

FRANCISCO CARLOS BRAGANÇA DE SOUZA  
ENGENHEIRO CIVIL

ORIENTAÇÃO ABSOLUTA DE MODELOS  
UTILIZANDO FEIÇÕES DIGITALIZADAS COMO CONTROLE

Dissertação apresentada ao Curso de  
Pós-Graduação em Ciências Geodésicas  
para obtenção do Grau de Mestre em  
Ciências pela Universidade Federal  
do Paraná.

UNIVERSIDADE FEDERAL DO PARANÁ

1 9 8 2

ORIENTAÇÃO ABSOLUTA DE MODELOS UTILIZANDO  
FEIÇÕES DIGITALIZADAS COMO CONTROLE

DISSERTAÇÃO

Apresentada ao Curso de Pós-Graduação em Ciências  
Geodésicas para obtenção do Grau de Mestre em Ciên-  
cias pela Universidade Federal do Paraná.

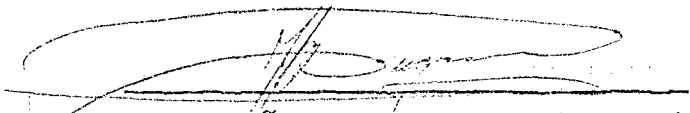
por

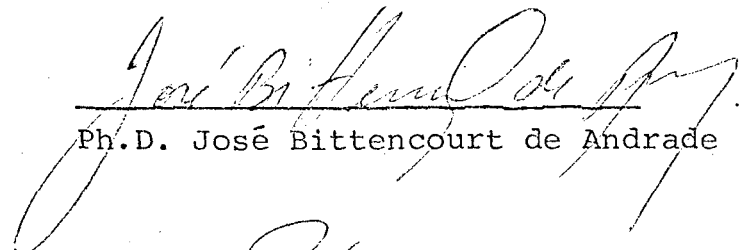
FRANCISCO CARLOS BRAGANÇA DE SOUZA, Engenheiro Civil

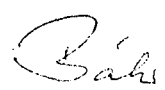
UNIVERSIDADE FEDERAL DO PARANÁ

1 9 8 2

BANCA EXAMINADORA:

  
Ph.D João Bosco Lugnani - Orientador

  
Ph.D. José Bittencourt de Andrade

  
Dr. Ing. Hans Peter Bähr

À minha mãe,  
Urânia Bragança de Souza,  
dedico este trabalho.

## AGRADECIMENTOS

O autor deseja externar seu profundo agradecimento:

- ao Professor João Bosco Lugnani pela orientação e incentivo;
- aos colegas, engenheiros Jorge Masson e Edson Mitishita, pela ajuda no desenvolvimento dos programas computacionais utilizados;
- ao CNPq, pela bolsa de estudos concedida;
- ao IPH-UFRGS, nas pessoas dos Professores Marc Pierre Bordas e Vitor Haertel; e
- aos Professores, colegas e amigos do Curso de Pós-Graduação em Ciências Geodésicas, que, de alguma forma, contribuíram durante a execução deste trabalho.

## S I N O P S E

Neste trabalho é tratada uma entre muitas das aplicações fotogramétricas da utilização de feições digitalizadas como controle. O estudo recaiu sobre o problema da orientação absoluta de modelos estereoscópicos para aerotriangulação. São apresentados e discutidos: novos modelos matemáticos para o cálculo dos parâmetros de orientação absoluta através de feições; comparação dos novos modelos matemáticos com os que utilizam controle convencional; comentários e sugestões para pesquisas futuras.

## S Y N O P S I S

This work tests one of the many photogrammetric applications using digital entities as control. The problem of absolute orientation of stereoscopic models in aerotriangulation was studied. New mathematical models are presented which compute absolute orientation parameters using digitized features. These new mathematical models are compared to conventional models using control points, and in addition, comments are made and suggestions are given for future research.

## CONTEÚDO

	Pág.
TÍTULO .....	i
DEDICATÓRIA .....	ii
AGRADECIMENTOS .....	iii
SINOPSE .....	iv
CONTEÚDO .....	v
LISTA DE FIGURAS .....	viii

## CAPÍTULO I

1.	INTRODUÇÃO .....	01
1.1.	Definição do Problema .....	01
1.2.	Tendências da Cartografia Moderna .....	02
1.3.	Objetivos da Pesquisa .....	02

## CAPÍTULO II

2.	REQUISITOS BÁSICOS .....	04
2.1.	Orientação Absoluta .....	04
2.2.	Conceito de Feição .....	08
2.3.	Orientação Absoluta Utilizando Feições .....	09

	pág.
2.4. Geração de Feições .....	10
2.4.1. Feições retas .....	11
2.4.2. Feições curvas .....	12
2.5. Feições Contínuas .....	15
2.5.1. Princípio básico e modelo matemático geral para interpolação utilizando splines paramétricos...	15
2.6. Qualidade de Interpolação .....	19
2.6.1. Erro de origem .....	19
2.6.2. Erro devido a Orientação da feição .....	22
2.6.3. Densidade e Posição do ponto .....	24

### CAPÍTULO III

3. MODELOS MATEMÁTICOS .....	27
3.1. Introdução .....	27
3.2. Feições Retas: Modelo Matemático .....	27
3.3. Feições Curvas: Utilização de Splines Paramé- tricos na Orientação Absoluta .....	44
3.4. Parâmetros Aproximados .....	47

### CAPÍTULO IV

4. TESTES .....	50
4.1. Introdução .....	50
4.2. Feições Curvas: Teste de Frequência .....	51
4.3. Feições Curvas: Número e Distribuição .....	53

	Pág.
4.4. Feições Curvas: Erro de Origem .....	55
4.5. Feições Retas : Número e Distribuição .....	56
4.6. Feições Curvas e Retas: Comparação com Pontos de Controle .....	58
4.7. Influência da Precisão dos Valores Aproximados dos Parâmetros .....	60

## CAPÍTULO V

5. CONCLUSÕES E RECOMENDAÇÕES .....	63
5.1. Conclusões .....	63
5.2. Recomendações .....	64

## APENDICE A

### SUGESTÕES PARA TRABALHOS FUTUROS

1. POSIÇÃO DAS FEIÇÕES NO MODELO .....	66
2. MATRIZ DE RODRIGUES .....	67

## APENDICE B

PROGRAMAS E SUB-ROTINAS .....	75
-------------------------------	----

REFERÊNCIAS BIBLIOGRÁFICAS .....	126
----------------------------------	-----

## LISTA DE FIGURAS

pág.

2.1	Esquema de orientação absoluta de um modelo....	06
2.2.1	Feição e sistema de eixos ao qual estão re- ferenciadas as coordenadas de seus pontos.....	08
2.2.2	Exemplo de feições encontradas na prática.....	09
2.4.2	Feições geradas inicialmente em 0 e de- pois transportadas para posições P1 e P2.....	13
2.5.1	Distância entre pontos lidos no modelo(di).....	16
2.5.2	Diagrama de feições tornadas contínuas mate- maticamente onde aparecem os pontos origi- nais (N1,N2,...Nn) e o ponto interpolado(Pi)...	17
2.6.1	Erro na determinação da origem da feição .....	20
2.6.2	Erro devido a orientação da feição em rela- ção a um dos eixos coordenados .....	22
2.6.3	Posições da Feição utilizadas no teste Nº 2 ....	23
2.6.4	Qualidade de interpolação. Desenho com dis- crepâncias exageradas .....	25
3.2.1	Modelo com pontos lidos sobre as feições retas (pfi) e feições retas definidas no campo (Fi) .....	30
3.3.1	Ponto (P) pertencente a feição A não digita- lizado .....	45
4.2.1	Localização das feições no modelo para o tes- te de frequência .....	51

4.3	Ilustração das feições utilizadas para testar número e distribuição no modelo .....	53
4.5	Distribuição de feições retas nos modelos testados .....	56
4.6	Tipos de controle utilizados nos testes comparativos .....	58
A1	Pesos controlados pela posição da feição .....	67

## CAPÍTULO I

### 1. INTRODUÇÃO

#### 1.1. Definição do Problema

Na solução dos problemas fotogramétricos o estabelecimento da correspondência entre pontos do espaço-imagem e do espaço-objeto, além de ser economicamente dispendioso, constitui-se também em fonte de erro. O planejamento do controle de campo, a marcação dos pontos, a sua enumeração, transferência e identificação, bem como a preservação de monumentos visando sua utilização em trabalhos posteriores, trazem consigo dificuldades bem conhecidas pelo fotogrametrista

Nesta pesquisa irá se estudar a utilização de um novo tipo de controle (feições digitalizadas) para complementar ou mesmo substituir totalmente o controle convencional necessário à orientação absoluta de modelos estereoscópicos para aerotriangulação. Estas novas entidades (feições), prescindindo da correspondência ponto a ponto, à qual se juntam as dificuldades já salientadas, teriam um caracter mais ou menos perene no que diz respeito à sua utilização em trabalhos futuros.

O estudo destas entidades, visando complementar ou substituir o controle convencional, parece oportuno, pois a tendência moderna para a criação de bancos de dados

|04|, assim como o lançamento no mercado de restituidores analíticos (Analytical Plotters) abre um horizonte amplo para a Fotogrametria. (|06|, |14|, |15|, |17|, |19|, |20|, |21|, |22|, |23|).

O trabalho concentrar-se-á na orientação absoluta analítica de modelos definidos no espaço tridimensional, salientando-se que as potencialidades do emprego de feições digitalizadas como controle são bem mais amplas |18|.

### 1.2. Tendências da Cartografia Moderna

O grande desenvolvimento dos computadores digitais tem tornado possível o armazenamento, processamento e reaproveitamento, de enormes conjuntos de informações, contribuindo assim para o crescimento da cartografia digitalizada. A criação de bancos de dados vem facilitar a utilização destas informações, pois a sistemática de consulta e a garantia de integridade, bem como a versatilidade operacional, tornam atraente sua implantação.

Sensível a isto, a comunidade cartográfica trata de adaptar-se às novas técnicas, dando já um passo importante com seus restituidores analíticos. O emprego da cartografia digitalizada e a organização de bancos de dados representam uma nova linha na pesquisa cartográfica.

### 1.3. Objetivos da Pesquisa

Implementar em programas computacionais o modelo matemático de orientação absoluta analítica, utilizando

feições como controle.

Testar a eficiência e aplicabilidade teórica do modelo matemático e dos programas, utilizando dados fictícios.

Realizar experiências e comparar os resultados obtidos pelo novo conceito com os obtidos pelo procedimento clássico.

Recomendar, se a pesquisa assim o permitir, o uso de feições digitalizadas como controle para a orientação absoluta de modelos na aerotriangulação.

## CAPÍTULO II

## 2. REQUISITOS BÁSICOS

2.1. Orientação Absoluta

Depois de completada a orientação relativa de um par de fotografias, um modelo tridimensional do terreno existe. A adequação da escala, o nivelamento e a orientação deste modelo, em relação ao sistema de terreno utilizado, são os objetivos da orientação absoluta.

O procedimento matemático utilizado para adequar a escala e nivelar o modelo, sem variar sua forma, é a **Transformação de Similaridade** (2.1.1). Nesta transformação o fator de escala ( $\lambda$ ), as rotações ( $K, \phi, w$ ) em torno dos eixos coordenados e as translações ( $x_0, y_0, z_0$ ), perfazendo sete parâmetros de orientação, são solucionadas analiticamente através do sistema de equações que passaremos a expor:

## TRANSFORMAÇÃO DE SIMILARIDADE

$$\bar{X} = \lambda M \bar{x} + \bar{X}_0 \quad (2.1.1)$$

onde:

$\bar{X}$  ; vetor das coordenadas no sistema do terreno;

$\bar{x}$  ; vetor das coordenadas de modelo;

M ; matriz das rotações;

$\lambda$  ; fator de escala

$\vec{X}_0$ ; vetor das translações

A matriz das rotações  $M$  é expressa pelo produto de três matrizes ortogonais que representam cada uma as rotações em torno dos três eixos cartesianos (X, Y, Z) (fig. 2.1).

$$M = R(\kappa) R(\phi) R(\omega) \quad (2.1.2)$$

sendo:

$$R(\kappa) = \begin{bmatrix} \cos\kappa & \text{sen}\kappa & 0 \\ -\text{sen}\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(\phi) = \begin{bmatrix} \cos\phi & 0 & -\text{sen}\phi \\ 0 & 1 & 0 \\ \text{sen}\phi & 0 & \cos\phi \end{bmatrix} \quad (2.1.3)$$

$$R(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \text{sen}\omega \\ 0 & -\text{sen}\omega & \cos\omega \end{bmatrix}$$

Ficamos então com:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (2.1.4)$$

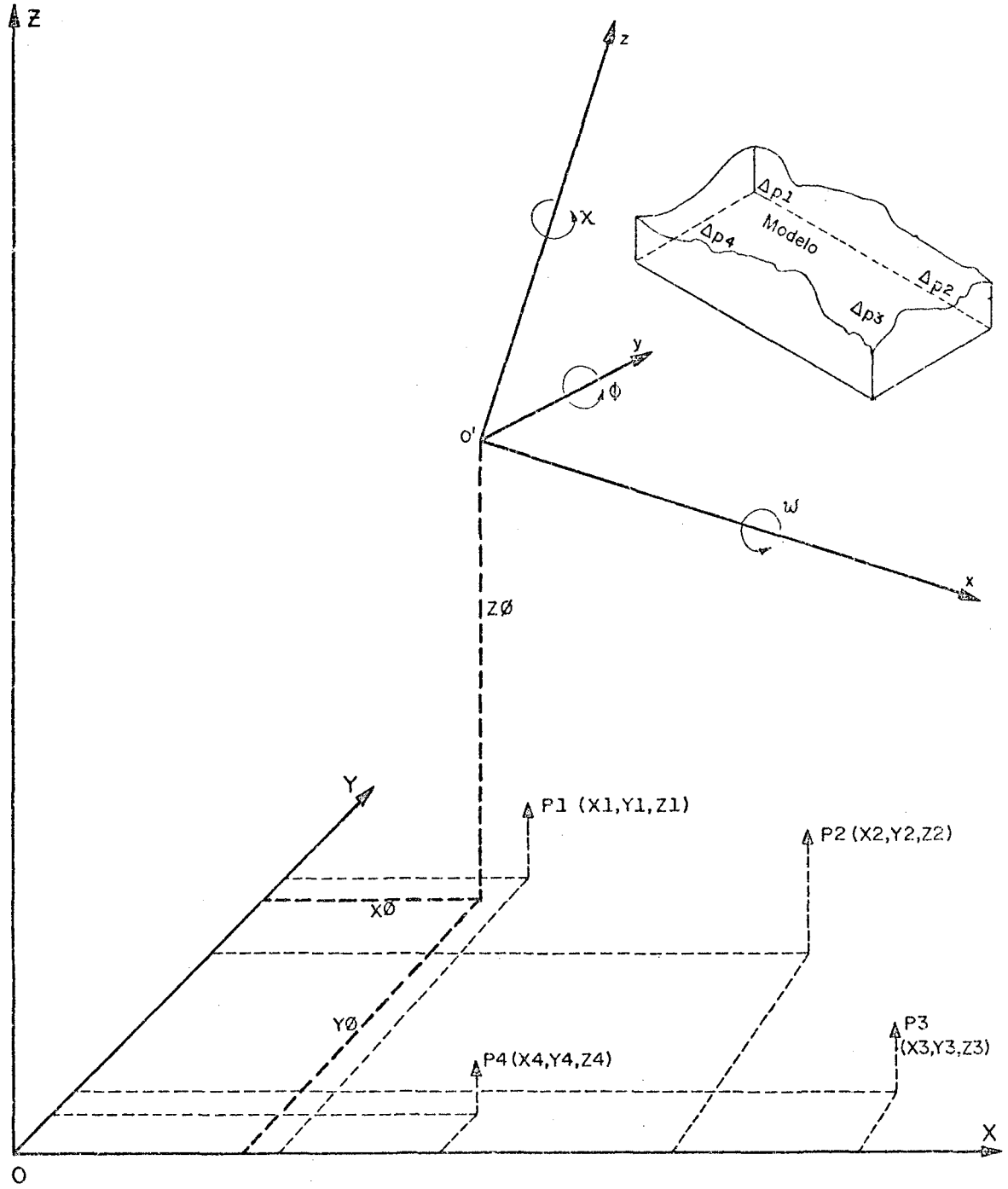


Figura 2.1 Esquema de orientação absoluta de um modelo.

onde:

$$m_{11} = \cos\phi \cos\kappa$$

$$m_{12} = \cos\omega \operatorname{sen}\kappa + \operatorname{sen}\omega \operatorname{sen}\phi \cos\kappa$$

$$m_{13} = \operatorname{sen}\omega \operatorname{sen}\kappa - \cos\omega \operatorname{sen}\phi \cos\kappa$$

$$m_{21} = -\cos\phi \operatorname{sen}\kappa$$

$$m_{22} = \cos\omega \cos\kappa - \operatorname{sen}\omega \operatorname{sen}\phi \operatorname{sen}\kappa$$

$$m_{23} = \operatorname{sen}\omega \cos\kappa + \cos\omega \operatorname{sen}\phi \operatorname{sen}\kappa$$

$$m_{31} = \operatorname{sen}\phi$$

$$m_{32} = \operatorname{sen}\omega \cos\phi$$

$$m_{33} = \cos\phi \cos\omega$$

Escrevendo a transformação de similaridade em forma matricial desenvolvida teremos:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X0 \\ Y0 \\ Z0 \end{bmatrix} \quad (2.1.5)$$

Executando as multiplicações vem:

$$X = \lambda (m_{11}x + m_{12}y + m_{13}z) + X0$$

$$Y = \lambda (m_{21}x + m_{22}y + m_{23}z) + Y0$$

$$Z = \lambda (m_{31}x + m_{32}y + m_{33}z) + Z0$$

(2.1.6)

O modelo matemático apresentado em (2.1.6) requer a solução de sete parâmetros (incógnitos), e como cada ponto proporciona três equações, necessitamos de, no mínimo, dois pontos com coordenadas planialtimétricas ( $x, y, z$ ) e um terceiro ponto, não alinhado com os dois anteriores, de coordenada altimétrica ( $z$ ) conhecida, para resolver o sistema.

## 2.2. Conceito de Feição

O termo feição será usado para denominar qualquer objeto ou parte deste que possa ser representado por um único segmento de curva especial (fig. 2.2.1). Uma rodovia, ferrovia, a aresta de um prédio, são exemplos de feições que podem ser encontradas na prática (fig. 2.2.2).

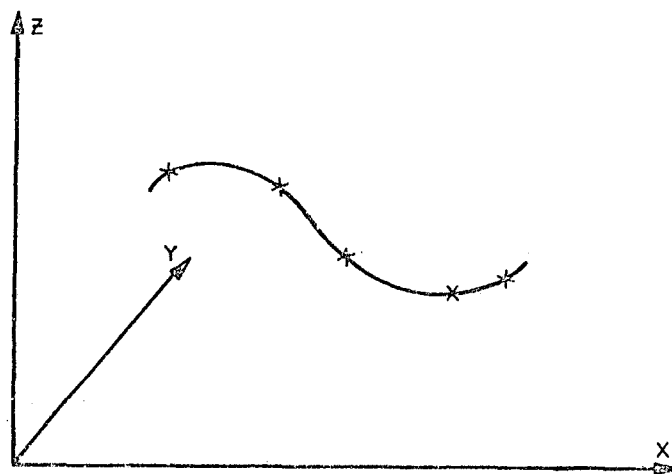


FIGURA 2.2.1 Feição e sistema de eixos ao qual estão referenciadas as coordenadas de seus pontos.

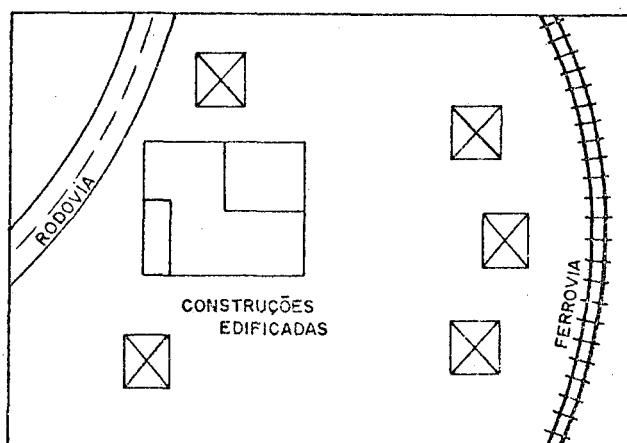


FIGURA 2.2.2. Exemplo de feições encontradas na prática

Quando armazenarmos em um arquivo de dados as coordenadas dos pontos que definem uma feição estaremos em presença de uma feição agora digitalizada, que será a matéria prima desta pesquisa.

### 2.3. Orientação Absoluta Utilizando Feições

Após a orientação relativa de um estereopar obtemos um modelo tridimensional da região imageada. Admitindo que nesta região apareçam feições, cujas coordenadas dos pontos que as definem estão armazenadas em um arquivo (parte de um banco de dados), podemos executar a orientação absoluta do modelo utilizando agora a correspondência entre feições ao invés de pontos isolados. A orientação visual aproximada usando feições que é feita quando se utiliza um sketchmaster, exemplifica esta idéia.

O modelo matemático fica então da forma seguinte:

$$F = T \cdot f \quad (2.3.1)$$

onde:

T ; Transformação de similaridade adaptada para o uso de feições.

f ; Coordenadas da feição no modelo.

F ; Coordenadas digitalizadas da feição (terreno).

Uma apresentação mais detalhada do modelo matemático anterior será discutida em capítulo posterior, pois, dependendo do tipo de feição, algumas mudanças serão introduzidas no mesmo.

Enquanto uma entidade de campo (F) é representada por n pontos, somente alguns pontos, não correspondentes a estes, precisam ser observados na feição (f) para utilizá-la como controle.

#### 2.4. Geração de Feições

A inexistência de bancos de dados, ou mesmo pequenos arquivos, com coordenadas de pontos definidores de feições, ensejou o uso de dados fictícios.

Conscientes da lacuna deixada pela falta de dados reais apresentaremos algumas vantagens advindas do uso de dados fictícios.

Quais sejam:

- o verdadeiro valor dos resultados pode ser calculado com a precisão desejada, consequentemente o verdadeiro valor do erro pode ser detectado;
- desvios normalmente distribuídos podem ser introduzidos nas observações através de sub-rotinas especiais;
- as feições podem ser geradas em qualquer posição e densidade no modelo;
- uma vez implementados os programas e sub-rotinas de geração, o tempo gasto para testes com situações distintas fica bastante reduzido.

#### 2.4.1. Feições Retas

Chamaremos de feições retas as entidades físicas que são representadas em um arquivo de dados através das coordenadas de seus pontos, as quais devem satisfazer a equação da reta (2.4.1).

$$\frac{X-X_1}{X_2-X_1} = \frac{Y-Y_1}{Y_2-Y_1} = \frac{Z-Z_1}{Z_2-Z_1}$$

(2.4.1)

Este tipo de feição é facilmente encontrado em regiões onde aparecem em número maior as obras feitas pelo homem (cidades, complexos industriais, grandes barragens, etc).

#### 2.4.2. Feições Curvas

Na prática nem sempre encontraremos retas em uma região fotografada, em número e disposição que satisfaçam os requisitos dos modelos matemáticos utilizados na Fotogrametria. A linha central de um trecho reto de uma estrada, à primeira vista pode parecer reta, mas a variação da altitude do terreno pode não ser linear o que fará com que os pontos que a representam não satisfaçam a equação (2.4.1). No sentido de generalizar as feições encontradas nos modelos fictícios foi desenvolvido um programa computacional que gera feições curvas. O programa gera a feição sempre na origem estabelecida para as coordenadas de terreno e analiticamente desloca o conjunto de pontos para local desejado. As feições foram geradas com vinte e cinco pontos.

