

## 수치사진측량을 위한 객체지향클래스 라이브러리 개발

### Development of Object Oriented Class Library for Digital Photogrammetry

유복모\*  
Yeu, Bock-Mo

이석군\*\*  
Lee, Seok-Gun

김의명\*\*  
Kim, Eui-Myoung

김기홍\*\*  
Kim, Gi-Hong

#### 要 旨

수치영상의 실시간 자료처리를 위해서는 문제 영역의 모든 것을 각각의 독립적 객체로 파악하여, 모든 문제를 객체간의 메시지 전달에 의한 객체들의 동작으로 파악하려는 객체지향기법이 필요하다. 본 연구에서는 객체지향기법을 이용하여 수치사진측량용 클래스를 설계하고 클래스들 간의 계층구조를 확립하였다. 또한 클래스 라이브러리를 이용하여 근거리 사진측량에 적용하므로써 쉽게 수치사진측량 시스템이 구현됨을 알 수 있었다.

#### ABSTRACT

For real-time data process of digital images, all things that related to problem-domain are understood individual object, and then it is necessary to program using object oriented approach to grasp all problem as a objects operation by means of message transmission among objects.

In this study, for digital photogrammetry class is designed using object oriented programming and class hierarchy is established. Also, it is very useful to construct to digital photogrammetry system using class library when it is applied to close range photogrammetry.

\*정회원 · 연세대학교 토목공학과 교수

\*\*정회원 · 연세대학교 토목공학과 박사과정

## 1. 서론

수치사진측량은 전산기술의 발달과 더불어 그 가능성과 실현성을 넓혀가고 있으며, 특히 사진측량의 일련의 과정에 대한 자동화를 목적으로 연구가 진행되어, 관측과정의 자동화와 실시간 3차원 측량기법 개발 등이 활발히 이루어지고 있다.

수치영상의 실시간 자료처리를 위해서는 문제 영역의 모든 것을 각각의 독립적 객체(object)로 파악하여, 모든 문제를 객체간의 메시지 전달에 의한 객체들의 동작으로 파악하려는 객체지향 기법(Object Oriented Programming)이 필요하다.

객체지향기법(OOP)은 자기자신의 상태를 나타내는 애트리뷰트(attribute)와 동작을 나타내는 메소드(method)를 하나의 단위로 하는 클래스를 제작하고 이 객체에 적절한 메시지(message)를 전달함으로써 모든 동작이 이루어지도록 하는 것이다. 객체지향 프로그래밍의 특징은 추상성(abstraction)과 상속성(inheritance), 다형성(polymorphism)을 들 수 있다. 추상성은 캡슐화(encapsulation)와 정보은폐 기능으로 대표되며 캡슐화는 데이터와 함수를 통해 하나의 캡슐을 만들 듯이 데이터형을 구성하는 모습을 말하는 것이고, 정보은폐는 그러한 멤버중에서 외부에서 직접 접근이 불가능한 특성을 가진 영역을 말하는 것이다. 상속이라는 것은 기초클래스로부터 특성을 물려받은 파생클래스를 만들게 됨을 뜻한다. 다형성은 동일한 이름을 가진 여러 형태의 모습을 구현하는 것이다.

야의 일련의 과정들의 계층도를 확립하고 주제별로 각각의 수치사진측량용 클래스 라이브러리를 개발하고자 한다.<sup>1,2)</sup>

## 2. 객체지향개념의 적용

객체지향설계는 클래스(class)와 그 클래스들 간의 계층구조를 확립하는 것으로 각 클래스는 데이터구조(attribute)와 행동양식(method)이 함께 정의된 모듈이다.

객체지향설계의 방법은 먼저 클래스를 인식하고, 정의하고, 구성하는 법칙을 세우고 이를 통해 클래스 라이브러리를 구성하며 이렇게 구현된 클래스나 객체를 사용하여 새로운 클래스를 파생시키거나 요구조건에 맞게 수정을 행하여 클래스 전체의 계층도를 완성하는 것이다. 본 연구에서는 행렬연산을 위한 Matrix 클래스를 기초클래스로 하여 공선조건, 광속조정, 좌표변환, 직접선형변환(DLT) 등에 대한 클래스를 설계하고 BMP영상을 처리할 수 있는 CBmp 클래스를 기초클래스로 하여 Hough 변환, Matching 클래스를 설계하였으며 클래스의 계층도는 그림2.1과 같다.

이와 같이 계층관계를 명시함으로써 프로그램 전반에 관한 골격을 형성할 수 있게 되어 프로그램을 알기 쉽고 명확하게 하여 직관적인 이해성을 높일 수 있다. 클래스 상속을 통해 추가적인 기능만을 덧붙일 수 있으므로 재사용성의 증대가 가능하게 되며, 구축

