

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การค้นหามาจากการอ้างอิงของผู้ชมในรายการทีวี

End-user Customizable Scene Search for TV Programs



เลขหมู่.....  
เลขทะเบียน..... 72927  
วัน,เดือน,ปี 26 ส.ย. 2550

b. 1121 1912  
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การค้นหาลากจากการอ้างอิงของผู้ชมในรายการทีวี**

**End-user Customizable Scene Search for TV Programs**



**ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต**

**สาขาวิศวกรรมคอมพิวเตอร์**

**คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2549**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การค้นหาฉากจากการอ้างอิงของผู้ชมในรายการทีวี

End-user Customizable Scene Search for TV programs

ผู้จัดทำ

1. นายทศพร เพ็ญรอด รหัสนักศึกษา 45010298
2. นายโรมัส โรเบิร์ต โคลเวอร์ รหัสนักศึกษา 46010328



อาจารย์ที่ปรึกษา

(อาจารย์ อานาจ ขาวเน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ การค้นหาจากภาพการอ้างอิงของผู้ชมในรายการทีวี  
นักศึกษา นายทศพร เพ็ญรอด และ นายโรมัส โรเบิร์ต โกลเวอร์  
รหัสประจำตัว 45010298 และ 46010328  
ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
พ.ศ. 2549  
อาจารย์ที่ปรึกษา อ.อำนาจ ขาวเน

### บทคัดย่อ

ในปัจจุบันความต้องการของผู้ชมรายการทีวีนั้นมีความหลากหลายมากขึ้น นอกจากความต้องการคุณภาพของสื่อเองแล้ว ก็ยังคงมีความต้องการที่จะเข้าถึงฉากที่ตนเองอยากดูได้ง่ายด้วยเช่นกัน จุดนี้เองทำให้ผู้พัฒนาได้นำเสนอวิธีการค้นหาจากไฟสวิติโอที่อัดมาจากรายการทีวี โดยทดสอบเฉพาะวิดีโอกีฬา โดยนำเสนอเทคนิคพื้นที่คงที่ (Fixed-image Region) บนวัตถุกราฟฟิกตัวหนังสือ (Superimposed Text) มาช่วยในการค้นหา โดยหลักการนั้นผู้ชมต้องออกแบบรูปแบบการค้นหาเอง เพื่อให้ได้สิ่งที่ต้องการที่ใกล้กับคนได้มากที่สุด

ทีมพัฒนาได้พัฒนาออกมาใน 2 รูปแบบ คือ พัฒนาทางด้านซอฟต์แวร์ด้วย VC++.net และ OpenCV Library ซึ่งเป็นการจำลองการค้นหาจากโดยอ้างอิงจากผู้ใช้ โดยผู้พัฒนาจะมีการจำลอง Pattern ของกีฬาชนิดต่างๆ คือ เบสบอล, ฟุตบอล และ แบดมินตัน เป็นต้น แล้วทดลองการค้นหาจากตามเหตุการณ์ดังกล่าว และแนวทางที่สองคือ พัฒนาทางด้านฮาร์ดแวร์โดยเริ่มจากการจำลองบน Simulink ของ MATLAB และมีการพัฒนาลงบนบอร์ดทดลอง Viretux-II Pro ของ Xilinx โดยมีการศึกษาเครื่องมือที่ช่วยในการพัฒนา 2 ชนิด คือ System Generator ของ MATLAB และ EDK 8.1 ของ Xilinx โดยในส่วนของการพัฒนาทางด้านฮาร์ดแวร์นั้นจะเป็นการนำอัลกอริทึม Superimposed Text Detection ลงบนบอร์ดทดลอง ผลการทดลองความถูกต้องของอัลกอริทึมโดยวัดประสิทธิภาพเฉลี่ย (Precision เท่ากับ 97.67% และ Recall เท่ากับ 78.72%)

<b>Topic</b>	End-user Customizable Scene Search for TV programs
<b>Student</b>	Todsaporn Fuangrod and Thomas Robert Clover
<b>Student ID</b>	45010298 and 46010328
<b>Degree</b>	Bachelor of Engineering
<b>Faculty</b>	Computer Engineering
<b>Year</b>	2006
<b>Advisor</b>	AJ. Amnacd Khawne

### บทคัดย่อ

The technology of searching is a crucial demand of viewers in TV programs recently. Since semantic content of video is difficult to automatically extract, thus our issue focuses on fixed-image region presented in superimposed-text often an intuitive clue for an end-user for specifying a certain scene in TV program.