O comprimento das feições em modelos na escala 1:10.000 foi limitado a 2 centímetros. O procedimento utilizado para a geração de feições curvas é representado (embora em duas dimensões) pela fig. 2.4.2 e o quadro I mostra a saída do programa gerador. (Apêndice B. GENER.FOR). A amplitude da curva gerada (A) e a frequência (F) tornam possível optar por uma feição bastante irregular ou plana conforme estes valores forem combinados.

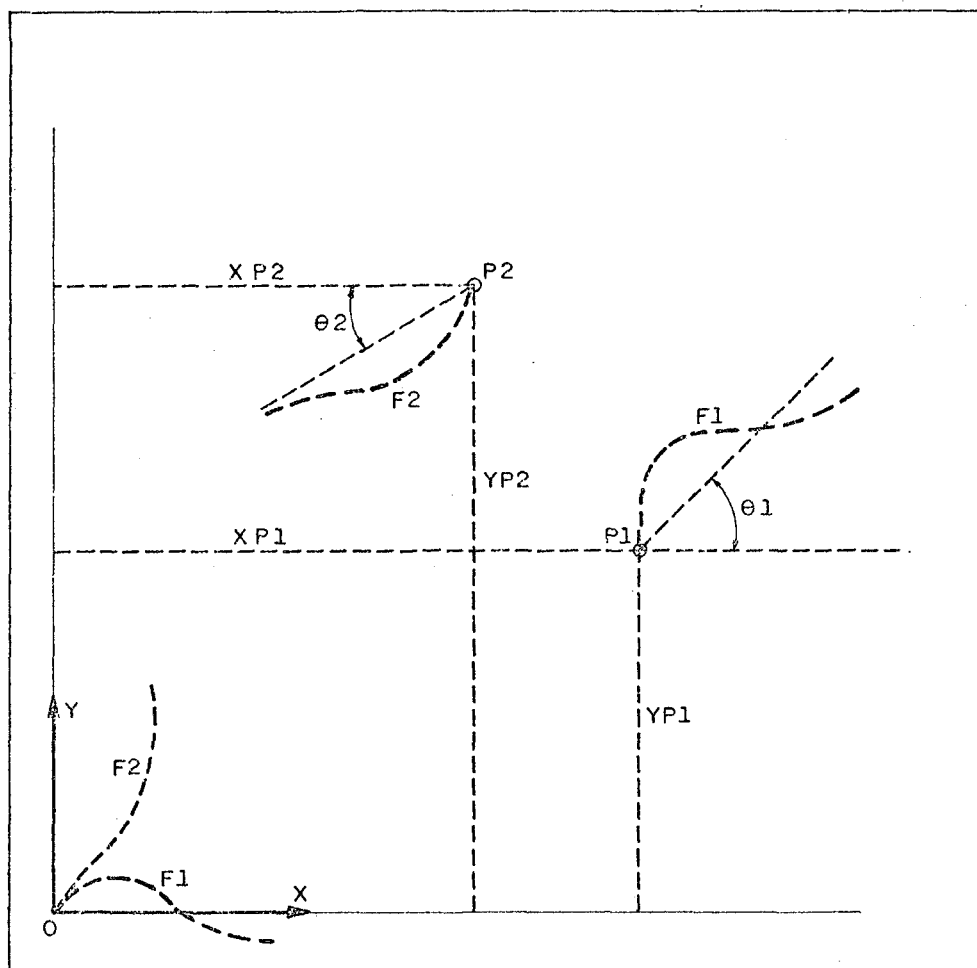


FIGURA 2.4.2 Feições geradas inicialmente em  $O$  e depois transportadas para posições  $P1$  e  $P2$

## QUADRO I

PONTO CONSIDERADO III			
VALOR DE XP: 18400.00			
VALOR DE YP: 9200.000			
VALOR DE ZP: 300.0000			
VALOR DE "A" PARA Y: 0.5000000			
VALOR DE "A" PARA Z: 0.3000000			
VALOR DE "F" PARA Y: 0.3000000			
VALOR DE "F" PARA Z: 0.2000000			
VALOR DE TETA: 1.570796			
ORDEM	XN	YN	ZN
1	18400.00	9200.00	300.00
2	18403.89	9191.67	301.56
3	18407.54	9183.33	303.09
4	18410.73	9175.00	304.52
5	18413.28	9166.67	305.84
6	18415.02	9158.33	307.00
7	18415.85	9150.00	307.97
8	18415.72	9141.67	308.72
9	18414.64	9133.33	309.24
10	18412.67	9125.00	309.51
11	18409.94	9116.67	309.52
12	18406.60	9108.33	309.28
13	18402.86	9100.00	308.78
14	18398.95	9091.67	308.05
15	18395.10	9083.33	307.10
16	18391.55	9075.00	305.96
17	18388.51	9066.67	304.66
18	18386.17	9058.33	303.23
19	18384.67	9050.00	301.72
20	18384.09	9041.67	300.15
21	18384.48	9033.33	298.59
22	18385.80	9025.00	297.06
23	18387.99	9016.67	295.61
24	18390.91	9008.33	294.28
25	18394.37	9000.00	293.11

## 2.5. Feições Contínuas

Mesmo em se tratando de um banco de dados, uma determinada entidade física será representada pela digitalização das coordenadas de um número discreto de pontos que a compõem. Quando observamos esta entidade (feição) em um modelo, geralmente não identificamos tais pontos e sim a entidade em si. A digitalização não necessita nenhuma sinalização de pontos na feição. Digitaliza-se qualquer ponto desde que pertença à feição. A leitura de um ponto qualquer de uma feição em um modelo será útil então, se adaptarmos uma função que represente de forma contínua o conjunto discreto de pontos armazenados. Adaptada esta função, o ponto de terreno correspondente a um ponto lido sobre a feição será encontrado por interpolação.

Uma representação tridimensional de uma feição requer um modelo também tridimensional de interpolação. As funções Spline [01] mostram-se satisfatórias para interpolar sobre curvas espaciais [18].

### 2.5.1. Princípio básico e modelo matemático geral para interpolação utilizando splines paramétricos

O parâmetro necessário para interpolar um ponto utilizando splines paramétricos é a distância deste a uma origem pré-estabelecida (fig. 2.5.1). Transportando a idéia para o caso de um ponto observado em um modelo, inicialmente se estabelecerá uma origem para as observações na feição e

a seguir a cada ponto lido seria calculada a distância cumulativa ( $s$ ) deste à origem. Esta distância ( $s$ ), transformada para a escala de terreno (considera-se que os pontos digitalizados estão na escala do terreno), é o parâmetro de entrada da função spline que calcula por interpolação as coordenadas do ponto lido agora em escala natural (fig. 2.5.2).

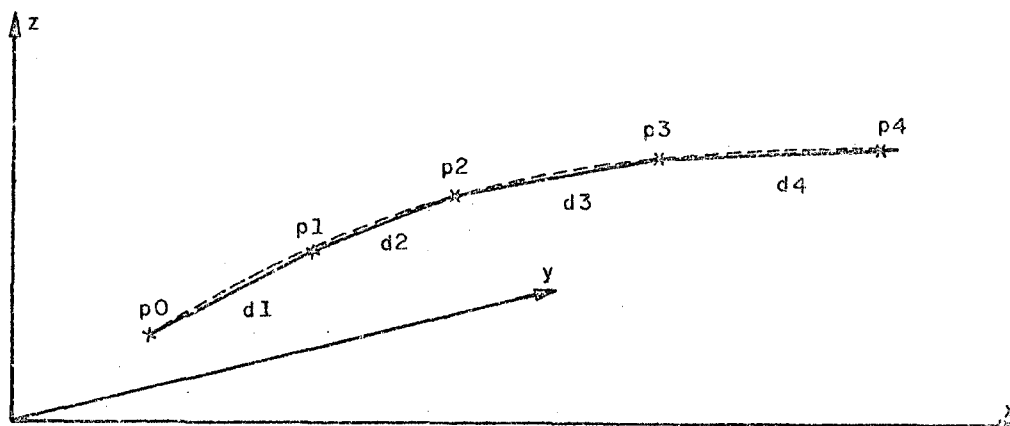


FIGURA 2.5.1 Distância entre pontos lidos no modelo ( $d_i$ ).

As distâncias cumulativas ( $s_i$ ), quando transformadas para a escala de terreno ( $S_i$ ), servem para interpolar as coordenadas ( $X, Y, Z$ ) (fig. 2.5.2).

$$d_i = \overline{p_i p_{i-1}}$$

$$s_i = \sum_{j=1}^i d_j \quad (2.5.1)$$

$$S_i = s_i \times \lambda \quad (\text{fator de escala})$$

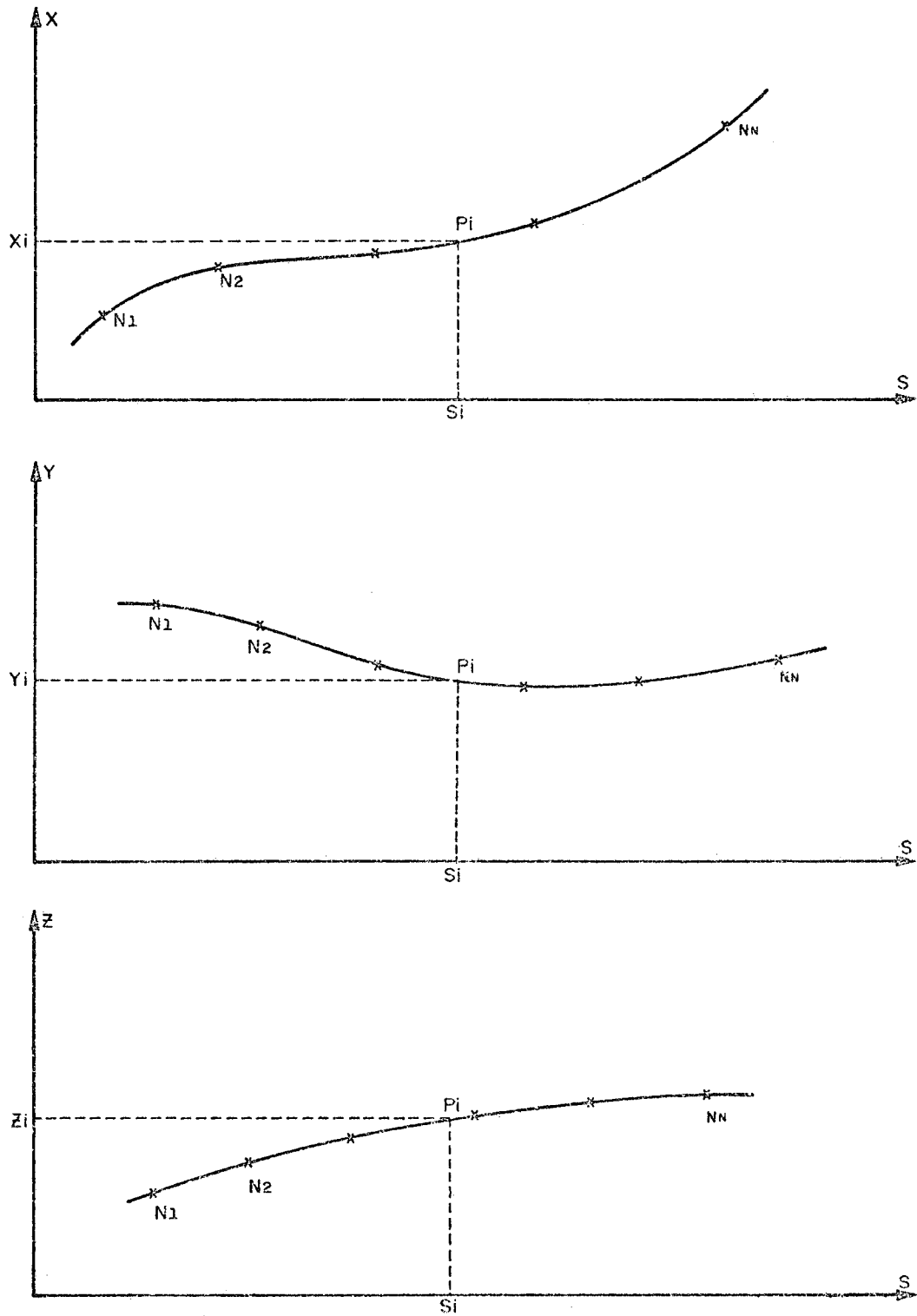


FIGURA 2.5.2 Diagrama de feições tornadas contínuas matematicamente onde aparecem os pontos originais ( $N_1, N_2, \dots, N_N$ ) e o ponto interpolado ( $P_i$ ).

O modelo matemático geral é expresso em função de S:

$$\begin{aligned} X &= f(S) \\ Y &= f(S) \\ Z &= f(S) \end{aligned} \quad (2.5.2)$$

ou ainda:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^{(T)} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^{(S)} \quad (2.5.3)$$

onde:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^{(S)} ; \text{ coordenadas calculadas através do modelo matemático de interpolação,}$$

e

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^{(T)} ; \text{ coordenadas calculadas através dos parâmetros da transformação de similaridade.}$$

As coordenadas de um ponto situado a uma distância (S) de uma origem pré-estabelecida calculadas pelo modelo matemático de interpolação são iguais àquelas calculadas através dos parâmetros da transformação (T) de similaridade.

## 2.6. Qualidade de Interpolação

Algumas dificuldades encontradas na aplicação do modelo matemático de interpolação serão agora estudadas e testadas, com o intuito de proporcionar uma visão crítica das limitações e potencialidade do uso de splines para a presente aplicação.

Atenção especial durante os testes foi dada a:

- precisão com que se estabelece a origem a partir da qual se iniciará a contagem das distâncias;
- posição dos pontos que definem a feição; e
- posição do ponto na feição, quando comparadas as discrepâncias de interpolação para os pontos de uma mesma feição.

Evidentemente os fatores que afetam a qualidade da interpolação interagem simultaneamente; entretanto, para facilidade de teste, quando cada fator foi testado os demais foram tomados como ideais.

Para os testes foi utilizado o programa TE3. FOR (Apêndice B).

### 2.6.1. Erro de origem

O modelo matemático de interpolação, quando afetado de erro grosseiro no estabelecimento da origem de contagem das distâncias cumulativas ( $E_o$ , fig. 2.6.1), não deteriora substancialmente a qualidade de interpolação.

Tendo em mente a precisão requerida nos pon-

tos de controle para diversas escalas de modelos [05] e a precisão dos instrumentos fotogramétricas atuais, erros de 100  $\mu\text{m}$ , 200  $\mu\text{m}$  e 300  $\mu\text{m}$ , na determinação da origem em modelos na escala 1:10.000, podem ser considerados grosseiros.

O teste nº 1 apresenta resultados para três valores de erro na determinação da origem em um modelo na escala 1:10.000. Para maior clareza os resultados são apresentados em escala do terreno.

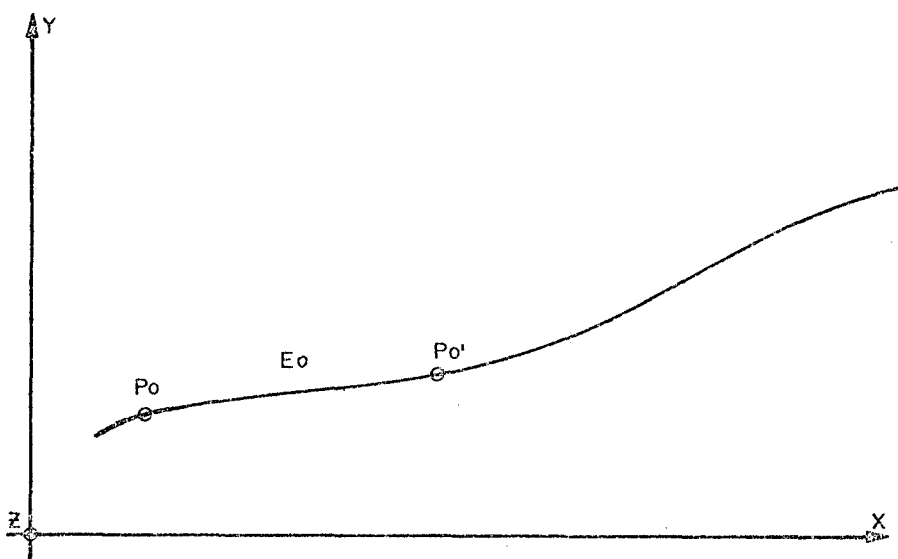


FIGURA 2.6.1 Erro na determinação da origem da feição.

T E S T E N º 1

DISCREPÂNCIAS DE INTERPOLAÇÃO

E R R O D E O R I G E M								
1 m			2 m			3 m		
X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)
0,57	0,83	0,19	1,14	1,64	0,36	1,70	2,45	0,54
0,60	0,81	0,14	1,21	1,60	0,27	1,82	2,39	0,40
0,67	0,76	0,05	1,34	1,50	0,11	2,01	2,24	0,16
0,73	0,68	-0,03	1,45	1,36	-0,07	2,19	2,03	-0,11
0,78	0,61	-0,12	1,56	1,22	-0,24	2,33	2,83	-0,36
0,81	0,58	-0,17	1,61	1,14	-0,34	2,41	1,70	-0,51
0,80	0,54	-0,19	1,62	1,10	-0,37	2,43	1,66	-0,55
0,83	0,61	-0,15	1,63	1,20	-0,30	2,43	1,79	-0,45
0,70	0,54	-0,11	1,47	1,19	-0,19	2,23	1,84	-0,27
0,97	1,08	0,07	1,67	1,79	0,08	2,37	2,50	-0,09

### 2.6.2. Erro devido a orientação da feição

Vamos imaginar inicialmente uma determinada feição (A, fig. 2.6.2) que tem as coordenadas Y de seus pontos crescendo quase que linearmente com a distância cumulativa S entre estes. A interpolação das coordenadas Y da feição A, em função de S, será mais sensível a erros em S ( $E_s$ , fig. 2.6.2) do que se considerarmos agora uma feição que se desenvolva aproximadamente perpendicular ao eixo Y (B, fig. 2.6.2).

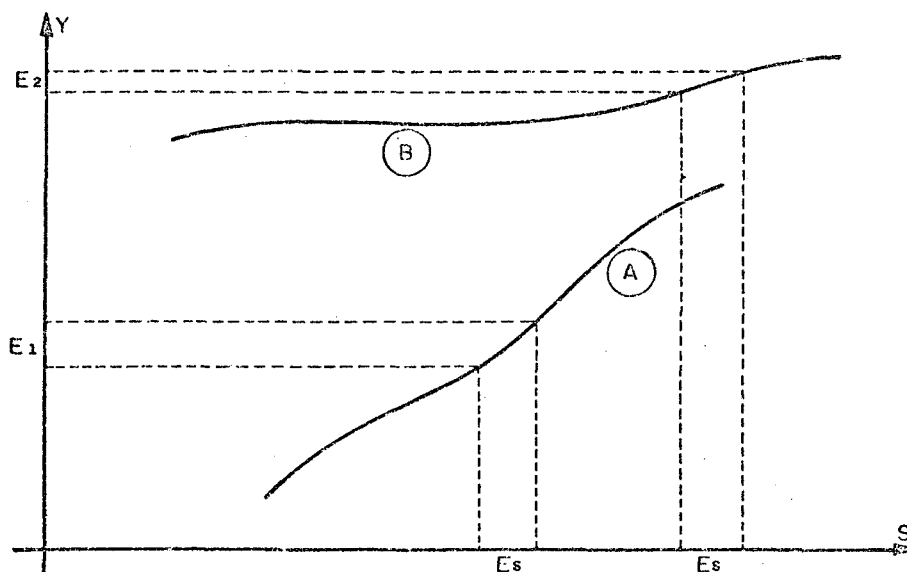


FIGURA 2.6.2 Erro devido a orientação da feição em relação a um dos eixos coordenados.

Observando a fig. 2.6.2 podemos constatar:

$E_s \approx E_1$  na feição A e

$E_2 \ll E_s$  na feição B.

O teste nº 2 utiliza feições orientadas como ilustra a fig. 2.6.3. Os resultados mostraram que as menores discrepâncias de interpolação apareceram para as coordenadas cujo eixo era aproximadamente perpendicular à feição testada (feições A e B fig. 2.6.3). Em contrapartida, enquanto as coordenadas X da feição B (fig. 2.6.3) foram as mais precisas, o mesmo não ocorreu com suas coordenadas Y, pois a feição estava posicionada quase que paralelamente ao eixo Y. Para a feição A (fig. 2.6.3) ocorreu exatamente o contrário, ou seja, coordenadas precisas em relação ao eixo perpendicular (Y) e coordenadas com discrepâncias grandes em relação ao eixo paralelo (X). A posição da feição C (fig. 2.6.3) apresentou resultados intermediários.

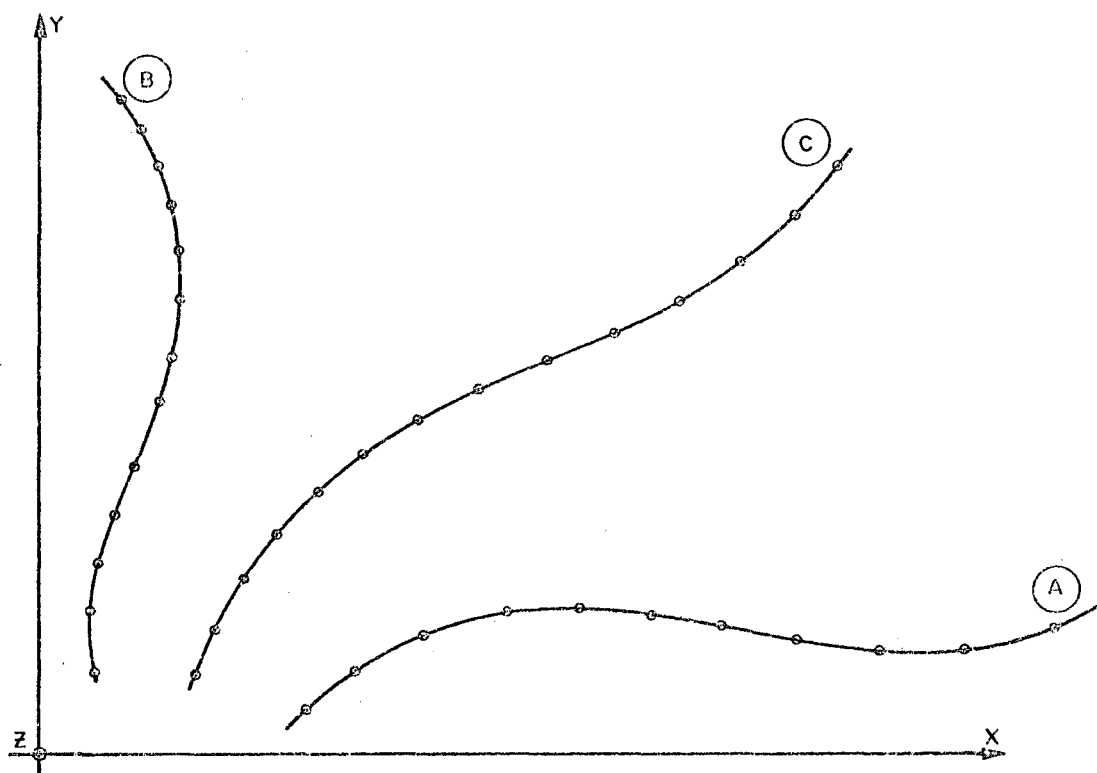


FIGURA 2.6.3 Posições da feição utilizadas no teste Nº 2

A ausência de variação nas coordenadas Z no teste de posição se justifica pela pequena variação no valor desta coordenada dentro da feição, quando comparada com as mudanças ocorridas nas coordenadas X e Y.

