

논문 2009-46SP-2-7

## 계층적 KLT 특징 추적기의 하드웨어 구현

(A Hardware Implementation of Pyramidal KLT Feature Tracker)

김 현 진\*, 김 경 환\*\*

(Hyunjin Kim and Gyeonghwan Kim)

### 요 약

본 논문에서는 계층적 KLT 특징 추적기의 하드웨어 구조를 제안한다. 계층적 KLT 특징 추적기(pyramidal Kanade-Lucas-Tomasi feature tracker)는 주로 MPU를 기반으로 구현되어 왔으나 반복연산 과정이 많아 실시간으로 처리하기 어려우므로, 실시간 수행을 위하여 FPGA(Field Programmable Gate Array)를 이용하여 구현하였다. 본 논문에서는 추출되는 특징점의 수를 일정하게 유지하기 위해 입력 영상의 밝기에 적응적으로 임계값을 설정하는 특징점 추출 알고리즘을 제안한다. 또한 계층적 KLT 추적 알고리즘을 메모리의 용량 및 대역폭의 한계를 극복하고, FPGA의 병렬처리 특성에 적합한 구조로 변환한다. 소프트웨어로 실행한 결과와의 비교를 통하여 특징점의 추출 및 추적이 유사한 양상으로 이루어짐을 검증하였고, 720x480 영상 입력에 대해 초당 30 프레임의 full frame rate로 추적이 수행됨을 확인하였다.

### Abstract

This paper presents the hardware implementation of the pyramidal KLT(Kanade-Lucas-Tomasi) feature tracker. Because of its high computational complexity, it is not easy to implement a real-time KLT feature tracker using general-purpose processors. A hardware implementation of the pyramidal KLT feature tracker using FPGA(Field Programmable Gate Array) is described in this paper with emphasis on 1) adaptive adjustment of threshold in feature extraction under diverse lighting conditions, and 2) modification of the tracking algorithm to accommodate parallel processing and to overcome memory constraints such as capacity and bandwidth limitation. The effectiveness of the implementation was evaluated over ones produced by its software implementation. The throughput of the FPGA-based tracker was 30 frames/sec for video images with size of 720x480.

**Keywords :** Feature tracking, KLT tracker, FPGA implementation, Feature point extraction

### I. 서 론

특징점의 추출 및 추적은 자동 운항(autonomous navigation), 감시시스템, 삼차원 개체의 구조 해석을 통한 물체 형태 또는 지형의 이해 등의 다양한 컴퓨터 비전 응용 분야에 적용되고 있다.

대부분의 특징점 추적 알고리즘은 일련의 영상 시퀀스로부터 특징점을 추출한 후, 인접하는 프레임 간의

변위를 계산하여 대응 관계를 추적한다. 하지만 변위 계산 시 반복연산 과정이 많아 일반 PC환경에서 구동시킬 경우 자동주차시스템<sup>[1]</sup>과 같이 실시간으로 추적 및 제어가 이루어져야 하는 시스템에 적용하는 데는 제한이 따른다. 따라서 특징점의 추출 및 추적을 실시간으로 수행하기 위해 FPGA와 같은 하드웨어를 이용한다.

특징점 추적 알고리즘 중 KLT 특징 추적기<sup>[2]</sup>는 평균 제곱오차(mean squared error)를 기반으로 하는 상관관계(correlation) 분석을 통해 특징점의 유사도를 측정하여 추적하는 방법이다. 이 때, 움직임이 큰 변위를 얻기 위해서는 특징점 윈도우의 크기가 커져야 하는 반면, 특징점 윈도우의 크기가 커질 경우 윈도우에 대해

\* 학생회원, \*\* 정회원, 서강대학교 전자공학과  
(Dept. of Electronic Engineering, Sogang Univ.)

※ 본 연구는 부품소재개발사업 VISS 개발 중 (주)만도과제인 '반자동 주차를 위한 영상처리 하드웨어 시스템 개발'과 IDEC의 지원으로 수행됨.

접수일자: 2008년10월14일, 수정완료일: 2009년1월3일

평균을 취하는 연산들로 인해 세부적인 변위에 대한 추적 성능이 저하되고, 연산량 또한 증가하므로 이에 대한 절충이 필요하다.

이러한 문제를 해결하기 위해 Bouguet<sup>[3]</sup>은 계층식 구현을 제안하였다. KLT 특징 추적기는 계층식 추적을 수행함으로써 변위가 큰 특징점에 대응하며 세밀한 움직임에 대한 추적도 가능하게 된다.

다른 한편으로는 KLT 특징 추적기를 실시간 시스템에 적용하기 위해 하드웨어와 소프트웨어로 분리하여 반복적인 연산이 많은 부분을 FPGA에서 처리하는 방법들이 연구되었다<sup>[4~5]</sup>.

