

論文2002-39SP-3-8

# 항공영상에서 상대 위치 추정 알고리즘의 실시간 구현

## (Real-Time Implementation of the Relative Position Estimation Algorithm Using the Aerial Image Sequence)

朴宰弘\*, 金官奭\*, 金麟澈\*\*, 朴來弘\*, 李商郁\*\*\*,

(Jae-Hong Park, Kwan Suk Kim, Rin-Chul Kim, Rae-Hong Park, and Sang Uk Lee)

### 요약

본 논문에서는 TMS320C80 멀티미디어 MVP(multimedia video processor)를 이용한 항법 변수 추출의 구현 기법에 관하여 연구하였다. 특히, 항법 변수 추출 시스템의 실시간 구현에 중요한 역할을 하는 상대 위치 추정 알고리즘의 실시간 구현 방법에 관하여 고찰한다. 두 지점에서 취득된 영상을 이용하는 상대 위치 추정 알고리즘을 근간으로 하여, 방대한 양의 계산량을 감축하면서 고정 소수점 프로세서에 적합한 고속 알고리즘을 개발한다. 그런 다음, MVP 내의 4개의 병렬 프로세서(PP; parallel processor)를 이용하여 병렬 처리할 수 있도록 알고리즘을 재구성한다. 그 결과, MVP를 이용한 항법 변수 추출 시스템은 초당 30 프레임의 처리할 수 있음을 확인하여, 실시간 구현 조건을 만족시킴을 알 수 있었다.

### Abstract

This paper deals with an implementation of the navigation parameter extraction technique using the TMS320C80 multimedia video processor (MVP). Especially, this paper focuses on the relative position estimation algorithm which plays an important role in real-time operation of the overall system. Based on the relative position estimation algorithm using the images obtained at two locations, we develop a fast algorithm that can reduce large amount of computation time and fit into fixed-point processors. Then, the algorithm is reconfigured for parallel processing using the 4 parallel processors in the MVP. As a result, we shall demonstrate that the navigation parameter extraction system employing the MVP can operate at full-frame rate, satisfying real-time requirement of the overall system.

**Keyword** : Relative Position Estimation, Implementation, Aerial Image Sequence, Navigation Parameters.

\* 正會員, 西江大學校 電子工學科

(Dept. of Electronic Eng., Sogang Univ.)

\*\* 正會員, 서울市立大學校 電子電氣工學部

(School of Electrical Eng., Univ. of Seoul)

\*\*\* 正會員, 서울大學校 電氣工學部

(School of Electrical Eng., Seoul National Univ.)

※ 본 연구는 국방과학연구소 및 자동제어 특화연구센터의 연구비지원에 의한 결과입니다.

接受日字:1999年10月21日, 수정완료일:2002年4月15日

### I. 서론

영상 항법 시스템<sup>[1]</sup>은 비행체에 부착된 카메라로부터 취득된 영상을 이용하여 비행체의 위치를 추정하는 시스템으로, 비행체 외부의 정보를 이용하지 않기 때문에 외부 교란에 견실한 특성을 보이는 장점을 가진다. 본 논문에서 고려하는 영상 항법 시스템<sup>[2]</sup>은 간단한 자이로스코프 등을 통해 자세 정보를 취득할 수 있고, 비행체 하단에 카메라가 부착된 상태에서 지정된 궤도를

따라 비행하는 무인 비행체에 적용하기 위한 것으로, 크게 상대 위치 추정과 절대 위치 추정 등 두 알고리즘으로 구성된다. 상대 위치 추정에서는 인접한 프레임 간의 영상 및 해당 비행체의 자세 정보(roll, pitch, head 등의 비행체의 각도와 비행 고도)를 이용하여 프레임간 위치 변화를 추정함으로써 비행체의 위치를 판별한다. 반면에, 절대 위치 추정에서는 현재 취득된 영상으로부터 지형 정보를 재구성하고, 이를 미리 마련된 지형 데이터 베이스와 비교함으로써 비행체의 기준 위치를 판별하게 된다. 이때 지형 데이터 베이스로 위성 영상을 사용한다면 자세 보정이 된 입력 영상과 정합하여 위치를 찾을 수 있고, 디지털 고도 모델(DEM; digital elevation model)을 사용한다면 여러 장의 입력 영상들로부터 얻어지는 3차원 정보와 정합하여 결과를 얻을 수 있다. 절대 위치 추정은 넓은 지역에 대해 정합을 수행하기 때문에 많은 계산 시간이 소요되므로 전체 비행 궤적 상의 위치 중에서 지형적 특이성을 갖는 일부 위치에서만 간헐적으로 이루어지는 반면, 상대 위치 추정은 주기적으로 수행되어 비행체의 현재 위치 정보를 갱신하게 된다. 따라서, 영상 항법 시스템의 실제 구현을 위해서는 상대 위치 추정 알고리즘의 실시간 구현이 매우 중요함을 알 수 있다<sup>3-4)</sup>.

본 논문의 주된 초점은 상대 위치 추정 알고리즘을 실시간으로 구현할 수 있는 방법에 대하여 고찰하고 이를 실제로 구현하는 것을 목적으로 하였다. 본 논문에서는 실시간 영상 처리에 매우 적합한 구조를 가진 TI사의 TMS320C80 MVP(multimedia video processor)라는 전용 디지털 신호 처리 장치(DSP; digital signal processor)<sup>5)</sup>를 이용하여 시스템을 구현하였다. 먼저, 상대위치 추정 알고리즘에서 대부분의 계산 시간을 점유하는 특징점 추출 및 영상 정합 부분의 계산 성능을 향상시키기 위하여 고속 알고리즘을 제안하였다. 그런 다음, 알고리즘이 MVP 상에 적절히 구현될 수 있도록 고정 숫자점 연산에 의한 영향을 분석하였고, 알고리즘을 분석하여 병렬 처리하는 방법을 제안하였다. 그 결과, 본 논문에서는 상대 위치 추정 알고리즘을 MVP로 구현할 때, 초당 30 프레임 이상을 처리할 수 있음을 보이겠다.

II. 상대 위치 추정 알고리즘

상대 위치 추정<sup>2)</sup>은 연속적으로 입력되는 항공 영상

을 정합하고 정합점과 비행체의 자세 정보를 이용하여 항법변수를 추출하는 것으로 항법 시스템의 성능을 크게 좌우한다. 그림 1은 상대 위치 추정 알고리즘의 블록 선도를 보이고 있으며 그림 2는 상대위치 추정이 이루어지는 방식을 보이고 있다. 상대 위치 추정은 특징점 추출(feature point extraction), 3차원 영상 보정(3-D image compensation), 영상 정합(template matching), 속도 추출(velocity calculation), 위치 추정(position update) 부분으로 구성된다.

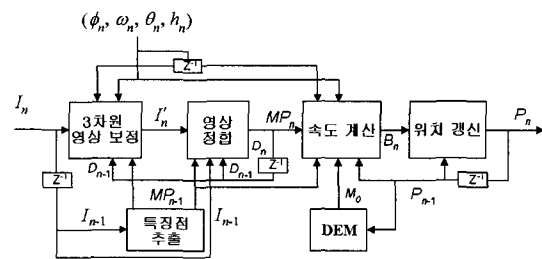


그림 1. 상대위치 추정의 블록선도  
Fig. 1. A block diagram of the relative position estimation.