### TESTE Nº 2

#### DISCREPÂNCIAS DE INTERPOLAÇÃO

X (m)			Y (m)			Z (m)		
POSIÇÃO			POSIÇÃO			POSIÇÃO		
A	B	C	A	B	C	A	B	C
0,03	-0,01	0,02	0,01	0,03	0,02	0,01	0,01	0,01
-0,12	0,02	-0,06	-0,02	-0,12	-0,09	-0,02	-0,02	-0,02
0,44	-0,07	0,25	0,07	0,44	0,35	0,07	0,07	0,07
-1,65	0,28	-0,97	-0,28	-1,65	-1,36	-0,28	-0,28	-0,28

#### 2.6.3. Densidade e posição do ponto

Dependendo da sinuosidade da feição, um número maior ou menor de pontos é necessário para bem defini-la. Podemos então testar a influência da densidade na qualidade de interpolação se considerarmos o número fixo de pontos que definem a feição e variarmos a sinuosidade da curva que a caracteriza. Raciocinando desta maneira cabe aqui salientar que aumentando-se a frequência (F de geração) de uma curva definida por n pontos, estaremos deteriorando a qualidade de interpolação de uma forma semelhante à deterioração causada por uma diminuição do número de pontos definidores da curva.

O teste nº 3 apresenta as discrepâncias de interpolação para três valores de frequência, tendo os resultados confirmado a idéia anterior.

Quanto à posição do ponto na feição, observa-se que as discrepâncias aumentam para pontos situados nos extremos da curva (fig. 2.6.4), chegando mesmo a valores grosseiros (ponto 12, Teste nº 3). Isto se deve ao fato de que nas regiões extremas da feição estaremos em presença da necessidade de extrapolação, procedimento para o qual as funções spline não são recomendadas [01].

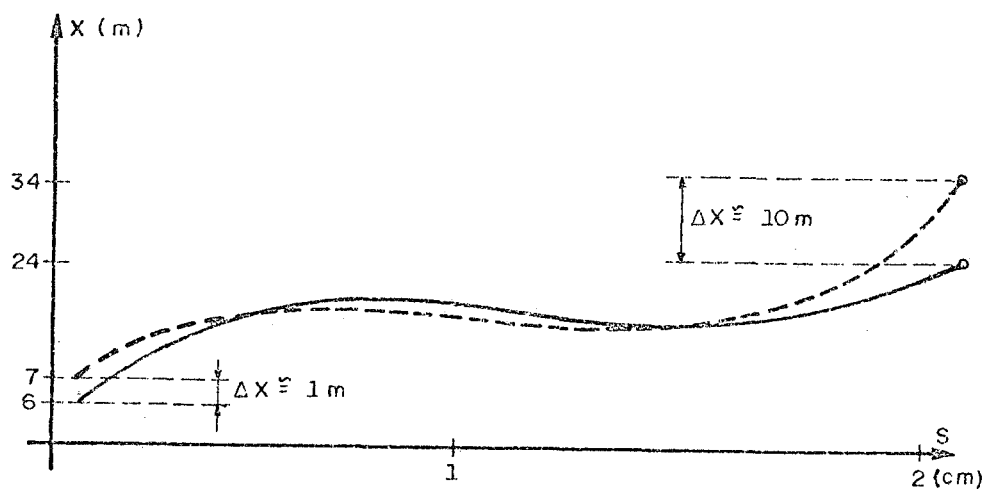


FIGURA 2.6.4 Qualidade de interpolação. Desenho com discrepâncias exageradas.

T E S T E Nº 3  
DISCREPÂNCIAS DE INTERPOLAÇÃO

P O N T O	X (m)			Y (m)			Z (m)		
	FREQUÊNCIAS			FREQUÊNCIAS			FREQUÊNCIAS		
	0,2	0,5	0,8	0,2	0,5	0,8	0,2	0,5	0,8
1	-0,01	0,01	0,56	0,00	0,16	0,70	0,01	0,11	0,09
2	0,01	0,04	-0,63	0,00	-0,10	0,05	0,01	-0,09	0,48
3	0,00	-0,03	0,96	0,00	-0,03	-0,14	0,00	0,04	-0,77
4	0,00	-0,01	-0,83	0,01	-0,01	0,11	0,00	0,00	0,52
5	0,00	0,08	-0,03	0,01	0,00	0,15	0,00	-0,05	0,14
6	0,00	-0,09	0,40	0,00	0,01	-0,53	0,01	0,08	-0,66
7	0,01	0,05	-0,53	0,01	-0,01	0,44	0,00	-0,03	0,68
8	0,03	0,01	0,84	0,03	-0,01	-0,14	-0,01	-0,00	-0,48
9	-0,10	-0,07	-0,35	-0,8	0,03	-0,43	0,02	0,07	-0,05
10	0,36	0,36	0,30	0,77	0,04	0,77	-0,05	-0,22	0,33
11	-0,36	-1,64	-1,28	-1,05	-0,57	-1,10	0,23	0,75	0,12
12	6,72	8,57	10,12	5,06	3,22	1,66	-1,17	-3,79	-5,99

## CAPÍTULO III

### 3. MODELOS MATEMÁTICOS

#### 3.1. Introdução

Na seção 2.1 desenvolvemos o modelo matemático de um dos métodos de orientação absoluta convencional. Como já dissemos ser nossa intenção escapar da necessidade do estabelecimento da correspondência ponto a ponto, é preciso desenvolver e testar novos modelos matemáticos que utilizem agora a relação feição a feição. Apresentaremos na seção 3.2 o modelo matemático para o caso de as feições serem retas, e na seção 3.3, quando abordarmos as feições curvas, haverá uma modificação. Enquanto pelos métodos convencionais os pontos de controle tinham suas coordenadas do campo conhecidas, agora isto não ocorrerá necessariamente. Temos a feição digitalizada; então um ponto (não necessariamente digitalizado) lido sobre ela em um modelo estereoscópico, terá suas coordenadas de campo calculadas por interpolação utilizando splines paramétricos.

#### 3.2. Feições Retas: Modelo Matemático

Sempre que uma entidade física (feição) for definida por dois de seus pontos em um arquivo de dados, será considerada feição reta. (As feições curvas terão sempre mais de dois pontos armazenados).

Para cada feição definida por dois pontos  $p_0$  demos aplicar as equações paramétricas da reta e calcular as coordenadas de um ponto genérico que satisfaçam as equações 2.4.1 e 2.5.2.

Estas equações são:

$$X = X_1 + t(X_2 - X_1) \quad (3.2.1)$$

$$Y = Y_1 + t(Y_2 - Y_1) \quad (3.2.2)$$

$$Z = Z_1 + t(Z_2 - Z_1) \quad (3.2.3)$$

onde

$X_1, Y_1, Z_1, X_2, Y_2, Z_2$ ; são coordenadas dos pontos que definem a feição reta.

$X, Y, Z$ , coordenadas do ponto genérico

$t$ , parâmetro da reta.

As equações (3.2.1), (3.2.2) e (3.2.3) permitem que se trabalhe agora com a reta em si (feição) e não mais com as coordenadas do ponto genérico  $(X, Y, Z)$ .

O modelo matemático de orientação absoluta (eq. 2.6) fica então:

$$\begin{array}{l}
 \lambda |m_{11}x + m_{12}y + m_{13}z| + X_0 = X_1 + t(X_2 - X_1) \\
 \lambda |m_{21}x + m_{22}y + m_{23}z| + Y_0 = Y_1 + t(Y_2 - Y_1) \\
 \lambda |m_{31}x + m_{32}y + m_{33}z| + Z_0 = Z_1 + t(Z_2 - Z_1)
 \end{array} \quad (3.2.4)$$

onde:

$\lambda$ ; fator de escala

$m_{ij}$ ; elementos da matriz das rotações

$X_0, Y_0, Z_0$ ; parâmetros de translação

$X_1, Y_1, Z_1, X_2, Y_2, Z_2$ ; pontos definidores da feição (terreno)

$t$ ; parâmetro da reta.

$x, y, z$ ; coordenadas do ponto genérico lidas sobre a feição (observações) no modelo.

Não é estabelecida a correspondência ponto a ponto (fig. 3.2.1), pois as coordenadas  $(x, y, z)$  são lidas no modelo estereoscópico em qualquer posição da feição.

As equações (3.2.4) escritas de forma extensa (3.2.5) servirão para o desenvolvimento do modelo matemático de ajustamento (3.2.6).

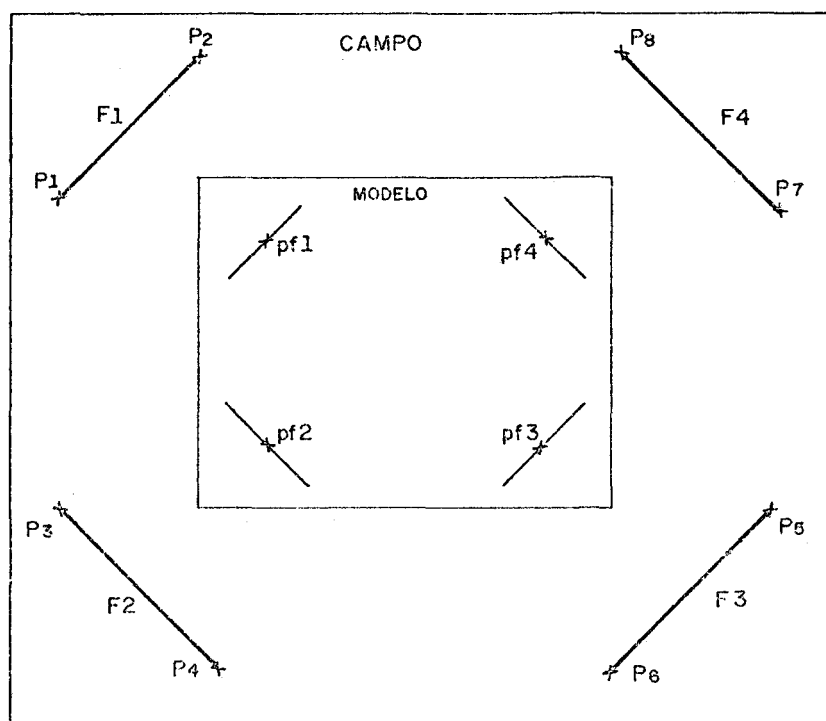


FIGURA 3.2.1 Modelo com pontos lidos sobre as feições retas (pfi) e feições retas definidas no campo (Fi).

$$FX = \lambda \left[ x(\cos\phi \cos\kappa) + y(\cos\omega \operatorname{sen}\kappa + \operatorname{sen}\omega \operatorname{sen}\phi \cos\kappa) + z(\operatorname{sen}\omega \operatorname{sen}\kappa - \cos\omega \operatorname{sen}\phi \cos\kappa) \right] + X_0 - X_1 - t(X_2 - X_1) = 0$$

$$FY = \lambda \left[ x(-\cos\phi \operatorname{sen}\kappa) + y(\cos\omega \cos\kappa - \operatorname{sen}\omega \operatorname{sen}\phi \operatorname{sen}\kappa) + z(\operatorname{sen}\omega \cos\kappa + \cos\omega \operatorname{sen}\phi \operatorname{sen}\kappa) \right] + Y_0 - Y_1 - t(Y_2 - Y_1) = 0$$

$$FZ = \lambda \left[ x(\operatorname{sen}\phi) + y(-\operatorname{sen}\omega \cos\phi) + z(\cos\phi \cos\omega) \right] + Z_0 - Z_1 - t(Z_2 - Z_1) = 0$$

(3.2.5)

$$F(X_a, L_a) = 0 \quad (3.2.6)$$

Na equação acima aparecem simultaneamente parâmetros desconhecidos e quantidades observadas, fato este que caracteriza o método combinado de ajustamento. (Equações de condição com incógnitas → MÉTODO GAUSS-HELMERT).

Para uma maior compreensão definiremos inicialmente os símbolos utilizados no desenvolvimento do método combinado de ajustamento.

Quais sejam:

$X$ ; vetor  $u \times 1$  das correções

$X_0$ ; vetor  $u \times 1$  dos valores aproximados

$X_a$ ; vetor  $u \times 1$  dos parâmetros ajustados

$L_b$ ; vetor  $n \times 1$  das quantidades observadas

$L_0$ ; vetor  $n \times 1$  das quantidades calculadas em função dos parâmetros aproximados ( $L_0 = F(X_0)$ )

$L_a$ ; vetor  $n \times 1$  das quantidades observadas ajustadas

$V$ ; vetor  $n \times 1$  dos resíduos ( $V = L_b - L_a$ )

$u$ ; nº incógnitas (parâmetros)

$r$ ; nº de equações

$n$ ; nº de observações

Linearizando (3.2.6) pela série de Taylor vem:

$$\frac{\partial F}{\partial La_i} \Big|_{V_i} + \frac{\partial F}{\partial X_{ai}} \Big|_{X_i} + F(La_{i-1}, Xa_{i-1}) + \frac{\partial F}{\partial La_i} \Big|_{(Lb-La_{i-1})} = 0$$

$$\begin{array}{lll} La_{i-1} = Lb & Xa_{i-1} = X0 & Xa_{i-1} = X0 \\ Xa_{i-1} = X0 & La_{i-1} = Lb & La_{i-1} = Lb \end{array} \quad (3.2.7)$$

onde:

$$\begin{array}{l} \frac{\partial F}{\partial La_i} \Big|_{\substack{La_{i-1} = Lb \\ Xa_{i-1} = X0}} = B_{i^*} \end{array} \quad (3.2.8)$$

$$\begin{array}{l} \frac{\partial F}{\partial Xa_i} \Big|_{\substack{Xa_{i-1} = X0 \\ La_{i-1} = Lb}} = A_i \end{array} \quad (3.2.9)$$

$$\begin{array}{l} F(La_{i-1}, Xa_{i-1}) + \frac{\partial F}{\partial La_i} \Big|_{\substack{Xa_{i-1} = X0 \\ La_{i-1} = Lb}} (Lb-La_{i-1}) = W_i \end{array} \quad (3.2.10)$$

(\*) O índice  $i$  caracteriza o desenvolvimento iterativo do método combinado, onde:  $Xa_i = Xa_{i-1} + X_i$  e  $La_i = La_{i-1} + V_i$

ou ainda, sob a forma matricial já dimensionada:

$$\begin{array}{cccccc} B_i & V_i & + & A_i & X_i & + & W_i & = & 0 \\ r \times n & n \times 1 & & r \times u & u \times 1 & & r \times 1 & & \end{array} \quad (3.2.11)$$

Considerando P a matriz dos pesos associada às observações (Lb) podemos escrever a fórmula para a solução das equações normais:

$$X_i = - [A_i^T (B_i P^{-1} B_i^T)^{-1} A_i]^{-1} \cdot A_i^T (B_i P^{-1} B_i^T)^{-1} \cdot W_i \quad (3.2.12)^*$$

A matriz A será então:

$$\begin{array}{cccccccccccc} \frac{\partial FX}{\partial \lambda} & \frac{\partial FX}{\partial \kappa} & \frac{\partial FX}{\partial \phi} & \frac{\partial FX}{\partial \omega} & \frac{\partial FX}{\partial X_0} & \frac{\partial FX}{\partial Y_0} & \frac{\partial FX}{\partial Z_0} & \frac{\partial FX}{\partial t_1} & \frac{\partial FX}{\partial t_2} & \dots & \frac{\partial FX}{\partial t_{NF}} \\ \frac{\partial FY}{\partial \lambda} & \frac{\partial FY}{\partial \kappa} & \frac{\partial FY}{\partial \phi} & \frac{\partial FY}{\partial \omega} & \frac{\partial FY}{\partial X_0} & \frac{\partial FY}{\partial Y_0} & \frac{\partial FY}{\partial Z_0} & \frac{\partial FY}{\partial t_1} & \frac{\partial FY}{\partial t_2} & \dots & \frac{\partial FY}{\partial t_{NF}} \\ \frac{\partial FZ}{\partial \lambda} & \frac{\partial FZ}{\partial \kappa} & \frac{\partial FZ}{\partial \phi} & \frac{\partial FZ}{\partial \omega} & \frac{\partial FZ}{\partial X_0} & \frac{\partial FZ}{\partial Y_0} & \frac{\partial FZ}{\partial Z_0} & \frac{\partial FZ}{\partial t_1} & \frac{\partial FZ}{\partial t_2} & \dots & \frac{\partial FZ}{\partial t_{NF}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{array}$$

Cada feição observada gerará três linhas e oito colunas não nulas na matriz A, cujos elementos (derivadas parciais em relação aos parâmetros) serão:

$$\frac{\partial FX}{\partial \lambda} = m_{11}x + m_{12}y + m_{13}z$$

(\*) O desenvolvimento matemático para chegar até esta equação pode ser fartamente encontrado na literatura referente a ajustamento de observações [24].

$$\frac{\partial FX}{\partial \kappa} = \partial |m_{21}x + m_{22}y + m_{23}z|$$

$$\frac{\partial FX}{\partial \phi} = \lambda |x(-\text{sen}\phi \cos\kappa) + y(\text{sen}\omega \cos\phi \cos\kappa) + z(-\cos\omega \cos\phi \cos\kappa)|$$

$$\frac{\partial FX}{\partial \omega} = \lambda |m_{12}z - m_{13}y|$$

$$\frac{\partial FX}{\partial X_0} = 1$$

$$\frac{\partial FX}{\partial Y_0} = 0$$

$$\frac{\partial FX}{\partial Z_0} = 0$$

$$\frac{\partial FX}{\partial t} = - (X_2 - X_1)$$

$$\frac{\partial FY}{\partial \lambda} = m_{21}x + m_{22}y + m_{23}z$$

$$\frac{\partial FY}{\partial \kappa} = - \lambda |m_{11}x + m_{12}y + m_{13}z|$$

$$\frac{\partial FY}{\partial \phi} = \lambda |x(\text{sen}\phi \text{sen}\kappa) + y(\text{sen}\omega \cos\phi \text{sen}\kappa) + z(\cos\omega \cos\phi \text{sen}\kappa)|$$

$$\frac{\partial FY}{\partial \omega} = \lambda |m_{22}z - m_{23}y|$$

$$\frac{\partial FY}{\partial X_0} = 0$$

$$\frac{\partial FY}{\partial Y_0} = 1$$

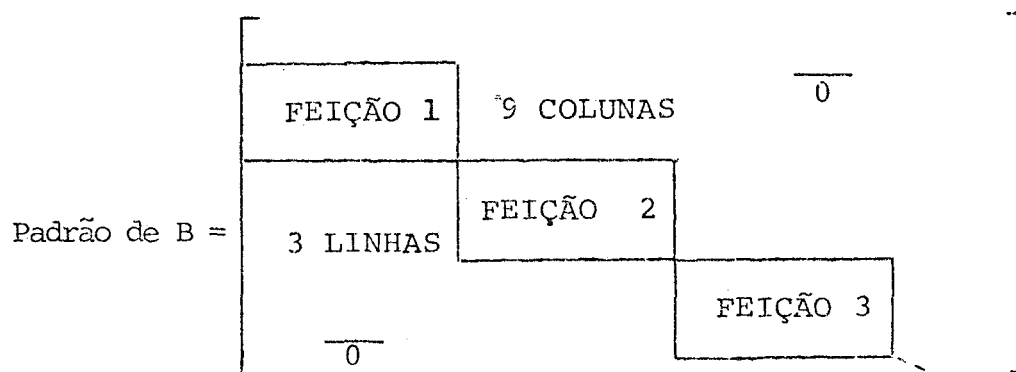
$$\frac{\partial FY}{\partial Z_0} = 0$$

$$\frac{\partial FY}{\partial t} = - (Y_2 - Y_1)$$

$$\frac{\partial FZ}{\partial \lambda} = m_{31}x + m_{32}y + m_{33}z$$

$$\frac{\partial FZ}{\partial \kappa} = 0$$





Os elementos de B serão as derivadas parciais em relação às observações:

$$\frac{\partial FX}{\partial x} = \lambda m11$$

$$\frac{\partial FX}{\partial y} = \lambda m12$$

$$\frac{\partial FX}{\partial z} = \lambda m13$$

$$\frac{\partial FX}{\partial X1} = (t-1)$$

$$\frac{\partial FX}{\partial Y1} = 0$$

$$\frac{\partial FX}{\partial Z1} = 0$$

$$\frac{\partial FX}{\partial X2} = -t$$

$$\frac{\partial FX}{\partial Y2} = 0$$

$$\frac{\partial FX}{\partial Z_2} = 0$$

$$\frac{\partial FY}{\partial x} = \lambda_{m21}$$

$$\frac{\partial FY}{\partial Y} = \lambda_{m22}$$

$$\frac{\partial FY}{\partial z} = \lambda_{m23}$$

$$\frac{\partial FY}{\partial x_1} = 0$$

$$\frac{\partial FY}{\partial Y_1} = (t - 1)$$

$$\frac{\partial FY}{\partial Z_1} = 0$$

$$\frac{\partial FY}{\partial X_2} = 0$$

$$\frac{\partial FY}{\partial Y_2} = -t$$

$$\frac{\partial FY}{\partial Z_2} = 0$$

$$\frac{\partial FZ}{\partial x} = \lambda_{m31}$$

$$\frac{\partial FZ}{\partial y} = \lambda_{m32}$$

$$\frac{\partial FZ}{\partial z} = \lambda_{m33}$$

$$\frac{\partial FZ}{\partial X1} = 0$$

$$\frac{\partial FZ}{\partial Y1} = 0$$

$$\frac{\partial FZ}{\partial Z1} = (t - 1)$$

$$\frac{\partial FZ}{\partial X2} = 0$$

$$\frac{\partial FZ}{\partial Y2} = 0$$

$$\frac{\partial FZ}{\partial Z2} = -t$$

Quando da aplicação da equação (3.2.12) consideraram-se como observações as coordenadas de campo e do modelo atribuindo-lhes, porém, pesos diferentes.

Os demais termos que aparecem em (3.2.12) são:

$$W_i = F(La_{i-1}, Xa_{i-1}) + B_i(Lb - La_{i-1}); \text{ vetor do erro de fechamento}$$

onde :

Lb =  
nX1

x,
y,
z,
x1
y1
z1
x2
y2
z2
⋮
⋮
⋮
xn
yn
zn
x1n
y1n
z1n
x2n
y2n
z2n

;

$$\begin{array}{l} \text{La} = \\ \text{nXl} \end{array} \left[ \begin{array}{c} \text{xla} \\ \text{yla} \\ \text{zla} \\ \text{Xla} \\ \text{Yla} \\ \text{Zla} \\ \text{X2a} \\ \text{Y2a} \\ \text{Z2a} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \text{xlan} \\ \text{ylan} \\ \text{zlan} \\ \text{Xlan} \\ \text{Ylan} \\ \text{Zlan} \\ \text{X2an} \\ \text{Y2an} \\ \text{Z2an} \end{array} \right]$$

;

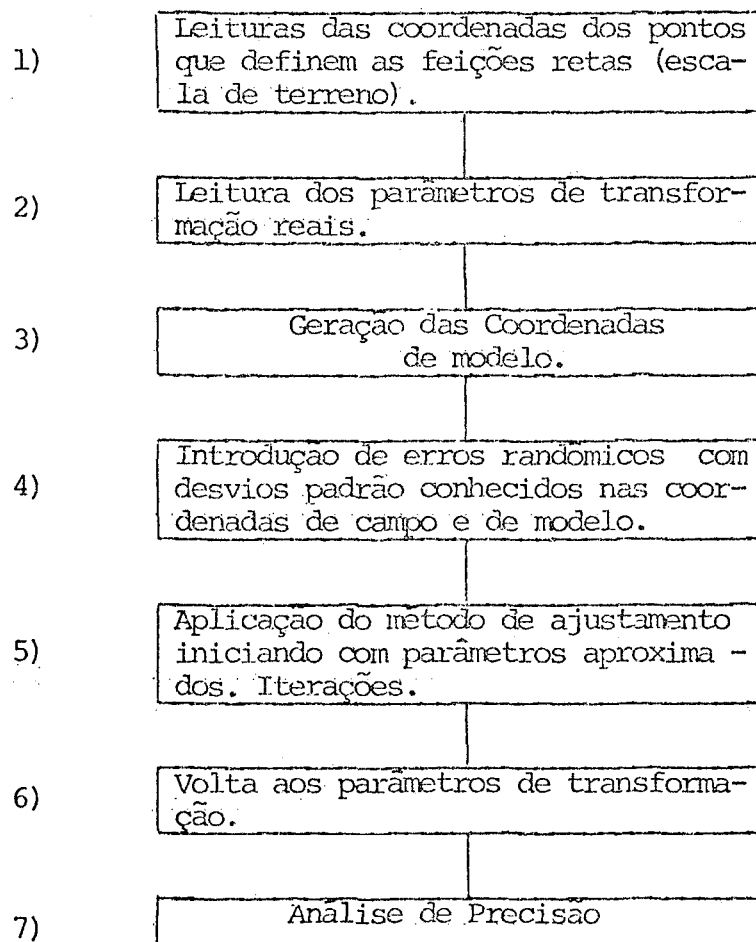


$$\hat{\sigma}_0^2 = \frac{V^T P V}{r-u} \quad (3.2.14)$$

onde:

$\hat{\sigma}_0^2$  ; variância da unidade de peso a-posteriori  
 $r-u$  ; número de graus de liberdade.