본 논문에서는 전체 계층적 KLT 추적 알고리즘을 하드웨어로 설계하는 방법을 제안한다. 먼저 해리스 코너 추출기(Harris corner detector)<sup>[6]</sup>와 누적 히스토그램을 이용하여 특징점의 수를 일정하게 유지하기 위해 외부 환경에 따라 변화하는 입력 영상의 밝기에 적응적인 특징점 추출 알고리즘을 제안한다. 그리고 기존의 MPU를 기반으로 구현되었던 계층적 KLT 특징 추적 알고리즘을 메모리 용량 및 대역폭의 한계를 극복하고, 병렬처리에 적합한 하드웨어 구조로 변환한다.

본 논문의 구성은 다음과 같다. II장에서 계층적 KLT 특징 추적기 알고리즘에 관하여 소개하고, III장에서는 KLT 추적 알고리즘의 하드웨어 구현을 위한 구조를 제안한다. IV장에서는 구현된 하드웨어 시스템의 실험 결과에 대해 분석하고, 마지막으로 V장에서 본 논문의 결론을 맺는다.

## II. 계층적 KLT 특징 추적기 알고리즘

KLT 추적 알고리즘은 영상 밝기의 항상성 제약조건(image brightness constancy constraint)을 만족하고 영상의 움직임이 충분히 작을 때, 연속하는 프레임 사이의 특징점의 변위를 계산하여 특징점의 대응 관계를 얻는다.

카메라의 움직임으로 인해 특징점 윈도우는 연속하는 프레임 사이에서 일정 거리만큼 움직이고, 모양 또한 근소하게 변화하기 때문에 움직임 모델은 이미지 밝기와 관련하여 식 (1)과 같이 나타낼 수 있다.

$$J(\mathbf{x}) = I(\mathbf{x} - \mathbf{d}) - \eta(\mathbf{x}) \quad (1)$$

이 때, 함수  $I$ 는 특정 프레임에서의 밝기, 함수  $J$ 는

다음 프레임에서의 밝기,  $\mathbf{d}$ 는 변위 벡터,  $\eta$ 는 형태의 변화로 인한 차이이다. 특징점의 추적은 두 영상에서 대응되는 특징점 윈도우  $W$ 의 평균제곱오차(mean squared error)  $\epsilon$ (식(2))을 최소화하는 변위를 결정하는 것을 의미한다.

$$\epsilon = \sum_{j \in W} (I(\mathbf{x}_j - \mathbf{d}) - J(\mathbf{x}_j))^2 \quad (2)$$

밝기 함수를 Taylor series expansion에 의해 근사함으로써 식 (3)과 같이 변위  $\mathbf{d}$ 에 대한 선형 시스템을 얻을 수 있으며, 이의 반복해를 계산하여  $\epsilon$ 를 최소화하는 변위  $\mathbf{d}$ 를 찾을 수 있다.

$$G\mathbf{d} = \bar{\mathbf{b}} \quad (3)$$

여기서  $G$ 는 영상  $I$ 의  $x$ 와  $y$  방향 미분치 ( $I_x, I_y$ )의 2차 모멘텀으로 이루어진  $2 \times 2$  행렬로  $\sum_{j \in W} \begin{bmatrix} I_x^2(\mathbf{x}_j) & I_x(\mathbf{x}_j)I_y(\mathbf{x}_j) \\ I_x(\mathbf{x}_j)I_y(\mathbf{x}_j) & I_y^2(\mathbf{x}_j) \end{bmatrix}$ 로 정의되며,  $\bar{\mathbf{b}} = \sum_{j \in W} \begin{bmatrix} I_x(\mathbf{x}_j)(I(\mathbf{x}_j) - J(\mathbf{x}_j)) \\ I_y(\mathbf{x}_j)(I(\mathbf{x}_j) - J(\mathbf{x}_j)) \end{bmatrix}$ 이다.

추적에 사용될 특징점의 선택은 각 프레임에 대한 해리스 코너 추출기(Harris-corner detector) 적용으로 이루어진다. 코너 응답,  $R$ 은 2차 모멘텀 행렬  $G$ 로부터 식 (4)와 같이 정의된다. 여기서  $k$ 는 민감도 계수(sensitivity parameter)로  $0.04 < k < 0.06$  이다.

$$R = \det(G) - k(\text{trace}(G))^2 \quad 0.02 \leq k \leq 0.04 \quad (4)$$

코너 응답이 큰 부분은 단일 픽셀이 아닌 넓은 영역에 걸쳐 나타나는 경향을 띠므로 세밀한 코너를 표현하지 못한다. 이를 해결하기 위해 코너 응답  $R$ 에 대한 임계값 적용과  $N \times N$  영역의 최대 코너 응답 선택의 과정(non-maximum suppression)<sup>[7]</sup>을 거친다.