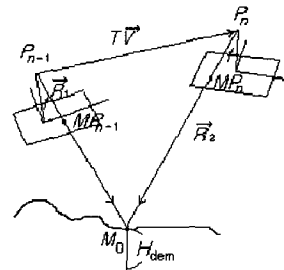


그림 2. 상대위치 추정  
Fig. 2. The relative position estimation.

먼저, 특징점 추출 부분에서는 이전 영상  $I_{n-1}$ 에서 분산이 최대가 되는 블록을 특징점  $MP_{n-1}$ 로 추출한다. 여기서 특징점이라 함은 영상 내의 블록들 중에서 가장 우수한 정합 성능을 보이게 하는 블록으로, 본 논문에서 사용하는 정합 척도와 계산 효율성을 고려하여 분산이 최대가 되는 블록을 특징점으로 선정하였다. 본 논문에서는  $31 \times 15$  크기의 블록 단위로 분산을 구하는데, 영상의 움직임이 위에서 아래로 움직이고 인접한 두 영상이 50% 이상 겹친다는 가정 하에,  $320 \times 240$  영상의 상반부에 대해 탐색한다. 그리고, 계산 시간 및 추정 성능을 참조하여, 수평 방향으로 화소 단위, 그리고

수직 방향으로 블록크기의 반(7 lines) 단위로 특징점을 검색한다.

한편, 이전 프레임의 영상은 현재 프레임의 영상과 다른 비행 자세 정보를 가지고 있으므로, 보다 정확한 영상 정합을 위해서 정합 하고자 하는 두 영상의 자세를 동일하게 맞추는 영상 보정을 수행한다. 본 논문에서는, 그림 2에 보인 바와 같이, 두 영상간의 자세 정보(roll, pitch, heading)의 차이를 이용하여, 이전 프레임에서 추출한 특징점  $MP_{n-1}$ 을 navigation-to-body 변환을 수행하여 보정된 특징점  $MP'_{n-1}$ 을 얻는다.

즉,  $MP_{n-1}$ 내의 위치  $(X, Y, Z)$ 를  $MP'_{n-1}$ 내의 위치  $(X', Y', Z')$ 로 변환하고자 할 때, 먼저, 초점과 위치  $(X', Y', Z')$  간의 방향 벡터  $(N_x, N_y, N_z)$ 를 다음과 같이 얻는다.

$$\begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = N_B \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (1)$$

여기서, navigation-to-body 좌표 변환 행렬  $N_B$ 는 다음과 같이 두 프레임간의 roll, pitch, heading의 변화  $(\theta, \phi, \omega)$ 의 함수로 주어진다.

$$N_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \omega & 0 & -\sin \omega \\ 0 & 1 & 0 \\ \sin \omega & 0 & \cos \omega \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

위치  $(X, Y, Z)$ 는 영상 평면 상의 한 점이므로, 깊이 정보  $Z$ 는 초점 거리  $f$  와 같다. 따라서, 위치  $(X', Y', Z')$ 에 대응되는  $(X, Y, Z)$ 는 식 (3)과 같은 비례식을 통하여 얻을 수 있다.

$$\frac{X'}{N_x} = \frac{Y'}{N_y} = \frac{Z'}{N_z} = \frac{f}{N_z} \quad (3)$$

그러면,  $MP_{n-1}$ 내의 위치  $(X', Y')$ 에서의 밝기 값은  $MP_{n-1}$ 내의 위치  $(X, Y)$ 에서의 밝기 값으로 얻어진다. 그런데, 식 (1)과 식 (3)에서의 좌표값들은 미터(meter) 단위이므로 영상 평면상에서 실제 화소 위치를 계산하기 위해서는 단위를 변환시켜주어야 한다. 즉, 영상의 중심으로부터 수평 혹은 수직 방향으로의 화소 단위 거리 차를  $XP$ 라고 하면, 이에 대응되는 미터 단위의 좌표  $X$ 는 식 (4)를 통해 얻을 수 있다.

$$X = \text{PIXEL}X \times \text{ratio} \times XP \quad (4)$$

여기서  $\text{PIXEL}X$ 는 화소 단위를 미터 단위로 보정해주는 인자이고  $\text{ratio}$ 는 이전 영상과의 고도 차를 보정해주는 인자이다. 이와 같은 과정을 화소 단위로 반복 실행함으로써 보정된 특징점  $MP'_{n-1}$ 를 얻을 수 있다.

세 번째로, 영상 정합에서는 이전 프레임의 특징점  $MP'_{n-1}$ 과 가장 유사한 위치를 현재 프레임에서 검색한다. 이때, 유사도 판단은 높은 정합 성능을 유지하기 위해 식 (5)로 정의되는 정규화 상관도(NCC; normalized cross correlation)를 이용한다.

$$\text{NCC}(p, q) = \frac{\sum_i \sum_j W_j W_i MP_{n-1}(i, j) I_n(i-p, j-q)}{\sqrt{\sum_i \sum_j W_j W_i MP_{n-1}^2(i, j) \sum_i \sum_j I_n^2(i-p, j-q)}} \quad (5)$$

여기서  $W_j \times W_i$ 는 블록의 크기로 본 논문에서는  $31 \times 15$ 를 선정하였다. 정규화 상호 상관도는 일반적으로 널리 사용되는 평균 제곱 오차(MSE; mean squared error) 혹은 절대 평균 편차(MAD; mean absolute difference) 등의 유사도 척도에 비해 계산량은 많지만 정합 성능이 우수한 것으로 알려져 있다. 영상 정합에서는 이전 영상에서 추정된 움직임 벡터  $D_{n-1}$ 를 참조하여, 현재 영상의  $(MP_{n-1} + D_{n-1})$ 점 부근의  $71 \times 71$  크기의 검색 영역(search region) 내에서 유사도가 가장 높은 정합점  $MP_n$ 을 찾아낸다. 본 논문에서는 계산 및 정합 성능을 고려하여 2단계 정합을 수행한다. 1단계 정합에서는 4 화소 단위의 축소 영상(decimated image)으로 검색 범위 내에서 영상 정합을 수행하고, 2단계 정합에서는 1단계 정합 위치 근방의  $\pm 5$  화소 범위에서 원 영상에 대하여 정합을 수행한다. 2단계 정합 방법의 성능에 대해서는 Sim<sup>[2]</sup>에 소개되어 있다.

마지막으로, 속도 추출, 위치 추정 단계에서는 비행체의 이전 위치  $P_{n-1}$ 에서 바라본 특징점  $MP_{n-1}$ 과 이에 대응되는 현재 위치  $P_n$ 에서 바라본 정합점  $MP_n$ 을 비교하여 스테레오 정합 기법으로 현재 위치  $P_n$ 을 추정한다. 즉, 이전 프레임 영상에 대한 자세 정보와 지면 고도 정보  $H_{dem}$ (DEM; digital elevation model)를 이용해  $MP_{n-1}$ 에 대응되는 지상 대응점  $M_0$ 를 찾는다. 그런 다음, 지상 대응점과 정합점  $MP_n$ 을 이용하여 현재 위치  $P_n$ 을 얻는다.

