

논문 2014-51-11-23

FAST 하드웨어 가속기를 위한 임계값 제어기

(A Threshold Controller for FAST Hardware Accelerator)

김 택 규*, 서 용 석*

(Taek-Kyu Kim[Ⓢ] and Yong-Suk Suh)

요 약

카메라와 같이 연속적인 영상을 제공하는 환경에서 특징 점들을 추출하기 위해 다양한 알고리즘들이 연구되고 있다. 특히, FAST (Feature from Accelerated Segment Test) 알고리즘은 연산 구조가 간단하고 실시간 특징 점 추출이 용이하여 FPGA 기반 하드웨어 가속기로 구현되어 사용되고 있다. 사용된 FAST 하드웨어 가속기는 특징 점을 추출하기 위해 임계값을 필요로 한다. 임계값은 영상에서 추출되는 특징 점의 기준이 되는 값으로, 값의 크기에 따라 추출되는 특징 점의 개수가 정해질 뿐만 아니라 전체 수행시간에도 영향을 주기 때문에, 일정한 수행시간 동안에 많은 특징 점들을 추출하기 위해서는 적절한 임계값 제어 방법이 요구된다. 본 논문에서는 임계값 제어를 위해 PI 제어기를 제안한다. 제안한 PI 제어기는 시험 영상들을 통해 기능 및 성능을 검증하였고, Xilinx Vertex IV FPGA 기반의 로직으로 구현 비용을 계산하였다. 제안한 PI 제어기는 47개의 Flip Flops, 146개의 LUTs, 그리고 91개의 Slices을 사용해, FAST 하드웨어 가속기 2.1%의 Flip Flop, 4.4%의 LUTs, 그리고 4.6%의 Slice에 해당하는 적은 비용으로 구현되었다.

Abstract

Various researches are performed to extract significant features from continuous images. The FAST algorithm has the simple structure for arithmetic operation and it is easy to extraction the features in real time. For this reason, the FPGA based hardware accelerator is implemented and widely applied for the FAST algorithm. The hardware accelerator needs the threshold to extract the features from images. The threshold is influenced not only the number of extracted features but also the total execution time. Therefore, the way of threshold control is important to stabilize the total execution time and to extract features as much as possible. In order to control the threshold, this paper proposes the PI controller. The function and performance for the proposed PI controller are verified by using test images and the PI control logic is designed based on Xilinx Vertex IV FPGA. The proposed scheme can be implemented by adding 47 Flip Flops, 146 LUTs, and 91 Slices to the FAST hardware accelerator. This proposed approach only occupies 2.1% of Flip Flop, 4.4% of LUTs, and 4.5% of Slices and can be regarded as a small portion of hardware cost.

Keywords : PI controller, threshold controller, FAST hardware accelerator, feature from accelerated segment test

I. 서 론

* 정회원, 한국원자력연구원 연구로계통설계부
(Research Reactor System Design Department,
Korea Atomic Energy Research Institute)

Ⓢ Corresponding Author(E-mail: taekkyukim@kaeri.re.kr)

※ 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(원자력연구개발사업, No. 2012M2C1A1026912)

접수일자: 2014년08월19일, 수정일자: 2014년10월05일
게재확정: 2014년10월27일

카메라처럼 실시간으로 입력받은 영상을 활용해 정보를 처리하는 분야에서는 특징 점들을 추출하기 위해 다양한 알고리즘들을 연구하고 있다. 대부분의 알고리즘들은 소형 내장형 시스템에서 디지털 회로로 구현해 사용하기에는 수식이 복잡하거나 계산 양이 많아 구현 비용이 높기 때문에^[1~2] 비교적 연산구조가 단순하고

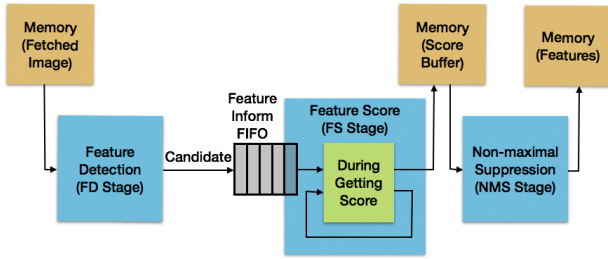


그림 1. 간략한 FAST 하드웨어 가속기 구조
Fig. 1. Simplified structure of FAST hardware accelerator.