선형 근사는 변위가 작은 경우에만 적용되므로, 좀 더 큰 변위를 다루기 위해서 계층적인 추적을 수행한다. 영상 피라미드를 구성한 후, 저해상도 영상에서 추적을 시작한다. 저해상도 영상에서 고해상도 영상, 즉 상위 레벨에서 하위 레벨로 각 레벨에서의 변위 추정 결과를 누적하여 가장 하위 레벨인 원영상에서 추정된 변위가 최종 변위 벡터가 된다. 이러한 계층적인 구현은 각 레벨의 변위 벡터는 작게 유지하면서 최종 변위

```
// pyramidal tracking algorithm

Detect feature points :Harris corner detector
Build pyra mid repres entations of I and J :{IL}L=0...Lm and {JL}L=0...Lm
Initializa tion of py ramidal gu ess: gLm = [gxLm gyLm]T = [0 0]T
for L = Lm down to 0 with step of -1
  Location o fpoint u on image IL: uL = [px py]T = u/2L
  Derivative of IL:
    Ix(x,y) = (IL(x+1,y) - IL(x-1,y))/2
    Iy(x,y) = (IL(x,y+1) - IL(x,y-1))/2
  Spatial gr adient mat rix:
    G = ∑x=px-wx, y=py-wypx+wx, py+wy [ IxL(x,y) Ix(x,y)Iy(x,y)
    Ix(x,y)Iy(x,y) IyL(x,y) ]
  Initializa tion of it erative L- K: v0 = [0 0]T
  for k = 1 to K with step of 1 (or until ||vk|| < accuracy threshold )
    Image diff erence :
      δIk(x,y) = IL(x,y) - JL(x + gxk-1 + vxk-1, y + gyk-1 + vyk-1)
    Image mism atch vecto r:
      bk = ∑x=px-wx, y=py-wypx+wx, py+wy [ δIk(x,y)Ix(x,y)
      δIk(x,y)Iy(x,y) ]
    Optical fl ow(Lucas-K anade):
      ηk = G-1bk
    Guess for next itera tion:
      vk = vk-1 + ηk
  end of for -loop on k
  Final opti cal flow a t level L:
    dL = vk
  Guess for next leve l L-1:
    gL-1 = [gxL-1 gyL-1]T = 2(gL + dL)
  end of for -loop on L
  Final opti cal flow v ector:
    d = g0 + d0
  Location o fpoint on J:
    v = u + d
```

그림 1. 계층적 KLT 특징 추적기 알고리즘의 의사코드  
Fig. 1. Pseudo code of pyramidal KLT feature tracker algorithm.

벡터는 큰 움직임에 대해서도 다룰 수 있다는 장점이 있다. 그림 1은 계층적 KLT 추적 알고리즘의 의사 코드이며, Bouguet의 방법<sup>[3]</sup>을 참조하여 작성하였다.

### III. 하드웨어 구조 설계

제안하는 전체 하드웨어의 구조를 그림 2에 나타내었다. 전체 구조는 해리스 코너 검출기를 이용한 특징점 추출 부분과 계층적 KLT 특징 추적기를 이용한 추적 부분으로 크게 나눌 수 있으며, 이 두 부분은 병렬로 동작한다. 입력 영상에 대하여 해리스 코너 검출기를 이용하여 특징점의 초기 목록을 특징 테이블(feature table)에 저장하는 한편, 영상의 피라미드를 구성하여 외부 메모리에 저장한다. 특징 테이블에 저장된 특징점의 위치 정보를 이용하여 외부 메모리에서 해당 이미지 패치의 픽셀 값을 읽어, 변위를 각 레벨에서 반복하여 구한 후 최종적으로 추정된 변위를 이용하여 특징점의 위치 정보를 갱신하도록 한다.

$$S = \sum_{k=1}^{\infty} ar^{k-1} = \frac{a}{1-r}, \quad (|r| < 1) \quad (5)$$

a : 첫째항, r : 등비

영상 피라미드는 레벨의 수가 많아지더라도 식 (5)의 무한등비급수 공식에 의해 영상의 총화소수의 4/3를 넘지 않으므로 영상 피라미드 전체를 외부 메모리에 저장하여도 메모리의 대역폭에 의한 문제가 크지 않다. 또한, FPGA의 병렬처리 특성을 이용하면 피라미드 레벨의 수에 관계없이 동시에 피라미드 영상을 얻을 수