Para aplicarmos esta forma de ajustamento ao modelo matemático dado em (3.2.5) seguimos a sistemática de programação cujos passos são:



No passo 5 são necessárias de três a quatro i terações para atingirmos a convergência da solução.

Como o parâmetro  $t$  varia de feição para feição, necessitamos no mínimo de 4 feições observadas para proceder a orientação absoluta, as quais nos darão 12 equações para solucionar as 11 incógnitas ( $\lambda, \kappa, \phi, W, X_0, Y_0, Z_0, t_1, t_2, t_3, t_4$ ) do sistema.

O programa para orientação absoluta de modelos utilizando feições retas como controle se encontra no Apêndice B (RETA.FOR).

### 3.3. Feições Curvas: Utilização de Splines Paramétricos na Orientação Absoluta

Embora o modelo matemático aqui utilizado seja o mesmo desenvolvido na seção 2.1 (7 parâmetros de orientação) as coordenadas de campo de um ponto não digitalizado, lidas sobre uma feição, serão encontradas por interpolação.

Como mostra a fig. 3.3.1, ao observarmos uma feição em um modelo estereoscópico, o ponto lido sobre esta pode ou não estar armazenado em um banco de dados. Nossa observação deve recair sobre a feição em si e não sobre pontos individualizados da mesma.

A distância  $s$  (escala de modelo) entre um ponto lido e a origem desta (vide 2.5.1), quando multiplicada pelo fator de escala ( $\lambda$ ), é o parâmetro de entrada ( $s$ ) do modelo matemático de interpolação.

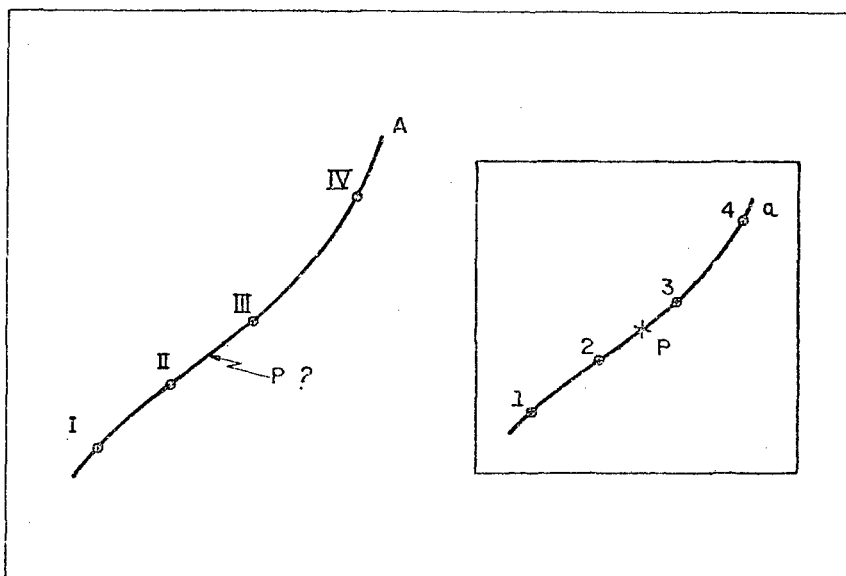


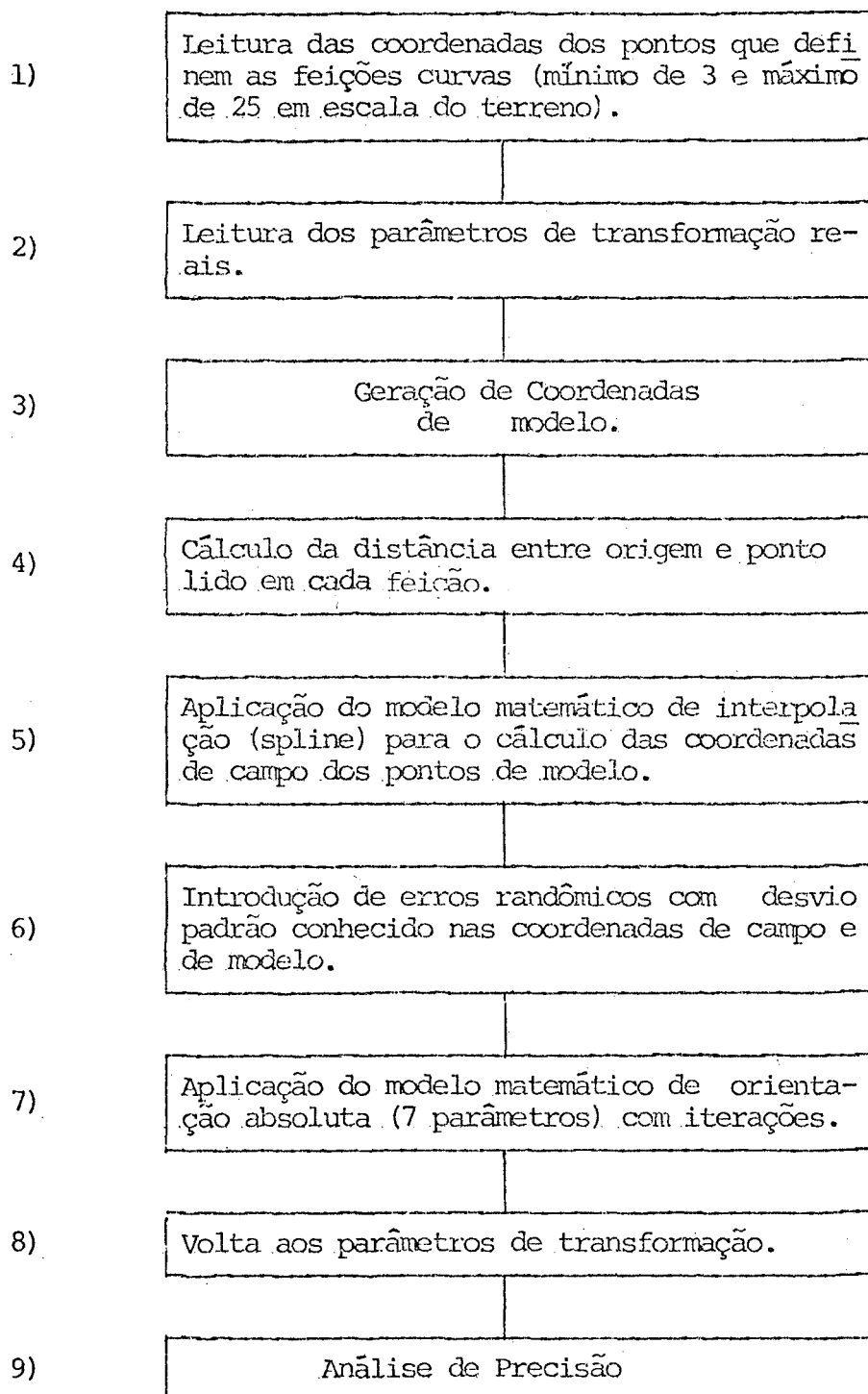
FIGURA 3.3.1 Ponto (P) pertencente a feição A não digitalizado.

Como se pode notar, não se faz a correspondência ponto a ponto neste procedimento, pois o ponto lido pode estar em qualquer posição na feição, sendo suas coordenadas de campo interpoladas com a ajuda de (s).

Uma das críticas que poderia ser feita a este processo é o fato de que para a origem das feições é necessária a correspondência ponto a ponto, o que comprometeria este trabalho. Entretanto como mostramos na seção 2.6.1, os erros que podem ser cometidos nesta etapa do processo, são bem menores que a precisão dos instrumentos fotogramétricos atuais. Isto nos liberta de uma identificação precisa\* do ponto origem, podendo até considerá-la irrelevante. Alguns testes serão levados a efeito no capítulo IV onde serão estudadas as influências de diversos valores de erros de origem na qualidade da orientação absoluta.

\* Estamos considerando que a precisão na identificação aqui referida está relacionada com futuras operações de medição no modelo estereoscópico.

A sistemática para orientação absoluta utilizando splines paramétricos foi programada (Curva.FOR, Apêndice B) na sequência seguinte:



No passo 7, de 3 a 4 iterações são necessárias.

Foi excluído o desenvolvimento do modelo matemático de ajustamento, já que, ao contrário do que aconteceu com feições retas (adição do parâmetro  $t$ ), quando nos deparamos com feições curvas, não houve modificações nas equações (2.6), cujo método de ajustamento já é bastante conhecido.

#### 3.4. Parâmetros Aproximados

O ângulo  $\kappa$  terá seu valor inicial ( $\kappa^{(0)}$ ) bem definido se considerarmos a relação:

$$\kappa^{(0)} = 90^\circ - AN \quad (3.4.1)$$

onde  $AN$  é o azimute de vôo.

Os ângulos  $\phi^{(0)}$  e  $\omega^{(0)}$  serão considerados inicialmente nulos, já que seus valores são geralmente pequenos (fotos verticais).

Para as translações podemos considerar:

$XO$  = média das coordenadas  $X$  (digitalizadas)

$YO$  = média das coordenadas  $Y$  (digitalizadas)

$ZO$  = média das coordenadas  $Z$  (digitalizadas)

ou ainda utilizar a fórmula:

$$\overline{x^{(0)}} = \overline{XG} - \lambda^{(0)} \left[ R(\kappa=\kappa^{(0)}) \quad R(\phi=\phi^0) \quad R(\omega=\omega^0) \right] \overline{x} \quad (3.4.2)$$

$$\overline{x^{(0)}} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} ; \text{ vetor dos parâmetros de translação aproximados.}$$

$$\overline{XG} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} ; \text{ coordenadas de campo (interpoladas)}$$

$$\overline{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} ; \text{ coordenadas de modelo}$$

$\lambda^{(0)}$ ; fator de escala aproximado.

Convém lembrar que para proceder à interpolação das coordenadas de campo ( $\overline{XG}$ ) é necessário inicialmente o conhecimento de  $(\lambda^{(0)})$ , o que não acarreta maiores problemas, pois geralmente se tem a escala média das fotos que compõe os modelos, e conseqüentemente, dependendo do instrumento, uma escala aproximada do modelo.

No caso de estarmos tratando com feições retas, o parâmetro  $t$ , que aparece em adição aos 7 parâmetros convencionais, pode ter seu valor aproximado  $t^{(0)}$  igual a 0.5. Não nos interessam extrapolações nas retas definidas

pelos pontos  $x_1$  e  $x_2$  (Seção 3.2).

## CAPÍTULO IV

### 4. TESTES

#### 4.1. Introdução

Para testarmos a utilização de feições como controle para orientação absoluta, idealizamos um conjunto de dados (vide seção 2.4) com as seguintes características:

- arquivo contendo feições curvas definidas por 25 pontos (escala de terreno);
- arquivo contendo feições retas;
- arquivo contendo pontos de controle;
- desvio padrão das coordenadas de terreno igual a 1,5 m;
- desvio padrão das coordenadas de modelo igual a 0,015 mm (15 micrômetros);
- modelos gerados na escala 1:10.000

O desvio padrão de 1,5 m, considerado para as observações de terreno, tem a finalidade de criar uma situação bastante desfavorável aos experimentos, possibilitando assim uma melhor avaliação dos modelos matemáticos.

Utilizamos para os testes:

- 4.2, 4.3 e 4.4 o program CURVA.FOR (Apêndice B)
- 4.5, 4.6 e 4.7 os programas RETA.FOR e POINT.FOR (Apêndice B).

#### 4.2. Feições Curvas: Teste de Frequência

Para a execução deste teste foi considerada uma localização das feições no modelo como mostra a fig. 4.2.1.

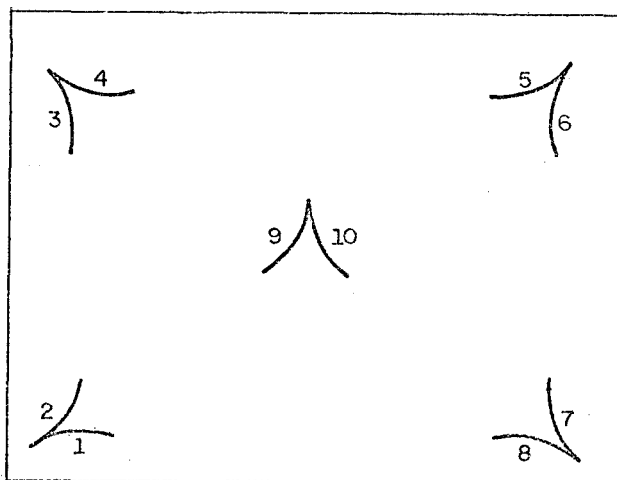


FIGURA 4.2.1 Localização das feições no modelo para o teste de frequência.

Como mostramos na seção 2.6.3 um aumento na frequência de geração das feições curvas (aumento da sinuosidade das curvas) deteriora a qualidade de interpolação. O estudo da influência desta deterioração, na precisão dos valores ajustados dos parâmetros de orientação absoluta, é o objetivo deste teste.

## TESTE DE FREQUENCIA

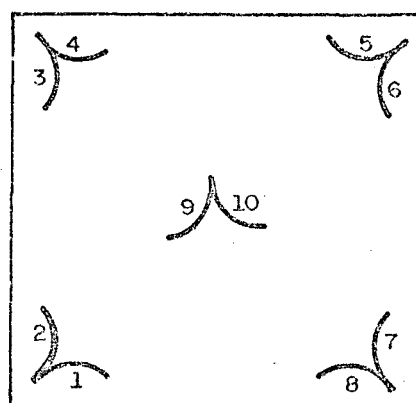
VALORES REAIS DOS PARÂMETROS		PARÂMETROS CALCULADOS		
		I	II	III
		NF= 10 F=0.1	NF= 10 F=0.3	NF= 10 F=0.8
$\kappa$ (rad)	0,78539	0,78539	0,78538	0,78538
$\phi$ (rad)	0,00000	0,00000	0,00000	0,00000
$\omega$ (rad)	0,00000	0,00000	0,00000	0,00000
XO ( m)	50,00000	50,009	50,193	50,199
YO ( m)	300,000	299,856	299,762	299,760
ZO ( m)	100,000	100,107	99,890	99,889
$\lambda$	10,0000	9,9999	9,9999	9,9999

NF = nº de feições curvas

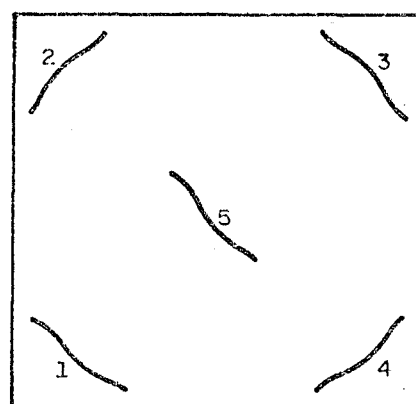
F = frequência de geração

### 4.3. Feições Curvas: Número e Distribuição

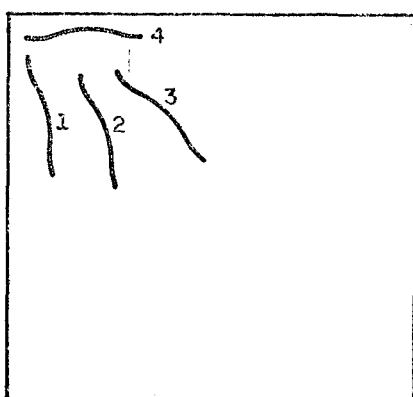
A figura 4.3 ilustra as condições de número e distribuição das feições curvas utilizadas nestes testes.



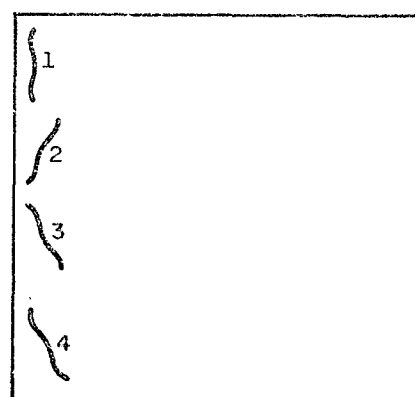
4.3 a



4.3 b



4.3 c



4.3 d

FIGURA 4.3 Ilustração das feições utilizadas para testar número e distribuição no modelo.

## TESTE DE NÚMERO E DISTRIBUIÇÃO

VALORES REAIS DOS PARÂMETROS		PARÂMETROS CALCULADOS			
		Fig. 4.3.a NF= 10 A=0.2 F=0.3	Fig. 4.3.b NF= 5 A=0.2 F=0.3	Fig. 4.3.c NF= 4 A=0.2 F=0.3	Fig. 4.3.d NF= 4 A=0.2 F=0.3
$\kappa$ (rad)	0,78539	0,78538	0,78540	0,78541	0,78544
$\phi$ (rad)	0,00000	0,00000	0,00001	0,00163	0,00340
$\omega$ (rad)	0,00000	0,00000	0,00001	0,00167	0,00342
X0 (m)	50,000	50,199	50,479	50,405	50,786
Y0 (m)	300,000	299,760	300,550	302,262	301,381
Z0 (m)	100,000	99,890	100,291	99,445	99,992
$\lambda$	10,0000	9,9999	9,9997	9,9991	9,9992

NF = nº de feições curvas

A = amplitude de geração

F = frequência de geração

#### 4.4. Feições Curvas: Erro de Origem

A influência do erro ( $\epsilon$ ) no estabelecimento da origem da feição na interpolação utilizando splines paramétricos já foi testada na seção 2.6.1. Apresentaremos a seguir os efeitos do erro ( $\epsilon$ ) na precisão dos parâmetros calculados na orientação absoluta do modelo ilustrado na fig. 4.3.a.

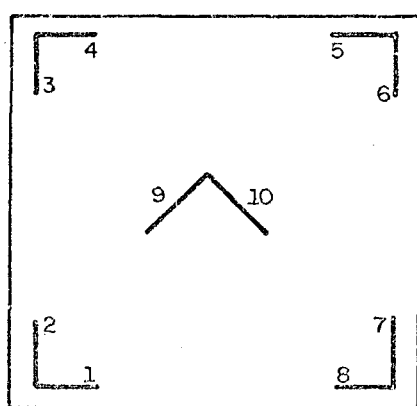
#### TESTE COM ERROS DE ORIGEM

VALORES REAIS DOS PARÂMETROS		PARÂMETROS CALCULADOS			
		NF= 10 A=0.2 F=0.3 $\epsilon=0.0$	NF= 10 A=0.2 F=0.3 $\epsilon=1m$	NF= 10 A=0.2 F=0.3 $\epsilon=2m$	NF= 10 A=0.2 F=0.3 $\epsilon=3m$
$\kappa$ (rad)	0,78539	0,78538	0,78537	0,78536	0,78536
$\phi$ (rad)	0,00000	0,00000	0,00000	0,00000	0,00000
$\omega$ (rad)	0,00000	0,00000	0,00000	0,00000	0,00000
X0 (m)	50,000	50,199	50,954	51,711	52,468
Y0 (m)	300,000	299,760	300,278	300,797	301,317
Z0 (m)	100,000	99,890	99,713	99,536	99,360
$\lambda$	10,0000	9,9999	9,9996	9,9993	9,9990

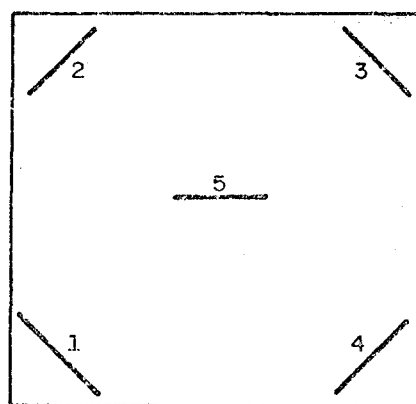
NF = nº de feições curvas  
 A = amplitude de geração  
 F = freqüência de geração  
 $\epsilon$  = erro de origem

#### 4.5. Feições Retas: Número e Distribuição

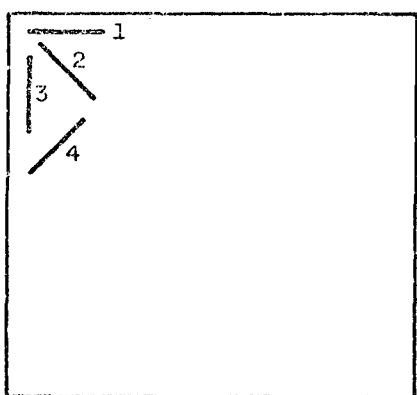
Para testar o modelo matemático de orientação absoluta que utiliza feições retas como controle, utilizamos as feições distribuídas como mostra a fig. 4.5



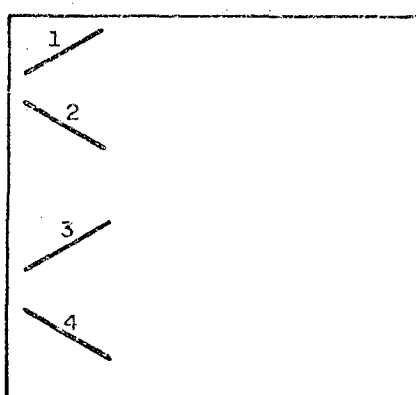
4.5 a



4.5 b



4.5 c



4.5 d

FIGURA 4.5 Distribuição de feições retas nos modelos testados.

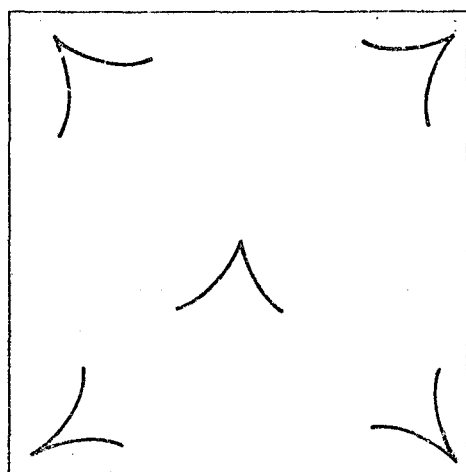
TESTE DE NÚMERO E DISTRIBUIÇÃO

VALORES REAIS DOS PARÂMETROS		PARÂMETROS CALCULADOS			
		Fig.4.5a NF=10	Fig.4.5b NF=5	Fig.4.5c NF=4	Fig.4.5d NF=4
$\kappa$ (rad)	0,87266	0,87250	0,87286	0,88680	0,87272
$\phi$ (rad)	0,00000	0,00025	0,00080	0,02593	0,01185
$\omega$ (rad)	0,00000	0,00039	0,00173	0,00424	0,01317
X0 (m)	3500,000	3499,637	3498,801	3428,998	3509,444
Y0 (m)	2000,000	2000,428	1999,508	1976,147	2000,903
Z0 (m)	700,000	699,727	699,592	574,967	639,132
$\lambda$	10,0000	9,9979	10,0005	9,8841	9,9973

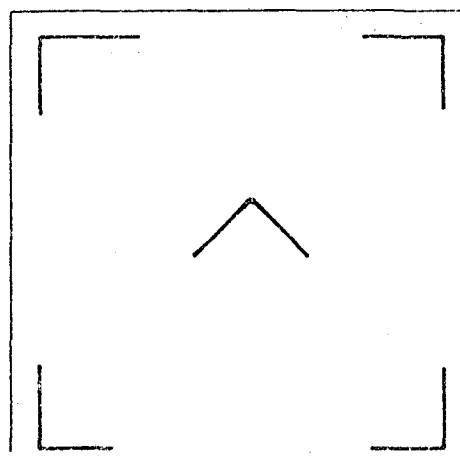
NF = nº de feições retas

#### 4.6. Feições Curvas e Retas: Comparação com pontos de Controle

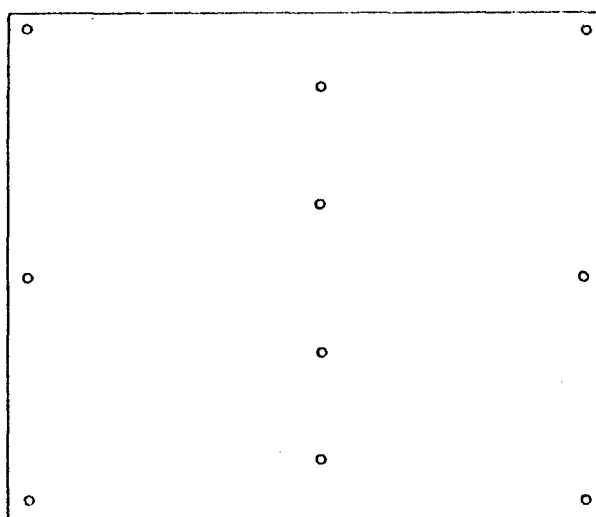
A fig. 4.6 ilustra os modelos pesquisados neste teste comparativo.



