri ve bağımsız modeller blok dengeleme programlarında da bir resim ve bir model hesaplamalarda birim olarak kabul edilmiştir. Ayrıca blok dahilindeki bütün birimlerin yapılarının homojen olduğu varsayılmıştır. Başka bir deyişle, bütün resimlerin aynı özelliklere sahip olduğu ve değerlendirilmeye tabi tutulacak maksimum sayıdaki resim/model noktalarının bütün resimler için eşit olduğu kabul edilmiştir. % 60 boyuna ve % 25 enine indirmeli olarak çekilen resimlerle paralel işlem yapabilme için gerekli düzenli veri yapılarının oluşturulabileceği tespit edilmiştir.<sup>10</sup>

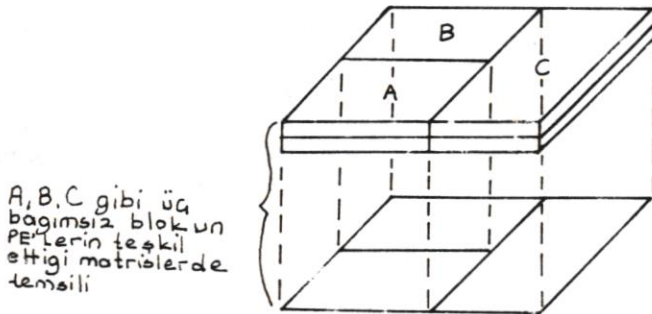
PE'ler paralel değerlendirmeyi vektör veya matris formları biçiminde saklanan veri değerleri üzerine yaptığından, blok dengeleme birimlerince oluşturulacak veri yapıları düzenli formlar halinde olmak zorundadır. Paralel bilgisayarlarda düzenli veri yapıları PE'lerin oluşturduğu 1 bit derinliğinde matris yapıların alt-matrislere bölünmesi ile sağlanabilmektedir. Bölümlenme sonunda bütün alt-matrisler eşit sayıda ve eşit boyutlarda PE bulundurmaya zorundadır. Örnek olarak ışın demetleri yöntemi için geliştirilen algoritmayı verebiliriz. Bu algoritma ile 1 bit derinliğindeki matris yapıları PE'ler bölünerek her biri 6x3 boyutunda toplam 18 PE'den ibaret alt-matris yapılarına ayrıştırılmaktadır. Buradaki 18 sayısı bir resim için kabul edilebilecek maksimum nokta sayısıdır. Bağımsız modeller için geliştirilen algoritmada ise her bir model için 4x4 boyutunda toplam 16 PE'den ibaret alt-matris yapıları teşkil edilmektedir.<sup>10</sup> Her bir PE'de kaç bitlik bir yerin işgal edileceği saklanan veri değerlerinin özelliğine bağlıdır. Alt-matrislerin oluşturulması tamamlandığında bir matris, fiziki görünüm olarak, bir bloktaki birimlerin yanyana/alt

alta dizilmiş halini ifade etmektedir. Komparatör veya çift resim değerlendirme aletleri ile yapılan ölçmelerden oluşturulan blok verilerinin hiç bir düzenleme yapılmadan paralel işlem görmesi mümkün değildir. Buna neden de verilerin gelişigüzel bir yapıda ve sırada olabileceğidir. Her bir resim/model için ölçülen noktaların sayısı ve bunların resim/model içerisindeki dağılımları gelişigüzel olabilmektedir. Bu nedenle, özel bir yol takip edilerek öncelikle ölçülen blok verilerinin düzenli bir yapı oluşturması sağlanmalıdır. Eğer paralel değerlendirmeye uygun olmayan yapıda bir blok değerlendirilmesi söz konusu ise, bu takdirde gelişigüzel yapıdaki blok verileri paralel işlem görebilecek ve alt-matris yapılarına uyum sağlayabilecek şekilde yeniden düzenlenmelidir.

Şekil 3'te 4 kolon ve her bir kolonu 8 resim/modelden oluşan bir blokun verilerinin alt-matrisler halinde gösterimi sunulmaktadır. Şekilde boş görülen kısımlardaki PE'ler hiçbir veri değeri taşımamaktadır, yani boştur. Önce de belirtildiği gibi, PE'ler verilen bir komutun işlemini hep birlikte yerine getirirler. Bu durumda boş PE'lerin hepsi sıfır değerleri ile boş yere işlem yapmaktadır. Bu takdirde, boyut olarak yerleştirilebildikleri takdirde, birden fazla sayıda ve birbirlerinden bağımsız blokların dengelenmesinin de yapılabilme imkanı olmaktadır. Şekil 4'te A, B, C gib bağımsız blokların paralel bir bilgisayarda oluşturacağı matris yapıları görülmektedir.