### III. MVP를 이용한 실시간 구현

본 절에서는 상대 위치 추정 알고리즘을 위한 MVP 시스템에 대하여 간략히 설명하고, 실시간 구현 방법에 대하여 설명한다. 상대 위치 추정 알고리즘은 Park<sup>[3]</sup>에서 소개한 바와 마찬가지로 모든 연산이 병렬 프로세서에서 효과적으로 수행될 수 있어야 하므로, 입력 데이터의 크기를 참조하여 고정 소수점 연산에 의한 영향이 최소가 되도록 계산 정밀도를 조정하였다. 즉, 병렬 프로세서에서는 16비트×16비트 곱셈과 32비트 덧셈을 수행한다는 점을 참조하여 각 계산 과정에서 이에 적합하게 연산 결과들을 크기 조정해야한다. 본 논문에서 적용한 크기 조정 방법은 각 데이터의 값의 범위(dynamic range)를 참조한다면, 간단히 얻을 수 있으므로 구체적인 설명은 생략한다. 또한, 특징점 추출 부분에서 슬라이딩 윈도우(sliding window) 방식의 고속 알고리즘을 사용하였고, 영상 정합 부분에서 2단계 탐색 알고리즘을 사용하여 계산 성능을 향상시킨다. 그러나, 계산량이 비교적 적은 영상 보정에 대해서는 증분 계산 방법을 적용하지 않았다.

#### 1. MVP 시스템 소개

본 논문에서 사용하는 TMS320C80 MVP는 하나의 칩에 1개의 주 프로세서(MP ; master processor)와 4개의 병렬 프로세서(PP ; parallel processor) 등 5개의 프로세서가 집적되어 있다. 이 칩에는 2킬로바이트 단위의 50킬로바이트 칩 내장 램이 있는데 이들은 크로스바 스위치(cross-bar switch)를 통해 5개의 프로세서에 독립적으로 연결될 수 있어, 다양한 병렬 처리 알고리즘을 수행할 수 있다. 병렬 프로세서는 발전된 형태의 32비트 DSP로, 고정 소수점 방식의 연산을 수행하며 곱셈이 많은 신호처리 응용 분야나 화소 관련 처리가 많은 응용 분야에서 좋은 성능을 발휘한다. 그리고, 주 프로세서는 32비트 축소 명령 집합 컴퓨터(RISC ; Reduced Instruction Set Computer) 프로세서로, IEEE-754 호환의 부동 소수점 유닛을 탑재했다. 모든 5개의 프로세서는 C나 어셈블리 언어로 프로그래밍할 수 있다.

이외에도 칩 내장 메모리와 칩 외장 메모리간의 데이터 교환을 전담하는 하드웨어인 전송 제어기(TC ; transfer controller)가 있다. 전송 제어기는 블록 데이터

를 이동하는데 효율적인 패킷 전송과 워드 단위의 데이터를 이동시키는 직접 외부 접근(DEA ; direct external access), 그리고, 내부 캐시 메모리 갱신 등을 지원한다. 또한, 비디오 인터페이스를 담당하는 영상 제어기(VC ; video controller)가 있어 멀티미디어 응용 프로그램에 알맞은 기능을 제공한다.

MVP 칩 내의 자원들은 크로스바에 의해 병렬적 조합으로 연결되어 있고 자원 공유 및 데이터 전송이 매우 빠른 속도로 이루어진다. 본 논문에서는 40MHz의 MVP를 사용하는데, 이 경우 초당 20억 번의 RISC-like 연산을 수행할 수 있다. 또한 초당 1.8기가바이트의 명령어, 초당 2.4기가바이트의 데이터를 동시에 칩 내부에서 이동시킬 수 있으며, 초당 400메가바이트의 속도로 칩 외장 메모리와 데이터를 교환할 수 있다<sup>[5]</sup>.

표 1. MVP 보드 사양  
Table 1. System specification of MVP board.

system clock	40MHz
host interface	PCI host interface
external memory	40 Mbytes DRAM
video memory	4 Mbytes VRAM
display output	RGB
video input	NTSC

본 논문에서는 IBM 호환 PC에 PCI 버스로 접속한 보드에 40MHz의 MVP를 탑재한 모듈을 사용하였으며, 이 모듈의 중요한 사양은 표 1에 제시하였다. 그림 3에서는 MVP 모듈을 포함한 전체적인 영상 항법 시스템의 블록선도와 MVP 보드를 보여주고 있다. 현재, 이 시스템은 개발 중인 관계로 PC와 접속되어 있지만, 향후 독립된 시스템으로 운용할 수 있다. PC와 MVP 모듈간의 접속은 PC에서의 프로그램과 데이터 다운로드 및 결과 확인을 위한 것으로, 시스템 개발을 용이하게 한다. 여기서, 영상 항법에 관련된 모든 처리는 부착된 카메라를 통해 입력된 영상을 이용하여 MVP 모듈에서 수행되고, MVP 모듈에서 직접 모니터로 결과가 도시된다. 따라서, PC는 알고리즘 수행에는 영향을 주지 않도록 되어 있다.

실제, 그림 3의 시스템은 미리 정해진 비행 궤적 상의 디지털 고도 정보를 시스템 메모리에 저장한 다음,

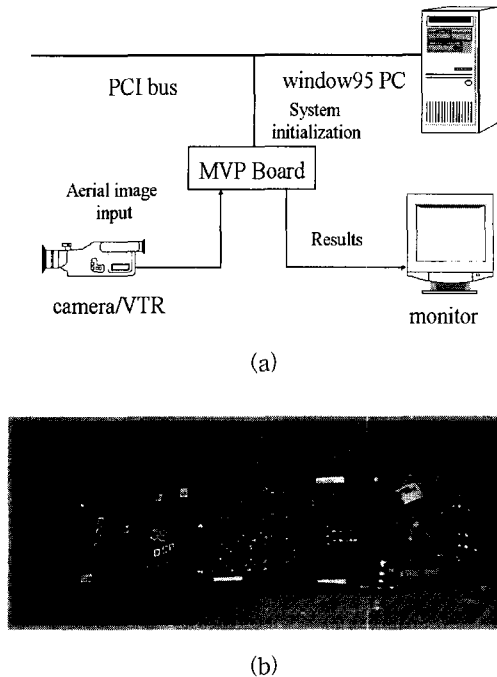


그림 3. 영상 항법변수 추출 시스템, (a) 블록선도, (b) MVP board  
 Fig. 3. A navigation parameter extraction system. (a) block diagram, (b) MVP board

비행기에 부착된 카메라와 자이로스코프로부터 각각 입력되는 영상과 자세 정보를 실시간으로 입력받아 상대위치 추정 알고리즘을 동작시켜 비행체의 위치를 반복적으로 추정하게 된다. 본 논문에서는 실험의 편의상 영상 신호와 그에 대응되는 자세 정보를 Hi-8 캠코더 등을 이용하여 녹화한 다음, 자세 정보는 영상을 취득하는 주기 즉 프레임 율(frame rate) 단위로 시스템 메모리에 저장하였다. 그런 다음, Hi-8 캠코더를 재생시키면서 입력 영상을 MVP에서 취득하고, 해당 입력 영상에 따른 자세 정보를 메모리에서 선택하여 알고리즘을 동작시킴으로써 실제 상황을 모사하였다. 이때, MVP에서는 미리 표시된 영상 신호의 첫 부분을 인식함으로써 해당 자세 정보 선택이 용이하도록 하였다.