계산 양이 적은 SUSAN이나^[3] Featured from Accelerated Segment Test^[4] (이하: FAST) 알고리즘들이 각광받고 있다. 이들은 비록 수식의 복잡도가 높은 알고리즘들과 비교하여 추출된 특징 점들의 고유성이나 정확도가 떨어지지만, 수식이 간단해 적은 비용의 디지털 회로로 실시간 처리가 용이한 장점을 가진다.

이중 E. Rosten이 제안한 FAST 알고리즘은 가/감산기와 비교기를 가지는 연산 구조로 적은 비용의 디지털 회로로 하드웨어 가속기 구현이 가능하다. 본 연구에서 활용한 FAST 하드웨어 가속기는^[5-6] 특징 점 후보들을 찾는 Feature Detection(이하: FD), 후보들의 중요도를 정하는 Feature Score(이하: FS), 그리고 중요도가 높은 후보들을 찾아 특징 점으로 결정하는 Non-maximal Suppression(이하: NMS)으로 구성되어 약 4,000개의 특징 점을 3.142 ms 동안에 추출한다.

FAST 알고리즘은 0에서 255까지 임의의 정의 가능한 임계값을 이용해 특징 점을 추출하며 하드웨어 가속기 또한 동일하다. 임계값은 영상에서 추출되는 특징 점의 특징 정도와 추출 개수에 영향을 미친다. 너무 작은 임계값은 추출된 특징 점들의 특징 정도가 낮을 뿐만 아니라 추출된 특징 점 개수를 크게 증가시켜 과도한 연산량으로 전체 수행 처리 시간을 늘린다. 반면, 너무 큰 임계값은 특징 점의 특징 정도는 강하나 추출된 특징 점을 개수가 적어 다른 후처리 영상 알고리즘들에게 불충분한 정보를 제공하게 된다. 그렇기 때문에 FAST 알고리즘 기반 하드웨어 가속기의 적절한 임계값 조절이 필요하다.

본 연구에서는 차량 주행 영상처럼 입력 영상이 다양하게 변하는 조건에서 하드웨어 가속기의 임계값을 조절하기 위해 제어분야에서 자주 사용되는 PI 제어기를 제안한다. 설계된 PI 제어기를 시험하기 위해 주행 시에 발생 가능한 영상의 민감한 변화나 둔감한 변화들을

가정하고 실제 주행을 통해 시험 영상들을 획득하였다. 이후 구현한 PI 제어기와 FAST 하드웨어 가속기를 연동하여 동작 및 성능을 검증하였다.

II. 본 론

1. FAST-9 알고리즘

본 연구에서 사용한 FAST 하드웨어 가속기는 그림 1과 같은 구조를 가진다. 기준 픽셀과 16개의 인접한 픽셀들 간의 비교를 통해 특징 점을 추출하는 기법으로, 특징 점 후보들을 추출하는 FD 단계, 후보들의 특징 정도에 따라 점수를 부여하는 FS 단계, 그리고 인접한 후보들 간에 점수를 비교해 점수가 가장 높은 후보를 특징 점으로 정의하는 NMS 단계로 구성된다.

$$S_{p \rightarrow x} = \begin{cases} S_{dark}, & I_{p \rightarrow x} \leq I_p - t \\ S_s, & I_p - t < I_{p \rightarrow x} < I_p + t \\ S_{bright}, & I_p + t \leq I_{p \rightarrow x} \end{cases} \quad (1)$$

FD 단계에서는 메모리에 저장된 영상을 읽어 수식 (1)을 사용해 영상 내의 픽셀들에 대해 기준 픽셀 I_p 와 16개의 인접한 픽셀들($I_{p \rightarrow x}$, $x \in \{1, 2, \dots, 15, 16\}$)이 임계값 t 를 기준으로, $I_{p \rightarrow x}$ 가 I_p 와 임계값 t 의 차보다 작으면 S_{dark} 로, I_p 와 임계값 t 의 합보다 크면 S_{bright} 로 정의한다. 그리고 정의된 $I_{p \rightarrow x}$ 들이 9개 이상 연속으로 S_{dark} 이거나 S_{bright} 이면 기준 픽셀을 특징 점 후보로 정의하고 Feature Inform FIFO에 저장한다.

$$V = \max \left(\sum_{x \in S_{bright}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t \right) \quad (2)$$

이후 FS 단계에서는 수식 (2)에 따라 각각 특징 점 후보의 점수를 매긴다. 임계값 t 를 기준으로 값을 증가시키면서 최대 특징 점 상태가 유지되는 임계값을 찾아 점수로 정의하고 Score Buffer 메모리에 저장한다.