4.6. Feições Curvas



4.6.b Feições Retas



4.6.c Pontos de controle

FIGURA 4.6 Tipos de controle utilizados no testes comparativos

## TESTE COMPARATIVO

VALORES REAIS DOS PARÂMETROS		PARÂMETROS CALCULADOS		
		Fig. 4.6a NFC=10	Fig. 4.6b NFR=10	Fig. 4.6c NP=10
$\kappa$ (rad)	0,78539	0,78538	0,78523	0,78541
$\phi$ (rad)	0,00000	0,00000	0,00020	0,00001
$\omega$ (rad)	0,00000	0,00000	0,00023	0,00000
X0 (m)	50,000	50,199	49,634	50,148
Y0 (m)	300,000	299,760	300,430	300,731
Z0 (m)	100,000	99,890	99,728	100,062
$\lambda$	10,0000	9,9999	9,9989	9,9997

NFC = nº de feições curvas

NFR = nº de feições retas

NP = nº de pontos de controle

#### 4.7. Influência da Precisão dos Valores Aproximados dos Parâmetros

No que diz respeito a valores aproximados dos parâmetros, os modelos matemáticos para orientação absoluta, utilizando feições curvas e feições retas, mostraram-se bastante flexíveis.

Os ângulos  $\phi$  e  $\omega$  tiveram seus valores aproximados pequenos, pois as fotos que compõe os modelos estereos cópicos foram idealizadas como verticais. O ângulo  $\kappa$  teve seu valor aproximado oscilando com diferenças de até 30° do seu valor real.

Para as translações (X0, Y0, Z0) os valores aproximados foram tomados com erros iguais à metade do valor verdadeiro, o mesmo acontecendo com o fator de escala ( $\lambda$ ).

Os processos matemáticos de ajustamento, apesar do fato de iniciarem com valores aproximados grosseiramente errados, convergiram em três iterações, sem comprometer significativamente a precisão dos parâmetros calculados.

TESTES COM VALORES APROXIMADOS  
DOS PARÂMETROS

VALORES REAIS DOS PARÂMETROS		P A R Â M E T R O S			
		FEIÇÕES CURVAS		FEIÇÕES RETAS	
		APROX.	CALC.	APROX.	CALC.
$\kappa$ (rad)	0,87266	1,39626	0,87265	1,39626	0,87250
$\phi$ (rad)	0,00000	0,03580	0,00000	0,03580	0,00025
$\omega$ (rad)	0,00000	0,05266	0,00000	0,05266	9,00039
X0 (m)	3500,000	1750,000	3500,146	1750,000	3499,637
Y0 (m)	2000,000	1000,000	1999,783	1000,000	2000,428
Z0 (m)	700,000	1050,000	699,850	1050,000	699,727
$\lambda$	10,0000	15,0000	9,9999	15,0000	9,9979

QUADRO RESUMO DOS TESTES FEITOS NO CAPÍTULO IV

VALORES REAIS	FEIÇÕES CURVAS						ERRO DE ORIGEM			FEIÇÕES RETAS				PONTOS
	NF=10 A=0.2 F=0.1	NF=10 A=0.2 F=0.3	NF=10 A=0.2 F=0.8	NF=5 A=0.2 F=0.3	NF=4 A=0.2 F=0.3	NF=4 A=0.2 F=0.3	ε =1m	ε =2m	ε =3m	NF=10	NF=5	NF=4	NF=4	NP=10
∞ (rad)														
0,78539	0,78539	0,78538	0,78538	0,78540	0,78541	0,78544	0,78537	0,78536	0,78536	0,78523	0,78559	0,79953	0,78545	0,78541
∅ (rad)														
0,00000	0,00000	0,00000	0,00000	0,00001	0,00163	0,00340	0,00000	0,00000	0,00000	0,00020	0,00077	0,02601	0,01182	0,00001
ω (rad)														
0,00000	0,00000	0,00000	0,00000	0,00000	0,00167	0,00342	0,00000	0,00000	0,00000	0,00023	0,00158	0,00429	0,01321	0,00001
XO(m)														
50,000	50,009	50,193	50,199	50,479	50,405	50,786	50,954	51,711	52,684	49,634	48,834	51,009	50,135	50,148
YO(m)														
300,000	299,856	299,762	299,760	300,550	302,262	301,381	300,278	300,797	301,3172	300,430	299,510	303,578	300,234	300,731
ZO(m)														
100,000	100,107	99,890	99,869	100,291	99,445	99,992	99,713	99,536	99,360	99,728	99,593	82,138	95,305	100,062
λ														
10,0000	9,9999	9,9999	9,9999	9,9997	9,9991	9,9992	9,9996	9,9993	9,9990	9,9989	10,0015	9,8863	9,9978	9,9997

NF — N° de feições ; A — amplitude ; ε — erro de origem  
 NP — N° de pontos ; F — frequência

## CAPÍTULO V

## 5. CONCLUSÕES E RECOMENDAÇÕES

5.1. Conclusões

Uma análise dos resultados dos testes apresentados nos capítulos anteriores permite-nos concluir o que se segue:

No teste de frequência com feições curvas (seção 4.2) os resultados mostraram que, embora na prática não se encontrem feições que necessitem de curvas definidoras extremamente sinuosas (frequência de geração grande), a densidade dos pontos na feição deve possibilitar a reprodução precisa da curva através do modelo matemático de interpolação.

Quanto ao número e disposição de feições curvas e retas os resultados foram os esperados. Como os dados utilizados (coordenadas de campo e de modelo) não possuíam erros sistemáticos (somente randômicos), uma diminuição do número de observações se mostrou prejudicial à precisão dos parâmetros calculados, assim como uma distribuição que comprometesse a geometria do modelo estereoscópico. Os resultados apresentados nas seções 4.3 e 4.5 ilustram estas observações.

Os resultados obtidos na seção 4.3 apenas vêm confirmar as observações feitas na seção 2.6.1, onde, através da análise dos resultados, ficou constatada uma flexibilidade na identificação da origem (início) da feição para

interpolação com splines paramétricos. Os parâmetros calculados (seção 4.3) não tiveram a sua precisão significativamente comprometida.

Uma comparação entre os modelos matemáticos, para feições retas, que utilizam pontos interpolados e que utilizam pontos de controle convencionais, foi apresentada na seção 4.6. Os resultados mostraram que é perfeitamente possível obter precisão equivalente utilizando feições retas, feições curvas e pontos como controle na orientação absoluta de um modelo estereoscópico.

Em se tratando da parte computacional, o modelo matemático que utiliza feições retas, por sua simplicidade, é comparável ao que utiliza pontos de controle. O mesmo não ocorre com o modelo para uso de feições curvas, onde aparece também o modelo matemático para interpolação através de splines paramétricos, o que causa uma necessidade maior de espaço de memória e tempo de processamento.

## 5.2. Recomendações

Conscientes de que este trabalho não esgota o assunto em pauta e fazendo uso da experiência adquirida durante a elaboração do mesmo, recomendamos aos futuros pesquisadores do tema que:

- a) Seja estimulada na comunidade cartográfica brasileira a idéia da criação de um banco de dados, onde, entre outras informações afins, tenhamos arquivos com fei-

ções digitalizadas principalmente de áreas urbanas;

- b) Sejam melhorados e otimizados os programas computacionais utilizados nesta dissertação;
- c) O modelo matemático para feições curvas inclua, na parte do ajustamento, os parâmetros necessários para interpolação com splines, sofisticação esta que não foi considerada neste trabalho;
- d) Seja desenvolvido um modelo matemático que faça a orientação absoluta de um modelo estereoscópico utilizando simultaneamente e sem distinção feições curvas, feições retas e pontos de controle.

## APENDICE A

## SUGESTÕES PARA TRABALHOS FUTUROS

## 1. POSIÇÃO DAS FEIÇÕES NO MODELO

Quando testamos em 2.6.2 o erro de interpolação devido à posição da feição e analisamos os resultados, verificamos que, quando uma determinada feição estivesse, por exemplo, aproximadamente paralela ao eixo X (posição A, fig. 2.6.3), as coordenadas interpoladas X dos pontos da feição apresentavam discrepâncias bastante grandes. Entretanto, as coordenadas interpoladas Y da mesma feição mostravam discrepâncias bem menores. Comparando-se os resultados dos testes para as posições A, B e C (fig. 2.6.3) verificamos que as melhores coordenadas interpoladas X foram conseguidas com a feição aproximadamente paralela a Y e as melhores coordenadas Y com a feição agora aproximadamente paralela a X.

Para um trabalho posterior sugerimos que este fato seja aproveitado considerando-se diferentes pesos para as coordenadas dos pontos da feição conforme a sua posição no modelo (fig. A.1).

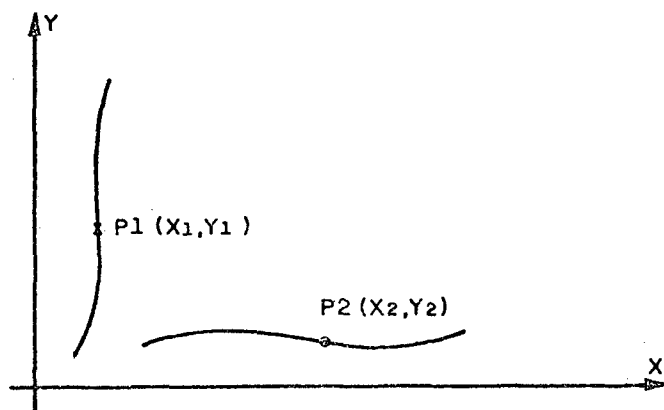


FIGURA A1 Pesos controlados pela posição da feição.

Um exemplo, considerando-se a fig. A.1 seria:

$P_1$       Peso das coordenadas  $Y_1 = p$   
             Peso das coordenadas  $X_1 = 2p$

$P_2$       Peso das coordenadas  $Y_2 = 2p$   
             Peso das coordenadas  $X_2 = p$

Nestas observações não foi considerada a coordenada  $Z$  pois dificilmente utilizaríamos feições aproximadamente paralelas ao eixo das altitudes.

## 2. MATRIZ DE RODRIGUES

Como recomendamos anteriormente, seria interessante desenvolver um modelo matemático que, além de considerar os parâmetros do spline como incógnitas, processasse simultaneamente e sem distinção feições curvas, feições retas e pontos de controle. É fácil perceber que um modelo ma

temático com essa capacidade seria bastante complexo. No sentido de facilitar os trabalhos futuros apresentaremos a seguir uma linearização utilizando a matriz de Rodrigues e que não nos pareceu muito clara na bibliografia consultada.

Seja a transformação de similaridade:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \quad (\text{A.2.1})$$

onde :

$[ X \ Y \ Z ]^T$  = coordenadas de ponto no terreno

$[ x \ y \ z ]^T$  = coordenadas de ponto no modelo

$\lambda$  = fator de escala;

$[ X_0 \ Y_0 \ Z_0 ]^T$  = translações; e

$$R = \frac{1}{1 + \frac{1}{4}(a^2 + b^2 + c^2)} \begin{bmatrix} 1 + \frac{1}{4}(a^2 - b^2 - c^2) & -c + \frac{1}{2}ab & b + \frac{1}{2}ac \\ c + \frac{1}{2}ab & 1 + \frac{1}{4}(-a^2 + b^2 - c^2) & -a + \frac{1}{2}bc \\ -b + \frac{1}{2}ac & a + \frac{1}{2}bc & 1 + \frac{1}{4}(-a^2 - b^2 + c^2) \end{bmatrix}$$

(A.2.2)

que é a Matriz de Rodrigues [28].

A Matriz de Rodrigues é muito útil pois é uma matriz ortogonal de terceira ordem, em termos de três parâmetros independentes, e sem as funções trigonométricas [28].

Fazendo:

$$1 + \frac{1}{4} (a^2 + b^2 + c^2) = k$$

a expressão (A.2.2.) fica

$$R = \frac{1}{K} \cdot \bar{R} \quad (\text{A.2.3})$$

Temos então:

$$\frac{\partial R}{\partial a} = \frac{1}{2K} \begin{bmatrix} a & b & c \\ b & -a & -2 \\ c & 2 & -a \end{bmatrix} - \frac{a\bar{R}}{2K^2};$$

$$\frac{\partial R}{\partial b} = \frac{1}{2K} \begin{bmatrix} -b & a & 2 \\ a & b & c \\ -2 & c & -b \end{bmatrix} - \frac{b\bar{R}}{2K^2}; \text{ e}$$

$$\frac{\partial R}{\partial c} = \frac{1}{2K} \begin{bmatrix} -c & -2 & a \\ 2 & -c & b \\ a & b & c \end{bmatrix} - \frac{c\bar{R}}{2K^2}.$$

Para valores aproximados de  $a$ ,  $b$  e  $c$  iguais a zero vem que:

$K = 1$  e portanto:

$$\frac{\partial R}{\partial a} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} ;$$

$$\frac{\partial R}{\partial b} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} ; e$$

$$\frac{\partial R}{\partial c} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

Considerando  $a$ ,  $b$  e  $c$  como segue:

$$a \rightarrow \omega$$

$$b \rightarrow \phi$$

$$c \rightarrow \kappa$$

e fazendo as rotações na ordem;

$$R = R(\omega) R(\phi) R(\kappa)$$

para valores dos ângulos próximos a zero podemos escrever:

$$dR = \begin{bmatrix} 0 & -dc & db \\ dc & 0 & -da \\ -db & da & 0 \end{bmatrix} \quad (\text{A.2.4})$$

Sejam agora as equações de observação:

$$\begin{bmatrix} VX \\ VY \\ VZ \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \lambda R \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \quad (\text{A.2.5})$$

onde

$$[ VX \quad VY \quad VZ ]^T$$

é o vetor dos resíduos.

O uso da linearização da matriz de Rodrigues apresentada em (A.2.4) pode ser utilizada em (A.2.5) se inicialmente fizermos uma transformação aproximada dos sete parâmetros de similaridade como segue:

$\lambda_1 \rightarrow$  constante próxima ao valor de  $\lambda$  verdadeiro.

$a \rightarrow \omega_1 = 0$

$b \rightarrow \phi_1 = 0$

$c \rightarrow \kappa_1 =$  constante próxima ao valor de  $\kappa$  verdadeiro

Aplicando estes valores aproximados em (A.2.1) obtemos:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \lambda I \quad R I \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} X_0^1 \\ Y_0^1 \\ Z_0^1 \end{bmatrix} \quad (\text{A.2.6})$$

$[X_0^1 \quad Y_0^1 \quad Z_0^1]^T$  é o vetor de translações também próximos dos valores verdadeiros.

Após esta transformação e de posse das coordenadas  $X'$ ,  $Y'$  e  $Z'$  as equações de observação linearizadas tomando:

$$\begin{aligned} \lambda_0 &= 1 \\ a, b \text{ e } c &= 0 \end{aligned}$$

que nos dá  $R_0$  e ainda pequenas translações:

$$[dx_0 \quad dy_0 \quad dz_0]^T$$

ficamos com:

$$\begin{bmatrix} VX \\ VY \\ VZ \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - (\lambda I + \lambda_0 d\lambda) \quad (R I + R_0 dR) \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} X_0^1 + dx_0 \\ Y_0^1 + dy_0 \\ Z_0^1 + dz_0 \end{bmatrix} \quad (\text{A.2.7})$$

A equação (A.2.7) expressa a transformação, total, ou seja, a transformação inicial mais a diferencial. Como já dispunhamos das coordenadas  $[X' \quad Y' \quad Z']^T$ , oriundas da transformação inicial, é correto então escrever:

$$\begin{bmatrix} VX \\ VY \\ VZ \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - (\lambda_0 R_0 + R_0 d\lambda + \lambda_0 dR) \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} - \begin{bmatrix} X_0^1 + dX_0 \\ Y_0^1 + dY_0 \\ Z_0^1 + dZ_0 \end{bmatrix}$$

(A.2.8)

Esta expressão é encontrada retirando-se de (A.2.7) os valores aproximados  $R_1$  e  $\lambda_1$  e desprezando-se o produto diferencial  $dRd\lambda$ .

Os produtos que aparecem entre parenteses são

$$R_0 = I \quad \text{e} \quad \lambda_0 R_0 = I$$

$$R_0 d\lambda = d\lambda I \quad \text{e}$$

$$\lambda_0 dR = \begin{bmatrix} 0 & -dc & db \\ dc & 0 & -da \\ -db & da & 0 \end{bmatrix}$$

Desenvolvendo (A.2.8) vem:

$$\begin{array}{c} \left[ \begin{array}{c} VX \\ VY \\ VZ \end{array} \right] = \left[ \begin{array}{c} X \\ Y \\ Z \end{array} \right] - \begin{array}{c} \lambda 0 R 0 \\ 0 \ 1 \ 0 \\ 0 \ 0 \ 1 \end{array} \left[ \begin{array}{c} X' \\ Y' \\ Z' \end{array} \right] - \begin{array}{c} R 0 d\lambda \\ d\lambda \ 0 \ 0 \\ 0 \ d\lambda \ 0 \end{array} \left[ \begin{array}{c} X' \\ Y' \\ Z' \end{array} \right] - \begin{array}{c} \lambda 0 dR \\ 0 \ -dc \ db \\ dc \ 0 \ -da \\ -db \ da \ 0 \end{array} \left[ \begin{array}{c} X' \\ Y' \\ Z' \end{array} \right] - \left[ \begin{array}{c} dx_0 \\ dy_0 \\ dz_0 \end{array} \right] \end{array}$$

onde:

$$-(R 0 d\lambda + \lambda 0 dR) \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} -d\lambda X' - dbZ' + dcY' \\ -d\lambda Y' + daZ' - dcX' \\ -d\lambda Z' - daY' + dbX' - 0 \end{bmatrix} =$$

$$= \begin{bmatrix} -X' & 0 & -Z' & Y' \\ -Y' & Z' & 0 & -X' \\ -Z' & -Y' & X' & 0 \end{bmatrix} \begin{bmatrix} d\lambda \\ da \\ db \\ dc \end{bmatrix} \quad e:$$

$$- \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = - \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}$$

$$- \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = - \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}$$

Acrescentando as translações diferenciais ficamos com as equações de observação agora linearizadas:

$$\begin{bmatrix} VX \\ VY \\ VZ \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} -X' & 0 & -Z' & Y' & -1 & 0 & 0 \\ -Y' & Z' & 0 & -X' & 0 & -1 & 0 \\ -Z' & -Y' & X & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} d\lambda \\ da \\ db \\ dc \\ dX_0 \\ dY_0 \\ dZ_0 \end{bmatrix} - \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (\text{A.2.9})$$

onde:

$$\begin{bmatrix} X' & Y' & Z' \end{bmatrix}^T$$

são as coordenadas calculadas com a primeira aproximação.

APÊNDICE B  
PROGRAMA E SUB-ROTINAS

**NOME DO PROGRAMA:**

GENER. FOR

**AUTOR (ES) :**

JOÃO BOSCO LUGNANI &amp; JORGE MASSON

**COMENTÁRIOS:**

Precisão simples.

Este programa gera feições definidas por  
25 pontos.

**OBSERVAÇÃO**

Não há necessidade de dados pois o programa solicita informações na forma de diálogo através do terminal.

GENER.FOR            FORTRAN V.6(1144)            11-OCT-82            14:06    PAGE 1

```

        DIMENSION X(50),Y(50),Z(50),XN(50),YN(50),ZN(50)
        TYPE 13
13      FORMAT(/,9X,'VALOR DE "A" PARA AS COORDENADAS Y? ',)$)
        ACCEPT 5,AY
5       FORMAT(G)
        TYPE 14
14      FORMAT(/,9X,'VALOR DE "A" PARA AS COORDENADAS Z? ',)$)
        ACCEPT 5,AZ
        TYPE 17
17      FORMAT(/,9X,'VALOR DE "F" PARA AS COORDENADAS Y? ',)$)
        ACCEPT 5,FY
        TYPE 18
18      FORMAT(/,9X,'VALOR DE "F" PARA AS COORDENADAS Z? ',)$)
        ACCEPT 5,FZ
        CALL DELTA(X,Y,Z,AY,FY,AZ,FZ)
        TYPE 21
21      FORMAT(/,9X,'INTRODUZA OS VALORES PARA OS PONTOS (P/ TERMINAR
* TECLE ?):',/))
100     TYPE 51
51      FORMAT(/,9X,'PONTO? ',)$)
        ACCEPT 11,PT
11      FORMAT(A5)
        IF(PT.EQ.'?') GO TO 200
        TYPE 22
22      FORMAT(/,9X,'VALOR DE XP? ',)$)
        ACCEPT 5,XP
        TYPE 23
23      FORMAT(/,9X,'VALOR DE YP? ',)$)
        ACCEPT 5,YP
        TYPE 28
28      FORMAT(/,9X,'VALOR DE ZP? ',)$)
        ACCEPT 5,ZP
        TYPE 29
29      FORMAT(/,9X,'VALOR DE TETA(GRAUS DECIMAIS)? ',)$)
        ACCEPT 5,ALFA
        CALL NCD(XN,YN,ZN,X,Y,Z,ALFA,XP,YP,ZP)
        WRITE(3,400)PT,XP,YP,ZP,AY,AZ,FY,FZ,ALFA
400     FORMAT('1',///,9X,'PONTO CONSIDERADO ** ',A5,' **',///,9X,
* 'VALOR DE XP: ',G,/,9X,'VALOR DE YP: ',G,/,9X,'VALOR
* DE ZP: ',G,///,9X,'VALOR DE "A" PARA Y: ',G,/,9X,'VALOR DE
* "A" PARA Z: ',G,///,9X,'VALOR DE "F" PARA Y: ',G,/,9X,'VA
* LOR DE "F" PARA Z: ',G,///,9X,'VALOR DE TETA: ',G,///)
        WRITE(3,402)
402     FORMAT(/,1X,'ORDEM',14X,'XN',15X,'YN',15X,'ZN',/)
        DO 54 I=1,25
        WRITE(3,401)I,XN(I),YN(I),ZN(I)
401     FORMAT(3X,I2,2X,3(5X,F12.2))
54      CONTINUE
        GO TO 100
200     CALL EXIT
        END)

```

GENER.FOR

FORTRAN V.6(1144)

11-OCT-82

14:06

PAGE 1

```

SUBROUTINE DELTA(X,Y,Z,AY,FY,AZ,FZ)
DIMENSION X(50),Y(50),Z(50),S(50)
PI=3.1415926536
DL=2.*PI/24
DO 50 I=1,25
M=I-1
S(I)=M*DL
X(I)=S(I)/PI
Y(I)=AY*SIN(FY*PI*S(I))/PI
Z(I)=AZ*SIN(FZ*S(I)*PI)/PI
50 CONTINUE
RETURN
END
```

GENER.FOR

FORTRAN V.6(1144)

11-OCT-82

14:06

PAGE 1

```
      SUBROUTINE NCD(XN,YN,ZN,X,Y,Z,ALFA,XP,YP,ZP)
      DIMENSION X(50),Y(50),YN(50),XN(50),VN(5),A(5,5),V(5)
      *,VP(5),ZN(50),Z(50)
      ALFA=(3.1415926536/180.)*ALFA
      DO 40 I =1,3
      DO 40 J=1,3
      A(I,J)=0.
40    CONTINUE
      A(1,1)=COS(ALFA)
      A(1,2)=SIN(ALFA)
      A(2,2)=A(1,1)
      A(2,1)=-A(1,2)
      A(3,3)=1.
      VP(1)=XP
      VP(2)=YP
      VP(3)=ZP
      BETA=100.
      DO 51 I=1,3
      DO 51 J=1,3
      A(I,J)=A(I,J)*BETA
51    CONTINUE
      DO 90 I=1,25
      V(1)=X(I)
      V(2)=Y(I)
      V(3)=Z(I)
      CALL PROD(VN,A,V,3,VP)
      XN(I)=VN(1)
      YN(I)=VN(2)
      ZN(I)=VN(3)
90    CONTINUE
      RETURN
      END
```

GENER.FOR            FORTRAN V.6(1144)            11-OCT-82            14:06    PAGE 1

```
      SUBROUTINE PROD(VN,A,V,N,VP)
      DIMENSION VN(5),V(5),A(5,5),VP(5)
      DO 30 I=1,N
      VN(I)=0.
      DO 30 J=1,N
      VN(I)=VN(I)+A(I,J)*V(J)
30    CONTINUE
      DO 40 I=1,N
      VN(I)=VN(I)+VP(I)
40    CONTINUE
      RETURN
      END
```

**NOME DO PROGRAMA:**

TE3. FOR

**AUTOR:**

JOÃO BOSCO LUGNANI

**COMENTÁRIOS:**

Precisão Dupla.

