

# VASM 알고리즘을 이용한 건축물 CAD 자료의 수치지도 건물 객체와의 형상 정합 및 지도좌표 부여 방법의 제안

## A Proposal of a Shape Matching and Geo-referencing method for Building Features in Construction CAD Data to Digital Map using a Vertex Attributed String Matching algorithm

허용<sup>1)</sup> · 유기윤<sup>2)</sup> · 김형태<sup>3)</sup>

Huh, Yong · Yu, Ki-Yun · Kim, Hyung Tae

### Abstract

An integration between construction CAD data and GIS data needs geo-referencing processes of construction CAD data whose coordinate systems are their own native or even unknown. Generally, these processes are based on manually detected conjugate-vertices. In this study, we proposed an semi-automated conjugate -vertices detection method for building features between construction CAD data and a digital map using a vertex attributed string matching algorithm. A geo-referencing function for construction CAD data based on the similarity transform could be derived with those conjugate-vertices. Using our proposed method, we overlaid geo-referenced CAD data to a digital map of the College of Engineering, Seoul National University and evaluated our method.

Keywords : Conjugate-Vertices, Geo-referencing, Construction CAD data, Digital map, VASM algorithm

### 초 록

건설 CAD 자료와 GIS 자료를 연계하기 위해서는 임의의 좌표체계로 표현되거나 경우에 따라 좌표정보를 가지고 있지 않은 CAD 자료에 지도좌표를 부여하는 과정이 필요하다. 일반적으로 이러한 과정들은 수작업에 의하여 결정된 공액 꼭지점을 이용하지만 많은 시간이 소요되는 문제점을 가지고 있다. 본 연구는 VASM 알고리즘을 이용하여 건설 CAD 자료와 수치지도에서 건물 객체의 형상 정합을 수행함으로써 공액 꼭지점을 반자동 추출할 수 있는 기법을 제안한다. 이렇게 추출된 공액 꼭지점을 이용하여 사상변환에 기반한 지도좌표 부여 함수를 유도할 수 있었다. 본 연구에서 제안한 기법을 이용하여 지도좌표가 부여된 서울대학교 공과대학 건물들의 건설 CAD 도면을 수치지도에 중첩해보았고, 그 결과를 바탕으로 제안된 기법을 평가해 보았다.

핵심어 : 공액 꼭지점, 지도좌표부여, 건설 CAD 자료, 수치지도, VASM 알고리즘

## 1. 서 론

제3차 국가 GIS 사업을 통하여 공간정보의 활용을 고도화하고 다양한 공간 및 비공간정보를 연계하여 제공함으로써 통합적인 국토공간정보의 활용을 위한 움직임이 활발하게 진행되고 있다. 이러한 변화 속에서 건설도면과

GIS 자료를 연계함으로써 공간정보에 대한 기존 개념 및 적용범위를 획기적으로 확대시키기 위한 다양한 시도가 이루어지고 있다. 즉 지형지물과 같은 거시적인 공간정보를 개별 시설물의 정확한 구조 및 내부를 표시하는 건설도면과 같은 미시적인 공간정보와 연계함으로써 실내의 위치기반 서비스와 같은 새로운 활용을 가능케 할 수 있다.

1) 정희원 · 서울대학교 공과대학 건설환경공학부 박사과정(E-mail:hy7808@snu.ac.kr )

2) 연결저자 · 정희원 · 서울대학교 공과대학 건설환경공학부 부교수(E-mail:kiyun@snu.ac.kr)

3) 한국토지공사 책임연구원(E-mail:kht@iklc.co.kr )

Ayman과 Dunham(1999)은 위와 같은 기술적 발전을 위해서는 CAD자료로 구축된 건설도면과 GIS 자료간의 원활한 변환이 선행되어야 한다는 점을 지적하였으나, CAD와 GIS 자료는 본래의 활용 목적에 따라 동일한 실세계의 객체일지라도 위치 및 형상의 표현에 있어 차이를 피할 수 없다는 문제가 해결되어야 한다. 이에 따라 이들 자료들의 연계를 위한 자료 구축에 있어 명시적인 표준이 제공되어야 하나 아직 미흡한 실정이다. 특히 CAD로 작성된 건물 공간정보의 대부분은 실세계의 좌표를 추정할 수 없는 임의의 좌표계로 작성되어 있는 경우가 많아 건축물의 지번이나 주소 등을 통하여 CAD 도면상의 공간 객체가 지도상의 어느 건물을 표현한다는 것은 추정할 수 있지만 정확한 위치로 GIS 자료상에 중첩을 할 수 없는 것이 현실이다.

일반적으로 건설 CAD 자료의 좌표 문제를 해결하기 위하여 수작업에 의하여 결정된 공액 꼭지점을 이용하여 좌표변환식을 유도하지만 많은 시간이 소요되며, 사용자에게 주관적인 해석에 따라 균일하지 못한 결과를 가져오는 문제점을 가지고 있다. 이러한 문제를 해결하기 위하여 본 연구에서는 문자열 정합(string matching) 알고리즘에 기반한 형상 정합 알고리즘의 하나인 VASM (vertex attributed string matching) 알고리즘을 이용하여 건설 CAD 자료와 수치지도에서 폴리곤으로 표현된 동일한 건물 객체의 공액 꼭지점을 반자동 추출하고, 이렇게 추출된 정보를 이용하여 상사변환에 근거한 좌표변환식을 도출하는 기법을 제안하였다. 제안된 기법을 적용 및 평가하기 위하여 서울대학교 공과대학의 9개 건물의 건설 CAD 도면을 1:1,000 수치지도와 연계될 수 있도록 개별 건물의 외곽선 폴리곤을 각각의 자료에서 추출하고 공액 꼭지점을 탐색, 이를 이용하여 좌표부여함수식을 도출하고 건설 CAD 도면을 수치지도에 중첩하여 평가해보았다.