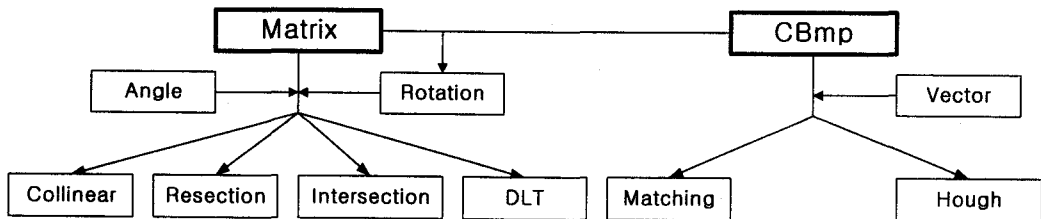


그림 2.1 수치사진측량 클래스의 계층도

본 연구에서는 수치사진측량의 각 과정을 클래스라 된 계층구조에서 가장 적합한 클래스들을 선택하여는 사용자 정의 자료형으로 정의하여 수치사진측량본 그들의 새로운 하위 클래스를 설정하고, 여기에 새로

은 애트리뷰트(attribute)나 메소드(method)를 첨가하거나 대치하여 새로운 클래스들을 계속적으로 구축해 나갈 수 있게 된다.

### 3.1 Matrix 클래스와 Rotation 클래스

수치사진측량용 클래스의 기초클래스인 Matrix 클래스는 데이터 멤버로 행과 열을 나타내는 정수형 r, c와 행렬의 내용을 가질 포인터 m이 있다. 또한 멤버

### 3. 수치사진측량 클래스 제작

표 3.1 행렬 클래스의 정의와 구조

Matrix 클래스의 구조		프로그램 형태
Class Name	Matrix	<code>class Matrix</code>
Attribute	행렬의 요소 행의 수 열의 수	<code>{</code> <code>private:</code>  <code>double ** m;</code> <code>int r;</code> <code>int c;</code>
Method	다형성 생성자 생성자 생성자 소멸자	<code>public:</code>  <code>Matrix();</code> <code>Matrix(int, int);</code> <code>Matrix(int, int, double);</code> <code>Matrix(Matrix&amp;);</code> <code>~Matrix();</code>
	연산자중복 행렬연산 행렬의 대입 행렬의 덧셈 행렬의 곱셈	<code>double&amp; operator()(int, int);</code> <code>Matrix&amp; operator=(const Matrix&amp;);</code> <code>Matrix operator+(Matrix&amp;);</code> <code>Matrix operator*(double);</code>
	행렬의 입력 전지행렬 역행렬 행렬의 출력	<code>Matrix&amp; read();</code> <code>virtual Matrix trans();</code> <code>Matrix inverse();</code> <code>void print();</code>
	프렌드함수 행렬의 덧셈 행렬의 곱셈 행렬의 출력 행렬의 파일입력 행렬의 파일출력	<code>friend Matrix operator+(Matrix&amp;, Matrix&amp;);</code> <code>friend Matrix operator-(Matrix&amp;, Matrix&amp;);</code> <code>friend Matrix operator*(const Matrix&amp;, const Matrix&amp;);</code> <code>friend ostream&amp; operator&lt;&lt;(ostream&amp;, Matrix&amp;);</code> <code>friend BOOL MatFileRead(CString, Matrix&amp;);</code> <code>friend BOOL MatFileOut(CString, Matrix&amp;);</code> <code>friend void TextPrint(int, int, CString, COLORREF, CDC*);</code>
		<code>};</code>

본 연구에서는 먼저 Matrix 클래스에서 행렬의 일 반계산을 다루며 하위 클래스에서는 이를 계승받아 상위클래스의 내용을 그대로 가지면서 각각의 모듈에 맞는 계산을 행하게 된다. 또한 클래스별로 주제를 나누어 정리하는 객체지향개념의 장점으로 여러 명의 작업자가 나누어 미리 합의하거나 제시된 헤더파일(header file)에 맞게 프로그램을 작성함으로써 각자의 개성과 특징을 살리며 공동주제의 분담연구를 가능케 할 수 있다.

함수에는 표 3.1과 같이 생성자, 소멸자, 행렬고유함수, 프렌드함수, 연산자중복함수 등이 있다.<sup>3)</sup>

수치사진측량에서 3축방향으로 경사진 좌표계를 기준좌표계로 변환하기 위해 표 3.2와 같이 기초클래스인 Matrix 클래스에서 파생된 Rotation 클래스를 설계 하였다. 회전변환의 요소는 행렬의 성질을 모두 가지 게 되므로 행렬에 대한 일반 성질은 Matrix 클래스에서 정의하고 회전에 필요한 요소만 Rotation 클래스에 정의한다. 이와 같이 객체지향개념으로 상속을 통해 이

미 구성되어 있는 행렬 클래스의 애트리뷰트와 메소드를 물려받고 필요한 부분만을 재구성할 수 있으므로 프로그램의 재사용성과 재구성이 매우 용이하게 된다.

Hough 변환은 영상의 모든 유효한 영상소에 대해서 그 점을 지나는 모든 직선의 방정식을 직선 변수 공간으로 투영시켜, 변수공간에서의 교차점은 동일선상에 있는 점들이라는 사실을 이용한 것으로 영상으로

표 3.2 Rotation 클래스

Rotation 클래스의 구조		프로그램 형태
Class Name	Rotation	<pre>class Rotation : public Matrix { public:     Rotation();     Rotation(int, int);     ~Rotation();      Rotation rotation_x(double&amp;);     Rotation rotation_y(double&amp;);     Rotation rotation_z(double&amp;);     Matrix rotation_zyx(double&amp;, double&amp;, double&amp;); };</pre>
Method	생성자 생성자 소멸자  X축회전 Y축회전 Z축회전 3축회전	

### 3.2 Hough 클래스

스캐닝된 수치영상에 대한 입출력을 위해서 Visual C++의 기초클래스인 CObject에서 파생된 CBmp 클래스를 설계하였으며 내부표정의 자동화의 일부에 해당하는 사진지표에 대한 영상좌표취득을 위해서 표 3.3 과 같이 Hough 클래스를 설계하였다.