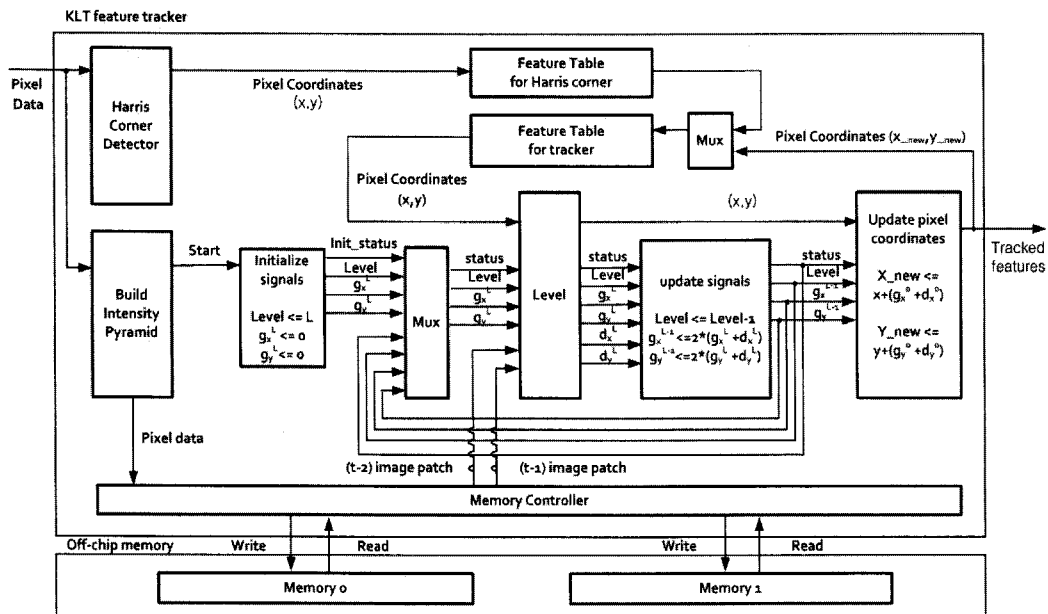


그림 2. 하드웨어 구조  
Fig. 2. Hardware architecture.

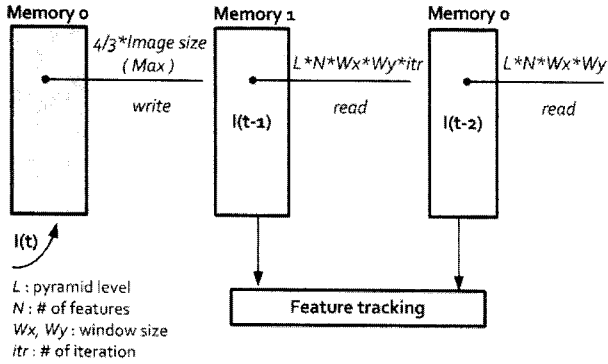


그림 3. 외부 메모리 사용

Fig. 3. Usage of off-chip memories.

있다.

그림 3은 피라미드식의 특징점 추적을 위한 메모리 관리 방법을 도식화한 것이다. 영상의 피라미드가 가장 상위 레벨까지 생성된 후 가장 상위 레벨에서부터 하위 레벨 쪽으로 변위 벡터가 누적되기 때문에 현재 입력되고 있는 영상에 대해서는 특징점을 추적할 수 없으며, 이전의 두 프레임  $I(t-2)$ 와  $I(t-1)$ 에서 특징점의 추적을 수행한다. 이러한 특성에 적합하도록 더블 버퍼링(double buffering) 기법을 이용하여 영상 피라미드를 저장한다.

더블 버퍼링 방법은 두 개의 메모리를 각각 입력 데이터의 저장과 데이터에 대한 접근의 두 역할로 분담하는 방식으로, 그림 3과 같이 현재 입력되는 영상의 피라미드는  $I(t-2)$ 가 저장되어 있는 메모리와 동일한 메모리에 저장된다. 메모리 0에  $I(t-2)$ 의 영상 피라미드, 메모리 1에  $I(t-1)$ 의 영상 피라미드가 저장되어 있을 때 현재 프레임의 입력이 메모리 0에 저장되는 동안  $I(t-2)$ 에서  $I(t-1)$ 로의 추적이 진행되고, 추적이 끝난 이후 메모리 0은  $I(t)$ 의 영상 피라미드로 완전히 대체된다. 새로운 프레임이 입력될 때마다 두 메모리의 역할을 바꾸어 영상 입력과 처리를 동시에 수행함으로써 메모리를 효율적으로 사용할 수 있게 된다.

$$\begin{aligned} N_R(t-2) &= L \times N \times W_x \times W_y \\ N_R(t-1) &= L \times N \times W_x \times W_y \times itr \end{aligned} \quad (6)$$