Şekil 3. Toplam 32 modelden oluşan bir blok'un paralel bir bilgisayarın hafızasında temsili olarak saklanmış şekli.



Şekil 4. Farklı blokların paralel bir bilgisayarda oluşturacağı matris yapıları.

## 6. BLOK DENGELEME PROGRAMLARI PSATB VE PSATIM

Yazar tarafından ICL distributed Array Processor (ICL DAP) için geliştirilen paralel algoritma ve programlar blok dengeleme probleminin paralel çözümü amacıyla yapılan çalışmaların ilkidir. Geliştirilen programlardan ışın demetleri yöntemine göre olanı PSAT (Parallel Solution of Aerial Triangulation by Bundles) ve bağımsız modeller yöntemine göre olanı da PSATIM (Parallel Solution of Aerial Triangulation by Independent Models) olarak adlandırılmışlardır.<sup>11,12</sup>

Çalışmaların yapıldığı ICL DAP bilgisayarı 64x64 elemanlık bir matris yapısı oluşturan toplam 4096 PE'den ibarettir. Her bir PE için 16 K bit'lik bir kapasite (hafıza) ayrılmıştır. PE'lerin toplam kapasitesi 8 M Byte'lık bir hafıza oluşturmaktadır. Sistem bir bütün olarak bir işlem matrisi ve her bir elemanı 1 bit kapasiteli 64x64'lük 16384 adet hafıza matrisinden oluşan bir yapı teşkil etmektedir. Şekil 2'de ICL DAP bilgisayarının genel matris yapısı görülmektedir. Kontrol birimince yayınlanan bir komut  $64 \times 64 = 4096$  PE tarafından aynı anda işlem görmektedir.

Daha önce belirtildiği gibi, düzenli veri yapıları oluşturabilmek için ışın demetleri ve bağımsız modeller yöntemlerinde sırasıyla 6x3 ve 4x4 PE'lik alt-matris yapıları oluşturulmakta idi. Buna göre bir DAP matrisinin ışın demetleri ve bağımsız modeller yöntemlerinde kabul edebileceği bloklar sırasıyla en fazla 210 resim (10 kolon x 21 resim) ve 256 model (16 kolon x 16 model)'den oluşmak zorundadır. Bunlardan daha büyük bloklar parçalar halinde dengelenmek zorundadır. Blokların küçük olması halinde ise, daha önce de belirtildiği gibi, birden fazla sayıda blok'un birlikte dengelenmesi de mümkün olabilmektedir. PSATB ve PSATIM programlarınca kabul edilebilecek blok türleri şu şekilde sıralanabilir.

1. Maksimum 10 kolon x 21 resim veya 16 kolon x 16 model'den ibaret tek bir blok.
2. Her biri maksimum 5 kolon x 21 resim veya 8 kolon x 16 model'den ibaret 2 bağımsız blok.
3. Her biri sırasıyla 5 kolon x 21 resim, 5 kolon x 11 resim ve 5 kolon x 10 resim veya 6 kolon x 16 model, 5 kolon x 16 model ve 5 kolon x 16 model'den ibaret 3 bağımsız blok.
4. Her biri sırasıyla maksimum 5 kolon x 11 resim, 5 kolon x 10 resim, 5 kolon x 11 resim ve 5 kolon x 10 resim veya 4 kolon x 16 model'den ibaret 4 bağımsız blok.

Konvensiyonel programlama dillerinde iki veya daha fazla sayıdaki resim veya modeldeki ortak noktalar nokta tanım numaralarının kıyaslanması sonunda belirlenebilmektedir. Kıyaslanma işlemi her bir resim/model noktası için ayrı ayrı tekrarlanmak zorundadır. Geliştirilen paralel algoritmada ise, resimlerin veya modellerin tamamı bindirilmeli (ortak) olabilecekleri resim veya modellerle aynı anda kıyaslanarak hangi noktaların ortak olduğu belirlenmekte ve gerekli işlemler yapılmaktadır. Bu nedenle, bir resim veya model noktasına ait tanım numarası ve diğer veri değerleri okutulurken bu verilerin hangi PE'nin hafızasında saklanacağına önceden kesin olarak belirlenmesi gerekir. Şekil 5'te 2 kolon ve her bir kolonu 3 resimden ibaret bir blok için nokta tanım numaralarının alt-matrislerde temsili olarak saklanması görülmektedir. Aynı tanım numarasına sahip noktalar programların tasarımına uymayan PE'lerde saklanıldığında kıyaslanma sonunda bu noktaların farklı noktalar olarak işlem görme tehlikesi ortaya çıkar.