먼저, 캠코더에 기록된 항공 영상은 정해진 시간 간격으로 MVP에서 취득된다. 영상은 640×480 크기로 입력이 되는데 이 영상은 한 줄 건너 삭제하는 방식으로 320×240 영상으로 줄여 사용된다. 영상의 메모리간 이동에는 MVP 프로세서에 내장된 전송 제어기에 의해 수행되는 패킷 전송(PT; packet transfer)이라는 방법이 사용되었다. 패킷 전송은 프로세서가 아닌 전송 제

어기에서 수행되기 때문에 프로세서가 그 동안에 다른 연산을 할 수 있고 대용량의 데이터를 메모리에서 이동시키는 데 적합하다. 패킷 전송은 다양한 방법으로 데이터를 이동시킬 수 있는데 영상의 크기를 320×240으로 줄이는 경우나 카메라의 촬영 각도가 달라질 경우 영상의 위쪽이 진행 방향으로 맞추기 위해서 영상을 회전시킬 때 패킷 전송의 몇 가지 옵션을 변경하여 간단히 수행할 수 있다. 본 논문에서는 프레임 율은 영상의 종류에 따라 초당 0.2 혹은 1 프레임으로 하였으며, 시간 간격은 영상 제어기의 타이머를 이용하여 맞추었다.

디지털 고도 모델 정보는 지상의 고도값을 영상으로 만든 정보로 40메가바이트의 D램 공간 대부분을 차지하는 큰 용량의 데이터이다. 디지털 고도 모델 정보는 영상정합 후 최종적으로 속도벡터를 계산하는 과정에서 사용되며 프로그램 실행 전에 로드된다. 알고리즘 실행 후 계산된 위치 결과는 D램에 저장되며, PC가 이 결과에 접근할 수 있어 간단히 결과를 확인할 수 있다.

## 2. 주 프로세서와 병렬 프로세서의 운용

MVP를 효율적으로 운용하기 위해서는 주 프로세서와 병렬 프로세서들에게 적절히 작업을 분담하여야 한다. 주 프로세서는 IEEE-754 부동 소수점 유닛으로 구성된 32비트 축소 명령 집합 컴퓨터 프로세서이다. 주 프로세서는 주로 MVP 칩 내장 프로세서들간의 제어를 담당하는 프로세서로 주 프로세서 자신을 포함한 4개의 병렬 프로세서 및 전송 제어기, 영상 제어기 등의 제어기에게 초기화(reset), 중지(halt), 캐시 초기화(cache reset), 작업 할당, 메시지 인터럽트 전달 등의 제어 명령을 내릴 수 있다<sup>[5]</sup>.

MVP로 구현된 상대 위치 추정 알고리즘에서 주 프로세서는 변환 행렬 구하기 등 일부 부동 소수점 연산이 필수적인 계산 이외에는, 병렬 프로세서들에게 작업 할당, 작업 결과의 조합, 영상 제어기 등을 통한 외부 데이터 입출력 등을 담당한다.

주 프로세서의 가장 중요한 역할은 병렬 프로세서의 제어이다. 주 프로세서는 명령어 버퍼(command buffer)를 통하여 병렬 프로세서에 프로그램을 실행시킨다. 이때, 프로그램 실행에 필요한 매개 변수는 매개 변수 버퍼(command argument buffer)를 통하여 전달한다. 병렬 프로세서는 해당 프로그램을 실행시킨 후 주 프로세서에 message interrupt를 가함으로써 프로그램 실행이 끝났음을 알려주고, 이때, 실행 결과는 매개 변수 버

퍼 혹은 미리 약속된 메모리에 수록함으로써 전달한다. 여기서, 주 프로세서 및 병렬 프로세서 간의 명령어 및 매개변수는 일반적으로 각 병렬 프로세서의 칩 내장 매개변수(parameter) 램에 수록된다.

네 개의 병렬 프로세서는 MVP의 계산 능력의 큰 부분을 담당하고 있다. 병렬 프로세서는 비트 필드나 다중 화소 조작 등의 DSP에 관련된 계산을 수행하는 데 뛰어난 능력을 발휘하는 프로세서이다. 병렬 프로세서의 작업은 주로 칩 내장 램에 저장된 데이터에 대해 이루어진다. 칩 외장 메모리와 칩 내장 메모리간의 블록 데이터 전송은 병렬 프로세서의 작업과 독립적으로 전송 제어기에 의해 수행되므로, 병렬 프로세서는 병렬 처리 시스템에서 자주 일어나는 데이터의 분배나 병합과는 관계없이 그 처리 능력을 데이터의 계산에만 초점을 맞출 수 있다. 이러한 점에서 칩 내장 메모리의 효율적 이용은 MVP의 계산 성능에 중요한 역할을 하게 된다.

3. 고속 알고리즘

본 논문에서 고려하는 상대위치 추정 기법은 MVP를 이용하더라도 2절에서 설명한 알고리즘으로는 실시간 구현하기 어려우므로 계산 시간을 감축할 수 있는 고속 알고리즘의 개발이 필수적이다. 특히 Sim<sup>[2]</sup>에서 소개한 형태의 하이브리드 위치 추정 알고리즘과 같이 절대위치 추정이 상대위치 추정보다 훨씬 더 많은 처리 시간을 필요로 하기 때문에 영상 샘플링 속도보다도 더 많은 시간 감축이 요구된다. 상대위치 추정 기법에서 계산량이 많은 부분은 영상 정합, 특징점 추출, 영상 보정 등 3부분인데, Sim<sup>[2]</sup>에서는 2단계 정합을 수행함으로써 영상 정합의 계산 시간을 크게 감축하였다. 그리고, Park<sup>[3]</sup>에서는 영상 정합뿐만 아니라 특징점 추출, 영상 보정을 고속으로 처리할 수 있는 알고리즘을 개발하였다. 즉, 특징점 추출에서는 슬라이딩 윈도우 방식의 고속 알고리즘을 제안하였고, 영상 보정에서는 증분 가산 방법을 제안함으로써 계산시간을 대폭 감축하였다.

본 논문에서는 II장에서 설명한 바와 같이, 검색 영역이 현재 영상이 아니라 반대로 이전 영상에서 특징점을 추출함으로써 영상 보정 시 소요되는 계산 시간을 대폭 감축하였다. 즉, 이전 영상에서 특징점을 추출하고, 현재 영상에 대한 영상 보정은 특징점 부근으로 국한하였다. 이와 같은 방법은 현재 영상 전체를 영상 보정하지 않기 때문에, 증분 가산 방법을 사용하지 않

더라도, 기존의 방법에 비해 계산 시간을 대폭적으로 감축할 수 있다. 따라서, 본 논문에서의 영상 보정은 구현 방법이 비교적 복잡한 증분 가산 방법을 사용하지 않는다.

한편, 특징점 추출에서는 Park<sup>[3]</sup>에서 제안한 슬라이딩 윈도우 기법을 사용하여 계산 효율을 높였다. 특징점 추출에서는 이전 영상의 상반부에 대해 수평 방향으로 화소 단위, 그리고 수직 방향으로 7 줄 단위로 특징점을 검색하게 된다. 이는 특징점 블록을 수평 방향으로 한 화소씩 이동시키면서 각 위치에서의 분산을 반복적으로 구하는 것으로 구현된다. 이때, 각 특징점 블록마다 블록 내 화소들의 자승 합(squared sum)과 평균을 구하게 되는데, 자승 합 및 평균은 인접한 특징점 블록에서의 연산과 대부분 중복되게 계산된다. 슬라이딩 윈도우 기법은 이렇게 블록이 한 방향으로 연속적으로 이동하며 계산할 때 중복되는 부분을 다시 계산하지 않고, 이전 블록에서 계산한 결과에서 갱신되는 부분만 첨삭시킴으로써, 특징점 블록의 분산을 효율적으로 계산하는 방법이다. 그림 4에서는 슬라이딩 윈도우 기법의 예를 보여주고 있다. 화살표로 표시된 방향으로 이동하면서 각 블록들의 분산을 계산하게 되는데, 첫 번째 블록에 대한 분산을 계산한 다음, 두 번째 블록에 대한 분산을 계산할 때의 예를 보여주는 것이다. 이때, 가운데 검은 색 부분은 첫 번째 블록과 중복되는 부분이고, 왼쪽의 흰색 부분은 첫 번째 블록에 대한 분산에서 삭제할 부분이고, 오른쪽의 회색 부분은 첫 번째 블록에 대한 분산에 추가해야 할 부분을 나타낸다. 따라서, 슬라이딩 윈도우 기법에서는 인접한 블록에 대한 분산에서, 왼쪽 1줄에 대한 자승합 및 평균을 삭제하고, 새로이 오른쪽 1줄에 대한 자승합 및 평균을 더함으로써 새로운 블록에 대한 분산을 계산할 수 있다. 슬라이딩 윈도우 기법을 도입하면, 각 줄에 대한 자승