NMS 단계에서는 인접한 특징 점 후보들 간에 FS 단계에서 얻어진 점수를 비교하여, 가장 점수가 큰 특징 점 후보를 특징 점으로 결정해 Features 메모리에 저장한다.

2. 하드웨어 가속기 문제점 도출

하드웨어 가속기의 FS 단계는 입력받은 특징 점 후보들을 초기에 설정된 임계값 t 를 기준으로 증가시키면

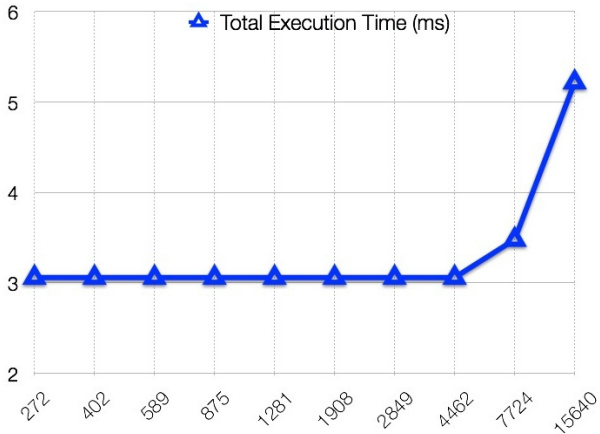


그림 2. 추출된 특징 점 후보 개수에 따른 수행 시간
Fig. 2. Total execution time depending on the number of extracted feature candidates.

서 특징 점 상태가 유지 가능한 최대 임계값을 찾을 때까지 반복 수행된다. 최대 임계값 t 를 찾을 때까지 임계값을 바꾸면서 반복 수행하는 동안, 다른 특징 점 후보들은 처리를 기다려야한다. 그래서 FD 단계가 많은 특징 점 후보를 생성하면 FS 단계에서 각각의 특징 점 후보에 대해 반복연산을 수행하여 오랜 시간이 소요되고, 그로인해 가속기의 Feature Inform FIFO에 대기 중인 특징 점 후보들이 가득 차게 된다. 특징 점 후보가 가득 차면 FD 단계는 Feature Inform FIFO가 저장가능한 상태가 될 때까지 일시적으로 중단되어 전체 수행 시간에 영향을 미친다. 그림 2는 하드웨어 가속기 검증에 사용된 시험 영상을^[5] 활용해 임계값을 바꾸면서 추출된 특징 점 후보의 개수와 수행을 마치는데 소요된 전체 수행시간을 보여준다. 추출된 특징 점 후보의 개수가 4,462 이하에서는 3.142ms로 안정적인 수행 시간을 보였으나, 특징 점 후보의 개수가 4,462 개를 초과하면서부터 전체 수행 시간이 증가하였다. 이처럼 임계값 t 는 하드웨어 가속기 전체 수행 시간에 영향을 미치기 때문에 적절한 조절이 필요하다.

3. 임계값 제어기 구현

본 연구에서는 하드웨어 가속기의 전체 수행 시간이 3.142ms를 초과하지 않는 범위에서 자동차 주행처럼 영상이 연속적으로 변화하는 환경에서 임계값을 제어하기 위해 제어 응용 분야에서 사용되는 PI 제어기를^[7] 제안한다. 제안한 PI 제어기는 제어가 필요한 다양한 응용 분야에서 사용되는 기법으로, 비례(Proportional), 적분

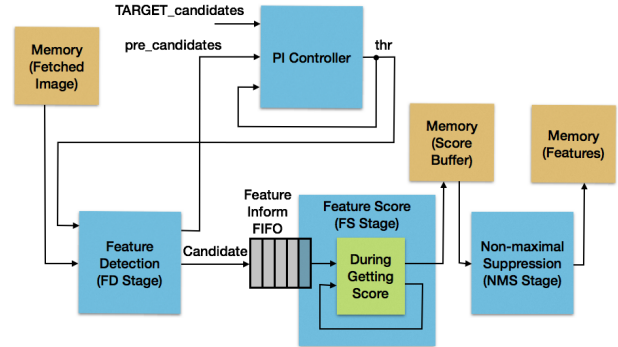


그림 3. 제안하는 임계값 제어기 연결구조
Fig. 3. Configuration for the proposed threshold controller.