$L$ : 피라미드 레벨,  $N$ : 특징점의 수

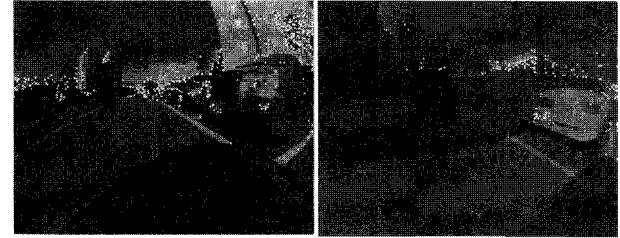
$W_x, W_y$ : 윈도우의 크기,  $itr$ : 반복 횟수

메모리 참조 횟수(reference counting)는 식 (6)과 같이 피라미드 레벨의 수, 특징점의 수, 윈도우의 크기에



영상 (a)

영상 (b)



영상 (a)의 특징점 : 398

영상 (b)의 특징점 : 167

그림 4. 고정된 임계값을 설정할 경우 추출되는 특징점의 수

Fig. 4. Number of selected features using fixed threshold value.

영향을 받는다. 따라서 실시간 처리를 위해서는 메모리의 동작 주파수에 따라 특징점과 레벨의 수, 윈도우의 크기를 결정해야 한다.

### 1. 해리스 코너 검출기 모듈

해리스 코너 검출기를 이용하여 특징점을 추출할 때 사용되는 코너 응답은 영상 밝기에 따라 변화하므로, 그에 적용되는 임계값 또한 적응적일 필요가 있다.

그림 4의 영상 (a)는 전체적으로 밝고 에지와 코너 성분이 많아 코너 응답이 큰 반면, 영상 (b)와 같이 어두운 영상은 밝기와 그의 미분치가 작은 값을 가지게 되어 코너 응답 또한 작은 값을 가진다. 따라서 고정된 임계값을 지정하여 사용하는 경우 코너 응답의 차이에 의해 추출되는 특징점의 수가 영상 (a)의 경우 398개, (b)의 경우 167개로 크게 변화하게 되므로, 특징점의 수를 일정하게 유지하기 위해서는 적응적인 설정이 필요하다. 그림 5는 영상 밝기에 적응적인 임계값 설정을 위해 제안하는 하드웨어 구조이다.

입력 영상으로부터 코너 응답,  $R$ 과 영상 전체에 대한  $R$ 의 최대값,  $R_{max}$ 를 구한다.  $R$ 의 최대값은 전체 영상을 입력받은 후에 얻을 수 있기 때문에 현재 프레임의 임계값을 설정하는데 즉각적으로 이용하려면 큰 저장 공간과 지연 시간이 필요하다. 따라서 이전 프레임과 현재 프레임이 크게 변하지 않는다는 가정을 도입하여 이전 프레임에서 구한  $R$ 의 최대값으로 현재 영상

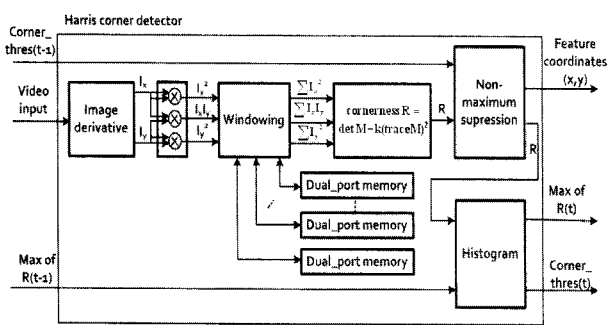


그림 5. 해리스 코너 검출기의 하드웨어 구조  
Fig. 5. Hardware architecture of Harris corner detector.

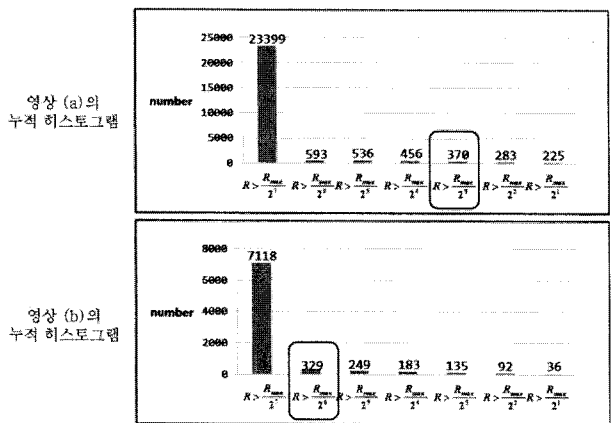


그림 6. <그림 4>의 영상 (a)와 (b)의 누적 히스토그램  
Fig. 6. Accumulative histogram of Fig. 4(a)-(b).



영상 (a)의 특징점 : 357      영상 (b)의 특징점 : 335

그림 7. 적응적인 임계값을 적용한 경우 추출된 특징점의 수  
Fig. 7. Number of selected features using adaptive threshold value.