표 3.3 Hough 클래스

Hough 클래스	
Class Hough : public CBmp	
함수	<pre>friend class vector; hough&amp; process_image(hough&amp;); int look_up_table(vector&amp;, vector&amp;, int*); int hough(hough&amp;, hough&amp;, vector&amp;, int*, int*); int thres_par(hough&amp;, int*, int*, int*); int ihough(hough&amp;, hough&amp;, hough2&amp;, vector&amp;, vector&amp;, int*, int*);</pre>

영상에서의 Hough 변환은 수행할 영상소를 매개변수(parameter space)영역으로 변환하는 것으로 임의의 물체의 외곽선추출(직선, 원, 타원 등)을 할 수 있는 강력한 도구이다. 본 연구에서는 거리-각( $\rho-\theta$ )좌표계에서의 하프변환을 구현하였다.

부터 직선을 추출하는데 활용된다.<sup>45)</sup>

일반적인 계산과정은 다음과 같이 5 단계로 나눌 수 있다.

- (1) 수치영상입력
- (2) 변위  $\rho$ 의 최대크기는 처리하고자 하는 영상의 크기에 의해 결정되고, 각도  $\theta$ 는  $-90^\circ \sim 90^\circ$  까지로 정한다,
- (3) 직선 성분을 검출 하고자 하는 대상 영역의 모든 특징 영상소에 대해  $\rho = x \cos \theta + y \sin \theta$  로 하프변환을 수행한다.
- (4) 누적배열을 조사하여 적당한 임계값을 정하고 이 값을 넘는 누적배열 성분만을 구해낸다. 임계값은 처리하고자 하는 영상내에서 의미를 가지는 최소 직선의 길이로 결정한다.(그림 3.1)
- (5) 위의 과정으로 정해진  $\rho$  와  $\theta$  로서 대상 영상 내에 존재하는 직선을 추출해 낸다.

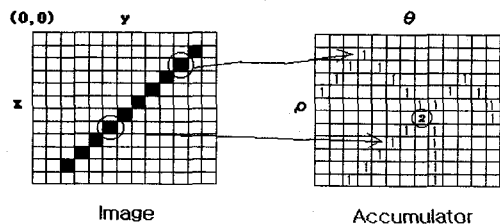


그림 3.1 누적 배열에서 특징저장소의 저장과정

### 3.3 Matching 클래스

영상정합에 앞서 영상좌표계에서 취득된 기준점에 대한 영상좌표를 사진좌표계로 변환해야 하며 이 과정에서 부등각사상변환(affine transformation)을 가장 많이 이용한다. 부등각사상변환은 사진측량의 가장 기본이 되는 공선조건식을 클래스로 설계한 Collinear 클래스의 멤버함수로 되어있다.

좌우영상에서의 모든 점의 좌표를 사진좌표계로 변환한 다음 좌우영상에 대한 영상정합을 실시한다. 영상정합은 두 개의 영상에서 밝기값을 비교하여 공통되는 점을 결정하는 것으로 수치영상을 이용한 영상정합은 좌우 수치영상에 존재하는 공액점들을 탐색하는 방법으로 영상정합의 정도에 따라 수치사진측량의 정확도가 좌우된다.<sup>6)</sup>

영상정합과정을 객체지향프로그래밍하는 과정은 그림 3.2와 같이 하나의 Document와 2개의 View로 구성할 수 있다.

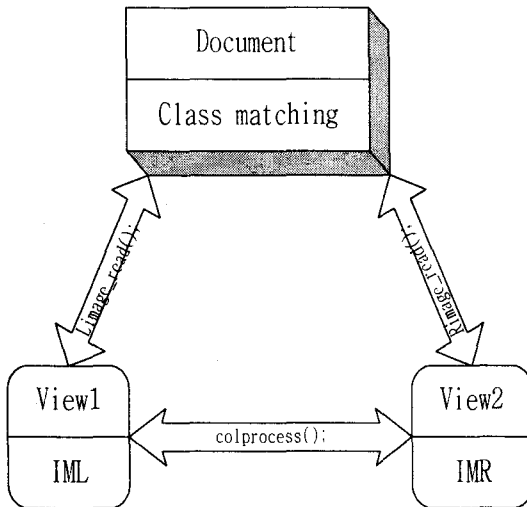


그림 3.2 Document와 View의 측면에서 본 영상정합과정

View1에서 탐색영역을 설정하면 설정된 탐색영역의 자료는 Document를 통해서 View2의 대상영역을 검색하는 구조로 간략화할 수 있다. 본 연구에서는 영상상관기법으로 영상정합을 클래스화하고 실행하였다.