## 2. 본 연구의 건설 CAD 자료 지도좌표 부여 과정

본 연구에서는 제안하는 건설 CAD 자료 지도좌표 부여 과정 및 기법은 아래 그림 1과 같이 세 단계로 구성된다. 먼저 GIS와 CAD 자료에서 동일 건물의 외곽 폴리곤을 추출하는 전처리 과정이 수행된다. 이 과정에서 건설 도면 레이어명의 표준화 미비 및 수치지도의 개별 건물을 고유하게 인식할 수 있는 방법에 있어 현실적인 한계로

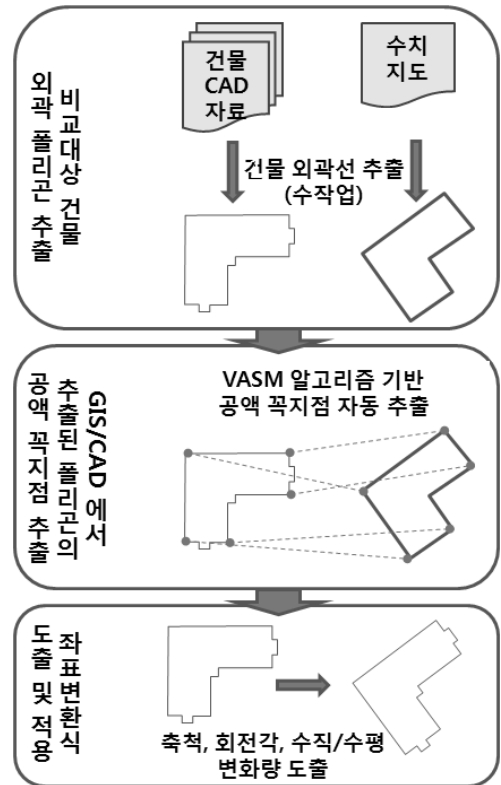


그림 1. 제안된 건설 CAD 자료 지도좌표 부여 과정

인하여 수작업이 필요하다. 표준화되지 않은 건설 CAD 자료 레이어명으로 인하여 건물의 외곽선을 표현하는 하나의 폴리곤을 추출하는 과정은 사용자의 레이어 선택은 물론 경우에 따라 편집 과정이 필요할 수 있다. 수치지도의 경우에도 학교와 같이 동일한 명칭으로 많은 건물이 표현되어 있거나 단순히 주기(annotation)를 해당 건물 내에 표시함으로써 개별 건물의 동변이 객체 기반으로 제공되지 않는 경우 역시 사용자의 검색 및 편집과정을 통하여 건물 외곽선을 추출해야 한다.

이렇게 추출된 두 건물 외곽선에 VASM 알고리즘을 적용하여 형상 정합을 수행한다. 원래의 VASM 알고리즘은 어떤 한 폴리곤 형상을 비교대상이 되는 다른 폴리곤 형상으로 변환하는데 있어 폴리곤을 구성하는 꼭지점의 편집 연산 과정과 그 비용을 얻기 위한 알고리즘이다. 본 연구에서는 편집 연산 과정에서 비교 대상 폴리곤의 꼭지점과 교환되는, 즉 꼭지점간의 대응 관계가 성립할 경우 이들 꼭지점을 공액 꼭지점으로 사용할 수 있다는 가정하에 연구를 수행하였다(Kruskal, 1983). 이렇게 도

출된 공액 꼭지점을 이용한 좌표변환을 수행하기 위하여 축척, 회전각, 수직/수평 변화량을 계산하고 원시 건물 CAD 자료에 적용함으로써 건설 CAD 자료 지도좌표 부여 과정이 완료된다.

### 3. 문자열 정합을 이용한 폴리곤 정합 기법

VASM 알고리즘은 문자열 정합을 이용한 폴리곤 정합 기법을 바탕으로 개발되었다. 문자열 정합은 두 개의 연속된 문자열  $X$ 와  $Y$ 가 있을 때  $X$ 의 문자열을 문자열  $Y$ 로 변환하는 데 필요한 최소한의 편집 거리(edit distance), 즉 변환에 필요한 연산 비용을 계산한다(Duda 등, 2001). 일반적으로 변환에 사용되는 연산에는 삭제(delete), 교환(substitution), 삽입(insert)의 세 가지 종류가 있다. 이들 세 가지 연산의 비용은 연산이 수행될 문자쌍  $x_i, y_j$ 의 속성값에 따라 미리 정의된 함수에 의하여 결정되며, 변환이 완료되었을 때 적용된 모든 편집 연산의 총비용으로 두 문자열의 편집 거리가 결정된다.

Wagner와 Fischer(1974)는 동적 프로그래밍 기법을 적용하여 효과적으로 문자열을 정합할 수 있는 기법을 개발하였다. 이 기법은 두 개의 문자열  $X=x_1 \cdots x_n$ 와  $Y=y_1 \cdots y_m$ 이 있을 때, 행과 열의 개수가 각각  $(n+1)$ 와  $(m+1)$ 인 누적 비용 행렬  $D(i, j)$ 를 계산한다. Wagner와 Fischer가 제안한 알고리즘의 기본 아이디어는  $D(0,0)$ 에서  $D(n,m)$ 까지의 최소 비용 경로를 순차적으로 탐색하는 것이며, 이때  $D(i, j)$ 의 원소값은  $X' = x_1 \cdots x_i$ 를  $Y' = y_1 \cdots y_j$ 로 변환하는 최소 누적 비용을 의미한다.

$D(i, j)$ 의 원소값, 즉 누적 비용을 결정하는 방법은 다음과 같다. 문자열 정합과정에서  $D(i, j)$ 의 누적 비용을 결정하기 위해서는  $x_i$ 와  $y_j$ 간의 연산이 발생할 수 있는 모든 경로와 그에 해당하는 경로별 마지막 누적 비용이 필요하다. 삭제연산일 경우  $D(i-1, j)$ 의 누적 비용에  $x_i$ 의 삭제연산 비용을 더하여  $D(i, j)$ 에 저장하고, 교환연산에 의하여  $D(i, j)$ 의 값이 결정된다면  $D(i-1, j-1)$ 의 누적 비용에  $x_i$ 와  $y_j$ 간의 교환연산의 비용을, 삽입연산일 경우  $D(i, j-1)$ 의 누적 비용에  $y_j$ 의 삽입연산의 비용 더하면 된다. 이렇게 세 가지의 잠재 누적 비용 중에서 가장 작은 값을 가지는 연산이 선택되어 누적 비용이  $D(i, j)$ 에 저장된다. 물론 이와 같은 연산을 수행하기 위해서는 초기값이 미리 결정되어야 한다. 이때의 초기값이란  $D(0,0)$ ,  $\{D(i, 0)|i=1, \dots, n\}$ ,  $\{D(0, j)|j=1, \dots, m\}$ 을 말한다. 이렇게 모

든 행렬의 위치에서 누적 비용을 계산한 후에  $D(n, m)$ 에서 시작하여  $D(0,0)$ 로 경로를 역행함으로써 실제 편집 연산의 순서를 알 수 있으므로 두 문자열 사이의 유사도 뿐만 아니라  $x=x_1 \cdots x_n$ 의 문자열이  $y=y_1 \cdots y_m$ 의 문자열로 변환하는 과정도 알 수 있다.