O programa testa a qualidade da interpolação que utiliza splines paramétricos para pontos de uma determinada feição.

**OBSERVAÇÃO:**

Exemplo de arquivo de dados:

NP	X	Y	Z
1	3.4	2.5	8.0
2	7.2	0.3	2.7
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
n	Xn	Yn	Zn

```

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION S(30),X(30),Y(30),Z(30),SC(15),SI(15),SX(15),SY(15),
1SZ(15),INDEX(30),DELX(15),DELY(15),DELZ(15),XI(15),YI(15),
1ZI(15),XII(15),YII(15),ZII(15),XNOD(15),YNOD(15),ZNOD(15)
DISTA(X1,Y1,Z1,X2,Y2,Z2)=((X2-X1)**2+(Y2-Y1)**2+
1(Z2-Z1)**2)**0.5
I=1
S(I)=0.0
READ(2,10)NF,X1,Y1,Z1
X(I)=X1
Y(I)=Y1
Z(I)=Z1
5 CONTINUE
READ(2,10,END=20)NF1,X2,Y2,Z2
10 FORMAT(4G)
D=DISTA(X1,Y1,Z1,X2,Y2,Z2)
X1=X2
Y1=Y2
Z1=Z2
S(I)=S(I)+D
I=I+1
S(I)=S(I-1)
X(I)=X1
Y(I)=Y1
Z(I)=Z1
GO TO 5
20 CONTINUE
K=I-1
WRITE(3,30)K
30 FORMAT(T20,'NUMERO DE SEGMENTOS ',I2, '//')
WRITE(3,40)
40 FORMAT(T20,'DISTANCIAS CUMULATIVAS', '//')
WRITE(3,50)(S(J),J=1,K)
50 FORMAT(5(F8,3,2X))
CALL DIVISI(I,S,X,Y,Z,SC,XNOD,YNOD,ZNOD,SI,XI,YI,ZI,NN)
CALL SPCDEF(NN,SC,XNOD,SX,INDEX,SPAR)
CALL SPCDEF(NN,SC,YNOD,SY,INDEX,SPAR)
CALL SPCDEF(NN,SC,ZNOD,SZ,INDEX,SPAR)
NNM1=NN-1
DO 100 I=1,NNM1
SPAR=SI(I)
XII(I)=SPLINE(NN,SC,XNOD,SX,INDEX,SPAR)
YII(I)=SPLINE(NN,SC,YNOD,SY,INDEX,SPAR)
ZII(I)=SPLINE(NN,SC,ZNOD,SZ,INDEX,SPAR)
DELX(I)=XII(I)-XI(I)
DELY(I)=YII(I)-YI(I)
DELZ(I)=ZII(I)-ZI(I)
100 CONTINUE
WRITE(3,110)
110 FORMAT(T30,///,10X,'DISCREPANCIAS DE INTERPOLACAO EM X,Y,Z',//)
WRITE(3,120)(DELX(I),DELY(I),DELZ(I),I=1,NNM1)
120 FORMAT(3F8,2)
CALL EXIT
END

```

TE3.FOR FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```
      SUBROUTINE DIVISI(N,S,X,Y,Z,SC,XNOD,YNOD,ZNOD,  
      1SI,XI,YI,ZI,NN)  
      IMPLICIT REAL*8(A-H,O-Z)  
      DIMENSION X(30),Y(30),Z(30),S(30),SC(15),SI(15),  
      1XNOD(15),YNOD(15),ZNOD(15),XI(15),YI(15),ZI(15)  
      J=0  
      I=0  
10     CONTINUE  
      I=I+1  
      IF(I.GT.N)GO TO 20  
      J=J+1  
      SC(J)=S(I)  
      XNOD(J)=X(I)  
      YNOD(J)=Y(I)  
      ZNOD(J)=Z(I)  
      I=I+1  
      IF(I.GT.N)GO TO 20  
      SI(J)=S(I)  
      XI(J)=X(I)  
      YI(J)=Y(I)  
      ZI(J)=Z(I)  
      GO TO 10  
20     CONTINUE  
      NN=J  
      RETURN  
      END
```

TE3.FOR FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```

SUBROUTINE SPCDEF(N,XN,FN,S,INDEX)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XN(N),FN(N),S(N),INDEX(N),
1RHO(100),TAU(100)
NM1=N-1
DO 1 I=1,N
INDEX(I)=I
DO 3 I=1,NM1
IP1=I+1
DO 2 J=IP1,N
II=INDEX(I)
IJ=INDEX(J)
IF(XN(II).LE.XN(IJ))GO TO 2
ITEMP=INDEX(I)
INDEX(I)=INDEX(J)
INDEX(J)=ITEMP
2CONTINUE
3CONTINUE
NM2=N-2
RHO(2)=0.DO
TAU(2)=0.DO
DO 4 I=2,NM1
IIM1=INDEX(I-1)
II=INDEX(I)
IIP1=INDEX(I+1)
HIM1=XN(II)-XN(IIM1)
HI=XN(IIP1)-XN(II)
TEMP=(HIM1/HI)*(RHO(I)+2.DO)+2.DO
RHO(I+1)=-1.DO/TEMP
4D=6.DO*((FN(IIP1)-FN(II))/HI-(FN(II)-FN(IIM1))/HIM1)/HI
TAU(I+1)=(D-HIM1*TAU(I)/HI)/TEMP
S(1)=0.DO
S(N)=0.DO
DO 5 I=1,NM2
IB=N-I
5S(IB)=RHO(IB+1)*S(IB+1)+TAU(IB+1)
RETURN
END

```

TE3.FOR FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```

FUNCTION SPLINE(N,XN,FN,S,INDEX,X)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XN(N),FN(N),S(N),INDEX(N)
I1=INDEX(1)
IF(X,GE,XN(I1))GO TO 1
I2=INDEX(2)
H1=XN(I2)-XN(I1)
SPLINE=FN(I1)+(X-XN(I1))*((FN(I2)-FN(I1))/H1-H1*S(2)/6.DO)
RETURN
1  IN=INDEX(N)
   IF(X,LE,XN(IN))GO TO 2
   INM1=INDEX(N-1)
   HNM1=XN(IN)-XN(INM1)
   SPLINE=FN(IN)+(X-XN(IN))*((FN(IN)-FN(INM1))/HNM1+
1HNM1*S(N-1)/6.DO)
   RETURN
2  DO 3 I=2,N
   II=INDEX(I)
   IF(X,LE,XN(II))GO TO 4
3  CONTINUE
4  L=I-1
   IL=INDEX(L)
   ILP1=INDEX(L+1)
   A=XN(ILP1)-X
   B=X-XN(IL)
   HL=XN(ILP1)-XN(IL)
   SPLINE=A*S(L)*(A**2/HL-HL)/6.DO+B*S(L+1)*(B**2/HL-HL)/
16.DO+(A*FN(IL)+B*FN(ILP1))/HL
   RETURN
END

```

## NOME DO PROGRAMA:

POINT. FOR

## AUTOR (ES):

FRANCISCO C.B. DE SOUZA

## COMENTÁRIOS:

O programa calcula os parâmetros de orientação absoluta de um modelo estereoscópico utilizando pontos de controle.

As coordenadas do modelo são geradas internamente.

Precisão Simples.

## OBSERVAÇÃO:

Exemplo de arquivo de dados:

			Nº de pontos
5			↙
380.	400.	200.	
100.	100.	200.	
50.	50.	250.	
70.	20.	300.	
90.	1000.	400.	
↑	↑	↑	
X	Y	Z	

```

C      PROGRAMA PARA ORIENTACAO ABSOLUTA
C      ANALITICA UTILIZANDO PONTOS DE CONTROLE
      DIMENSION XCB(10),YCB(10),ZCB(10),XB(10),YB(10),ZB(10),
      1F(60,1),R(3,3),A(30,7),B(30,60),W(30,1),BT(60,30),XM(30,30),
      1TB(30),ATMA(7,7),ATMW(7,1),C(7),RX(7,1),RXC(7,1),AX(30,1),
      1AXW(30,1),AK(30,1),AKT(1,30),UTPV(1,1),XMV(7,7),P1(60,60),
      1W1(30,1),W2(30,1),V(60,1),DIF(60,1),PBT(60,30),
      1XCB1(10),YCB1(10),ZCB1(10),XB1(10),YB1(10),ZB1(10),
      1R1(3,3),RT1(3,3),XI(10),YI(10),ZI(10),XX(10),YY(10),ZZ(10)
      DATA E,X0,Y0,Z0/0.1,50.,300.,100./
      DATA E1,X01,Y01,Z01/9.,45.,340.,94./
      DATA DPC,DPM/0.15,0.015/
      DATA IK,MK,SK/45,0,0./
      DATA IK1,MK1,SK1/38,7,45.4/
      DATA IF,MF,SF/0,0,0./
      DATA IF1,MF1,SF1/2,3,4./
      DATA IW,MW,SW/0,0,0./
      DATA IW1,MW1,SW1/3,1,2.5/
      CALL GMSR(IK,MK,SK,XK)
      CALL GMSR(IF,MF,SF,XF)
      CALL GMSR(IW,MW,SW,XW)
      CALL TRIGO(XK,XF,XW,SK,CK,SF,CF,SW,CW)
      CALL XKFW(SK,CK,SF,CF,SW,CW,R1)
      CALL TRANSP(R1,RT1,3,3)
      READ(2,5)NF
5      FORMAT(I)
      DO 10 I=1,NF
10     READ(2,11)XCB(I),YCB(I),ZCB(I)
11     FORMAT(3G)
      CALL HOMOL(NF,XCB,YCB,ZCB,E,X0,Y0,Z0,RT1,XB,YB,ZB)
      CALL RANDOM(NF,XCB,YCB,ZCB,DPC,XCB,YCB,ZCB)
      CALL RANDOM(NF,XB,YB,ZB,DPM,XB,YB,ZB)
      CALL GMSR(IK1,MK1,SK1,XK1)
      CALL GMSR(IF1,MF1,SF1,XF1)
      CALL GMSR(IW1,MW1,SW1,XW1)
      VA1=DPM**2
      VA2=DPC**2
      IC1=3*NF
      IC2=6*NF
      IC3=(6*NF)-5
      DO 12 I=1,IC3,6
12     F(I,1)=VA1
      F(I+1,1)=VA1
      F(I+2,1)=VA1
      F(I+3,1)=VA2
      F(I+4,1)=VA2
      F(I+5,1)=VA2
100    CONTINUE
      DO 100 I=1,IC2
      DIF(I,1)=0.
100    CONTINUE
      DO 200 I=1,NF
      XCB1(I)=XCB(I)
      YCB1(I)=YCB(I)
      ZCB1(I)=ZCB(I)
      XB1(I)=XB(I)

```

POINT.FOR

FORTRAN V.6(1144)

11-OCT-82

14:09

PAGE 1-1

```

        YB1(I)=YB(I)
        ZB1(I)=ZB(I)
200     CONTINUE
        DO 220 I=1,IC2
        DO 220 J=1,IC2
        P1(I,J)=0.
220     CONTINUE
        DO 230 I=1,IC2
        P1(I,I)=P(I,1)
230     CONTINUE
C
C     INICIO DO LOOP ITERATIVO
C
        DO 14 K=1,4
        CALL TRIGO(XK1,XF1,XW1,SK,CK,SF,CF,SW,CW)
        CALL XKFW(SK,CK,SF,CF,SW,CW,R)
        CALL GERA(NP,IC1,XB,YB,ZB,R,SW,CW,E1,A)
        CALL GERAB(NP,IC1,IC2,E1,R,B)
        CALL AB(B,DIF,W2,IC1,IC2,1)
        CALL VW(NP,IC1,XCB,YCB,ZCB,XB,YB,ZB,R,E1,X01,Y01,Z01,W1)
        DO 250 I=1,IC1
        W(I,1)=W1(I,1)+W2(I,1)
250     CONTINUE
        CALL TRANSF(B,BT,IC1,IC2)
        CALL BDBT(IC2,IC1,BT,P,XM)
        CALL VERSOL(XM,TR,IC1)
        CALL ATPAU(IC1,7,A,XM,W,ATMA,ATMW)
        CALL VERSOL(ATMA,C,7)
        CALL AB(ATMA,ATMW,RX,7,7,1)
        CALL EPM(RX,RXC,7,1,-1.)
        E1=E1+RXC(1,1)
        XK1=XK1+RXC(2,1)
        XF1=XF1+RXC(3,1)
        XW1=XW1+RXC(4,1)
        X01=X01+RXC(5,1)
        Y01=Y01+RXC(6,1)
        Z01=Z01+RXC(7,1)
        WRITE(3,13)E1,XK1,XF1,XW1,X01,Y01,Z01
13     FORMAT(///30X,'FATOR DE ESCALA = ',F9.4,///30X,
        1'KAPPA = ',F11.8,///30X,'FI = ',F11.8,///30X,'OMEGA = ',
        1F11.8,///30X,'X0 = ',F9.4,///30X,'Y0 = ',F9.4,///30X,'Z0 = ',
        1F9.4)
        CALL AB(A,RXC,AX,IC1,7,1)
        CALL SOMAT(AX,W,AXW,IC1,1)
        CALL AB(XM,AXW,AK,IC1,IC1,1)
        CALL EPM(AK,AK,IC1,1,-1.)
        CALL AB(P1,BT,FRT,IC2,IC2,IC1)
        CALL AB(FRT,AK,V,IC2,IC1,1)
        DO 300 I=1,NP
        XB(I)=XB1(I)+V(6*I-5,1)
        YB(I)=YB1(I)+V(6*I-4,1)
        ZB(I)=ZB1(I)+V(6*I-3,1)
        XCB(I)=XCB1(I)+V(6*I-2,1)
        YCB(I)=YCB1(I)+V(6*I-1,1)
        ZCB(I)=ZCB1(I)+V(6*I,1)
300     CONTINUE

```

POINT.FOR            FORTRAN V.6(1144)            11-OCT-82            14:09            PAGE 1-2

```

DO 400 I=1,NP
DIF(6*I-5,1)=XB1(I)-XB(I)
DIF(6*I-4,1)=YB1(I)-YB(I)
DIF(6*I-3,1)=ZB1(I)-ZB(I)
DIF(6*I-2,1)=XCB1(I)-XCB(I)
DIF(6*I-1,1)=YCB1(I)-YCB(I)
DIF(6*I,1)=ZCB1(I)-ZCB(I)
400 CONTINUE
C
C   FIM DO LOOP ITERATIVO
C
14 CONTINUE
CALL AB(A,RXC,AX,IC1,7,1)
CALL SOMAT(AX,W,AXW,IC1,1)
CALL AB(XM,AXW,AK,IC1,IC1,1)
CALL EPM(AK,AK,IC1,1,-1.)
CALL TRANSP(AK,AKT,IC1,1)
CALL AB(AKT,W,UTPV,1,IC1,1)
CALL EPM(UTPV,UTPV,1,1,-1.)
SIGMA=(UTPV(1,1))/(IC1-7)
WRITE(3,15)SIGMA
15 FORMAT(//30X,'SIGMA A POSTERIORI = ',F7.4)
CALL EPM(ATMA,XMV,7,7,SIGMA)
DO 17 I=1,7
XMV(I,I)=XMV(I,I)**(0.5)
17 CONTINUE
WRITE(3,18)XMV(1,1)
18 FORMAT(//30X,'DP/ESCALA = ',F7.4)
WRITE(3,19)XMV(2,2)
19 FORMAT(/30X,'DP/KAPPA = ',F7.4)
WRITE(3,20)XMV(3,3)
20 FORMAT(/30X,'DP/FI = ',F7.4)
WRITE(3,21)XMV(4,4)
21 FORMAT(/30X,'DP/OMEGA = ',F7.4)
WRITE(3,22)XMV(5,5)
22 FORMAT(/30X,'DP/XO = ',F7.4)
WRITE(3,23)XMV(6,6)
23 FORMAT(/30X,'DP/YO = ',F7.4)
WRITE(3,24)XMV(7,7)
24 FORMAT(/30X,'DP/ZO = ',F7.4)
CALL EXIT
END

```

POINT.FOR

FORTRAN V.6(1144)

11-OCT-82

14:09

PAGE 1

```

SUBROUTINE HOMOL(NP,XI,YI,ZI,E,XO,YO,ZO,RT,XX,YY,ZZ)
DIMENSION XI(NP),YI(NP),ZI(NP),RT(3,3),
1XX(NP),YY(NP),ZZ(NP)
DO 10 I=1,NP
XX(I)=E*(RT(1,1)*(XI(I)-XO)+RT(1,2)*(YI(I)-YO)+
1RT(1,3)*(ZI(I)-ZO))
YY(I)=E*(RT(2,1)*(XI(I)-XO)+RT(2,2)*(YI(I)-YO)+
1RT(2,3)*(ZI(I)-ZO))
ZZ(I)=E*(RT(3,1)*(XI(I)-XO)+RT(3,2)*(YI(I)-YO)+
1RT(3,3)*(ZI(I)-ZO))
10 CONTINUE
RETURN
END
```

## NOME DO PROGRAMA:

CURVA . FOR

## AUTOR (ES):

FRANCISCO C.B. DE SOUZA

## COMENTÁRIOS:

Precisão Simples.

O Programa faz a orientação absoluta de um modelo estereoscópico utilizando feições curvas como controle. As coordenadas do modelo são geradas inter<sub>u</sub>namente.

## OBSERVAÇÃO:

3 ← número de feições (NF)

4 ← número de pontos na feição 1 (NPF)

FEIÇÃO 1

1	—	—	—
2	—	—	—
3	—	—	—
4	—	—	—

4 ← número de pontos na feição 2 (NPF)

FEIÇÃO 2

1	—	—	—
2	—	—	—
3	—	—	—
4	—	—	—

4 ← número de pontos na feição 3 (NPF)

FEIÇÃO 3

1	—	—	—
2	—	—	—
3	—	—	—
4	—	—	—
↑	↑	↑	↑
NP	X	Y	Z

NP = N<sup>o</sup> dos pontos em cada feição

```

C      PROGRAMA PARA ORIENTACAO ABSOLUTA
C      USANDO FEICDES CURVAS COMO CONTROLE
      DIMENSION R(3,3),RT(3,3),XX(15,10),YY(15,10),AUX6(15),
      1ZZ(15,10),XXX(15,10),YYY(15,10),ZZZ(15,10),
      1R1(3,3),A1(10),A2(10),A3(10),A4(10),A5(10),A6(10),
      1TA(25,3,10),X(25,10),Y(25,10),Z(25,10),XNOD(15,10),
      1YNOD(15,10),ZNOD(15,10),XI(15,10),YI(15,10),ZI(15,10),
      1SC(15,10),S(25,10),SX(15),SY(15),SZ(15),INDE1(30),INDE2(30),
      1INDE3(30),AUX1(15),AUX2(15),AUX3(15),AUX4(15),AUX5(15),
      1AUX7(15),RHO(100),TAU(100),XCB(10),YCB(10),ZCB(10),XB(10),
      1YB(10),ZB(10),P(60,1),A(30,7),B(30,60),W(30,1),BT(60,30),
      1XM(30,30),TB(30),ATMA(7,7),ATMW(7,1),C(7),RX(7,1),RXC(7,1),
      1AX(30,1),AXW(30,1),AK(30,1),AKT(1,30),VTPV(1,1),XMV(7,7),
      1P1(60,60),W1(30,1),W2(30,1),V(60,1),DIF(60,1),FBT(60,30),
      1XCB1(10),YCB1(10),ZCB1(10),XB1(10),YB1(10),ZB1(10),
      1XU(25,10),YU(25,10),ZU(25,10),SI(15,10)
      DATA E,X0,Y0,Z0/0.1,3500.,2000.,700./
      DATA E1,X01,Y01,Z01/15.,1750.,1000.,1050./
      DATA DPC,DFM/1.5,0.015/
      DATA IK,MK,SK/50,0,0./
      DATA IK1,MK1,SK1/80,0,0./
      DATA IF,MF,SF/0,0,0./
      DATA IF1,MF1,SF1/0,0,0./
      DATA IW,MW,SW/0,0,0./
      DATA IW1,MW1,SW1/0,0,0./
      CALL GMSR(IK,MK,SK,XK)
      CALL GMSR(IK1,MK1,SK1,XK1)
      CALL GMSR(IF,MF,SF,XF)
      CALL GMSR(IF1,MF1,SF1,XF1)
      CALL GMSR(IW,MW,SW,XW)
      CALL GMSR(IW1,MW1,SW1,XW1)
      CALL TRIGO(XK,XF,XW,SK,CK,SF,CF,SW,CW)
      CALL XKFW(SK,CK,SF,CF,SW,CW,R1)
      CALL TRANSP(R1,RT,3,3)
      CALL FACE(E,X0,Y0,Z0,RT,XX,YY,ZZ,XXX,YYY,ZZZ,NF)
      DO 10 I=1,NF
      A1(I)=XXX(7,I)
      A2(I)=YYY(7,I)
      A3(I)=ZZZ(7,I)
      A4(I)=XX(7,I)
      A5(I)=YY(7,I)
      A6(I)=ZZ(7,I)
10     CONTINUE
      CALL RANDOM(NF,A1,A2,A3,DPC,A1,A2,A3)
      CALL RANDOM(NF,A4,A5,A6,DFM,A4,A5,A6)
      DO 555 I=1,NF
      WRITE(3,666)A1(I),A2(I),A3(I),A4(I),A5(I),A6(I)
666    FORMAT(6(2X,F13.3))
555    CONTINUE
      IC1=3*NF
      IC2=6*NF
      IC3=(6*NF)-5
      WRITE(3,88)IC1,IC2,IC3
88     FORMAT(3G)
      CALL OAB(NF,IC1,IC2,IC3,E1,X01,Y01,Z01,XK1,XF1,XW1,
      1A1,A2,A3,A4,A5,A6,DFM,DPC)
      CALL EXIT
      END

```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```

SUBROUTINE MAGI(NF,XI,YI,ZI,E,X0,Y0,Z0,RT,NN,K,XX,YY,ZZ)
DIMENSION XI(15,10),YI(15,10),ZI(15,10),RT(3,3),
1XX(15,10),YY(15,10),ZZ(15,10)
NNM1=NN-1
DO 10 I=1,NNM1
XX(I,K)=E*(RT(1,1)*(XI(I,K)-X0)+RT(1,2)*(YI(I,K)-Y0)+
1RT(1,3)*(ZI(I,K)-Z0))
YY(I,K)=E*(RT(2,1)*(XI(I,K)-X0)+RT(2,2)*(YI(I,K)-Y0)+
1RT(2,3)*(ZI(I,K)-Z0))
ZZ(I,K)=E*(RT(3,1)*(XI(I,K)-X0)+RT(3,2)*(YI(I,K)-Y0)+
1RT(3,3)*(ZI(I,K)-Z0))
10 CONTINUE
RETURN
END
```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```