View1과 View2의 영상에 대한 밝기값의 상관성을 영상정합클래스인 matching 클래스의 colprocess() 멤버함수를 통해서 수행하였다. 영상상관기법은 좌우 수치영상의 동일점으로 예상되는 두 점의 유사성을 공분산과 상관계수를 통해 통계학적으로 추정하는 것이다.<sup>7)</sup>

중복 촬영된 한 쌍의 사진간에 비교가 이루어진다고 할 때 좌측영상에 있는 어떤 점을 중심으로 하는 기준영역을 설정하고 우측영상에 공액점이 존재한다고 가정이 되는 탐색영역을 설정한다. 그리고 탐색영역내의 임의의 한점 (n, m)에 대하여 그 점을 중심으로 하고 기준영역과 같은 크기의 영역 즉 대상영역이 있다고 가정하면, 기준영역의 크기를 N\*M 이라 할 때 두 영역간의 상관계수 cor(n, m)은 식 (3.1)과 같이 주어진다.

$$\begin{aligned}
 &cor(n, m) \\
 &= \frac{\sum_{x=1}^N \sum_{y=1}^M (G_u(x, y) - \bar{G}_u) \cdot (G_s(x, y) - \bar{G}_s)}{\left[ \sum_{x=1}^N \sum_{y=1}^M \{G_u(x, y) - \bar{G}_u\}^2 \cdot \sum_{x=1}^N \sum_{y=1}^M \{G_s(x, y) - \bar{G}_s\}^2 \right]^{\frac{1}{2}}} \\
 &.....(3.1)
 \end{aligned}$$

### 3.4 DLT 클래스

DLT 클래스에서는 DLT에서 사용하는 입력자료만을 쉽게 알아 볼 수 있도록 표 3.4와 같이 별도의 데이터 클래스를 구축하였고, 어떤 점을 기준점으로 선택할 지 검사점으로 할지의 선택에 대한 변수를 지정하여 기준점선택유무에 대한 적용을 자료입력부분에서 지정할 수 있도록 설계하였다.

표 3.4의 1)번에서 선택된 자료에 따라 3)번에서 지상기준점 좌표를 입력받고 4)번에서 좌측사진좌표와 기준점좌표로부터 L 계수를 산정하고 5)번에서는 우측사진좌표와 기준점좌표로부터 L 계수를 산정하게 된다. 6)번함수는 좌우측 사진좌표와 계산된 L 계수로 대상물의 3차원 좌표를 계산한다. 7)번은 4), 5), 6)번 함수들의 계산결과를 출력하는 함수로서 각 함수마다

표 3.4 DLT 클래스

DLT 클래스의 구조	
Class Name	class DLT : public Matrix
멤버	1) struct coord { double 좌, 우측 사진의 xy 좌표; double 지상기준점의 XYZ 좌표; int 지상기준점으로 사용유무; double 인터섹션에 대한 대상물의 XYZ 좌표; } nate(MAXPOINT); 2) double 좌, 우측의 DLT 계수;
함수	3) GCPin(); 4) Limagecoord(), Rimagecoord(); 5) LDLTressection(), RDLTressection(); 6) DLTinter(); 7) ALLprint();

별도의 출력함수를 만들도록 설계하였다.

본 연구에서 설계한 DLT 클래스는 특히 비측정용 카메라의 검정에 유용하도록 설계되었으며 실시간 사진측량의 입력자료를 위한 CCD 사진기에 쉽게 적용 되도록 하였다. DLT를 사용하여 CCD 사진기로 취득된 영상의 보정을 위한 모형식을 구성하여 CCD 영상 취득에서 발생하는 렌즈의 편심왜곡과 방사왜곡을 보정한다. 렌즈의 왜곡을 보정하기 위한 기하학적 모형은 다음과 같이 나타낼 수 있다.<sup>8)</sup>

$$\begin{aligned}
 F_1 &= x + \delta_x \\
 &= \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1} \\
 F_2 &= y + \delta_y \\
 &= \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1} \dots\dots\dots(3.2)
 \end{aligned}$$

$$\begin{aligned}
 \delta_x &= k_1(xr^2) + k_2(xr^4) + k_3(xr^6) \\
 &\quad + P_1(r^2 + 2x^2) + P_2(2xy) \\
 \delta_y &= k_1(yr^2) + k_2(yr^4) + k_3(yr^6) \\
 &\quad + P_1(2xy) + P_2(r^2 + 2y^2) \dots\dots\dots(3.3)
 \end{aligned}$$

$L_1 \dots L_{11}$ 를 선형변환계수로 정의하고

$k_1, k_2, k_3, P_1, P_2$ 를 왜곡보정계수라고 정의한다. CCD 사진기로 취득된 영상으로부터 선형변환계수를 결정하고 또 취득된 영상을 보정하기 위한 모형식을 결정하기 위한 왜곡보정계수를 구하는 과정에서 선형변환계수와 왜곡보정계수 사이에 높은 상관성이 있어 동시에 정확한 해를 구하기 어려우므로 먼저 선형변환계수를 결정하고 이를 이용하여 왜곡보정계수를 구하고 검증된 모형식을 바탕으로 다시 선형변환계수를 구하는 방식을 적용하였다.