의  $R_{max}$ 를 대체한다.  $R_{max}$ 의 값은 누적 히스토그램의 bin의 범위를 정의하기 위해 사용되며,  $R_{max}$ 에  $2^n$ 을 나누어 각각의 히스토그램 bin의 범위를 정한다. 어떤 수에  $2^n$ 을 나눌 수는 간단한 shift 연산으로 구할 수 있으므로 이 연산에 따른 부담은 무시할 수 있다.

그림 6은 그림 4의 영상 (a)와 (b)의 코너 응답에 대한 누적 히스토그램이다. 누적 히스토그램의 분포로부터 원하는 특징점의 수를 초과하지 않는 bin의 범위를 임계값으로 사용한다. 그림 7은 약 400개의 특징점을

얻을 수 있도록 적응적인 임계값 설정 방법을 이용한 특징점 추출 결과이다. 추출된 특징점의 수가 영상 (a)는 357개, 영상 (b)는 335개로 고정된 임계값을 사용하는 방법에 비해 추출되는 특징점의 수를 일정하게 유지하는 것을 확인할 수 있다.

## 2. 변위 벡터 계산 모듈

변위 벡터 계산(Computation of optical flow at level) 모듈은 특징점의 위치 정보와 이전 레벨에서 누적된 변위를 이용하여 외부 메모리로부터 이미지 패치의 픽셀 값을 읽어, 변위를 반복적으로 계산한 후 해당 레벨에서의 최종 변위 벡터를 추정하는 모듈이다. 이 모듈의 하드웨어 구조는 그림 8과 같다. 이미지 패치의 픽셀 값을 입력으로 받고, 해당 레벨에서의 추정된 변위와 상태 신호(status signal)를 결과로 한다.

특징점을 추적하는 과정에서 다음 세 경우에 해당될 경우 소실된 것으로 판단한다.

- 추적 중인 특징점이 영상의 범위 밖으로 이동한 경우
- 영상  $I(t-2)$ 와  $I(t-1)$ 에서 추적된 점 주위의 영역의 밝기 차이가 큰 경우
- $G$  행렬의 행렬식(determinant)이 0인 경우

각 계산 과정에서 특징점이 소실되었다고 판단될 경우, 반복(iteration) 블록으로 전달되는 상태 신호를 통해 더 이상의 반복 변위 추정이 진행되지 않도록 한다. 본 논문에서는 소실된 특징점의 대체에 대해서는 고려하지 않았다.

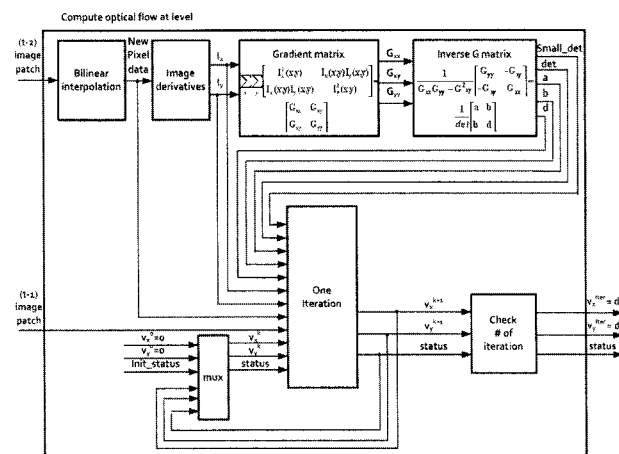


그림 8. 변위 벡터 계산 모듈의 구조  
Fig. 8. Architecture of module for computation optical flow at level.

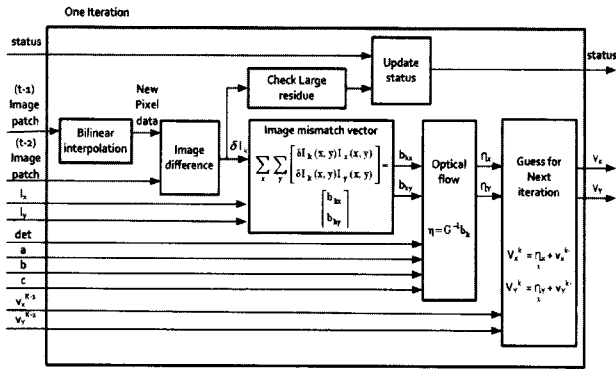


그림 9. 개별 반복 블록의 구조  
Fig. 9. Structure of individual iteration block.

추적된 특징점의 좌표는 실수의 형태이므로 선형 보간(bilinear interpolation)을 이용하여 부화소 위치(subpixel location)에 있는 영상의 밝기를 추정한다. 영상의 미분은 소벨 연산자(Sobel operators)를 이용하여 계산한다. 소벨 연산자는 덧셈기만을 이용하여 구현할 수 있으므로, 간단한 하드웨어 구조를 가진다.