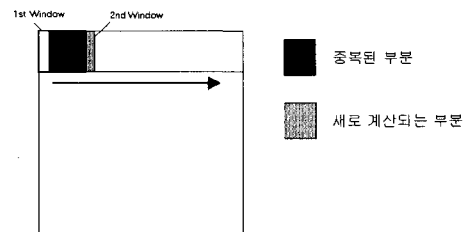


그림 4. 특징점 추출에서 슬라이딩 윈도우의 예  
Fig. 4. An example of the sliding window for the feature point extraction.

합과 평균을 저장하고 있어야 한다는 단점이 있으나, 계산량이 약 1/13으로 감축됨을 확인하였다.

#### 4. 상대 위치 추정의 병렬 처리

3절에서 상대위치 추정 기법에 대한 고속 알고리즘을 개발하였지만, MVP의 성능을 충분히 활용하여 계산 시간을 효과적으로 감축시키기 위해서는 MVP 내의 4개의 병렬 프로세서와 1개의 주 프로세서를 최대한 동시에 이용하는 방법을 모색하여야 한다.

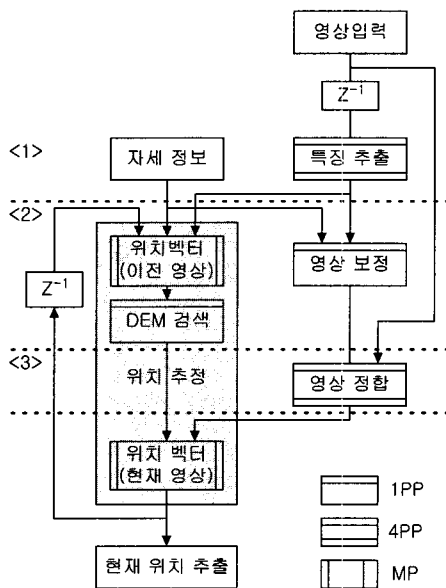


그림 5. 자료 의존성 그래프  
Fig. 5. Data dependency graph.

먼저, II장과 그림 1에서 나타난 상대 위치 추정의 다섯 부분을 병렬로 처리될 수 있는지를 검증하기 위하여, 자료 의존 문제를 그림 5와 같이 분석하였다. 그림 5에 보인 바와 같이 특징점 추출 결과를 이용하여 영상 보정이 수행되고, 영상 보정된 특징점을 이용하여 영상 정합이 이루어지므로, 특징점 추출, 영상 보정, 영상 정합은 순차적으로 실행되어야 한다. 한편, 속도 계산 및 위치 추정은 이전 영상의 특징점으로부터 지상 대응점을 찾는 부분과 이 지상 대응점으로부터 현재 영상에서 정합된 특징점을 이용하여 현재 위치 벡터를 얻는 부분으로 나눌 수 있다. 여기서, 현재 위치 벡터를 얻는 부분은 영상 정합 결과를 이용하기 때문에 순차 실행되어야 하지만, 지상 대응점을 찾는 부분은 영상 정합이 수행되기 이전에 실행될 수 있음을 알 수 있다. 이 점을 참조하여 본 논문에서는 비교적 계산 부담이

적은 영상 보정이 실행되는 동안 지상 대응점을 찾는 부분이 동시에 실행되도록 하였다. 위의 내용을 고려하면, 지상 대응점을 찾는 부분을 제외하고, 특징점 추출, 영상 보정, 영상 정합, 그리고 현재 위치 벡터를 얻는 부분은 반드시 순차적으로 실행되어야 함을 알 수 있다. 이러한 때에, 각 병렬 프로세서들을 파이프라인 형태로 연결하여 각 부분들을 전달시키는 파이프라인 다중 처리 방식(pipelined multi-processing)을 고려할 수 있으나, 이는 latency가 길어질 뿐만 아니라, 각 부분의 수행 시간이 상이하기 때문에 각 병렬 프로세서들에게 균등한 작업량 할당이 어려워 계산 시간이 많이 소요된다는 단점이 있다. 따라서, 본 논문에서는 상대 위치 추정 알고리즘을 특징점 추출, 영상 보정, 및 영상 정합 등 4단계 순차 실행으로 구현하였다. 그러나, 각 부분은 영상 데이터를 블록 단위로 반복 처리하는 것으로 이루어지므로 다중 데이터에 대한 병렬 처리 방법으로 구현할 수 있다. 즉, 각 부분은 단일 명령 복수 데이터(SIMD; single instruction multiple data) 방식으로 4개의 병렬 프로세서를 이용하여 병렬 처리한다. 이때 각 부분이 처리되는 순서를 그림 5의 자료 의존성 그래프에 점선으로 표시하여 함께 나타내었다.

먼저, 상대위치 추정은 입력된 영상에 대하여 그림 5에 <1>로 표시된 부분인 특징점 추출부터 수행되는데, 검색되는 1/2 프레임을 4개의 병렬 프로세서들을 이용하여 병렬 처리한다. 이를 위해 검색 영역을 수직 방향으로 4등분하고, 등분된 검색 영역을 4개의 병렬 프로세서들에게 할당하여 특징점 추출을 수행한다. 주 프로세서에서는 병렬 프로세서의 수행이 끝나는 대로 그 결과를 조합하여 전체 검색 영역에 대한 최적의 특징점을 선정한다.

두 번째로 그림 5에서 <2>로 표시된 영상 보정과 지상 대응점을 찾는 부분이 실행된다. 3차원 영상 보정 부분은 비교적 적은 수행시간을 가질 뿐만 아니라, 단일 명령 복수 데이터 형태의 병렬 처리를 통해 계산 성능을 향상시키기에 부적절한 구조를 가지고 있으므로, 1개의 병렬 프로세서로 실행시킨다. 단, 마지막 부분의 위치 추정/속도 계산 부분에서 특징점으로부터 지상 대응점을 찾는 부분은 영상 정합 결과와 무관하므로 추가로 1개의 병렬 프로세서와 주 프로세서를 이용하여 영상 보정 부분과 다중 처리한다.