SUBROUTINE FACE(E,X0,Y0,Z0,RT,XX,YY,ZZ,XXX,YYY,ZZZ,NF)
DIMENSION TA(25,3,10),X(25,10),Y(25,10),Z(25,10),
1XNOD(15,10),YNOD(15,10),ZNOD(15,10),XI(15,10),RT(3,3),
1YI(15,10),ZI(15,10),XX(15,10),YY(15,10),ZZ(15,10),S(25,10),
1XXX(15,10),YYY(15,10),ZZZ(15,10),SC(15,10),SX(15),SY(15),
1SZ(15),INDE1(30),AUX1(15),AUX2(15),AUX3(15),AUX4(15),
1AUX5(15),AUX6(15),AUX7(15),INDE2(30),INDE3(30),SI(15,10),
1XU(25,10),YU(25,10),ZU(25,10),RHO(100),TAU(100)
10 READ(2,10)NF
FORMAT(G)
DO 30 K=1,NF
READ(2,10)NPF
DO 20 I=1,NPF
11 READ(2,11)NF,(TA(I,J,K),J=1,3)
FORMAT(I,3G)
L=K
20 CONTINUE
CALL ORGAN(NF,NPF,L,TA,X,Y,Z)
CALL DISTA(X,Y,Z,NF,NPF,L,S)
CALL SEPAR(NF,NPF,S,X,Y,Z,XNOD,YNOD,ZNOD,XI,YI,ZI,
1NN,SC,SI,L)
CALL MAGI(NF,XI,YI,ZI,E,X0,Y0,Z0,RT,NN,L,XX,YY,ZZ)
DO 21 I=1,NN
AUX1(I)=XNOD(I,L)
AUX2(I)=YNOD(I,L)
AUX3(I)=ZNOD(I,L)
AUX4(I)=SC(I,L)
21 CONTINUE
CALL SPCDEF(NN,AUX4,AUX1,SX,INDE1)
CALL SPCDEF(NN,AUX4,AUX2,SY,INDE2)
CALL SPCDEF(NN,AUX4,AUX3,SZ,INDE3)
NNM1=NN-1
DO 22 I=1,NNM1
SPAR=SI(I,L)
XXX(I,L)=SPLINE(NN,AUX4,AUX1,SX,INDE1,SPAR)
YYY(I,L)=SPLINE(NN,AUX4,AUX2,SY,INDE2,SPAR)
ZZZ(I,L)=SPLINE(NN,AUX4,AUX3,SZ,INDE3,SPAR)
22 CONTINUE
30 CONTINUE
RETURN
END

```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```
      SUBROUTINE ORGAN(NF,NPF,K,TA,X,Y,Z)
      DIMENSION TA(NPF,3,NF),X(NPF,NF),Y(NPF,NF),Z(NPF,NF)
      DO 10 I=1,NPF
      X(I,K)=TA(I,1,K)
      Y(I,K)=TA(I,2,K)
      Z(I,K)=TA(I,3,K)
10    CONTINUE
      RETURN
      END
```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```
      SUBROUTINE SEPAR(NF,N,S,X,Y,Z,XNOD,YNOD,ZNOD,XI,  
1      1YI,ZI,NN,SC,SI,K)  
      DIMENSION X(N,NF),Y(N,NF),Z(N,NF),  
1      1XNOD(15,NF),YNOD(15,NF),ZNOD(15,NF),  
1      1XI(15,NF),YI(15,NF),ZI(15,NF),S(N,NF),SC(15,NF),SI(15,NF)  
      J=0  
      I=0  
10     CONTINUE  
      I=I+1  
      IF(I.GT.N)GO TO 20  
      J=J+1  
      SC(J,K)=S(I,K)  
      XNOD(J,K)=X(I,K)  
      YNOD(J,K)=Y(I,K)  
      ZNOD(J,K)=Z(I,K)  
      I=I+1  
      IF(I.GT.N)GO TO 20  
      SI(J,K)=S(I,K)  
      XI(J,K)=X(I,K)  
      YI(J,K)=Y(I,K)  
      ZI(J,K)=Z(I,K)  
      GO TO 10  
20     CONTINUE  
      NN=J  
      RETURN  
      END
```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1

```
      SUBROUTINE DISTA(XU,YU,ZU,NF,NN,K,S)
      DIMENSION XU(NN,NF),YU(NN,NF),ZU(NN,NF),
      1S(NN,NF)
      COMP(X1,Y1,Z1,X2,Y2,Z2)=((X2-X1)**2+(Y2-Y1)**2+
      1(Z2-Z1)**2)**0.5
      NNM1=NN-1
      I=1
      S(I,K)=0.0
5     CONTINUE
      X1=XU(I,K)
      Y1=YU(I,K)
      Z1=ZU(I,K)
      X2=XU(I+1,K)
      Y2=YU(I+1,K)
      Z2=ZU(I+1,K)
      D=COMP(X1,Y1,Z1,X2,Y2,Z2)
      S(I,K)=S(I,K)+D
      I=I+1
      S(I,K)=S(I-1,K)
      IF(I.EQ.NNM1)GO TO 10
      GO TO 5
10    CONTINUE
      RETURN
      END
```

```

SUBROUTINE OAB(NF,IC1,IC2,IC3,E,XO,YO,ZO,XK,XF,XW,XCB,YCB,ZCB,
1XB,YB,ZB,DPF,DFC)
DIMENSION XCB(10),YCB(10),ZCB(10),XB(10),YB(10),ZB(10),
1P(60,1),R(3,3),A(30,7),B(30,60),W(30,1),BT(60,30),
1TB(30),ATMA(7,7),ATMW(7,1),C(7),RX(7,1),RXC(7,1),AX(30,1),
1AXW(30,1),AK(30,1),AKT(1,30),UTPV(1,1),XMV(7,7),
1W1(30,1),W2(30,1),V(60,1),DIF(60,1),PET(60,30),
1XCB1(10),YCB1(10),ZCB1(10),XB1(10),YB1(10),ZB1(10),
1P1(60,60),XM(30,30)
VA1=DPF**2
VA2=DFC**2
DO 12 I=1,IC3,6
P(I,1)=VA1
P(I+1,1)=VA1
P(I+2,1)=VA1
P(I+3,1)=VA2
P(I+4,1)=VA2
P(I+5,1)=VA2
1_ CONTINUE
DO 100 I=1,IC2
DIF(I,1)=0.
100 CONTINUE
DO 200 I=1,NF
XCB1(I)=XCB(I)
YCB1(I)=YCB(I)
ZCB1(I)=ZCB(I)
XB1(I)=XB(I)
YB1(I)=YB(I)
ZB1(I)=ZB(I)
200 CONTINUE
DO 220 I=1,IC2
DO 220 J=1,IC2
P1(I,J)=0.
220 CONTINUE
DO 230 I=1,IC2
P1(I,I)=P(I,1)
230 CONTINUE
C
C INICIO DO LOOP ITERATIVO
C
DO 14 K=1,4
CALL TRIGO(XK,XF,XW,SK,CK,SF,CF,SW,CW)
CALL XKFW(SK,CK,SF,CF,SW,CW,R)
CALL GERA(NF,IC1,XB,YB,ZB,R,SW,CW,E,A)
CALL GERAB(NF,IC1,IC2,E,R,B)
CALL AB(B,DIF,W2,IC1,IC2,1)
CALL VW(NF,IC1,XCB,YCB,ZCB,XB,YB,ZB,R,E,XO,YO,ZO,W1)
DO 250 I=1,IC1
W(I,1)=W1(I,1)+W2(I,1)
250 CONTINUE
CALL TRANSP(B,BT,IC1,IC2)
CALL BDBT(IC2,IC1,BT,P,XM)
CALL VERSOL(XM,TB,IC1)
CALL ATPAU(IC1,7,A,XM,W,ATMA,ATMW)
CALL VERSOL(ATMA,C,7)
CALL AB(ATMA,ATMW,RX,7,7,1)

```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1-3

```

CALL EPM(RX,RXC,7,1,-1.)
E=E+RXC(1,1)
XK=XK+RXC(2,1)
XF=XF+RXC(3,1)
XW=XW+RXC(4,1)
XO=XO+RXC(5,1)
YO=YO+RXC(6,1)
ZO=ZO+RXC(7,1)
WRITE(3,13)E,XK,XF,XW,XO,YO,ZO
13  FORMAT(///30X,'FATOR DE ESCALA = ',F9.4,///30X,
1'KAPPA = ',F11.8,///30X,'FI = ',F11.8,///30X,'OMEGA = ',
1F11.8,///30X,'XO = ',F9.4,///30X,'YO = ',F9.4,///30X,'ZO = ',
1F9.4)
CALL AB(A,RXC,AX,IC1,7,1)
CALL SOMAT(AX,W,AXW,IC1,1)
CALL AB(XM,AXW,AK,IC1,IC1,1)
CALL EPM(AK,AK,IC1,1,-1.)
CALL AB(F1,BT,PBT,IC2,IC2,IC1)
CALL AB(PBT,AK,V,IC2,IC1,1)
DO 300 I=1,NF
XB(I)=XB1(I)+V(6*I-5,1)
YB(I)=YB1(I)+V(6*I-4,1)
ZB(I)=ZB1(I)+V(6*I-3,1)
XCB(I)=XCB1(I)+V(6*I-2,1)
YCB(I)=YCB1(I)+V(6*I-1,1)
ZCB(I)=ZCB1(I)+V(6*I,1)
300 CONTINUE
DO 400 I=1,NF
DIF(6*I-5,1)=XB1(I)-XB(I)
DIF(6*I-4,1)=YB1(I)-YB(I)
DIF(6*I-3,1)=ZB1(I)-ZB(I)
DIF(6*I-2,1)=XCB1(I)-XCB(I)
DIF(6*I-1,1)=YCB1(I)-YCB(I)
DIF(6*I,1)=ZCB1(I)-ZCB(I)
400 CONTINUE
C
C FIM DO LOOP ITERATIVO
C
14 CONTINUE
CALL AB(A,RXC,AX,IC1,7,1)
CALL SOMAT(AX,W,AXW,IC1,1)
CALL AB(XM,AXW,AK,IC1,IC1,1)
CALL EPM(AK,AK,IC1,1,-1.)
CALL TRANSP(AK,AKT,IC1,1)
CALL AB(AKT,W,VTFV,1,IC1,1)
CALL EPM(VTFV,VTFV,1,1,-1.)
SIGMA=(VTFV(1,1))/(IC1-7)
WRITE(3,15)SIGMA
15  FORMAT(///30X,'SIGMA A POSTERIORI = ',F7.4)
CALL EPM(ATMA,XMV,7,7,SIGMA)
DO 17 I=1,7
XMV(I,I)=XMV(I,I)**(0.5)
17  CONTINUE
WRITE(3,18)XMV(1,1)
18  FORMAT(///30X,'IP/ESCALA = ',F7.4)
WRITE(3,19)XMV(2,2)

```

CURVA.FOR

FORTRAN V.6(1144)

11-OCT-82

14:07

PAGE 1-2

```
19     FORMAT(/30X,'DP/KAPPA = ',F7.4)
      WRITE(3,20)XMV(3,3)
20     FORMAT(/30X,'DP/FI    = ',F7.4)
      WRITE(3,21)XMV(4,4)
21     FORMAT(/30X,'DP/OMEGA = ',F7.4)
      WRITE(3,22)XMV(5,5)
22     FORMAT(/30X,'DP/X0    = ',F7.4)
      WRITE(3,23)XMV(6,6)
23     FORMAT(/30X,'DP/Y0    = ',F7.4)
      WRITE(3,24)XMV(7,7)
24     FORMAT(/30X,'DP/Z0    = ',F7.4)
      RETURN
      END
```

SUB-ROTINAS COMUNS AOS PROGRAMAS

POINT . FOR E CURVA . FOR

```

SUBROUTINE GERA(NP,IC1,XB,YB,ZB,R,SW,CW,E,A)
C  ESTA SUBROTINA FORMA A MATRIZ (A)
  DIMENSION R(3,3),XB(NP),YB(NP),ZB(NP),A(IC1,7)
  DO 10 I=1,NP
    A(3*I-2,1)=R(1,1)*XB(I)+R(1,2)*YB(I)+R(1,3)*ZB(I)
    A(3*I-2,2)=E*(R(2,1)*XB(I)+R(2,2)*YB(I)+R(2,3)*ZB(I))
    A(3*I-2,3)=E*((-R(1,2)*SW+R(1,3)*CW)*XB(I)+R(1,1)*SW*YB(I)+
    1(-R(1,1)*CW)*ZB(I))
    A(3*I-2,4)=E*(-R(1,3)*YB(I)+R(1,2)*ZB(I))
    A(3*I-2,5)=1.
    A(3*I-2,6)=0.
    A(3*I-2,7)=0.
    A(3*I-1,1)=R(2,1)*XB(I)+R(2,2)*YB(I)+R(2,3)*ZB(I)
    A(3*I-1,2)=E*(-R(1,1)*XB(I)+R(1,2)*YB(I)+R(1,3)*ZB(I))
    A(3*I-1,3)=E*((-R(2,2)*SW+R(2,3)*CW)*XB(I)+R(2,1)*SW*YB(I)+
    1(-R(2,1)*CW)*ZB(I))
    A(3*I-1,4)=E*(-R(2,3)*YB(I)+R(2,2)*ZB(I))
    A(3*I-1,5)=0.
    A(3*I-1,6)=1.
    A(3*I-1,7)=0.
    A(3*I,1)=R(3,1)*XB(I)+R(3,2)*YB(I)+R(3,3)*ZB(I)
    A(3*I,2)=0.
    A(3*I,3)=E*((-R(3,2)*SW+R(3,3)*CW)*XB(I)+R(3,1)*SW*YB(I)+
    1(-R(3,1)*CW)*ZB(I))
    A(3*I,4)=E*(-R(3,3)*YB(I)+R(3,2)*ZB(I))
    A(3*I,5)=0.
    A(3*I,6)=0.
    A(3*I,7)=1.
10  CONTINUE
    RETURN
  END

```

```
C      SUBROUTINE GERAB(NF,L,M,E,R,B)
      ESTA SUBROTINA GERA A MATRIZ (B)
      DIMENSION R(3,3),B(L,M)
      DO 10 I=1,L
      DO 10 J=1,M
      B(I,J)=0.
10     CONTINUE
      DO 20 I=1,NF
      K=6*I-5
      B(3*I-2,K)=E*R(1,1)
      B(3*I-2,K+1)=E*R(1,2)
      B(3*I-2,K+2)=E*R(1,3)
      B(3*I-2,K+3)=-1.
      B(3*I-1,K)=E*R(2,1)
      B(3*I-1,K+1)=E*R(2,2)
      B(3*I-1,K+2)=E*R(2,3)
      B(3*I-1,K+4)=-1.
      B(3*I,K)=E*R(3,1)
      B(3*I,K+1)=E*R(3,2)
      B(3*I,K+2)=E*R(3,3)
      B(3*I,K+5)=-1.
20     CONTINUE
      RETURN
      END
```

```
C      SUBROUTINE VW(NP,IC1,XCB,YCB,ZCB,XB,YB,ZB,R,E,X0,Y0,Z0,W)
      SUBROTINA QUE CALCULA O VETOR (W)
      DIMENSION XCB(NP),YCB(NP),ZCB(NP),XB(NP),YB(NP),ZB(NP),
      1R(3,3),W(IC1,1)
      DO 10 I=1,NP
      W(3*I-2,1)=X0-XCB(I)+E*(XB(I)*R(1,1)+YB(I)*R(1,2)+
      1ZB(I)*R(1,3))
      W(3*I-1,1)=Y0-YCB(I)+E*(XB(I)*R(2,1)+YB(I)*R(2,2)+
      1ZB(I)*R(2,3))
      W(3*I,1)=Z0-ZCB(I)+E*(XB(I)*R(3,1)+YB(I)*R(3,2)+
      1ZB(I)*R(3,3))
10     CONTINUE
      RETURN
      END
```

## NOME DO PROGRAMA:

RETA . FOR

## AUTOR (ES):

FRANCISCO C.B.DE SOUZA

## COMENTARIOS:

Precisão Simples.

O programa faz a orientação absoluta de um modelo estereoscópico utilizando feições retas como controle. As coordenadas de modelo são geradas internamente.

## OBSERVAÇÃO:

Exemplo de arquivo de dados:

4 ← número de feições (NF)

—	]	Parâmetros da reta nas diferentes fei	
—		ções.	
—			

1	—	—	—	Feição 1
2	—	—	—	
1	—	—	—	Feição 2
2	—	—	—	
1	—	—	—	Feição 3
2	—	—	—	
1	—	—	—	Feição 4
2	—	—	—	
↑	↑	↑	↑	
NP	X	Y	Z	

RETA.FOR

FORTRAN V.6(1144)

22-FEB-83

15:00

PAGE 1

```

C      PROGRAMA PARA ORIENTACAO ABSOLUTA
C      ANALITICA USANDO FEICDES RETAS
C      COMO CONTROLE
      DIMENSION A(30,17),B(30,90),P(90,1),XB(10),YB(10),
      1ZB(10),XC1(10),YC1(10),ZC1(10),XC2(10),YC2(10),
      1ZC2(10),R(3,3),W(30,1),R1(3,3),RT(3,3),TA(2,3,10),
      1X(2,10),Y(2,10),Z(2,10),XA(10),YA(10),ZA(10),XX(10),
      1YY(10),ZZ(10),DIF(90,1),P1(90,90),XB1(10),YB1(10),
      1ZB1(10),XCB1(10),YCB1(10),ZCB1(10),XCB2(10),YCB2(10),
      1ZCB2(10),R2(3,3),W2(30,1),W1(30,1),BT(90,30),XM(30,30),
      1TB(30),ATMA(17,17),ATMW(17,1),C(17),RX(17,1),AX(30,1),
      1AXW(30,1),AK(30,1),PET(90,30),V(90,1),AKT(1,30),
      1VTPV(1,1),XMV(17,17),RXC(17,1),T(10),T1(10)
      DATA E,X0,Y0,Z0/0.1,3500.,2000.,700./
      DATA E1,X01,Y01,Z01/15.,1750.,1000.,1050./
      DATA DPC,DFM/1.5,0.015/
      DATA IK,MK,SK/50,0,0./
      DATA IK1,MK1,SK1/80,0,0./
      DATA IF,MF,SF/0,0,0./
      DATA IF1,MF1,SF1/0,0,0./
      DATA IW,MW,SW/0,0,0./
      DATA IW1,MW1,SW1/0,0,0./
      CALL GMSR(IK,MK,SK,XK)
      CALL GMSR(IK1,MK1,SK1,XK1)
      CALL GMSR(IF,MF,SF,XF)
      CALL GMSR(IF1,MF1,SF1,XF1)
      CALL GMSR(IW,MW,SW,XW)
      CALL GMSR(IW1,MW1,SW1,XW1)
      CALL TRIGO(XK,XF,XW,SEK,CK,SEF,CF,SEW,CW)
      CALL XKFW(SEK,CK,SEF,CF,SEW,CW,R1)
      CALL TRANSP(R1,RT,3,3)
      READ(2,5)NF
5      FORMAT(G)
      DO 8 I=1,NF
      READ(2,10)T(I)
8      CONTINUE
10     FORMAT(G)
      NPF=2
      DO 30 K=1,NF
      DO 20 I=1,NPF
      READ(2,11)(TA(I,J,K),J=1,3)
11     FORMAT(3G)
20     CONTINUE
      L=K
      CALL ORGAN(NF,NPF,L,TA,X,Y,Z)
      XA(L)=X(1,L)+T(L)*(X(2,L)-X(1,L))
      YA(L)=Y(1,L)+T(L)*(Y(2,L)-Y(1,L))
      ZA(L)=Z(1,L)+T(L)*(Z(2,L)-Z(1,L))
      XC1(L)=X(1,L);XC2(L)=X(2,L)
      YC1(L)=Y(1,L);YC2(L)=Y(2,L)
      ZC1(L)=Z(1,L);ZC2(L)=Z(2,L)
      G1=XA(L)-X0
      G2=YA(L)-Y0
      G3=ZA(L)-Z0
      XX(L)=E*(RT(1,1)*G1+RT(1,2)*G2+RT(1,3)*G3)
      YY(L)=E*(RT(2,1)*G1+RT(2,2)*G2+RT(2,3)*G3)

```

RETA.FOR

FORTRAN V.6(1144)

22-FEB-83

15:00

PAGE 1-1

```

30      ZZ(L)=E*(RT(3,1)*G1+RT(3,2)*G2+RT(3,3)*G3)
      CONTINUE
      CALL RANDOM(NF,XX,YY,ZZ,DFM,XX,YY,ZZ)
      CALL RANDOM(NF,XC1,YC1,ZC1,DFC,XC1,YC1,ZC1)
      CALL RANDOM(NF,XC2,YC2,ZC2,DFC,XC2,YC2,ZC2)
      VA1=DFM**2
      VA2=DFC**2
      K01=9*NF
      K02=3*NF
      K03=(9*NF)-8
      K04=NF+7
      DO 40 I=1,K03,9
      P(I,1)=VA1
      P(I+1,1)=VA1
      P(I+2,1)=VA1
      P(I+3,1)=VA2
      P(I+4,1)=VA2
      P(I+5,1)=VA2
      P(I+6,1)=VA2
      P(I+7,1)=VA2
      P(I+8,1)=VA2
40     CONTINUE
      DO 50 I=1,K01
      DIF(I,1)=0.
50     CONTINUE
      DO 60 I=1,K01
      DO 60 J=1,K01
      P1(I,J)=0.
60     CONTINUE
      DO 70 I=1,K01
      P1(I,I)=P(I,1)
70     CONTINUE
      DO 80 I=1,NF
      XB1(I)=XX(I)
      YB1(I)=YY(I)
      ZB1(I)=ZZ(I)
      XCB1(I)=XC1(I)
      YCB1(I)=YC1(I)
      ZCB1(I)=ZC1(I)
      XCB2(I)=XC2(I)
      YCB2(I)=YC2(I)
      ZCB2(I)=ZC2(I)
80     CONTINUE
      DO 85 I=1,NF
      T1(I)=2.3
85     CONTINUE
      C
      C      INICIO DO LOOP ITERATIVO
      C
      DO 888 II=1,4
      CALL TRIGO(XK1,XF1,XW1,SK1,CK1,SF1,CF1,SW1,CW1)
      CALL XKFW(SK1,CK1,SF1,CF1,SW1,CW1,R2)
      CALL GERA(NF,K02,K04,XX,YY,ZZ,XC1,YC1,ZC1,XC2,YC2,ZC2,
      IE1,R2,SK1,CK1,SF1,CF1,SW1,CW1,A)
      CALL GERAR(NF,K02,K01,E1,R2,T1,B)
      CALL AB(B,DIF,W2,K02,K01,1)

```