문자열을 구성하는 개별 문자, 즉 단위요소(primitive)에  $n$ 차원의 벡터 자료를 대입시킨 ASM(attributed string matching)은 폴리곤 정합에 다양하게 활용되어 왔다. Tsai와 Yu(1985)는 폴리곤으로 표현된 2차원 객체의 외곽선을 구성하는 선분을 단위요소로 이용하였으며, 선분의 길이와 방향을 속성값으로 활용하였다. 이후 다양한 분야에서 폴리곤 객체의 선분을 단위요소로 하는 ASM에 대한 연구가 수행되었다(Maes, 1991; Bunke와 Buhler, 1993). 하지만 이러한 방법들은 편집 거리를 통하여 형상의 유사도를 측정할 수는 있지만 어떤 선분에 편집 연산이 수행됐을 경우 앞뒤 선분과의 연결이 단절되는 문제점을 가지고 있었다. 즉 두 폴리곤의 어떤 선분  $x_i$ 와  $y_j$ 에 대하여 교환연산이 수행되었다면, 교환된  $y_j$ 의 시작점과 끝점은 각각  $x_{i-1}$ 의 끝점과  $x_{i+1}$ 의 시작점과 동일해야 하지만 이에 대한 고려가 반영되지 못하였다. Kaygin과 Bulut(2002)은 폴리곤 객체의 꼭지점을 단위요소로, 연속된 꼭지점 사이의 거리와 연산을 수행할 꼭지점을 기준으로 전후 선분의 방향각 변화량을 속성으로 이용하는 VASM 알고리즘을 제안하였다. 이 방법은 폴리곤을 구성하는 꼭지점의 위치를 직접 변화시킴으로써 연산 후 발생할 수 있는 선분의 연결성 문제를 해결할 수 있었다.

### 4. VASM 알고리즘

VASM 알고리즘도 Wagner와 Fischer가 제시한 알고리즘과 기본적인 구조는 동일하다. 단, 문자열의 문자에 해당하는 단위요소를 어떻게 설계하는가, 단위요소의 속성으로 어떤 값들을 사용하는가, 그리고 삭제, 교환, 삽입 연산에 대한 비용 함수를 어떻게 정의하는가에 따라 다양한 변형이 가능하다. 폴리곤 정합을 위한 VASM 알고리즘은 폴리곤 객체의 꼭지점을 단위요소로 이용하며, 순차적으로 나열된 꼭지점별로 이전 순서의 꼭지점과 함께 생성하는 선분의 길이 및 방향을 속성으로 이용한다. 본 연구에서는 Kaygin과 Bulut(2002)의 연구에서 제안한 비용 함수를 기반으로 다음과 같이 개선하여 사용하였다.

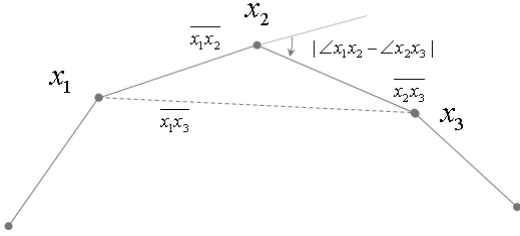


그림 2. VASM 알고리즘의 삭제연산

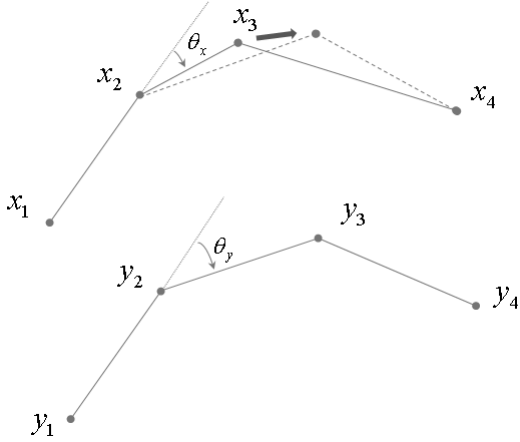


그림 3. VASM 알고리즘의 교환연산

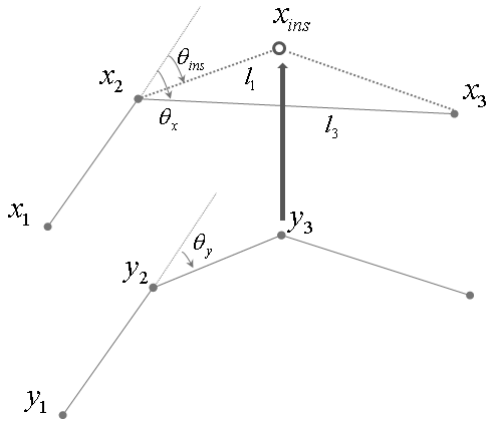


그림 4. VASM 알고리즘의 삽입연산

#### 4.1 삭제연산의 비용함수

삭제연산은 그림 2와 같이  $x_1x_2x_3$  라는 꼭지점 3개가 연속적으로 위치할 때, 꼭지점  $x_2$ 를 삭제함으로써 선분  $\overline{x_1x_2}$ 와  $\overline{x_2x_3}$ 을 새로운 선분  $\overline{x_1x_3}$ 로 변환시키는 결과를 가져온다.

이 연산의 비용함수는 식 (1)과 같이 새로운 선분으로 인한 길이와 방향의 변화량에 가중치를 부여하여 결정한다.

$$f_{delete}(x_1, x_2, x_3, L_X) = \sqrt{N_X N_Y} \frac{|\overline{x_1x_2}| + |\overline{x_2x_3}| - |\overline{x_1x_3}|}{L_X} + w \frac{|\angle x_1x_2 - \angle x_2x_3|}{180} \quad (1)$$

여기서  $L_X$  : 폴리곤  $X$ 의 둘레길이

$N_X, N_Y$  : 각각 폴리곤  $X$ 와  $Y$ 의 꼭지점 개수

$\angle x_i x_j$  : 선분  $\overline{x_i x_j}$ 의 방향각

$w$  : 선분과 각도에 의한 비용을 결합하기 위한 가중치

위의 식에서 가중치  $w$  최적값을 결정하기 위하여 0부터 1사이의 계수를 각각 적용하여 평가해 보았다. 그 결과 본 연구에서는 0.5를 최적 가중치로 결정하였다.