세 번째, 그림 5에서 <3>으로 표시된 영상 정합 부분은 가장 많은 시간을 소요하는 부분으로 이 부분의

병렬처리는 전체적인 계산 성능을 향상시키기 위해 필수적이다. 영상 정합의 2단계 정합은 1단계 정합 결과를 참조하여 수행해야 하므로 1 단계와 2 단계는 순차적으로 실행되어야 한다. 1/2단계 영상 정합은 검색 영역을 수직 방향으로 4등분하여 4개의 병렬 프로세서들에게 동일한 작업을 할당함으로써 병렬 처리한다. 그림 6은 2단계 정합 과정 중 제 1 단계 정합을 수행할 때에 분할되는 정합 영역을 나타낸다. 71×71 화소의 영역은 네 영역으로 나뉘어 각각의 병렬 프로세서가 각 영역의 정규화 상호 상관도를 통한 정합 결과를 산출한 후 주 프로세서에서 최적의 정합 영역을 선별하게 된다.

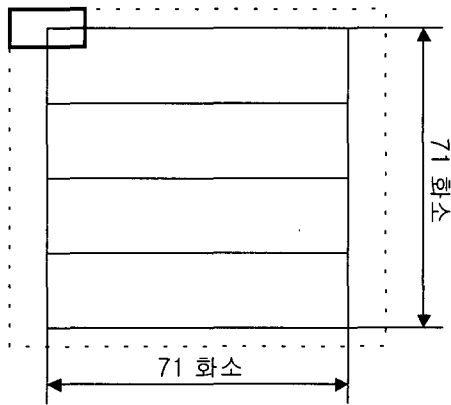


그림 6. 병렬처리를 위한 영상 분할의 예  
Fig. 6. An example of image segmentation for parallel processing.

마지막으로, 위치 추정/속도 검색 부분 중에서 지상 대응점으로부터 현재 위치를 추정하는 부분은 계산이 비교적 간단하므로 주 프로세서를 이용하여 결과를 얻는다.

IV. 실험 결과

본 절에서는 III장에서 설명한 고속 알고리즘 및 병렬 처리 기법으로 상대 위치 추정 기법을 MVP 모듈로 구현하였을 때의 수행 시간을 측정하여 실시간 구현 가능성을 확인한다. 그리고 비디오를 통해 입력되는 실제 NTSC 영상에 대하여 실험하여 상대 위치 추정 시스템의 성능을 평가한다.

1. 계산 성능 평가

본 절에서는 MVP 모듈을 이용한 상대 위치 추정의 계산 성능을 비교하고, 그 결과를 표 2에 제시하였다. 표 2에서는 Park<sup>[3]</sup>에서와 마찬가지로 1개의 병렬 프로세서만을 이용하였을 때 (1PP), 전 절에서 설명한 바와 같이 4개의 병렬 프로세서를 모두 이용하였을 때 (4PP)의 계산 시간을 1/1000초 단위로 측정하였다. 그리고, MVP 모듈을 이용한 시스템의 계산 성능을 객관적으로 평가하기 위해 Indigo 2(R4400 CPU, 175MHz) 워크스테이션에서 수행된 계산 시간도 함께 비교하였다. 워크스테이션에서의 계산 시간은 파일 입출력 및 그래픽 처리 등을 뺀 순수한 상대 위치 추정에 소요된 시간만을 측정하였다. 반면에, MVP 모듈에서의 시간은 결과 도시를 위한 그래픽 처리만을 제외한 모든 시간을 측정하는 것이다.

표 2. 평균 실행 시간 비교 (단위: 1/1000초)  
Table 2. Comparisons in terms of average processing time (Unit: msec)

	W/S	1PP	4PP
특징점 추출	21.0	8.69	3.28
영상 보정	2.9	2.25	2.87
1단계 영상 정합	75.4	26.97	7.13
2단계 영상 정합	29.0	9.32	2.63
위치 추정	1.4	2.34	0.05
총 실행 시간	129.7	49.57	15.96

표 2를 참조하면, 1개의 병렬 프로세서만 이용한 경우에도 워크스테이션에 비해 우수한 계산 성능을 보여 MVP 모듈의 우수한 계산 성능을 입증하고 있다. 이때, Park<sup>[3]</sup>에 비해 계산 성능이 우수하게 나타난 것은 MVP 칩 내장 메모리의 적절한 사용 등에 기인하는 것으로 MVP에서 칩 내장 메모리 사용 방법이 계산 성능에 큰 영향을 끼침을 알 수 있다.

영상 보정의 경우 4개의 병렬 프로세서를 이용하는 경우 더 많은 수행 시간이 걸린 것을 볼 수 있는데 이는 영상 보정 단계에서 그림 5에 보인 바와 같이 디지털 고도 모델 검색이 함께 이루어지기 때문이며 실제로 위치 추정의 경우 수행 시간이 비약적으로 줄어든 것은 위치 추정의 대부분이 영상 보정과 함께 이루어지기 때문이다.



전체적으로 4개의 병렬 프로세서를 이용함으로써 1개의 병렬 프로세서를 사용한 것에 비해 계산 성능이 약 3배 이상 향상됨을 있음을 알 수 있었다. 구체적으로 살펴보면, 특징점 추출 부분은 대부분 3개의 병렬 프로세서들로 실행되므로 약 2.65배의 계산 성능 향상을, 그리고, 영상 보정, 위치추정 부분은 2개의 병렬 프로세서들로 영상 보정 및 지상 대응점 추정을 수행하므로 약 1.57배의 계산 성능 향상을, 그리고, 1/2단계 영상 정합은 4개의 병렬 프로세서들을 모두 이용하므로 약 3.5배 이상으로 계산 성능을 향상시킬 수 있음을 볼 수 있다. 특히, 본 논문에서 구현한 상대 위치 추정 시스템은 프레임당 0.016초 이하의 시간이 소요되는 것으로 나타나, 절대 위치 추정을 포함하여 전체 영상 항법 시스템을 실시간 구현하는 것이 가능함을 보였다.

2. 추정 성능 평가

MVP 보드 상에서의 상대 위치 추정 시물레이션은

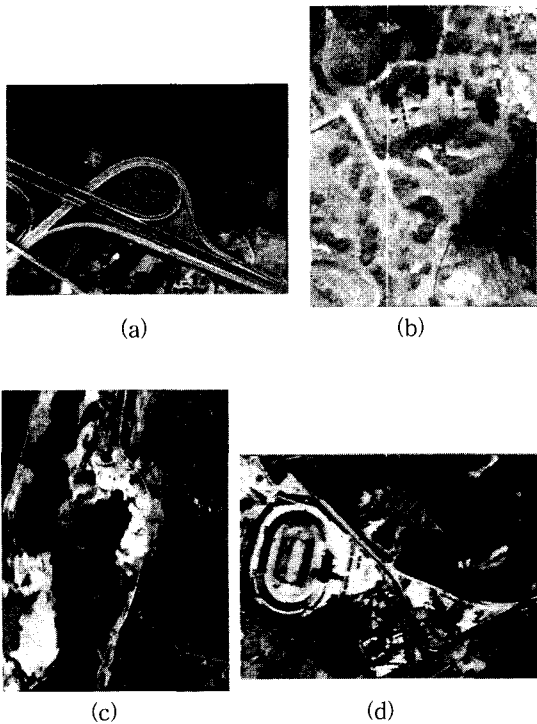


그림 7. 입력 영상 예 (a) 실험 영상 1 (b) 실험영상 2 (c) 실험영상 3 (d) 실험 영상 4

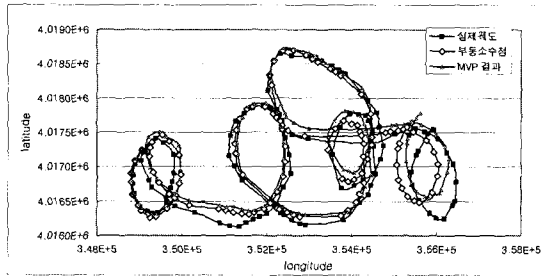
Fig. 7. An example of input sequence ; (a) test sequence 1 (b) test sequence 2 (c) test sequence 3 (d) test sequence 4.