(Integral)을 가지는 피드백 제어기 형태이다. 제어하고자 하는 대상의 출력값을 측정하여 원하는 설정값과 비교해 오차를 계산하고, 이 오차값을 이용해 제어에 필요한 제어값을 획득하는 구조를 지닌다.

본 연구에서는 임계값(thr)을 제어하기 위해 수식 (3, 4, 5, 6)으로 구성된 임계값 제어기를 설계하고, 그림 3 같이 FAST 하드웨어 가속기에 연결하였다.

$$thr = pre_thr - (e(t) \gg K_p) - \left(\int_0^t e(\tau) d\tau \gg K_i \right) \quad (3)$$

$$e(t) = TARGET_candidates - pre_candidates \quad (4)$$

$$e(t) \gg K_p = \begin{cases} Max, & (e(t) \gg K_p) > Max \\ K_p e(t), & Max \geq (e(t) \gg K_p) \geq Min \\ Min, & (e(t) \gg K_p) < Min \end{cases} \quad (5)$$

$$\int_0^t e(\tau) d\tau = \begin{cases} \int_0^t e(\tau) d\tau, & \left(\int_0^t e(\tau) d\tau \gg K_i \right) = 0 \\ 0, & \left(\int_0^t e(\tau) d\tau \gg K_i \right) \neq 0 \end{cases} \quad (6)$$

임계값 제어에 기본이 되는 수식 (3)에서 현재 영상 (t) 의 임계값(thr)은 이전 영상 $(t-1)$ 에서 사용한 임계값 (pre_thr) 에 $e(t)$ (이하: 비례항 오차값)와 $\int_0^t e(\tau) d\tau$ (이하: 적분항 오차값)를 뺀 값으로 정의된다. 비례항 오차값은 수식 (4)와 같이 목표치인 특징 점 후보 개수

(TARGET_candidates)와 이전 영상에서 추출된 특징 점 후보 개수(pre_candidates)의 차로 구해지고, 적분항 오차값은 비례항 오차값의 누적 값으로 구해진다. 추가적으로 비례항 오차값과 적분항 오차값을 조절하기 위해 K_p , K_i 이득값이 사용되었다.

임계값 제어는 비례항/적분항의 오차값과 이득값의 곱의 합이다. 이때 곱셈 연산은 분해능이 높은 즉, 정밀한 연산에 좋지만 디지털 회로로 구현시 높은 비용과 낮은 동작 속도를 가진다. 본 논문에서 사용하는 PI 제어기에서 조절해야하는 임계값은 정수로 0에서 255까지인 정수로서, 높은 분해능을 가지지 않기 때문에 저렴한 비용과 빠른 동작 속도로 처리가 가능한 비트 이동 연산으로 변경하였다.

여러 시험 영상을 통해 수식 (3)의 K_p 이득값을 결정하는 과정에서 대략적으로 임계값이 1만 변하더라도 특징 점 후보가 약 200개 이상으로 크게 변하는 것을 확인하였다. 이렇게 임계값 변화에 민감한 영상은 K_p 이득값을 작게 가져가게 되면, 순간적으로 추출된 특징 점 후보의 개수가 목표치와 크게 벗어나게 되어 $e(t) \gg K_p$ 값의 변화량이 커져 목표치에 수렴하지 못하고 임계값이 진동하는 문제가 발생하였다. 거꾸로 K_p 이득값을 크게 가져가면 진동하는 문제를 해결할 수 있으나, 비례항을 통해 오차값을 충분히 줄이기 힘든 단점이 발생한다. 본 연구에서는 이득값을 가능한 작게 잡으면서 $e(t) \gg K_p$ 값의 변화량을 줄이기 위해 제한치를 설정하는 수식 (5)를 추가하였다. 수식 (5)를 통해 $e(t) \gg K_p$ 값이 Max 를 초과하지 않고 Min 미만으로 내려가지 않게 하였다.