#### 4.2 교환연산의 비용함수

교환연산은 그림 3과 같이 폴리곤  $Y$ 의 꼭지점  $y_3$ 의 위치를 이용하여 폴리곤  $X$ 의 꼭지점  $x_3$ 를 새로운 위치로 이동시키는 결과를 가져온다. 이를 위해서는  $x_3$ 의 속성인  $\overline{x_2x_3}$ 의 길이와  $\theta_x$ 를 폴리곤  $Y$ 의 꼭지점  $y_3$ 의 속성을 참조하여 수정해야 하며, 비용함수는 식 (2)와 같이 결정한다.

$$f_{subst}(x_1, x_2, x_3, L_X, y_1, y_2, y_3, L_Y) = \sqrt{N_X N_Y} \left| \frac{|\overline{x_2x_3}|}{L_X} - \frac{|\overline{y_2y_3}|}{L_Y} \right| + w \frac{|\theta_x - \theta_y|}{180} \quad (2)$$

여기서  $\theta_x : \angle x_1x_2 - \angle x_2x_3$

$\theta_y : \angle y_1y_2 - \angle y_2y_3$

#### 4.3 삽입연산의 비용함수

삽입연산은 폴리곤  $X$ 에 원래 존재하지 않던 새로운 꼭지점  $x_{ins}$ 를 생성한다. 그림 4에서와 같이 꼭지점  $x_3$ 와  $y_3$ 를 비교한 결과  $y_3$ 에 대응되는 위치에 새로운 꼭지점  $x_{ins}$ 를  $x_2$ 와  $x_3$ 사이에 삽입하는 결과를 가져온다. 이때 비용함수를 계산하기 이전에  $x_2$ 와  $x_3$ 사이에 새로운 꼭지점을 생성하며 이때 생성의 비용은 생성될 꼭지점을 삭제했을 때의 비용으로 결정한다.

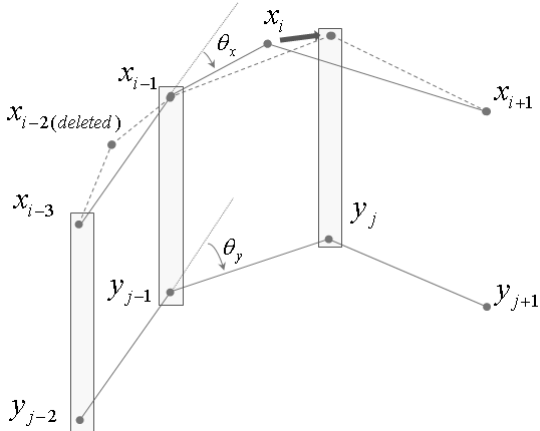


그림 5. 전단계의 연산을 고려한 교환연산

이 때 삭제될 꼭지점의 위치는 식 (3)을 이용하여 거리 ( $l_1$ )와 방향( $\theta_{ins}$ )을 이용하여  $x_2$ 로부터 계산한다.

$$\theta_{ins} = \begin{cases} \frac{|\theta_y(l_3 - l_1) + \theta_x l_1|}{l_3} & (l_3 \geq l_1) \\ \frac{|\theta_y(l_1 - l_3) + \theta_x l_3|}{l_1} & (l_3 < l_1) \end{cases} \quad (3)$$

여기서  $l_1 : (L_X/L_Y)l_y$

이렇게 생성된  $x_{ins}$ 를  $y_3$ 으로 교환함으로써 삽입연산이 완료된다. 이러한 삭제와 교환 연산을 반영한 삽입연산의 비용함수는 식 (4)와 같이 결정한다.

$$\begin{aligned} f_{insert}(x_1, x_2, x_3, L_X, y_1, y_2, y_3, L_Y) \\ = f_{delete}(x_2, x_{ins}, x_3, L_X) \\ + f_{subst}(x_1, x_2, x_{ins}, L_X, y_1, y_2, y_3, L_Y) \end{aligned} \quad (4)$$

#### 4.4 VASM 알고리즘의 구조 및 과정

VASM 알고리즘의 편집 연산 비용은 모두 연속된 꼭지점  $y_1, y_2, y_3$ 과 그에 대응되는 꼭지점  $x_1, x_2, x_3$ 을 이용하여 계산된다. 문자열 정합은 기본적으로 문자열  $X$ 를 문자열  $Y$ 로 변환하기 위한 최소한의 편집 연산과정과 총비용을 구하는 과정이다. 따라서 각 편집 연산식에서  $y_1, y_2, y_3$ 는 폴리곤  $Y$ 의 임의의 위치에서 연속되는 세 개의 꼭지점을 의미하지만  $x_1, x_2, x_3$ 은 경우에 따라 폴리곤  $X$ 의 어떤 위치에서 연속되는 세 개의 꼭지점이 아닌 이

전에 수행된 연산에 따라 다양한 경우가 발생할 수 있다.

그림 5와 같이  $x_i$ 와  $y_j$ 사이에 교환연산을 수행할 때, 이전의 연산 과정에서  $x_{i-3}$ 과  $y_{j-2}$ 사이에 교환연산을 수행하고  $x_{i-2}$ 의 삭제연산,  $x_{i-1}$ 과  $y_{j-1}$ 의 교환연산이 순차적으로 수행되었다고 가정한다. 이때  $x_{i-2}$ 의 삭제연산이 수행되었으므로  $\theta_x$ 는  $\angle x_{i-2}x_{i-1} - \angle x_i x_{i-1}$ 가 아닌  $\angle x_{i-3}x_{i-1} - \angle x_i x_{i-1}$ , 즉  $y_{j-2}$ 와  $y_{j-1}$ 에 대응되는  $X$ 의 꼭지점들을 이용해야 한다.