표 3. 4가지 실험 영상에 대한 추정 정확도 (단위: m)

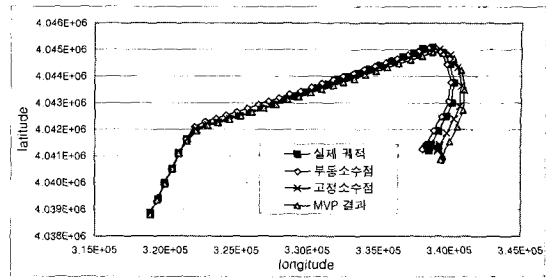
Table 3. Summary of the estimation accuracies on the four different test sequences (unit: m).

실험 영상 1	부동소수점	고정소수점	MVP 결과
총 비행 거리	28737	29011	29945
평균 위치 차이	192.02	170.58	176.05
최종 위치 차이	535.92	433.03	343.40
실험 영상 2	부동소수점	고정소수점	MVP 결과
총 비행 거리	28171.38	27829.13	35914.13
평균 위치 차이	537.36	481.2828	609.3437
최종 위치 차이	1057.47	1048.08	1293.18
실험 영상 3	부동소수점	고정소수점	MVP 결과
총 비행 거리	1264000	126783	125750
평균 위치 차이	453.91	509.28	443.68
최종 위치 차이	1028.16	1230.02	384.06
실험 영상 4	부동소수점	고정소수점	MVP 결과
총 비행 거리	129458.03	126129.01	124290.38
평균 위치 차이	934.6641	1044.162	846.61
최종 위치 차이	173.4799	185.8877	539.30

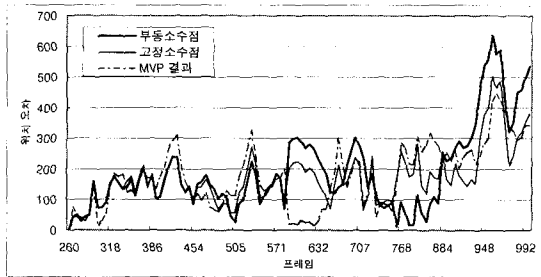
그림 7에 제시한 네 개의 실험 영상으로 실험하였다. 실험 영상은 UTM 좌표 상의 52지대에 해당하는 대진 이남 지역을 경비행기 혹은 헬기 상에서 Hi-8mm 비디 오 카메라로 촬영하여 취득하였다. 이 항공 영상들은 비디오로 재생되어 MVP 보드의 NTSC 입력포트로 입력받아 1초나 0.2초 간격으로 표본화한 다음, 시물레이션에 사용하였다. 따라서 이는 실제 이러한 시스템이 비행체에 부착되어 동작할 때와 같은 영상 입력 상황 이라고 할 수 있다. 움직임 추출을 위하여 사용되는 디지털 고도 모델 정보는 비행 지역의 지도에서 등고선 을 추출한 후 최종적으로 인터플레이션을 통해 안정화 된 지상의 고도 정보를 사용하였다. 자세 정보는 계기 판에서 얻어진 값을 사용하였지만 데이터 추출시 ±1/2 초 간격 내에서 영상과의 동기화가 이루어지지 않아 NTSC 방식의 ±1/2초 간격에서 추출된 30프레임 중 임의의 프레임을 선택하여 실험에 사용하였다. 실험 결과에서 비교한 세 가지 비행 궤적은 같은 프레임의 영 상을 사용하여 얻은 결과이다. 마지막으로 실험 결과에 서 나타낸 실제 궤적은 실험에 사용된 프레임들과 실



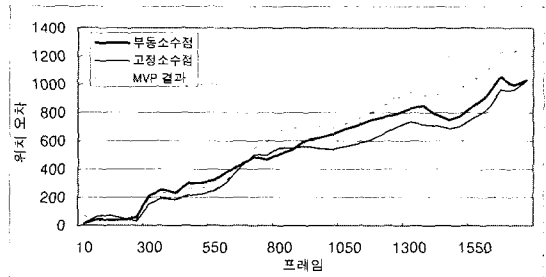
(a)



(a)



(b)



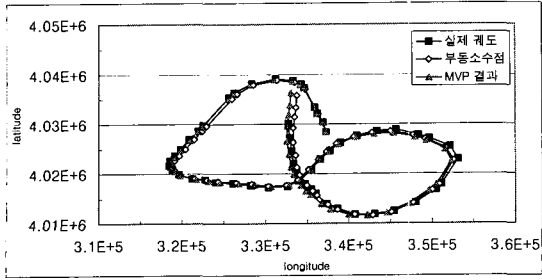
(b)

그림 8. 실험 영상 1에 대한 상대위치 추정 결과(단위 : m) (a) 추정 경로 (b) 추정 오차  
Fig. 8. Result of the relative position estimation on the test sequence 1 (unit : meter); (a) estimated trajectory; (b) position error.

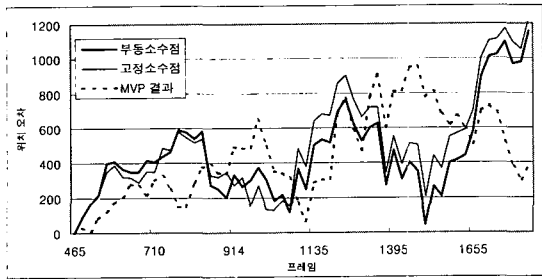
그림 9. 실험 영상 2에 대한 상대위치 추정 결과(단위 : m) (a) 추정 경로 (b) 추정 오차  
Fig. 9. Result of the relative position estimation on the test sequence 2 (unit : meter); (a) estimated trajectory; (b) position error.

제 지도를 육안으로 비교하여 얻은 위치로 얻어졌다.  
표 3과 그림 12에 나타난 각 실험 영상을 이용한 상대위치 추정 결과를 살펴보면, 실험 영상 1은 29km의 구간을 비행하는 헬기에서 12분간 촬영된 735장의 영상으로 많은 회전을 하여 roll, pitch 값의 변화가 많으며 다른 영상에 비해 낮은 고도에서 촬영된 영상이다. 하지만 많은 변화와 낮은 고도에 의한 3차원 효과에도 불구하고 안정된 시스템으로 동작하는 것을 알 수 있다. 6분간 1776장을 얻은 실험 영상 2는 96년 3월 촬영되었으며 낮은 고도에서 비행하는 경비행기에서 촬영되고 작은 화각 때문에 1초 간격으로 표본화할 경우 스테레오 매칭이 불가능하여 0.2초 간격으로 얻어진 샘플을 사용하였다. 따라서 27km의 짧은 거리를 이동하였음에도 많은 프레임이 사용되어 비교적 큰 오차를 나타내고 있다. 22분간 1391장을 얻은 실험 영상 3은 96년 10월에 촬영되었으며 이전 영상들에 비해서 매우 긴 구간인 126km를 비행하여 얻어진 결과인데 촬영 당시 비행이 안정되어 있어 비교적 적은 오차를 내었다. 반면 97년 1월에 26분간 1571장을 얻은 실험 영상 4는