### 3.5 Resection과 Intersection 클래스

표 3.5 Resection 클래스

Resection 클래스의 구조	
Class Name	Class Resection : public Matrix
멤버	1) double 사진좌표(xy), 대상물 좌표(X,Y,Z); 2) double 촬영점 초기값(X_0,Y_0,Z_0); 3) double 회전요소(k,p,w); 4) double kx, ky, kp, kw; 부가항
함수	5) Resection& operator=(const Matrix&); 6) BOOL Read_Resection_Data(CString); 7) void Bundle_Process(Resection&, CDC*); 8) Resection coeff_B(Resection&, Matrix&); 9) ResectionAdditional_B(Resection&, Matrix&); 10) Resection coeff_E(Resection&);

공간후방교선법을 수행하기 위해서는 표 3.5의 1), 2), 3) 번이 초기자료로서 입력되어지며, 시스템내에서는 화면상에 출력하여 자료를 변경하거나 대화식으로 볼수 있도록 대화 클래스를 별도로 설계하였다. 대화 클래스를 거친 자료는 상위 클래스로 정의한 Rotation 클래스에서 회전변환을 수행한다. 그 다음 8)번 함수에서 외부표정요소(X\_0, Y\_0, Z\_0, k, p, w)에 대한 편미분을 행하여 관측방정식을 형성하고, 10)번의 함수에서 이에 대한 정규방정식을 형성하게 된다. 그 다음 과정은 최소제곱조정과정으로 상위 클래스인 Matrix 클래스에서 정의한 대로의 행렬계산을 수행하

도록 설계하였다. 처리 결과는 콘솔(console)프로그램의 경우는 Matrix 클래스에서 정의된 출력함수로 출력할 수 있으며 윈도우 프로그램의 경우의 결과 저장과 화면상에서의 출력은 7)번의 함수에서 처리하도록 설계하였다. 또한 부가항을 이용할 경우에는 4)번과 9)번을 이용하면 된다.

사진좌표와 기준점의 3차원 좌표로부터 외부표정요소를 구하는 공간후방교회법(Resection) 과정과 이를 이용하여 대상물의 3차원 위치를 구하는 공간전방교회법(Intersection) 과정은 둘다 유사한 과정을 거치므로 관측방정식을 형성할 때 행렬구성을 같이 하거나 다른 클래스로 별도로 분리하여 Intersection 만의 과정으로 분리하여 계산할 수 있다. 본 연구에서는 Resection과정을 거쳐 일단 외부표정요소가 결정되면 이를 이용하여 Intersection를 수행할 수 있도록 클래스를 설계하였으며 Intersection에서 초기치를 결정하기 위해서 선형해석법을 이용하였으며 LinearXYZ()함수에서 이를 수행한다. Intersection 클래스는 표 3.6에 나타나 있으며 IntersectionProcess()에서 실제 공간전방교회법을 수행한다. 처리된 결과는 Draw()함수를 이용하면 View에 출력할 수 있다.

표 3.6 Intersection 클래스

Intersection 클래스의 구조	
Class Name	Class Intersection : public Matrix
멤버	int iter;//반복회수 double focal;//초점거리 double XL, YL, ZL, XR, YR, ZR;//좌측, 우측 투영 중심 double omega1, omega2, phi1, phi2, kappa1, kappa2;//좌우측 회전요소 int nimage;//상좌표의 수 int *num1, *num2;//사진좌표의 번호 Matrix *X, *Y, *Z;//초기치 또는 해 Matrix *x1, *y1, *x2, *y2;//좌우측사진좌표
함수	Intersection(); ~Intersection(); Intersection(int,int); Intersection& operator=(const Matrix&); BOOL IntersectionFileRead(CString); void LinearXYZ(); void coefft_BE(Matrix&, Matrix&, int); BOOL IntersectionProcess(CString); void Draw(CDC*, CString, CString, double);

객체지향개념으로 이루어진 MFC(Microsoft Foundation Class)와의 조합으로 전체 클래스를 구성하면 그림 3.3과 같은 연결구조를 만들 수 있다.

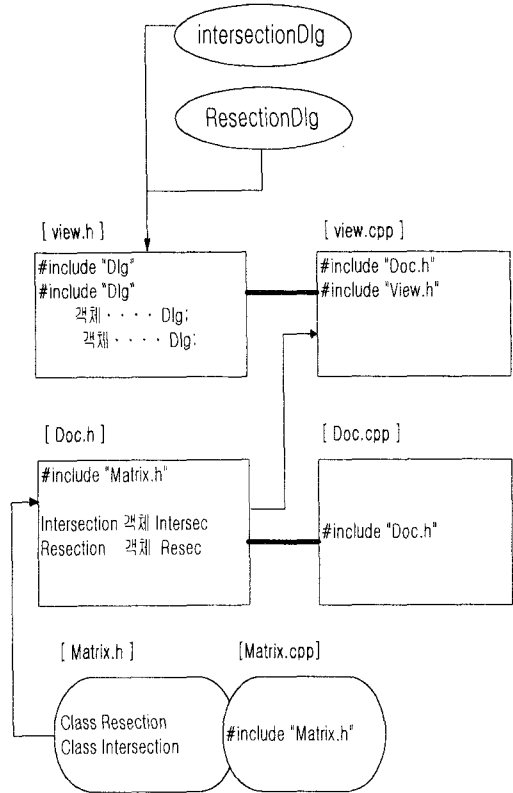


그림 3.3 Intersection, Resection, MFC 의 연결구조

입력자료는 키워드(Key Word)를 이용하여 자료의 순서에 상관없이 식별할 수 있도록 하였으며, 특히 키워드에 대한 대문자 소문자의 구별을 없앴다. 표3.7은 공간후방교회법의 수행을 위한 입력자료의 예를 보여 주고 있다.