PSATB programı ATRIG1 adlı bir ana program, 15 adet alt-program ve fonksiyondan oluşmaktadır. Program verilerle yüklenildiğinde hafızada her biri 1 bit derinliğinde toplam 12116 matrislik (yaklaşık 6 M Byte) yer işgal etmektedir. Program çalıştırıldığında alt-programlar ve fonksiyonlar çağırılış sırasına göre işlem görmektedir.

PSATIM programı ise MODEL1 adlı bir ana program, 13 adet alt-program ve fonksiyondan oluşmaktadır. Program verilerle yüklenildiğinde hafızada her biri 1 bit derinliğinde toplam 7640 matrislik (yaklaşık 4 M Byte) yer işgal etmektedir.

PSATB ve PSATIM programlarıyla yapılan çalışmaların sonuçları (10)'da yayınlanmıştır. Elde edilen sonuçlar paralel programların konvensiyonel programlama dillerinde yazılmış olan programlardan defalarca daha hızlı sonuç verdiğini göstermektedir.

	1	2	3			
1	1	7	13	25	26	
	2	8	14	19	20	
	3	9	10	15	21	22
	4	11	12	16	23	28
	5			23	29	30
	6			17	18	24
2	31	36	37	42	43	
	32	38	39	44	45	
	33	40	41	46	47	
	34	41	45	48	49	
	35			49	50	
				51	52	
			53	54	55	
			55	56		
	4	5	6			

	1	2	3			
1	1	7	13	7	13	19
	3	9	15	9	15	20
	5	11	17	11	17	23
	2	8	14	8	14	21
	4	10	16	10	16	22
	6	12	18	12	18	24
2	31	36	17	36	17	46
	32	38	42	38	42	47
	34	40	44	40	44	49
	6	37	18	37	18	24
	33	39	43	39	43	48
	35	41	45	41	45	50
			43	48	54	
			45	50	56	
	4	5	6			

Şekil 5. Toplam 6 modelden ibaret bir blok için nokta tanım numaralarının alt-matrislerde temsili olarak saklanması.

## 7. SONUÇ

PSATB ve PSATIM blok dengeleme programları paralel bilgisayarlarda havai nirenge probleminin çözümü için geliştirilen ilk programlardır. Jeodezi ve Fotogrametri bilim dallarının uğraşı alanlarından olan sayısal arazi modelleri, kartoğrafik verilerin vektör veya matris formlarında işlenmesi, vb. problemlerin çözümü, bilindiği gibi, aynı türden işlemlerin defalarca tekrarlanması sonucu elde edilebilmektedir. Geliştirilen paralel algoritmaların ışığı altında bu ve benzeri konularda paralel çözüm ve yöntemler geliştirilmesi imkanı vardır.



## YARARLANILAN KAYNAKLAR

- (1) Flynn, J.E.: Veri high speed computing systems, Proc. IEEE, 54(12), s. 1901-1900 1966.
- (2) Kodge, P.M.: The architecture of pipelined computers. McGraw-Hill Book Co., New York, 1981.
- (3) Kuck, D.J.: A survey of parallel machine organisation and programming. ACM Computing Surveys, Mart, s. 29-60, 1977.
- (4) Hockney, R., Jesshope, C.: Parallel computers, programming and algorithms. Hilger Press, Bristol, 1981.
- (5) Love, H.H.: The highly parallel super computers, National Computer Conference of the American Federation of Information Processing Society, Arlington, ABD, 1980.
- (6) Parkinson, D.: The Distributed Array Processor - DAP. IUCC Bulletin, Volume 2, 1980.
- (7) Gostic, R.W.: Software and algorithms for distributed array processors. ICL Technical Journal, Mayıs. s. 116-135, 1979.
- (8) ICL: DAP FORTRAN Language. International Computers Ltd., Herts, İngiltere, 1980.
- (9) Ghosh, G.H.: Analytical Photogrammetry, Pergamon Press, New York, 1979.
- (10) Baz, İ.: The solution of aerial triangulation on vector and parallel array processors, Ph.D. tezi, Glasgow Üniversitesi, Ekim, 1984.
- (11) Baz, İ.: Block adjustment with array and vector processors. Photogrammetric Record, 11(65), s. 543, 1985.
- (12) Julia, J.E.: Developments with the COBLO program. Photogrammetric Record, 12(68), s. 219-226, Ekim 1986.