수식 (6)은 적분항 오차값을 보정하기 위해 사용하였다. 적분항 오차값은 누적 오차의 특성으로 인해 누적 오차가 너무 커지면 제어하기가 힘들뿐만 아니라, 적분항 오차값이 임계값에 영향을 미치는 시점부터 영향을 미치지 않는 적분항 오차값 범위 이하로 줄어들 때까지 지속적으로 임계값을 증가 또는 감소시키게 된다. 지속적으로 임계값이 증가 또는 감소하게 되면 특징 점 후보의 개수가 목표치로부터 벗어나기 때문에, 수식 (6)과 같이 적분항 오차값이 임계값에 반영될 때마다 0으로 초기화하도록 했다. 비록 수식 (5, 6)에 의해 임계값 제어기의 급격한 추적 반응은 다소 떨어질 것으로 예상되지만, 안정적인 임계값 제어가 기대된다.



그림 4. 임계값 제어기 시험 영상 (좌측 상단부터 터널 통과, 톨게이트 통과, 신호대기, 고속도로표지판)
Fig. 4. Test images for the threshold controller (From the top of left. passing tunnel, passing tollgate, waiting for signal, passing highway sign).

III. 실험

본 장에서는 제안한 PI 제어기의 성능과 하드웨어의 구현비용을 설명하고, FAST 하드웨어 가속기에 추가되는 비용을 평가하였다.

1. 성능

본 연구에서 제안한 PI 제어기는 제어에 필요한 목표치, 이득값과 추가적으로 최대 허용 목표치와 임계값 제한치를 사용하였다. K_p 와 K_i 이득값은 그림 4의 시험 영상을 통해 결정하였다. 단, K_i 이득값의 찾는 과정에서는 항상 K_p 의 이득값보다 크게 설정하면서 K_i 이득값을 찾았다. 특징 점 후보가 순간적으로 과다하게 추출되어 전체 수행 시간에 영향을 주지 못하도록 최대 허용 목표치를 설정하였고, 초과 시에는 해당 영상의 특징 점 추출을 종료하고 다음 영상에서 특징 점을 추출한 준비를 하도록 하였다. 또한 FAST 알고리즘의 특성상 임계값이 너무 낮으면 불필요한 특징 점을 다수 추출하기 때문에 임계값의 최솟값을 제한하였다.

- TARGET_candidates 목표치: 3,000개
- K_p 이득값: 7
- K_i 이득값: 10
- 최대 허용 목표치: 3,500개
- 임계값 제한치: $25^{[8]}$ ~ 255

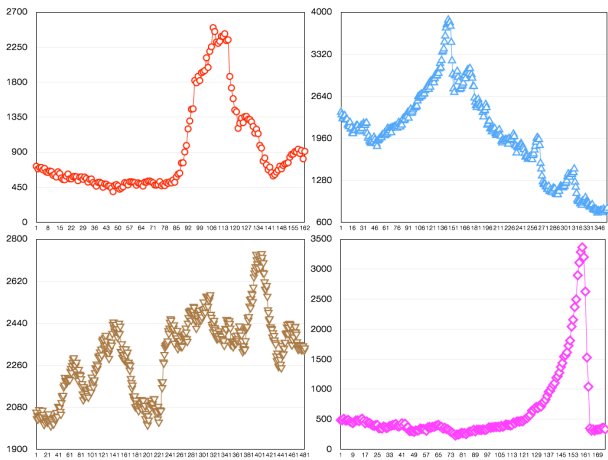


그림 5. 기존 하드웨어 가속기 시험 결과 (임계값 40, 좌측 상단부터 터널통과, 톨게이트통과, 신호대기, 고속도로표지판)

Fig. 5. Test results of original hardware accelerator (From the top of left. passing tunnel, passing tollgate, waiting for signal, passing highway sign).