따라서 VASM 알고리즘에서는 누적 비용 행렬  $D$ , 수행된 연산의 종류를 나타내는 편집 연산 행렬  $T$  이외에 연산 경로상에서 한 단계 이전의  $Y$ 의 꼭지점에 대응되는  $X$ 의 꼭지점의 위치를 저장하는 행렬  $P$ , 그리고 두 단계 이전의  $Y$ 의 꼭지점에 대응되는  $X$ 의 꼭지점의 위치를 저장하는 행렬  $Q$ 가 추가적으로 필요하다. 그림 5와 같은 연산을 수행한다면  $x_i$ 와  $y_j$ 사이에 교환연산을 수행할 때  $P(i-1, j-1)$ 에는  $x_{i-1}$ 의 위치와 속성, 그리고  $Q(i-1, j-1)$ 에는  $x_{i-3}$ 의 위치와 속성이 저장되어 있어야 한다.

표 1은 이와 같은 조건을 반영한 VASM 알고리즘의 의사코드이다. 행02에서 행15는  $D(0, 0)$ ,  $\{D(i, 0) | i=1, \dots, n\}$ ,  $\{D(0, j) | j=1, \dots, m\}$ 의 초기값을 결정하는 과정을 나타낸다.  $D(i, j)$ 과  $T(i, j)$ 은  $X' = x_1, \dots, x_i$ 를  $Y' = y_1, \dots, y_j$ 로 변환하는 총비용과 수행된 편집연산의 과정을 의미한다. 이때  $D(i, j)$ 과  $T(i, j)$ 을 결정하는 잠재 경로는 다음과 같이 세 가지가 존재한다.

- ① 삭제연산 :  $x_1, \dots, x_{i-1}$ 을  $y_1, \dots, y_j$ 로 변환하는 총비용과 그 과정에  $x_i$ 를 삭제하는 비용과 연산을 추가
- ② 교환연산 :  $x_1, \dots, x_{i-1}$ 을  $y_1, \dots, y_{j-1}$ 로 변환하는 총비용과 그 과정에  $x_i$ 를  $y_j$ 로 교환하는 비용과 연산을 추가
- ③ 삽입연산 :  $x_1, \dots, x_i$ 을  $y_1, \dots, y_{j-1}$ 로 변환하는 총비용과 그 과정에  $y_j$ 에 대응되는  $x_{ins}$ 를 삽입하는 비용과 연산을 추가

이들 세 가지 연산중 총비용이 가장 적은 연산이 선택되어  $D(i, j)$ 과  $T(i, j)$ 에 저장된다.

### 5. 공액 꼭지점을 이용한 좌표변환식 도출

본 연구에서 이용한 좌표 변환은 상사변환으로 Schaefer 등(2006)의 연구에서 설명하는 과정을 따라 축척, 회전량,

표 1. VASM 알고리즘 의사코드

```

01 function VASM (Polygon X, Polygon Y)
02 T(0,0) = 0;           % substitution
03 D(0,0) = 0;
04 P(0,0) = X(n);       Q(0,0) = X(n-1);
05 for i=1:n
06     T(i,0) = -1;      % delete
07     D(i,0) = D(i-1,0) + f_delete( P(i-1),X(i),X(i+1),L_X );
08     Q(i,0) = Q(i-1,0);   P(i,0) = P(i-1,0);
09 end
10 for j=1:m
11     T(0,j) = 1;       % insert
12     X_ins = V_ins( Q(0,j-1),X_ins,A(1),L_X,Y(j-2),Y(j-1),Y(j),L_Y );
13     D(0,j) = D(0,j-1) + f_delete( P(0,j-1),X_ins,X(1),L_X ) + F_subst( Q(0,j-1),P(0,j-1),X_ins,L_X,Y(j-2),Y(j-1),Y(j),L_Y );
14     Q(0,j) = P(0,j-1);   P(0,j) = X_ins;
15 end
16 for i=1:n
17     for j=1:m
18
19         D(1) = D(i-1,j-1) + f_subst( Q(i-1,j-1),P(i-1,j-1),X(i),L_X,Y(j-2),Y(j-1),Y(j),L_Y );
20         T(1) = 0;           Q(1) = P(i-1,j-1);   P(1) = X(i);
21
22         D(2) = D(i-1,j) + f_delete( P(i-1),X(i),X(i+1),L_X );
23         T(2) = -1;           Q(2) = Q(i-1,j);     P(2) = P(i-1,j);
24
25         X_ins = V_ins( Q(i,j-1),P(i,j-1),X(i+1),L_X,Y(j-2),Y(j-1),Y(j),L_Y );
26         D(3) = D(i,j-1) + f_delete( P(i,j-1),X_ins,X(i+1),L_X ) + f_subst( Q(i,j-1),P(i,j-1),X_ins,L_X,Y(j-2),Y(j-1),Y(j),L_Y );
27         T(3) = 1;           Q(3) = P(i,j-1);     P(3) = X_ins;
28
29         if (i=n)and(j=m)
30             k=1;
31         else
32             k=argmin( D(1),D(2),D(3) );
33         end
34         D(i,j) = D(k); T(i,j) = T(k); Q(i,j) = Q(k); P(i,j) = P(k);
35     end
36 end

```

수평/수직 변화량을 도출하였다. 먼저 좌표 변환식은 일  
반적인 부등각 상사변환식인 (5)와 같이 정의된다.

$$\sum_i |(p_i - \bar{p})M - (q_i - \bar{q})|^2 \quad (6)$$

$$f(x) = xM + T \quad (5)$$

여기서  $x$  : 건축도면의 특정 좌표 열벡터  
 $f(x)$  : 수치지도의 대응 지점 좌표의 열벡터  
 $M$  : 균일한 축척이 반영된 회전행렬  
 $T$  : 열벡터로 표현된 수직/수평 이동량

여기서  $p_i$  :  $i$ 번째 공액 꼭지점의 CAD 좌표  
 $\bar{p}$  :  $p_i$ 의 평균  
 $q_i$  :  $i$ 번째 공액 꼭지점의 수치지도 좌표  
 $\bar{q}$  :  $q_i$ 의 평균

이 때 최소제곱법은 식 (6)에 대한 최소값을 만족시키  
는  $M$ 을 얻는 과정을 의미한다.