표 3.7 Resection을 위한 입력자료의 예

photo id 3115	//키워드	//사진번호
focal length 153.24 (mm)	//키워드	//초점거리
initial X0 Y0 Z0 value 39800 27480 7570 대	//키워드	//지상기준점에 한 초기치
number of gcp 4 수	//키워드	//지상기준점의
1 36589.41 25273.32 2195.17	//지상기준점	
2 37631.08 31324.51 728.69		
3 39100.97 24934.98 2386.50		
4 40426.54 30319.81 757.31		
number of image coord 4	//키워드	//상좌표의 수
1 -86.15 -68.99		//상좌표
2 -53.4 82.21		
3 -14.78 -76.63		
4 10.46 64.43		
omega 0	//키워드	//X축회전요소의 초기
치(radian)		
phi 0	//키워드	//Y축회전요소의 초기
치(radian)		
kappa 0	//키워드	//Z축회전요소의 초기
치(radian)		

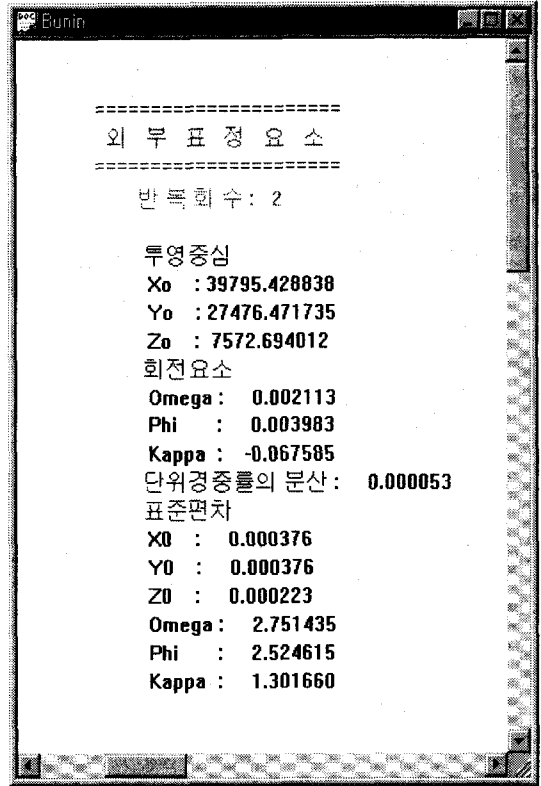


그림 3.4 Resection 처리의 예

#### 4. 클래스 라이브러리의 적용

파 일	편 집	뷰설정	자료처리		변환	표정	광속조정	영상정합	창	도움말
새파일	실행취소	틀 바	영 상 블러오기	좌측영상블러오기 우측영상블러오기	등각사상변환	내부표정	광속조정자료 블 러 오 기	영상상관기법	새창	Photo..
열기	잘라내기	상태바	확대	좌측영상확대 우측영상확대	부등각사상변환	상호표정	광속조정 자료처리		계단식	
닫기	복사하기				투영변환	절대표정	직접선형변환		마둑	
저장	붙여넣기								판식	
새이름 으로 출력 미리									아이콘	
보기									정렬	
프린터 설정										
최근작 업파일										
종료										

그림 4.1 수치사진측량시스템의 메뉴



수치사진측량용으로 설계된 클래스 라이브러리를 이용하여 항공사진을 스캐닝한 수치영상 또는 CCD 사진기에 의한 수치영상의 입력과 처리 및 대상물의 3차원 위치를 결정하기 위한 영상정합 등의 일련의 과정을 PC에서 수행할 수 있도록 수치사진측량시스템을 구현하였으며 메뉴는 그림 4.1와 같다.

본 연구에서 제작한 클래스 라이브러리를 이용하여 수치사진측량시스템을 구현하고 이를 CCD 사진기를 이용한 근거리 사진측량에 적용하였다. 또한 CCD 사진기를 통해 좌우측의 수치영상을 취득한 후 직접선형변환(Direct Linear Transformation)과 광속조정법(Bundle Adjustment)을 수행하여 대상물의 3차원 위치를 결정하고 정확도를 평가하였다.