시험에는 그림 4와 같은 영상들이 사용되었다. 시험에 사용된 영상들은 특징 점 후보들이 시간 변화에 따라 급격/완만하게 변하도록 고려하였다. 그림 5는 기존의 고정된 임계값을 가지는 FAST 하드웨어 가속기로부터 추출된 특징 점 후보 개수이고, 그림 6은 제안하는 임계값 제어기를 가지는 FAST 하드웨어 가속기가 추출한 특징 점 후보 개수이다. 두 개의 하드웨어 가속기 모두 특징 점 후보가 최대 허용 목표치를 초과하는 경우 하드웨어 가속기의 수행 속도에 영향을 미치지 않고 다음 입력 영상을 정상적으로 처리하도록 예외 처리를 하였다.

제안하는 PI 임계값 제어기를 가지는 FAST 하드웨어 가속기를 기존의 고정된 임계값을 가지는 FAST 하드웨어 가속기와 비교하기 위해 시험 영상을 처리하는 동안 최대 허용 목표치를 초과한 횟수와 추출된 특징 점 후보의 평균을 계산하였다. 기존의 하드웨어 가속기는 톨게이트통과 시험 영상에 대해서 10 프레임 동안 최대 허용 목표치를 초과하였고, 제안하는 임계값 제어기를 가지는 하드웨어 가속기는 터널통과 시험 영상에서 최대 허용 목표치를 1 프레임 동안 초과하였다. 다음은 시험 영상에 대한 프레임 당 평균 특징 점 후보 개수이다. 본 논문에서 사용한 FAST 하드웨어 가속기는 추출 중인 특징 점 후보가 최대 목표 허용치를 초과해 발생하는 예외 처리 요청 신호를 받으면 처리 중인

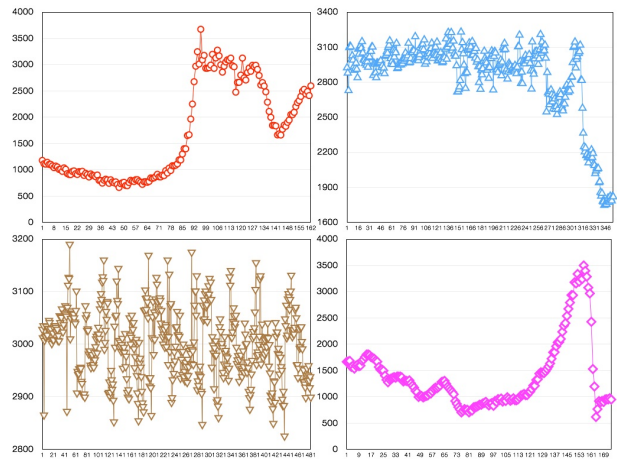


그림 6. 임계값 제어기 시험 결과 (좌측 상단부터 터널 통과, 톨게이트통과, 신호대기, 고속도로표지판)

Fig. 6. Test results of the threshold controller (From the top of left. passing tunnel, passing tollgate, waiting for signal, passing highway sign).

영상 프레임에 대해서는 더 이상 특징 점 후보를 추출하지 않는다. 그렇기 때문에 최대 목표 허용치를 초과한 경우에는 추출된 특징 점 후보의 개수를 3,500개로 간주하고 평균값을 계산하였다. 기존의 고정된 임계값을 가지는 FAST 하드웨어 가속기는 매 프레임마다 평균적으로 터널통과 시험 영상은 912개, 톨게이트통과 시험 영상은 2,016개, 신호대기 시험 영상은 2,305개, 그리고 고속도로 표지판 시험 영상은 628개의 특징 점 후보를 추출하였고, 제안한 PI 임계값 제어기를 가지는 FAST 하드웨어 가속기는 매 프레임마다 평균적으로 터널통과 시험 영상은 1,685개, 톨게이트통과 시험 영상은 2,850개, 신호대기 시험 영상은 2,998개, 그리고 고속도로 표지판 시험 영상은 1,405개의 특징 점 후보를 추출하였다. 시험 결과를 통해 제안하는 임계값 제어기를 추가하였을 시에 최대 허용치를 초과하는 횟수가 줄어들음을 확인하였고, 각 시험 영상에 대해서 특징 점 후보가 목표치인 3,000개에 좀 더 근접하도록 많이 추출되었음을 확인하였다.