상사변환이라는 조건을 만족시키기 위해서는 어떤 스  
칼라 값  $\lambda$ 에 대하여  $M^T M = \lambda^2 I$ 인 특징을 가져야한다.  
이때  $M = (M_1, M_2)$ 를 만족하는  $M_1, M_2$ 가 길이가 2인 열

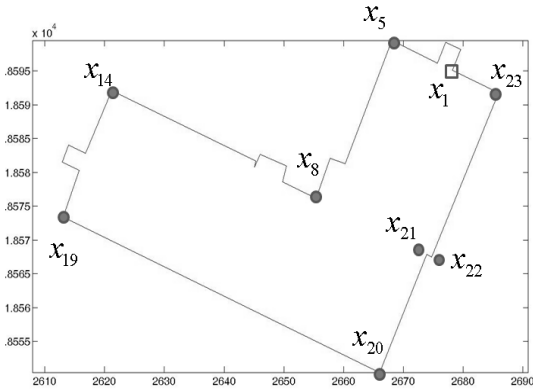


그림 6. 건설도면 CAD 자료로부터 추출된 건물 외곽선 폴리곤

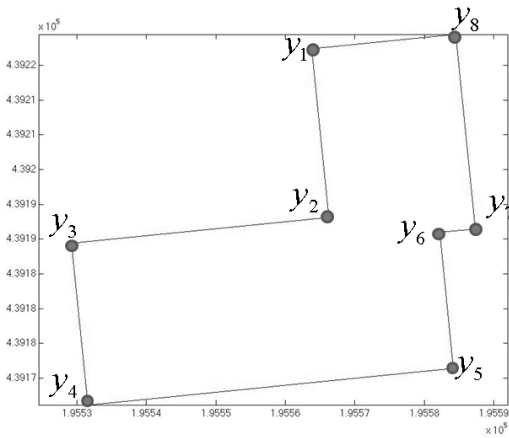


그림 7. 수치지도로부터 추출된 건물 외곽선 폴리곤

벡터일 때  $M$ 은  $M_1^T M_1 = M_2^T M_2 = \lambda^2$  이고,  $M_1^T M_2 = 0$  인 특성을 가져야 한다. 이러한 제약조건은  $\perp$  가 2차원 자료에서  $(x, y)^\perp = (-y, x)$  와 같은 연산자일 때,  $M_2 = M_1^\perp$  을 의미한다(Schaefer 등, 2006). 위 특성을 이용하면 식 (6) 은 다시 아래와 같은 식 (7)로 표현될 수 있다.

$$\sum_i \left| \begin{pmatrix} p_i - \bar{p} \\ -(p_i - \bar{p})^\perp \end{pmatrix} M_1 - (q_i - \bar{q})^T \right|^2 \quad (7)$$

이 2차 함수는 단일해를 가지게 되는데 이것은 최적의 변환 행렬  $M$ 을 도출한다. 이렇게 얻은  $M$ 을 식 (5)에 대입하여  $T$ 를 계산하면 사상 변환에 필요한 모든 상수를 도출하게 된다.

$$M = \frac{\sum_i \begin{pmatrix} p_i - \bar{p} \\ (p_i - \bar{p})^\perp \end{pmatrix} \left( (q_i - \bar{q}) \quad (q_i - \bar{q})^\perp \right)^T}{\sum_i (p_i - \bar{p})(p_i - \bar{p})^T} \quad (8)$$

## 6. 적용 및 평가

### 6.1 VASM 알고리즘을 이용한 공액 꼭지점 추출

본 연구의 VASM 알고리즘을 이용한 공액 꼭지점 도출 및 지도좌표부여에 사용된 자료는 서울대학교 공과대학 9개 건물의 1:1,000 수치지도 건물과 건설도면 CAD 자료 건물 외곽선이다. 그림 6과 그림 7은 적용된 건물 중 38동 건물의 건설도면 CAD 자료로부터 추출된 건물 외곽선 폴리곤과 수치지도로부터 추출된 건물 외곽선 폴리곤을 보여준다.

동일한 실세계의 객체일지라도 두 그림에서 알 수 있듯이 표현상에 있어 위치, 축척, 방향이 모두 상이한 상태이며, 건설도면상에서 일반화 과정을 거친 외곽선 폴리곤은 23개의 꼭지점으로 구성되어 있지만 수치지도에서는 8개의 꼭지점으로 표현되어 있다. 이들 꼭지점에 VASM 알고리즘을 통하여  $D$  행렬과  $T$  행렬을 얻은 다음  $T(23,8)$ 에서 시작하여 대응되는 연산에 따라  $T(0,0)$ 까지의 편집연산을 역순으로 탐색한 결과는 표 2와 같다. 표에서 각 열에서 교환연산이 선택된 위치의  $x_i$ 와  $y_j$ 가 대응되는 공액 꼭지점이 되며, 그림 6과 7에 표시하였다.

### 6.2 공액 꼭지점을 이용한 좌표부여 결과

그림 8은 교환연산이 수행된  $x_i, y_j$  공액 꼭지점을 이용하여 상기 좌표 변환식을 도출함으로써 건설도면 CAD 자료와 수치지도를 중첩한 결과를 보여준다. 시각적으로 평가해 보았을 때, 사상 변환에 있어 이동과 회전에 있어서는 타당한 결과를 얻었지만 건물폭에 있어서는 차이가 발생하였다. 이는 수치지도 작성에서 있어 지붕의 형상이 실제 건물의 실내의 구조 및 형상보다 더 큰 영향을 미쳤기 때문으로 추정된다. 또한 수치지도와 건설도면에서 실 객체를 표현하는데 있어 요구되는 위치와 형상 정확도의 차이도 영향을 미친 것으로 판단된다. 역시 나머지 8개 건물에 대하여 동일한 방법으로 적용한 결과 그림 9와 같이 대부분의 경우 건설도면에 올바른 지도좌표가 부여되었음을 확인할 수 있었다.