We have 2 ways to develop the system of end-user customizable scene search. Firstly, software based that was implemented using VC++.net was developed and cooperated OpenCV library as well. End-user has to define pattern for searching but for our experiment, we already constructed patterns for Japanese baseball game (NTV), football (ESPN), and badminton as examples. We started simulation our superimposed text detection with Simulink. Then, the XUP Virtex-II Pro had used in our experiment. We are not only studied System Generator of MATLAB but also leant EDK version 8.1 of Xilinx. Precision and Recall are an effective measure to superimposed text algorithm as precision rate and recall rate measure by 97.67 and 78.72 percentage.

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก อ. อานาจ ขาวเน ที่ให้ความช่วยเหลือให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ที่ดี

Many thanks to Ken Nakayama and Prof. M. Maekawa from the university of Electro-communication, Tokyo, Japan for all advise and gave a good opportunity when I am exchange student.

ขอขอบคุณ คุณทัศนีย์ สุวรรณทัต และ คุณ Juanda LOKMAN นักศึกษาปริญญาเอก ที่มหาวิทยาลัย Electro-Communicaiton โตเกียว ประเทศญี่ปุ่น ที่เคยให้ความรู้พื้นฐานด้านดิจิทัลอิมเมจโปรเซสซิ่ง และ การทำงานวิจัย รวมถึงการตีพิมพ์งานวิจัย

ขอขอบคุณ ศูนย์คอมพิวเตอร์และอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ห้องแล็บเทคโนโลยีภาพ ที่คอยให้ความรู้เกี่ยวกับเทคนิควิธีใหม่ๆ รวมถึงเทคโนโลยีทางด้านวิดีโอที่กำลังเป็นที่นิยมวิจัยในปัจจุบันและในอนาคต

ขอขอบคุณ ห้องแล็บ FPGA ของมหาวิทยาลัยมหานคร ที่คอยให้ความรู้ด้าน FPGA

ขอขอบคุณ นายวาทิตย์ วงวาทีน และ นายวรชัย กีก้อง นักศึกษาวิศวกรรมคอมพิวเตอร์ชั้นปี 3 ที่คอยให้ความช่วยเหลือตลอดเวลาทำโครงการ

ขอขอบคุณ บิศา มารดา ที่คอยสนับสนุนพวกเราโดยตลอดมา

ขอขอบคุณ คู่โปรเจก (ซึ่งกันและกัน) ที่ให้ความร่วมมือในการทำงานอย่างดีเยี่ยม ขอขอบคุณในความอุสาหะ และความพยายามทำงานวิจัยในโครงการนี้ ซึ่งนับว่าเป็น โครงการที่ต้องอาศัยความพยายามเป็นอย่างสูง

ขอขอบคุณ ชมรม โรบอท และ ชมรักบี้ฟุตบอล เป็นที่ให้ทำกิจกรรมตลอดเวลาที่เรียนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ขอขอบคุณ อาจารย์ทุกท่าน ในภาควิศวกรรมคอมพิวเตอร์ที่ให้ความรู้ ประสบการณ์ต่างๆ ตลอดจนสั่งสอนสิ่งที่ดีเสมอมา

ทศพร เพ็ญรอด

โรมัส โรเบิร์ต โคลเวอร์

# สารบัญ

หน้า

## บทที่ 1 บทนำ

1.1	ความเป็นมาและความสำคัญของปัญหา.....	1
1.2	วัตถุประสงค์ของโครงการ .....	3
1.3	ประโยชน์ที่คาดว่าจะได้รับ .....	3
1.4	ระเบียบวิธีวิจัย.....	3
1.5	ขอบเขตของโครงการ .....	4
1.6	ส่วนประกอบของรายงาน .....	4