경로가 복잡할 뿐만 아니라 실험 영상 3에 비하여 비행 시 바람이 심하여 비행자세가 불안하기 때문에 추정 오차가 크게 나는 것을 볼 수 있다.  
그림 8~그림 11에 나타난 오차 그래프와 궤적에 나타난 부동소수점과 고정소수점은 워크스테이션 상에서 얻어진 결과이며 초기 알고리즘의 개발은 부동소수점 연산을 하는 프로그램으로 구현되었으나 고속의 고정소수점 연산을 하는 MVP 상에서의 구현을 위해 고정소수점 연산의 프로그램을 거쳐서 MVP 보드 상에서 구현하였다. 워크스테이션 상에서의 실험 환경은 실시간 구동이 아니고 이전에 캡처되어 파일로 저장되어있는 항공 영상을 매 프레임을 처리할 때 읽어들이는 형식으로 되어 있으며 두 과정은 같은 영상을 사용하였다. 이때 고정소수점과 부동소수점의 차이는 초기에는 극히 미미하지만 프레임이 증가할수록 점점 벌어지는 양상을 띄게 된다. 이것은 상대위치 추정의 단점으로서 이전 추정의 오차가 다음 추정의 오차에 누적되기 때문이다. 이러한 오차는 때로는 증가하는 방향이기도 하고 감소하는 방향이기도 하기 때문에 고정소수점 결과



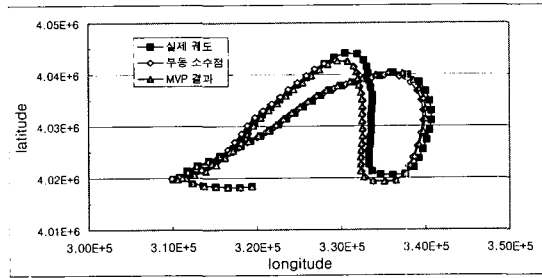
(a)



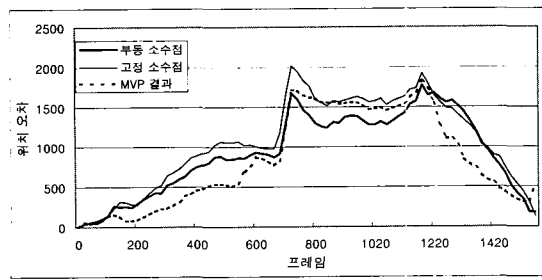
(b)

그림 10. 실험 영상 3에 대한 상대위치 추정 결과(단위 : m) (a) 추정 경로 (b) 추정 오차

Fig. 10. Result of the relative position estimation on the test sequence 3 (unit : meter) (a) estimated trajectory (b) position error.



(a)



(b)

그림 11. 실험 영상 4에 대한 상대위치 추정 결과(단위 : m) (a) 추정 경로 (b) 추정 오차

Fig. 11. Result of the relative position estimation on the test sequence 4 (unit : meter); (a) estimated trajectory; (b) position error.

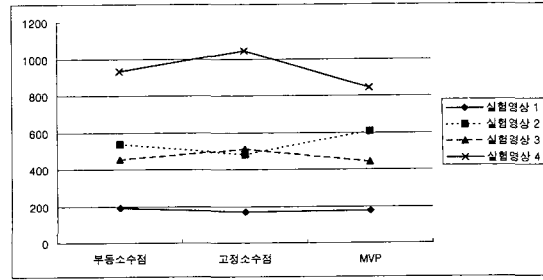


그림 12. 4가지 실험 영상에 대한 추정 정확도 결과 그래프(단위 : m)

Fig. 12. Result of the estimation accuracies on the four different test sequences(unit : m)

가 부동소수점 결과에 비해서 더 나은 성능을 가질 수도 있고 더 나쁜 성능을 가질 수도 있게 된다.

MVP 결과는 실제로는 연산 과정이 워크스테이션 상의 고정소수점 연산과 같기 때문에 고정소수점 결과와 같은 궤적과 예측 오차가 같아야 함에도 불구하고 입력 영상이 비디오를 통해서 실시간으로 입력되기 때문에 이때 표본화의 타이밍이 캡처된 영상과 정확히 일치하지 않아 다르게 나타났다.

### V. 결론

본 논문에서는 영상 항법 변수 추출 시스템 중 상대 위치 추정을 실시간 처리에 적합하게 개선하고, 구체적으로 MVP 모듈을 이용하여 구현한 후 이의 실시간 적용 가능성을 제시하였다. 먼저, MVP의 구조를 참조하여, 고정 소수점 연산에 의한 영향을 분석하고, 내부 메모리를 충분히 이용하는 방법을 모색하였다. 또한, 주 프로세서와 병렬 프로세서 간의 명령어 교환 방법을 이해하고, 알고리즘 분석을 통해 4개의 병렬 프로세서를 이용한 효율적인 병렬 처리가 가능하도록 알고리즘을 분해하였다.

그 결과, 4개의 병렬 프로세서들을 충분히 이용하여 상대 위치 추정 알고리즘을 MVP 모듈에 구현하였으며, 프레임당 0.016초 이하의 계산 시간을 나타냄을 보였다. 따라서, 본 논문에서 고려한 MVP 모듈은 상대 위치 추정뿐만 아니라 절대 위치 추정까지 포함한 전체 영상 항법 시스템을 실시간으로 구현할 수 있는 가능성을 보였다.

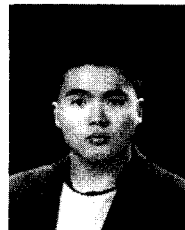
참 고 문 헌

- [1] C.-F. Lin, Modern Navigation, Guidance, and Control Processing, Englewood Cliffs, Prentice Hall, 1991.
- [2] D.-G. Sim, S.-Y. Jeong, D.-H. Lee, R.-H. Park, R.-C. Kim, S. U. Lee, and I. C. Kim, "Hybrid estimation of navigation parameters for aerial image sequence," IEEE Trans. Image Processing, vol. IP-8, no. 3, pp. 429-435, Mar. 1999.
- [3] J.-H. Park, K.-D. Hwang, S. B. Pan, R.-C. Kim, R.-H. Park, S. U. Lee, and I. C. Kim, "Implementation of the navigation parameter extraction from the aerial image sequence on the TMS320C80 DSP board," in Proc. Int. Conf. Signal Processing Applications & Technology, vol. 2, pp. 1562-1566, San Diego, CA, USA, Sept. 1997.
- [4] 김인철, 반성범, 김관석, 박래홍, 이상욱, "TMS 320C80을 이용한 상대위치 추정의 다중 처리", 제 11회 영상처리 및 이해에 관한 워크샵, pp. 213-217, 대전, 1999년 1월
- [5] TMS320C80 Multimedia Video Processor System-Level Synopsis, Texas Instruments Inc., 1995.

저 자 소 개



朴宰弘(正會員)  
 1996년 : 서강대학교 전자공학과 학사. 1998년 : 서강대학교 전자공학과 석사. 1998년 2월~현재 : 삼성전자 영상 디스플레이 사업부 연구원. <주관심분야 : computer vision>



金官奘(正會員)  
 1998년 : 서강대학교 전자공학과 학사. 2000년 : 서강대학교 전자공학과 석사. 2000년 2월~현재 : 네오웨이브 (주) 기술연구소 근무 중, 전임 연구원. <주관심분야 : computer vision>

朴來弘(正會員) 第36卷 S編 第6號 參照

金麟澈(正會員) 第30卷 B編 第4號 參照

李商郁(正會員) 第30卷 B編 第4號 參照