반복 블록은 각 레벨에서 계산된 미분치와  $G$  행렬을 이용하여 변위를 추정한다. 좀 더 정확한 해를 얻기 위해 일정 횟수만큼 변위 추정을 반복하는데, 반복 블록의 출력은 다음 변위 추정 시 이미지 패치를 이동하는데 사용되므로 반복 블록의 입력으로 되먹임된다. 개별 반복 블록의 하드웨어 구조는 그림 9에 나타내었다.

### IV. 실험결과

설계한 하드웨어 모듈을 FPGA에서 검증하기 위하여 Xilinx사의 ISE를 사용하여 합성을 하였다. 실험은 입력 시퀀스로부터 특징점 추출 및 추적이 올바르게 이루어지는지를 공개 알고리즘<sup>[8]</sup>과 비교하여 검증하였다. 검증에 사용한 공개 알고리즘은  $G$  행렬의 두 고유값에 임계치를 적용함으로써 특징점을 추출한다. 검증 결과 그림 10의 결과를 통해 서로 다른 특징점 추출 알고리즘의 적용과 구현시의 절단 오차(bit truncation error)등으로 인해 일부 차이를 보이지만 전반적으로 유사한 양상이 나타남을 확인할 수 있다. 소실점은 각각 2개씩으로, 서로 다른 특징점이 추출되었지만 추출된 특징점에 대해 유사한 추적 성능을 보였다.

계층적 KLT 특징 추적 하드웨어에 사용된 주요 자원은 표 1에 나타내었으며, 이 결과는 Xilinx사의 Virtex5 XC5VLX330에서 구현하였을 때의 사용량이다.

구현된 시스템은 이전 두 프레임을 이용하여 추적이 이루어지므로, 세 번째 프레임부터 실질적인 추적이 수행된다. 검출 속도는 추출하는 특징점의 수를 300여개로 설정하였을 때, 소프트웨어의 경우 펜티엄 3GHz의 프로세서로 720x480 영상에 대해 약 5.8 fps(frame/sec)의 성능을 보였다. 100MHz의 주파수로 동작하는 FPGA를 이용하여 테스트한 결과 동일한 영상 크기에

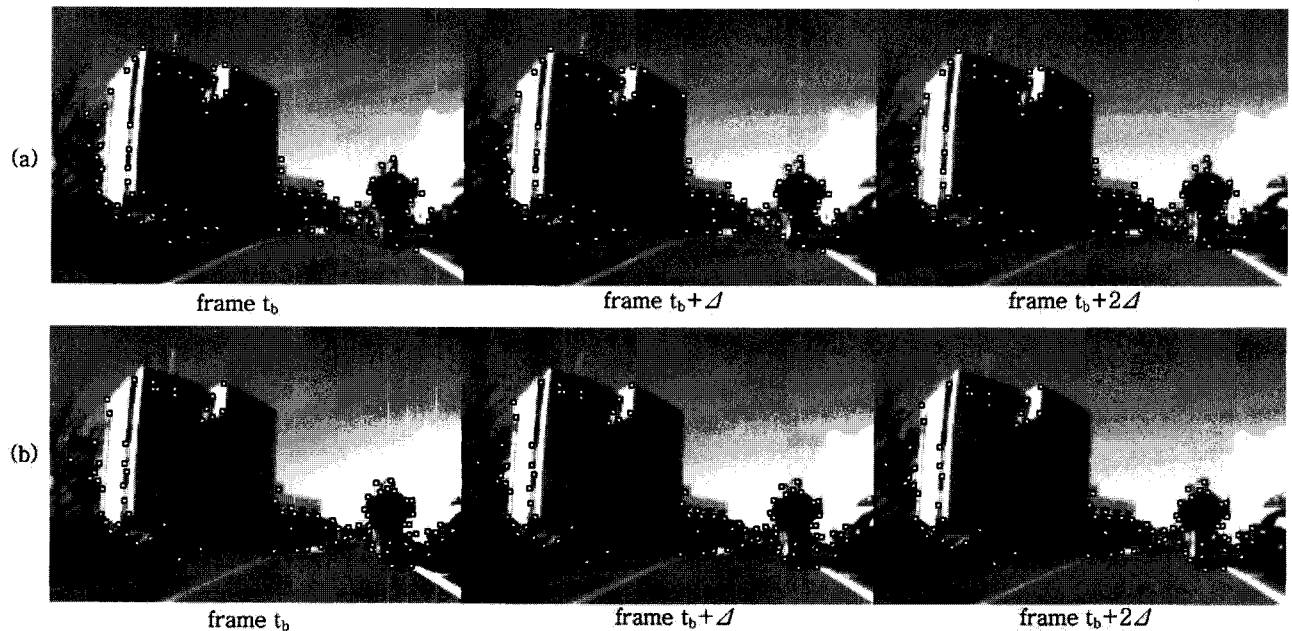


그림 10. 구현 결과 비교 : (a) 소프트웨어 구현 결과, (b) 하드웨어 구현 결과  
Fig. 10. Comparison of experimental results : (a) Result of software implementation, (b) Result of hardware implementation.