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1	ทฤษฎีพื้นฐานการค้นคืนวิดีโอ.....	5
2.1.1	การดึงลักษณะ (Feature Extraction).....	5
2.1.2	การวิเคราะห์โครงสร้าง (Structure Analysis).....	6
2.1.3	การสรุปวิดีโอ (Video Abstraction).....	6
2.1.4	การทำดัชนีและการค้นคืน (Indexing & Retrieval).....	8
2.2	ภาพรวม OpenCV library .....	8
2.3	Xilinx Virtex-II Pro.....	9
2.4	การใช้งาน FPGA ควบคู่กับ Image Processing.....	11
2.5	เริ่มต้นการใช้งาน Simulink.....	13
2.6	การออกแบบร่วมกันระหว่างฮาร์ดแวร์ และ ซอฟต์แวร์.....	16
2.7	ขั้นตอนออกแบบ FPGA.....	18

## บทที่ 3 การค้นหาจากวิดีโอโดยอ้างอิงจากผู้ใช้งาน

3.1	การค้นหาจากในวิดีโอโดยอาศัย Superimposed Text เป็นหลักในการค้นหา.....	21
3.2	การพัฒนาโปรแกรมจำลองการค้นหา.....	27

## บทที่ 4 การค้นหา Superimposed Text บนวิดีโอ

4.1	เทคนิคการค้นหา Superimposed text บนวิดีโอแบบอัตโนมัติ.....	37
4.2	การจำลองอัลกอริทึมบน Simulink ของ MATLAB.....	44
4.3	การพัฒนาอัลกอริทึมบนบอร์ดทดลอง Virtex-II Pro.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5 ผลการทดลอง

- 5.1 ผลการทดลองจากการพัฒนาระบบค้นหาภาคโดยอ้างอิงจากผู้ใช้โดยตรง.....76
- 5.2 ผลการทดสอบประสิทธิภาพของอัลกอริทึมการค้นหา Superimposed Text บนวิดีโอ.....76

## บทที่ 6 บทวิจารณ์และสรุป

- 6.1 บทสรุป.....79
- 6.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....80
- 6.3 ปัญหาอุปสรรค และ แนวทางการแก้ไข.....81
- 6.4 แนวทางการพัฒนาต่อ.....82



# สารบัญตาราง

	หน้า
ตารางที่ 2.1 การเปรียบเทียบข้อดี ข้อเสีย แนวทางในการพัฒนา FPGA.....	12
ตารางที่ 4.1 เซตค่าพื้นฐานของโมเดลใน Simulink.....	45
ตารางที่ 5.1 ตารางเปรียบเทียบ PR กับ RE ในขั้นตอนการหาขอบภายในอัลกอริทึม.....	77
ตารางที่ 5.2 ตารางเปรียบเทียบ PR กับ RE ในวิธีโอทีฟาต่างๆ กัน.....	77
ตารางที่ 5.3 ตารางเปรียบเทียบการใช้ Adaptive Threshold ภายในขั้นตอน Sobel Edge Detection.....	78



# สารบัญรูป

รูปที่	หน้า
1.1 แนวทางการพัฒนาระบบค้นหาจากโดยอ้างอิงจากผู้ใช้เป็นหลัก.....	2
2.1 โค้ดอะแกรมโปรเซสการวิเคราะห์และทำดัชนีวิดีโอ .....	5
2.2 ภาพโค้ดอะแกรมของ Virtex-II Pro FPGA.....	10
2.3 ภาพบอร์ด Xilinx Virtex-II Pro FPGA.....	10
2.4 แสดงการทำงานของ Simulink ในแต่ละรอบจังหวะเวลา.....	15
2.5 แสดงการพัฒนาแบบ Reconfigurable computing system .....	17
3.1 แสดง Superimposed Text.....	21
3.2 แสดง Non – fix image region .....	22
3.3 โค้ดอะแกรมของระบบค้นหาจากในไฟล์วิดีโอ .....	23
3.4 การเลือกรูปแบบค้นหาวิธีที่ 1 .....	24
3.5 การเปรียบเทียบหาพื้นที่คงที่ .....	24
3.6 การหาพิกเซลที่เคลื่อนที่ .....	25
3.7 การระบุตำแหน่งเฟรมที่ตรงกับ pattern .....	26
3.8 การตั้ง path ไลบรารีให้สามารถใช้งานร่วมกับ OpenCV .....	27
3.9 การตั้ง path include file ให้สามารถใช้งานร่วมกับ OpenCV .....	28
3.10 การตั้ง path source file ให้สามารถใช้งานร่วมกับ OpenCV .....	29
3.11 การสร้างโปรเจ็คใหม่ .....	29
3.12 ตัวอย่างการเพิ่มไลบรารีในโปรเจ็ค .....	30
3.13 โครงสร้าง IpImage ในไลบรารี OpenCV .....	31
3.14 การแตกเซรคของโปรแกรมค้นหาจากโดยอ้างอิงจากผู้ใช้อ .....	34
3.15 การพัฒนาโปรแกรมค้นหาจาก .....	36
4.1 ภาพจากต่างๆที่มี Superimposed Text ปรากฏ.....	37
4.2 ขั้นตอนการค้นหา Superimposed Text.....	38
4.3 ภาพค่าสูงสุด เบาะภาพค่าต่ำสุด หลังจากผ่าน Multiple Frame Integration .....	39
4.4 ภาพหลังจากการผ่าน Sobel Edge Detection ของ MaxIMAGE & MinIMAGE .....	41
4.5 วิธีการ Text box Classification .....	42
4.6 ผลลัพธ์จากการทำ Text box Classification .....	42