#### 4.1 실험모형 및 영상 취득

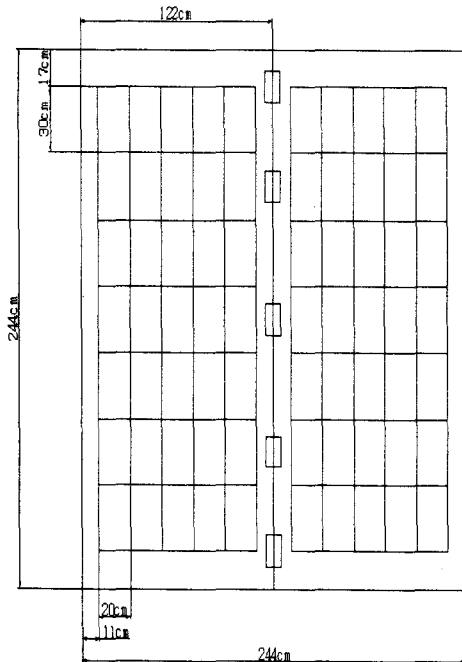


그림 4.2 실험모형

근거리 사진측량을 위해서 먼저 그림 4.2와 같은 실험모형을 제작하였으며 가로방향 122 cm, 세로방향 244 cm, 두께 1.2 cm 의 나무판 2장으로 구성하였다. 나무판은 경첩을 사용하여 연결하였으며 좌우 나무판

위에 가로방향 20 cm, 세로방향 30 cm 간격의 격자를 만들고 각 격자점에는 지름 5 mm 의 구멍을 뚫어서 총 70 개의 대상물(target)을 설치할 수 있도록 제작하였다. 실험모형내에 설치된 대상물은 나무봉이며, 길이는 각각 5, 10, 15, 20, 25, 30 cm 로 다양한 높이를 나타내도록 하였다. 봉의 끝에 원형의 사진지표를 부착하여 촬영 및 기준점 측량을 행하였다. CCD 사진기는 Hitachi 사의 VK-C370를 사용하였다. 이 사진기는 유효 영상크기가 768×494 이며 본 연구에서는 영상보드와의 호환성을 위해 640×480 영상소 크기의 영상을 취득하였다.

#### 4.2 CCD 영상의 보정

본 연구에서 사용된 CCD 사진기는 일반적으로 근거리 사진측량에 적용하기 위해서 렌즈의 왜곡을 보정하여야 한다. 영상의 영상소좌표에 대한 방사 및 편심왜곡을 보정하기 위하여 영상소좌표가 (320, 240)인 점을 원점으로 하고 한 픽셀의 크기를 6.4 $\mu$ m단위로 하는 좌표계로 변환한 좌표를 이용하여 보정좌표를 결정하였다. 실험모형의 3차원좌표는 간접삼각측량에 의한 기준점측량을 이용하였다.

보정용 영상좌표와 모형의 기준점좌표를 이용하여 직접선형변환식과 방사 및 편심왜곡 보정식을 결합되어 있는 식 (3.2)를 이용하여 보정을 실시하였으며 구해진 왜곡보정계수는 표 4.1과 같다.

표 4.1 왜곡보정계수

왜곡보정계수	16mm CCD 사진기
$k_1$	-0.092060
$k_2$	0.038826
$k_3$	-0.004834
$P_1$	-0.001196
$P_2$	0.005854

### 4.3 영상정합

본 연구에서는 좌우영상의 공액점의 좌표를 추출하기 위해 영상상관기법(image correlation method)을 이용하였다. 좌우영상의 상관성은 두 영역의 유사성을 나타내는 상관계수를 통계적으로 처리하여 추출하였으며 상관계수법을 이용하여 정합점을 탐색하였고, 각 정합점을 기준영역의 중심점으로 설정하였다. 기준영역의 크기는 영상에 나타나는 표지를 고려하여 11×11 영상소로 설정하였으며 탐색영역의 크기는 41×41 영상소 크기로 하여 실험을 수행하였다. 영상정합 결과는 대부분의 상관계수값이 0.9 이상으로 정합이 잘 이루어 졌다.

### 4.4 대상물의 3차원 위치결정 및 비교고찰

개발된 클래스 라이브러리를 이용하여 대상물의 3차원 위치결정에 있어 본 연구에서는 관측점의 정확도를 비교하기 위하여 직접선형변환식과 CCD 특성에 따른 부가항을 추가한 광속조정법을 개발하여 각각의 경우에 있어서 좌표결정의 정확도를 비교분석하였다. 실험모형에서 총 70개의 측정중 5, 10, 15, 20.....65, 70번의 14개 점을 기준점으로 활용하고 그외의 점은 3차원 위치의 정확도를 평가하는 검사점으로 사용하였다.

#### 4.4.1 DLT를 이용한 대상물 좌표 취득

모두 11개의 직접선형변환계수를 구하기 위해서는 최소 6개의 측점이 필요하며 본 연구에서는 정확도를 고려하여 8개의 측점을 잉여관측값으로 추가하였다. 그 외의 점들을 DLT계수를 이용하여 실제로 3차원 위치를 구한 후 이를 기준점 측량 성과와 비교하여 정확도를 검증하는 검사점으로 하였다. 비교를 위하여 광속조정과정에도 같은 기준을 적용하였다. 최소제곱법으로 구해진 직접선형변환계수는 표4.2와 같다.