```

CALL VW(NF,K02,XX,YY,ZZ,XC1,YC1,ZC1,XC2,YC2,ZC2,
1E1,R2,X01,Y01,Z01,T1,W1)
DO 90 I=1,K02
W(I,1)=W1(I,1)+W2(I,1)
90 CONTINUE
CALL TRANSP(B,BT,K02,K01)
CALL BDBT(K01,K02,BT,F,XM)
CALL VERSOL(XM,TB,K02)
CALL ATPAU(K02,K04,A,XM,W,ATMA,ATMW)
CALL VERSOL(ATMA,C,K04)
CALL AB(ATMA,ATMW,RX,K04,K04,1)
CALL EPM(RX,RXC,K04,1,-1.)
E1=E1+RXC(1,1)
XK1=XK1+RXC(2,1)
XF1=XF1+RXC(3,1)
XW1=XW1+RXC(4,1)
X01=X01+RXC(5,1)
Y01=Y01+RXC(6,1)
Z01=Z01+RXC(7,1)
DO 99 I=1,NF
T1(I)=RXC(7+I,1)+T1(I)
99 CONTINUE
WRITE(3,100)E1,XK1,XF1,XW1,X01,Y01,Z01
100 FORMAT(//30X,'FATOR DE ESCALA = ',F9.4,//30X,
1'KAPPA = ',F11.8,//30X,'FI = ',F11.8,//30X,'OMEGA =
1F11.8,//30X,'X0 = ',F9.4,//30X,'Y0 = ',F9.4,//30X,
1'Z0 = ',F9.4)
CALL AB(A,RXC,AX,K02,K04,1)
CALL SOMAT(AX,W,AXW,K02,1)
CALL AB(XM,AXW,AK,K02,K02,1)
CALL EPM(AK,AK,K02,1,-1.)
CALL AB(P1,BT,PBT,K01,K01,K02)
CALL AB(PBT,AK,V,K01,K02,1)
DO 110 I=1,NF
XX(I)=XB1(I)+V(9*I-8,1)
YY(I)=YB1(I)+V(9*I-7,1)
ZZ(I)=ZB1(I)+V(9*I-6,1)
XC1(I)=XCB1(I)+V(9*I-5,1)
YC1(I)=YCB1(I)+V(9*I-4,1)
ZC1(I)=ZCB1(I)+V(9*I-3,1)
XC2(I)=XCB2(I)+V(9*I-2,1)
YC2(I)=YCB2(I)+V(9*I-1,1)
ZC2(I)=ZCB2(I)+V(9*I,1)
110 CONTINUE
DO 120 I=1,NF
DIF(9*I-8,1)=XB1(I)-XX(I)
DIF(9*I-7,1)=YB1(I)-YY(I)
DIF(9*I-6,1)=ZB1(I)-ZZ(I)
DIF(9*I-5,1)=XCB1(I)-XC1(I)
DIF(9*I-4,1)=YCB1(I)-YC1(I)
DIF(9*I-3,1)=ZCB1(I)-ZC1(I)
DIF(9*I-2,1)=XCB2(I)-XC2(I)
DIF(9*I-1,1)=YCB2(I)-YC2(I)
DIF(9*I,1)=ZCB2(I)-ZC2(I)
120 CONTINUE
888 CONTINUE

```

RETA.FOR

FORTRAN V.6(1144)

22-FEB-83

15:00

PAGE 1-3

```
C
C      FIM DO LOOP ITERATIVO
C
      CALL TRANSP(AK,AKT,K02,1)
      CALL AB(AKT,W,UTPV,1,K02,1)
      CALL EPM(UTPV,UTPV,1,1,-1.)
      SIGMA=(UTPV(1,1))/(K02-K04)
130    WRITE(3,130)SIGMA
      FORMAT(//30X,'SIGMA A POSTERIORI = ',F7.4)
      CALL EPM(ATMA,XMV,K04,K04,SIGMA)
      DO 140 I=1,7
140    XMV(I,I)=XMV(I,I)**(0.5)
      CONTINUE
      WRITE(3,150)XMV(1,1),XMV(2,2),XMV(3,3),XMV(4,4),
150    1XMV(5,5),XMV(6,6),XMV(7,7)
      FORMAT(//30X,'DP/ESCALA = ',F7.4,/30X,'DP/KAPPA = ',
1F7.4,/30X,'DP/PI = ',F7.4,/30X,'DP/OMEGA = ',F7.4,
1/30X,'DP/X0 = ',F7.4,/30X,'DP/Y0 = ',F7.4,/30X,
1'DP/Z0 = ',F7.4)
      DO 180 I=1,NF
180    WRITE(3,200)T1(I)
200    CONTINUE
      FORMAT(G)
      CALL EXIT
      END
```

```

SUBROUTINE GERA(NF,K02,K04,XB,YB,ZB,XC1,YC1,ZC1,XC2,YC2,ZC2,
1E,R,SK,CK,SF,CF,SW,CW,A)
DIMENSION XB(NF),YB(NF),ZB(NF),XC1(NF),YC1(NF),ZC1(NF),
1XC2(NF),YC2(NF),ZC2(NF),R(3,3),A(K02,K04)
DO 10 I=1,NF
A(3*I-2,1)=R(1,1)*XB(I)+R(1,2)*YB(I)+R(1,3)*ZB(I)
A(3*I-2,2)=E*(XB(I)*R(2,1)+YB(I)*R(2,2)+ZB(I)*R(2,3))
A(3*I-2,3)=E*(XB(I)*(-SF*CK)+YB(I)*(SW*CF*CK)+ZB(I)*
1-CW*CF*CK)
A(3*I-2,4)=E*(R(1,2)*ZB(I)-R(1,3)*YB(I))
A(3*I-2,5)=1.
A(3*I-2,6)=0.
A(3*I-2,7)=0.
A(3*I-2,7+I)=- (XC2(I)-XC1(I))
A(3*I-1,1)=R(2,1)*XB(I)+R(2,2)*YB(I)+R(2,3)*ZB(I)
A(3*I-1,2)=-E*(XB(I)*R(1,1)+YB(I)*R(1,2)+ZB(I)*R(1,3))
A(3*I-1,3)=E*(XB(I)*(SF*SK)+YB(I)*(-SW*CF*SK)+ZB(I)*
1CW*CF*SK)
A(3*I-1,4)=E*(R(2,2)*ZB(I)-R(2,3)*YB(I))
A(3*I-1,5)=0.
A(3*I-1,6)=1.
A(3*I-1,7)=0.
A(3*I-1,7+I)=- (YC2(I)-YC1(I))
A(3*I,1)=R(3,1)*XB(I)+R(3,2)*YB(I)+R(3,3)*ZB(I)
A(3*I,2)=0.
A(3*I,3)=E*(XB(I)*CF+YB(I)*SW*SF-ZB(I)*SF*CW)
A(3*I,4)=E*(R(3,2)*ZB(I)-R(3,3)*YB(I))
A(3*I,5)=0.
A(3*I,6)=0.
A(3*I,7)=1.
A(3*I,7+I)=- (ZC2(I)-ZC1(I))
CONTINUE
RETURN
END

```

10

RETA.FOR

FORTRAN V.6(1144)

22-FEB-83

15:00

PAGE 1

```
      SUBROUTINE GERAB(NF,K02,K01,E,R,T,B)
      DIMENSION R(3,3),B(K02,K01),T(NF)
      DO 10 I=1,K02
      DO 10 J=1,K01
      B(I,J)=0.
10     CONTINUE
      DO 20 I=1,NF
      K=9*I-8
      B(3*I-2,K)=E*R(1,1)
      B(3*I-2,K+1)=E*R(1,2)
      B(3*I-2,K+2)=E*R(1,3)
      B(3*I-2,K+3)=(T(I)-1.)
      B(3*I-2,K+6)=-T(I)
      B(3*I-1,K)=E*R(2,1)
      B(3*I-1,K+1)=E*R(2,2)
      B(3*I-1,K+2)=E*R(2,3)
      B(3*I-1,K+4)=(T(I)-1.)
      B(3*I-1,K+7)=-T(I)
      B(3*I,K)=E*R(3,1)
      B(3*I,K+1)=E*R(3,2)
      B(3*I,K+2)=E*R(3,3)
      B(3*I,K+5)=(T(I)-1.)
      B(3*I,K+8)=-T(I)
20     CONTINUE
      RETURN
      END
```

RETA,FDR

FORTRAN V.6(1144)

22-FEB-83

15:00

PAGE 1

```

SUBROUTINE UW(NF,K02,XB,YB,ZB,XC1,YC1,ZC1,XC2,YC2,ZC2,
1E,R,X0,Y0,Z0,T,W)
DIMENSION XB(NF),YB(NF),ZB(NF),XC1(NF),YC1(NF),ZC1(NF),
1XC2(NF),YC2(NF),ZC2(NF),R(3,3),W(K02,1),T(NF)
DO 10 I=1,NF
W(3*I-2,1)=E*(R(1,1)*XB(I)+R(1,2)*YB(I)+R(1,3)*ZB(I))+
1X0-XC1(I)-T(I)*(XC2(I)-XC1(I))
W(3*I-1,1)=E*(R(2,1)*XB(I)+R(2,2)*YB(I)+R(2,3)*ZB(I))+
1Y0-YC1(I)-T(I)*(YC2(I)-YC1(I))
W(3*I,1)=E*(R(3,1)*XB(I)+R(3,2)*YB(I)+R(3,3)*ZB(I))+
1Z0-ZC1(I)-T(I)*(ZC2(I)-ZC1(I))
10 CONTINUE
RETURN
END
```

## ARQUIVO GERAL DE SUB-ROTINAS\*

(\*). As sub-rotinas, constantes neste arquivo, que aparecem sem o nome do autor, fazem parte do material didático distribuído aos alunos do C.P.G.C.G na disciplina de computação.

```
      SUBROUTINE RANDOM(N,X,Y,Z,SD,X1,Y1,Z1)
C      SUBROTINA QUE INTRODUZ ERROS RANDOMICOS
C      EM UM CONJUNTO DE DADOS ARMAZENADOS EM
C      FORMA VETORIAL
C      PARAMETROS
C      N=NUMERO DE ELEMENTOS DO VETOR(ORDEM)
C      X,Y,X,X1,Y1,Z1 VETORES(INPUT,OUTPUT)
C      SD=DESVIO PADRAO
      DIMENSION X(N),Y(N),Z(N),X1(N),Y1(N),Z1(N)
      IX=9513
      DO 10 I=1,N
      A=X(I)
      B=Y(I)
      C=Z(I)
      CALL GAUSS(IX,SD,A,E)
      X1(I)=E
      CALL GAUSS(IX,SD,B,F)
      Y1(I)=F
      CALL GAUSS(IX,SD,C,G)
      Z1(I)=G
10     CONTINUE
      RETURN
      END
```

```
                SUBROUTINE GMSR(IG,M,S,R)
C.....SUBROTINA PARA TRANSF. GMS EM RAJIANOS
C.....OBSERVACAO
C.....SE O ARCO FOR NEGATIVO O SINAL
C.....DEVE SER COLOCADO EM TODOS OS ARGUMENTOS
C.....NAO NULOS
                RAD=3.1415926535/180.0
                IF (IG) 50,10,70
70             R=(FLOAT(IG)+FLOAT(M)/60.00+S/3600.00)*RAD
                GO TO 100
50             R=-(ABS(FLOAT(IG))+ABS(FLOAT(M))/60.00+ABS(S)/3600.00)*RAD
                GO TO 100
10             IF (M) 11,20,51
11             GO TO 50
20             IF(S)50,51,51
51             GO TO 70
100            RETURN
                END
```

```

SUBROUTINE BDBT(N,M,B,D,BDBT)
C *****
C *****
C ** SUBROUTINE BDBT BOSCO-EDSON 25/06/81 **
C ** PROPOSITO: ESTA SUBROUTINE CALCULA O TRIPLO PRODUTO **
C ** MATRIZ B(N,M),MATRIZ D(M,M),MATRIZ BT(M,N) **
C ** CHAMADA: CALL BDBT(N,M,B,D,BDBT) **
C ** PARAMETROS: **
C ** N: NUMERO DE LINHAS DA MATRIZ B **
C ** M: NUMERO DE COLUNAS DA MATRIZ B **
C ** B: MATRIZ DE ENTRADA (IMPORTANTE: A MATRIZ **
C ** DE ENTRADA, TEM QUE SER A TRANSPOSTA DA **
C ** MATRIZ QUE SE QUEIRA MULTIPLICAR),OU SEIJA **
C ** SE DESEJA MULTIPLICAR B(N,M),DEVEMOS ENTRAR **
C ** COM B(M,N) **
C ** D: VETOR DA DIAGONAL DA MATRIZ DOS PESOS **
C ** BDBT:MATRIZ RESULTANTE DO TRIPLO PRODUTO **
C ** BDBT(N,N) **
C ** REQUISITOS: A MATRIZ DE ENTRADA 'B', TEM QUE SER A TRANS- **
C ** POSTA DA MATRIZ... **
C *****
C *****
DIMENSION B(N,M),D(N,1),BDBT(M,M)
DO 20 I=1,M
DO 20 J=1,I
S1=0.00
DO 10 K=1,N
S1=S1+B(K,I)*B(K,J)*D(K,1)
10 CONTINUE
BDBT(I,J)=S1
20 CONTINUE
MM=M-1
DO 30 I=1,MM
IP1=I+1
DO 30 J=IP1,M
BDBT(I,J)=BDBT(J,I)
30 CONTINUE
RETURN
END

```

```

SUBROUTINE ATPAU(N,M,A,F,XL,ATPA,U)
C *****
C *****
C ** SUBROUTINE ATPAU      BOSCO-EDSON  29/05/81      **
C ** PROPOSITO: ESTA SUBROUTINE CALCULA O TRIPLO PRODUTO      **
C **          ATPA=N,ATPL=U      **
C ** CHAMADA: CALL ATPAU(N,M,A,F,XL,ATPA,U)      **
C ** PARAMETROS:      **
C **          N: NUMERO DE LINHAS DA MATRIZ A      **
C **          M: NUMERO DE COLUNAS DA MATRIZ A      **
C **          A: MATRIZ DE ENTRADA, DIM A(N,M)      **
C **          F: MATRIZ DOS PESOS, DIM F(N,N)      **
C **          XL: VETOR L          DIM XL(N,1)      **
C **          ATPA: MATRIZ RESULTANTE DA MULTIPLICACAO      **
C **                AT,P,A,= N, DIM ATPA(M,M)      **
C **          U: VETOR RESULTANTE DA MULTIPLICACAO      **
C **                AT,P,XL,= U, DIM U(M,1)      **
C *****
C *****
      DIMENSION A(N,M),F(N,N),XL(N,1),ATPA(M,M),U(M,1)
      DO 30 I=1,M
      DO 30 J=1,I
      S2=0.D0
      DO 20 K=1,N
      S1=0.D0
      DO 10 K1=1,N
      S1=S1+F(K,K1)*A(K1,J)
10      CONTINUE
      S2=S2+A(K,I)*S1
20      CONTINUE
      ATPA(I,J)=S2
30      CONTINUE
C
C      COMPLEMENTA ATPA
C
      MM=M-1
      DO 40 I=1,MM
      IP1=I+1
      DO 40 J=IP1,M
      ATPA(I,J)=ATPA(J,I)
40      CONTINUE
C
C      CALCULO DE U
C
      DO 60 I=1,M
      S2=0.D0
      DO 50 K=1,N
      S1=0.D0
      DO 70 K1=1,N
      S1=S1+F(K,K1)*XL(K1,1)
70      CONTINUE
      S2=S2+A(K,I)*S1
50      CONTINUE
      U(I,1)=S2
60      CONTINUE
      RETURN
      END

```

```
SUBROUTINE VERSOL(A,B,I)
DIMENSION A(I,I),B(I)
IF(A(1,1)) 20,30,20
20 IF(I.EQ.1) GO TO 10
IM=I-1
DO 5 K = 1,I
DO 2 J = 1,IM
2 B(J)=A(1,J+1)/A(1,1)
B(I)=1./A(1,1)
DO 4 L = 1,IM
DO 3 J = 1,IM
3 A(L,J)=A(L+1,J+1) - A(L+1,1) * B(J)
4 A(L,I)=-A(L+1,1) * B(I)
DO 5 J = 1,I
5 A(I,J)=B(J)
RETURN
10 A(1,1) = 1./A(1,1)
30 WRITE(3,100)
100 FORMAT(1X,' A(1,1) = 0 ')
RETURN
END
```

```
C.....SUBROTINA PARA SOMAR MATRIZES (SOMAT),.....  
SUBROUTINE SOMAT(A,B,C,I,J)  
DIMENSION A(I,J),B(I,J),C(I,J)  
DO 20 L = 1,I  
DO 20 K = 1,J  
C(L,K) = A(L,K) + B(L,K)  
20 CONTINUE  
RETURN  
END
```

```
C.....SUBROTINA PARA MULTIPLICAR ESCALAR POR MATRIZ (EPM).....  
SUBROUTINE EPM(A,AE,I,J,E)  
DIMENSION A(I,J),AE(I,J)  
DO 15 L = 1,I  
DO 15 K = 1,J  
AE(L,K) = E * A(L,K)  
15 CONTINUE  
RETURN  
END
```

C.....SUBROTINA PARA MULTIPLICAR MATRIZES (AB).....

122

```
SUBROUTINE AB(A,B,R,L,M,N)
DIMENSION A(L,M),B(M,N),R(L,N)
DO 5 I = 1,L
DO 5 J = 1,N
R(I,J) = 0
DO 5 K = 1,M
5 R(I,J)=R(I,J) + A(I,K) * B(K,J)
RETURN
END
```

```
C.....SUBROTINA PARA TRANSPOR MATRIZ (TRANSP).....  
  SUBROUTINE TRANSP(A,AT,I,J)  
  DIMENSION A(I,J),AT(J,I)  
  DO 20 L = 1,I  
  DO 20 K = 1,J  
  AT(K,L)=A(L,K)  
20  CONTINUE  
  RETURN  
  END
```

```
SUBROUTINE TRIGO(XK, XF, XW, SK, CK, SF, CF, SW, CW)
SK=SIN(XK)
CK=COS(XK)
SF=SIN(XF)
CF=COS(XF)
SW=SIN(XW)
CW=COS(XW)
RETURN
END
```

```
C      SUBROUTINE XKFW(SK,CK,SF,CF,SW,CW,R)
      CALCULA A MATRIZ DE ROTACAO R=R(KAPPA)*R(FI)*R(OMEGA)
      DIMENSION R(3,3)
      R(1,1)=CK*CF
      R(1,2)=CK*SF*SW+SK*CW
      R(1,3)=-CK*SF*CW+SK*SW
      R(2,1)=-SK*CF
      R(2,2)=-SK*SF*SW+CK*CW
      R(2,3)=SK*SF*CW+CK*SW
      R(3,1)=SF
      R(3,2)=-CF*SW
      R(3,3)=CF*CW
      RETURN
      END
```

## REFERENCIAS BIBLIOGRÁFICAS

- |01| AHLBERG, J.H. et al. The Theory of Splines and their Applications. Ney York: Academic Press, 1967.
- |02| BROWN, Duane C. "Densification of Urban Geodetic Nets" Photogrammetric Engineering and Remote Sensing. Vol 43, Nº 4, (1977), 447-467.
- |03| COLLINS, Stanley H. "Terrain Parameters Directly form a Digital Terrain Model". The Canadian Surveyor. Vol.29, nº5, (1975), 507-518.
- |04| DATE, C.J. An introduction to database systems. 2.ed. Reading, mass., 1977. 536p.
- |05| DRUMMOND, Jane E. Optimum specifications for large scale urban mapping using conventional photogrammetry a study of existing techniques. Technical Report, Fredericton, N.B., (44): 1-197, 1977.
- |06| DUBUISSON, B.L.Y. "Automatic Photogrammetric Cartography". Photogrammetric Engineering and Remote Sensing. Vol. 41, Nº 1, (1975), 65-74.
- |07| \_\_\_\_\_. "Why analytical Plotters?" Photogrammetric Engineering and Remote Sensing. Vol.43, nº 11 (1977), 1367-1375.
- |08| EL-HAKIM, Sabry F. "Potentials and Limitations of Photogrammetry for Precision Surveying". Ph.D Thesis, Department of Surveying Engineering, UNB, 1979.
- |09| FAIG, W. "Investigation of the Performance of Ackermann's Program Package PAT-M-43 for Aerotriangulation with Independent Models". Technical Report Nº 20, Department of Surveying Engineering, UNB, 1972.

- |10| \_\_\_\_\_ . "Calibration of Close Range Photogrammetric Systems: Mathematical Formulation". Photogrammetric Engineering and Remote Sensing. Vol.41, Nº 12 (1975), 1479-1486.
- |11| GARDINER-HILL, Col. R.C. "The Development of Digital Maps". Professional Paper, New Serie Nº 23. Southampton, Ordnance Survey of Great Britain, 1972.
- |12| GHOSH, Sangib K. Phototriangulation. Lexington, mass Lexington Books, 1975. 241 p.
- |13| GRAFAREND, E.W. "Optimization of Geodetic Networks". Bollettino di Geodesia e Scienze Affini. XXXIII, 1974.
- |14| HELAVA, U.V. "The Analytical Plotter - Its Future". Photogrammetric Engineering and Remote Sensing. Vol.43, Nº 11 (1977), 1361-1362.
- |15| HOBBIE, I.D. "C100 PLANICOMP. The Analytical Stereoplotting System from Carls Zeiss". Photogrammetric Engineering and Remote Sensing.
- |16| KEATING, T.J. et al. "An improved Method of Digital Image Correlation". Photogrammetric Engineering and Remote Sensing. Vol. 41, Nº 8 (1975), 993,1002.
- |17| KONECNY, G. "Software Aspects of Analytical Plotters". Photogrammetric Engineering and Remote Sensing. Vol.43, Nº 11 (1977), 1363-1366.
- |18| LUGNANI, João Bosco. "Using Digital Entities as Control". PhD. Thesis, Department of Surveying Engineering, UNB, 1980.
- |19| MASRY, S.E. "Accuracy and Time Comparisons of Digital Maps. An International Test". Presented Paper at the XIVth International Congress for Photogrammetry, Hamburg, 1980.

- |20| \_\_\_\_\_ . "Method for Transferring Survey Detail to Photographs". Surveying and Mapping. Vol. 33, Nº 1 (1973), 23-26.
- |21| \_\_\_\_\_ . "Photogrammetric Digitising and Editing under Computer Control". Photogrammetric Record. Vol.7, Nº 41 (1973), 568-581.
- |22| \_\_\_\_\_ . "Digital Correlation Principles". Photogrammetric Engineering. Vol. 40, Nº 3 (1974), 303-308.
- |23| \_\_\_\_\_ . "Digital Mapping using entities: a new concept". Photogrammetric Engineering and Remote Sensing. Falls Church, vir 48 (11). 1561-5, November 1981.
- |24| MOFFITT, Francis H. & MIKHAIL, Edward N. Photogrammetry. 3. ed. New York, Harper & Row, 1980. 648p.
- |25| PACITTI, Tércio. Fortran - monitor; principios. 3. ed. Rio de Janeiro, Livros Técnicos e Científicos, 1978. 377p.
- |26| PEET, F.G. "The Use of Invariants in the Transformation and Registration of Images". The Canadian Surveyor. Vol. 29. Nº 5 (1975). 501-506.
- |27| POPE, A.J. "Some Pitfalls to be Avoided in the Iterative Adjustment of Nonlinear Problems". NOAA, Geodetic Research and Development Laboratory. Rockville, Md., 1974.
- |28| QUINTINO, Moacir José. Estudo Matemático e Comparativo dos Métodos de Orientação Relativa. Curitiba, 1975. 95p. Tese Mestrado, Universidade Federal do Paraná. Curso de Pós-Graduação em Ciências Geodésicas.

- |29| SCARANO, F.A. "A Digital Elevation Data Collection System". Photogrammetric Engineering and Remote Sensing. Vol. 42, Nº 4 (1976), 489-496.
- |30| STEINBRUCH, Alfredo. Algebra Linear e Geometria Analítica. São Paulo, McGraw Hill, 1973. 518p.
- |31| THOMPSON, E.H. Introduction to the Algebra of Matrices with Some Applications. Toronto: The University of Toronto Press, 1969.
- |32| ZIEMANN, H. "High Accuracy Photogrammetric Determinations Using Image Deformation Corrections". The Canadian Surveyor. Vol. 34, Nº 1 (1980).