표 1. FPGA 자원 사용률

Table 1. FPGA resource utilization.

Number of Slices Registers	21,502 out of 207,360
Number of Slice LUTs	20,880 out of 207,360
Number of occupied Slices	8,225 out of 51,840
Number of route-thrus	1,094 out of 414,720
Number of Block RAM/FIFO	22 out of 288

표 2. 성능 비교

Table 2. Comparison of performance.

성능 비교(720x480영상, 300개 특징점)		
실험 환경	동작 주파수	성능
Intel Pentium 4 CPU	3 GHz	5.8 fps(172 ms)
FPGA	100 MHz	30 fps

화소 클럭이 27MHz인 카메라 입력에 대해 30 fps의 full-frame rate로 동작하여 실시간 처리가 가능함을 확인하였고, 이를 표 2에 정리하였다. 구현된 시스템은 최대 209.249MHz까지 동작할 수 있으므로 더 높은 프레임률이 요구되는 시스템에 적용하는 것이 가능하다.

### V. 결 론

본 논문에서는 영상의 밝기에 적응적인 특징 추출 알고리즘을 제안하였고, 특징점 추출과 추적을 실시간으로 처리하기 위한 하드웨어 구현에 대하여 설명하였다. 계층적 KLT 특징 추적 알고리즘은 하드웨어의 병렬처리 특성에 적합하도록 변환되었으며, 메모리의 효율적인 사용 방안이 제시되었다.

본 연구를 통해 구현된 특징 추적 하드웨어는 주차공간탐색시스템, 감시시스템, 물체 추적 등 비전의 다양한 분야에 접목시켜 이용이 가능하다. 앞으로의 연구는 추적되는 특징점의 수가 일정하게 유지될 수 있도록 소실된 특징점을 새로운 특징점으로 대체하는 방안에 대한 연구가 진행될 것이다.

### 참 고 문 헌

[1] J. K. Suhr, K. H. Bae, J. H. Kim, "Free parking space detection using optical flow-based euclidean 3D reconstruction," IAPR Conf on Machine Vision Application(MVA'07), pp. 563-566, May 2007.

[2] C. Tomasi and T. Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[3] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," Intel Corporation, 2003.

[4] P. Arato, G. Kocza, I. Lovanyi and L. Vajta, "Hardware-software codesing of feature tracking algorithms," in Intelligent Engineering Systems(INES 2008), pp. 41-45, February 2008.

[5] P. D. Fiore, D. Kottke, W. Krawiec and C. David, "Efficient feature tracking with application to camera motion estimation," in Signals, Systems and Computers, pp. 949-953, November 1998.

[6] C. Harris and M. J. Stephens, "A combined corner and edge detector," in Alvey Vision Conference, pp. 147-152, 1988.

[7] C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of interest point detectors," International Journal of Computer Vision, pp. 151-172, June 2000.

[8] S. Birchfield, "KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker," <http://www.ces.clemson.edu/~stb/klt>, November 2005.

[9] C. Cabani and W. J. MacLean, "A proposed pipelined-architecture for FPGA-based affine-invariant feature detectors," in Proc. of IEEE Conf on Computer Vision and Pattern Recognition Workshop January 2006.

[10] S. Ghiasi, H. J. Moon, A. Nahapetian and M. Sarrafzadeh, "Collaborative and reconfigurable object tracking," Kluwer Journal of Supercomputing, Vol. 30, No 3, pp. 213-238, December 2004.

[11] W. J. MacLean, "An evaluation of the suitability of FPGAs for embedded vision systems," in the first IEEE Workshop on Embedded Computer Vision, CVPR 2005, June 2005.

---

 저 자 소 개
 

---



김 현 진(학생회원)  
 2007년 2월 서강대학교  
 전자공학과 학사  
 2009년 2월 서강대학교  
 전자공학과 석사  
 <주관심분야 : 영상신호해석, 컴  
 퓨터비전, 임베디드시스템 디자  
 인>



김 경 환(정회원)-교신저자  
 1984년 서강대학교  
 전자공학과 학사  
 1986년 서강대학교  
 전자공학과 석사  
 1996년 State University of New  
 York at Buffalo 전기 및  
 컴퓨터 공학과 박사  
 1997년 9월~현재 서강대학교 전자공학과 교수  
 <주관심분야 : 영상신호해석, 패턴인식, 문서영상  
 처리, 임베디드시스템 디자인>