4.3.2 CCD 특성에 따른 부가항을 추가한 광속조정법을 이용한 대상물 좌표취득  
 CCD 사진기는 일반적으로 근거리 사진측량에 적용

표 4.2 DLT계수

계수	좌측영상	우측영상
L <sub>1</sub>	0.0003346970030228975	0.00034542533753104902
L <sub>2</sub>	-0.0000108649084274087	-0.00000133785709742754
L <sub>3</sub>	0.00008747486927657318	0.00003891758776280411
L <sub>4</sub>	-2.27804477620523740000	-2.04803761041951620000
L <sub>5</sub>	0.00000851935200061339	0.00000691749517266520
L <sub>6</sub>	0.00030222307904917726	0.00031105378692085112
L <sub>7</sub>	0.00005866231319443644	0.00006773832073519092
L <sub>8</sub>	-0.38240486365589588000	-0.44866329962784768000
L <sub>9</sub>	-0.00000252616123260583	-0.00000785445738086738
L <sub>10</sub>	0.00001025398549457208	0.00000881688834193385
L <sub>11</sub>	-0.00012617907514430156	-0.00012165186496129599

하기 위해서는 렌즈의 왜곡을 보정하여야 하며 보정을 거친 후 실제로 촬영을 행할 시 비측정용이므로 사진측량에 필요한 사진지표가 없어 사진촬영시의 주점좌표를 정확히 구할 수 없으므로 이에 대한 보정항을 추가하였다. 본 연구에서 사용한 실험모형의 경우 공선조건식에 보정항을 추가하여 계산한 결과 좌측영상에서는 주점이동량은 l<sub>xp</sub> = 3.9910, l<sub>yp</sub> = 0.6721 이며 우측영상에서는 r<sub>xp</sub> = 3.7763, r<sub>yp</sub> = -0.2132을 얻을 수 있었다. 그리고 영상취득시 선명한 영상을 얻기 위해 CCD 렌즈초점의 미세한 변화가 동반되며 따라

표 4.3 외부표정요소

계수	좌측영상	우측영상
$\omega$	0.061616	0.117047
$\varphi$	0.135096	0.293438
$\kappa$	-0.009838	-0.017830
X <sub>0</sub>	3743.320525	5558.385365
Y <sub>0</sub>	-1115.595993	-970.675377
Z <sub>0</sub>	16025.637643	15952.925578

서 영상취득시의 정확한 초점거리를 알기위해 초점거리 보정항을 추가하여 공선조건식에 적용하면 위치결정의 정확도를 높일 수 있다. 이에 대한 보정항을 추가하여 계산한 결과 좌측영상의 경우 초점거리  $L_{focal} = 13.9040$ , 우측영상의 경우  $r_{focal} = 14.1710$ 의 값을 얻었다. CCD 사진기 특성에 따른 주점좌표 보정항과 초점거리 보정항을 추가한 부가매개변수 광속조정법으로 계산한 결과 좌우측영상의 외부표정요소는 표 4.3과 같다.

DLT 법과 CCD 특성에 따른 부가항을 첨가한 광속조정법의 경우에 대하여 선정된 대상점의 지상좌표를 간접삼각측량방법으로 구한 관측값을 기준으로 평균제곱근오차를 산정하였으며 그 결과는 표 4.4과 같다.

표 4.4 각 경우에 대한 좌표값의 표준오차

	DLT	자체검정 BUNDLE
X(mm)	38.7793	12.5073
Y(mm)	52.2080	21.0206
Z(mm)	33.4205	17.6492

## 5. 결 론

본 연구에서는 객체지향기법을 이용하여 수치사진측량의 각 단계에 대한 클래스(class)를 설계하고 설계된 클래스를 라이브러리로 만들고 근거리 사진측량에 적용하여 다음과 같은 결론을 얻었다.

1. 객체지향개념의 상속성을 이용하여 기초클래스인 Matrix와 CBitmap 클래스를 이용하여 클래스의 계층도를 구축하여 수치사진측량의 각 기능을 수행할 수 있는 파생클래스를 설계하였으며 이를 이용하여 수치사진측량시스템을 구축하였다.

2. 클래스 라이브러리를 이용하여 수치사진측량 시스템을 구현하고 CCD 사진기를 이용한 근거리 사진측량에 적용하였으며 이를 기본도구로 하여 GPS 등과 연계하면 실시간 사진측량에서의 많은 응용이 가능할 것으로 사료된다.

## 참 고 문 헌

1. Andrews, M., *Visual C++ Object-Oriented Programming*, SAMS Publishing, 1993.
2. Stroustrup, B., *The C++ Programming Language*, Addison-Wesley, 1986.
3. 유복모, 이석군, 김원대, 김의명, "C++ 언어를 이용한 객체지향 사진측량시스템 구축에 관한 연구", 토목학회 학술발표회 논문집(III), pp. 163-166, 1997
4. Adamos, C., and Faig, W. "Hough Transform in Digital Photogrammetry", *ISPRS*, Vol. XXIX, part B3, Commission III, pp. 250-254, 1992.
5. Xu, L., Oja, E., and Kultanen, P. "A new curve detection method : Randomized Hough Transform(RHT)", *Pattern Recognition Letters*, Vol. 11, No. 5, pp 331-338. 1990.
6. T. Schenk, "Concepts in Digital Photogrammetry", pp. 115-148, 1996.
7. Konecny, G. and D. Pope, "Correlation Technique and Devices", *P.E.& R.S.*, Vol 47, No.3, 1981, pp.323-333.
8. J. C. Li, T. Schenk, "Determination of Transformation Parameters between a Photograph and Its Digital Image through Camera Calibration", The Ohio State University, 1990.