2. 구현비용

제안한 PI 기반의 임계값 제어기는 자일링스 사의 Vertex IV FX FPGA에서 100MHz로 동작하도록 구현되었다. 표 1은 제안한 임계값 제어기의 디지털 회로 구현 비용을 나타낸다. 제안한 제어기는 FAST 하드웨어 가속기에 47개의 Flip Flops, 146개의 LUTs, 그리고

표 1. 하드웨어 구현 비용
Table 1. Hardware Implementation Cost.

Resource	FAST Hardware Accelerator[5]	PI Threshold Controller
Slice Flip Flops	2,215	47
4 input LUTs	3,315	146
Occupied Slices	1,750	91
Block RAMs	17	0
MULT18X18s	0	0

91개의 Slices가 추가로 사용되어 연동되었다. 이는 제안한 PI 임계값 제어기가 FAST 하드웨어 가속기의 2.1%의 Flip Flops, 4.4%의 LUTs, 그리고 4.6%의 Slices에 해당하는 적은 비용을 추가하여 임계값을 조절 가능함을 보여준다.

IV. 결 론

본 연구에서는 FAST 하드웨어 가속기의 임계값을 조절하기 위해 PI 제어를 활용한 임계값 제어기를 제안한다. 제안한 제어기는 자일링스 사의 Vertex IV FX FPGA에서 47개의 Flip Flops, 146개의 LUTs, 그리고 91개의 Slices를 사용해 구현이 가능하며, 시험 영상들을 통해 카메라와 같이 연속적인 동적 입력 영상에서 특징 점 후보들을 설정된 목표치에 근사하도록 조절해 가속기의 수행 시간에 영향을 미치지 않으면서 최대한 많은 특징 점을 추출하였다. 그렇기 때문에 적은 하드웨어 재원으로 실시간 처리가 요구되는 소형 내장형 응용 분야에서 FAST 하드웨어 가속기와 연동하여 적용 가능할 것으로 기대된다.

REFERENCES

- [1] L. Teixeira, W. Celes, and M. Gattass, "Accelerated Corner-detector Algorithms," in *BMVC08*, 2008. [Online]. Available: <http://www.comp.leeds.ac.uk/bmvc2008/proceedings/paper/45.pdf>
- [2] D. Bouris, A. Nikitakis, and J. Waters, "Fast and Efficient FPGA-Based Feature Detection Employing the SURF Algorithm," 18th IEEE

Annual International Symposium on Digital Object Identifier, pp. 3-10, 2010.

- [3] S. Smith and J. Brady, "SUSAN-A New Approach to Low-level Image Processing," *International Journal of Computer Vision*, vol. 23, pp. 45-48, 1997.
- [4] Edward Rosten et al, "Faster and better: A Machine Learning Approach to Corner Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105-119, 2010.
- [5] T.-K. Kim, "An Embedded FAST Hardware Accelerator for Image Feature Detection," *Journal of IEEK*, vol. 49, SP no. 2, pp. 28-34, 2012.
- [6] T.-K. Kim, Y.-S. Suh, and Y.-K. Kim, "Implementation of Feature Point Extraction Accelerator Based on FAST," *The IEEK Summer Conference*, vol. 36, no. 1, pp. 609-612, 2013.
- [7] Norman S. Nise, "Control Systems Engineering," Sixth Edition, John Wiley & Sons, 2011.
- [8] E. Rosten and T. Drummond, "Fusing Points and Lines for High Performance Tracking," *Proc. 10th IEEE International Conference Computer Vision*, vol. 2, pp. 1508-1515, 2005.

저 자 소 개



김택규(정회원)

2005년 고려대학교 전자 및 정보공학부 졸업.

2008년 고려대학교 전자정보공학과 석사 졸업.

2010년~현재 한국원자력연구원 선임 연구원

<주관심분야 : 마이크로 프로세서, 임베디드 시스템, 영상처리>



서용석(정회원)

1987년 광운대학교 전산학과 학사 졸업.

1996년 충남대학교 컴퓨터공학과 석사 졸업.

2011년 충남대학교 컴퓨터공학과 박사 졸업.

1987년~현재 한국원자력연구원 책임연구원

<주관심분야 : 컴퓨터 및 계측제어공학>