하지만 그림 9의 결과를 세부적으로 검토할 경우 두 가지 문제점을 찾을 수 있었다. 첫째, 건설도면 작성 시 본

표 2. 건설도면과 수치지도의 건물 외곽선에 VASM 알고리즘을 적용했을 때의 D행렬과 T행렬(T(23,8)에서 시작하여 T(0,0)까지 수행된 연산 과정과 비용)

	0	1	2	3	4	5	6	7	8
0	0.000	2.174	4.706	8.905	10.548	15.246	16.108	17.430	18.926
	0	1	1	1	1	1	1	1	1
1	0.413	0.841	3.148	6.338	9.878	12.658	15.411	16.465	17.924
	-1	0	0	0	0	0	1	1	0
2	0.483	1.195	2.703	5.828	7.932	11.873	13.888	15.611	17.798
	-1	-1	0	0	0	-1	0	1	0
3	1.010	1.173	2.894	5.428	7.168	12.164	13.069	14.556	17.723
	-1	0	0	0	0	0	0	0	0
4	1.011	1.586	2.582	5.721	6.726	11.359	13.222	13.223	16.158
	-1	1	0	0	0	0	0	1	0
5	1.950	<b>1.194</b>	2.475	4.470	5.625	10.120	10.795	11.949	15.107
	-1	<b>0</b>	1	0	1	0	1	1	0
6	2.092	1.560	1.585	4.721	4.798	8.939	10.208	11.044	13.739
	-1	-1	0	0	0	0	0	1	0
7	2.342	1.589	1.617	3.827	5.151	8.546	10.056	10.392	12.921
	-1	-1	1	0	-1	0	0	0	0
8	2.597	2.115	<b>1.616</b>	4.225	4.960	8.813	9.439	9.449	11.830
	-1	-1	<b>0</b>	0	0	0	0	1	0
9	3.178	2.824	1.948	3.653	5.292	8.485	9.771	9.478	11.474
	-1	-1	-1	0	0	0	0	0	0
10	3.335	3.163	2.027	3.970	5.371	8.714	9.850	9.816	11.616
	-1	-1	-1	0	0	-1	0	-1	0
11	3.509	3.238	2.420	3.758	5.376	8.555	9.879	10.231	11.612
	-1	-1	-1	0	0	0	0	0	0
12	3.930	3.743	2.695	4.147	5.149	9.128	9.705	10.560	11.926
	-1	-1	-1	0	0	0	0	0	0
13	4.236	4.374	2.712	3.510	5.427	8.770	8.805	9.446	12.204
	-1	-1	-1	1	-1	0	1	1	-1
14	4.734	4.757	3.489	<b>2.870</b>	5.101	7.763	9.089	9.062	11.313
	-1	-1	-1	<b>0</b>	0	0	1	0	0
15	4.845	4.917	3.531	3.267	3.760	8.159	8.412	9.201	10.266
	-1	-1	-1	-1	0	0	0	1	0
16	5.092	5.113	3.867	3.324	4.110	7.473	8.762	8.561	10.616
	-1	-1	-1	-1	-1	0	-1	0	0
17	5.843	5.817	4.680	3.823	4.063	7.824	8.571	8.911	10.204
	-1	-1	-1	-1	0	0	0	0	0
18	6.210	6.086	5.177	3.834	4.446	8.003	8.954	9.091	10.587
	-1	-1	-1	-1	-1	0	0	1	0
19	11.761	9.437	8.246	5.086	<b>4.129</b>	5.129	8.681	9.427	9.607
	-1	0	0	0	<b>0</b>	1	1	1	1
20	14.520	11.122	9.931	7.105	5.814	<b>5.157</b>	6.610	7.700	10.208
	-1	-1	-1	-1	-1	<b>0</b>	1	1	0
21	14.838	11.175	9.984	7.108	5.867	5.963	<b>6.286</b>	7.917	8.780
	-1	-1	-1	-1	-1	-1	<b>0</b>	0	0
22	18.508	12.782	11.591	9.100	7.475	5.971	6.981	<b>7.561</b>	9.503
	-1	-1	-1	-1	-1	-1	1	<b>0</b>	1
23	18.943	14.061	12.870	10.615	8.754	6.702	7.686	8.249	<b>8.782</b>
	-1	-1	-1	-1	-1	-1	-1	-1	<b>0</b>



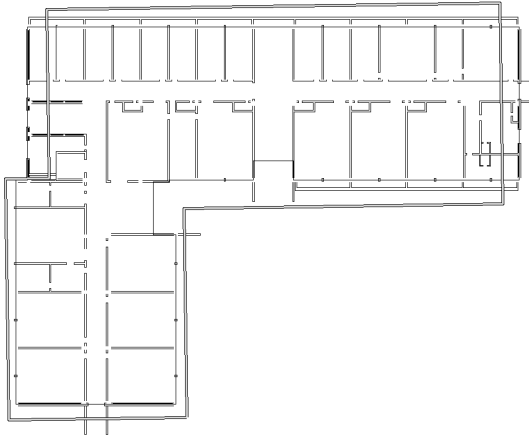


그림 8. 공액 좌표를 이용하여 건설도면을 수치지도 건물 외곽선(이중실선)에 좌표 변환하여 투영한 결과

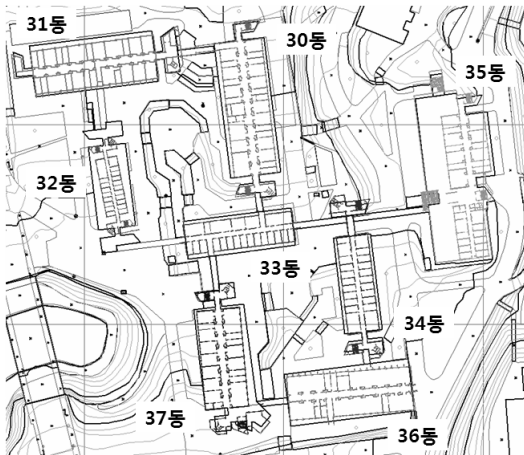


그림 9. 지도좌표가 부여된 8개 건물 건설도면의 1:1,000 수치 지도 중첩결과

건물과 함께 계단이나 복도 및 부속건물이 같이 표현되어 있지만 수치지도상에서 본건물과 부속건물 또는 연결 복도가 별도의 객체로 표현되어 있는 경우 사용자의 편집과정에 따라 공액 꼭지점 추출에 큰 영향을 미치게 된다. 그림 9의 좌측 중간에 위치한 32동 건물의 경우 설계도면상에서는 주건물과 상하의 두 부속건물이 하나의 폴리곤을 구성하였으나, 수치지도상에서 주건물만이 하나의 객체로 32동 건물로 표현되어 있었다. 그 결과 그림 10의 (a)와 같이 두 개의 부속건물과 주건물이 구성하는 큰 폴리곤이 수치지도상의 주건물만을 표현하는 폴리곤에 형상 정합이 되는 결과를 얻게 되었다. 둘째, 직사각형과 같은 회전 대칭성(rotational symmetry) 형상의 건물은 상사

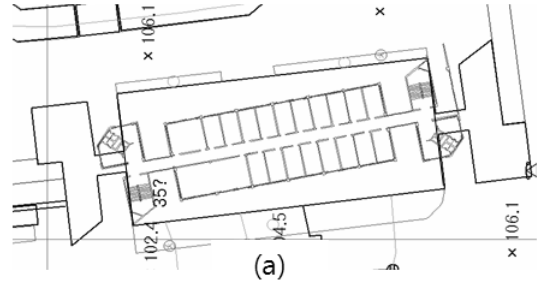


그림 10. 지도좌표부여 과정에서 발생하는 문제점 사례

변환에서 회전각의 해가 두 가지 이상 존재할 수 있다. 그림 10의 (b)의 경우 시각적인 결과는 우수하지만 실제 건축된 상태는 180도 회전되어 있었다.

첫 번째 문제 유형을 해결하기 위해서는 그림 1의 수동 편집과정에서 미리 수치지도상의 부속건물 폴리곤을 주 건물 폴리곤과 병합해야만 한다. 물론 일정 범위내에 존재하거나 인접한 폴리곤을 검색하고 조합가능한 모든 병합별로 편집비용을 분석하여 건설도면상의 폴리곤과 정합되는 최적의 폴리곤 병합을 도출하는 것도 가능하나 이를 위해서는 보다 많은 연구가 추후 진행되어야 한다.

두 번째 문제 유형은 형상만을 이용하는 VASM 알고리즘의 특성상 근본적인 해결은 불가능하다. 회전 대칭성 폴리곤 정합의 문제는 이미 많은 연구에서 언급되었으며, 문자열 정합을 이용하여 형상을 정합하는 과정에서 이러한 대칭성 존재 여부를 판단할 수 있는 기존 연구 성과들을 활용할 수 있으며(Lladós 등, 1997), 복수의 변환식이 존재할 수 있음을 판별하고 이를 기반으로 사용자의 판단 여부에 따라 반자동 처리가 가능할 것이다.

## 7. 결 론

본 연구에서는 건설도면 CAD 자료에 지도좌표를 부여하기 위하여 CAD 자료의 건물 외곽선의 형상을 수치지도상의 동일 건물 외곽선과 비교하여 대응되는 꼭지점,

즉 공역 꼭지점을 탐색하고 이를 이용하여 상사변환식을 유도하는 과정과 그 방법을 제안하였다. 제안된 방법을 서울대학교 공과대학의 9개 건물의 건설도면에 적용해 보았고 그 결과를 수치지도에 중첩해보았다. 적용결과 수치지도와 CAD 자료의 공역 꼭지점을 추출하기 위한 VASM 알고리즘의 활용 가능성을 확인할 수 있었지만, 비교대상 건물 외곽선 추출결과에 따라 많은 영향을 받을 수 있다는 것과 외곽선이 매우 단순한 건물들의 경우 회전 대칭성으로 인한 한계가 존재한다는 것도 확인할 수 있었다. 이러한 문제점들을 해결하기 위해서는 CAD와 GIS 자료를 연동하기 위한 많은 선행연구에서 언급된 것처럼 표준화된 레이어명을 가지고 건물의 외곽선을 하나의 단일 객체로 CAD 자료상에 작성하는 것이 필요할 것이다. 즉 객체기반으로 작성된 수치지도 2.0과 같이 CAD 자료가 표현하는 건물 객체를 고유하게 식별할 수 있는 어떤 속성 정보를 가지고 있는 공간정보와 건물의 외곽형상의 표현과 레이어명 표준이라는 조건을 충족한 건설도면을 이용할 경우 본 연구에서 제안된 방법을 통하여 신속하게 광범위한 지역의 건설도면과 공간정보를 연계할 수 있을 것이다. 또한 본 연구에서 벡터형 공간정보를 구성하는 가장 기본 단위인 꼭지점의 대응관계를 이용하여 폴리곤의 전역적인 좌표변환을 시도하였지만, 꼭지점을 연결한 선분과 선분의 대응관계를 이용하여 국지적인 좌표변환에도 응용할 수 있을 것이다. 이 경우 원자료의 형상에 왜곡이 발생하지만 시각적인 측면에서 보다 효과적인 정합 결과를 얻을 수 있을 것이다.

## 감사의 글

본 연구는 국토해양부 첨단도시기술개발사업 - 지능형

국토정보기술혁신 사업과제의 연구비지원(07국토정보C04)에 의해 수행되었습니다.

## 참고문헌

- Ayman, S.A. and Dunham, D. (1999), Utility system management: migration from CAD to GIS, *Proceeding of ESRI User Conference 99*.
- Bunke, H. and Buhler, U. (1993), Applications of approximate string matching to 2D shape recognition. *Pattern Recognition*, Vol. 26, No. 12, pp. 1797-1812.
- Duda, R.O., Hart, R.E. and Stork, D.G. (2001), *Pattern Classification*. John willy & sons, Inc., New York, NY.
- Kaygin, S. and Bulut, M.M. (2002), Shape recognition using attributed string matching with polygon vertices as the primitives, *Pattern Recognition Letters*, Vol. 23, pp. 287-294.
- Kruskal, J. B. (1983), An overview of sequence comparison : time warps, string edits, and macromolecules, *SIAM Review*, Vol. 25, No. 2, pp. 201-237.
- Lladós, J., Bunke H. and Mart E. (1997), Finding rotational symmetries by cyclic string matching, *Pattern Recognition Letters*, Vol. 18, No. 14, pp. 1435-1442.
- Maes, M. (1991), Polygonal shape recognition using string matching techniques, *Pattern Recognition*, Vol. 24, No. 5, pp. 433-440.
- Schaefer, S., McPhail, T. and Warren J. (2006), Image deformation using moving least squares, *ACM Transactions on Graphics*, Vol. 25, No.3, pp. 533-540.
- Tsai, W.H. and Yu, S.S. (1985), Attributed string matching with merging for shape recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 7, No. 4, pp. 453-462.
- Wagner, R.A. and Fisher, M.J. (1974), The string-to-string correction problem, *J. Assoc. Comput. Mach.*, Vol. 21, No. 1, pp. 168-173.

(접수일 2008. 7. 11, 심사일 2008. 8. 1, 심사완료일 2008. 8. 10)