4.7 ผลลัพธ์จากการทำ Area Boundaries .....	43
4.8 ผลลัพธ์จากการทำ Area Boundaries ในวิดีโออื่นๆ .....	43
4.9 กราฟที่เกิดจากการทำ projection .....	44
4.10 text box หลังจากการทำ Refinement .....	44
4.11 Model การจำลองระบบค้นหาพื้นที่ Superimposed Text บน Simulink .....	45
4.12 แสดงการตั้งค่าเริ่มต้นของ Block From Multimedia File .....	46
4.13 แสดงการตั้งค่าเริ่มต้นของ Block Color Space Conversion .....	47
4.14 แสดงการตั้งค่าเริ่มต้นของ Block Counter Limited .....	47
4.15 model sub-layer ข้อของบล็อก Split Min/Max Frame .....	48
4.16 model sub-layer ข้อของบล็อก Frame Memorized (50 frame) .....	49
4.17 เมื่อสามารถลง Driver Programming Cables สำหรับบอร์ดทดลองได้อย่างถูกต้อง .....	52
4.18 Dialog Box แรกเมื่อเปิดโปรแกรม iMPACT .....	53
4.19 หน้า Welcome to iMPACT เพื่อเลือกใช้งานที่ต้องการจากโปรแกรม .....	53
4.20 เมื่อโปรแกรมทำการ initialize chain เสร็จแล้ว .....	54
4.21 ให้เลือกไฟล์ .bit เพื่อนำมาดาวน์โหลดสู่บอร์ดทดลอง .....	54
4.22 เมื่อทำการดาวน์โหลดไฟล์ .bit ลงสู่บอร์ดทดลองได้สำเร็จ .....	55
4.23 หน้าจอต้อนรับของระบบ Build-In Self Test ผ่านโปรแกรม Hyper Terminal .....	57
4.24 หน้าแรกของ Base System Builder Wizard .....	57
4.25 หน้าจอต้อนรับผู้ใช้งานของ Base System Builder .....	58
4.26 เลือกบอร์ดทดลองที่ต้องการจะทำการออกแบบระบบ .....	59
4.27 ทำการเลือก Processor Core ที่จะใช้งาน .....	59
4.28 หน้าจอ Configure PowerPC ในส่วนแรก .....	60
4.29 ตั้งค่าสำหรับการใช้งาน Serial Port .....	61
4.30 ตั้งค่าสำหรับการใช้งาน LED ขนาด 4 บิตและ DipSwitch ขนาด 4 บิต .....	61
4.31 ตั้งค่าในส่วนของ Software Setup .....	62
4.32 ตั้งค่าระบบที่จะจัดเก็บข้อมูล Data, Instruction และ Stack/Heap ไว้ .....	63
4.33 หน้าจอสรุปการตั้งค่าทั้งหมดที่จะเกิดขึ้นในระบบ .....	63
4.34 ตั้งค่าสำหรับ XMD ก่อนเริ่มใช้งาน .....	64
4.35 หน้าต่าง XMD แสดงสถานะ Shell Prompt .....	65
4.36 หน้าต่าง XMD แสดงสถานะการ run ของ program อยู่ .....	66

4.37 รูปการต่ออินเตอร์เฟสระหว่างจอภาพ, บอร์ด, แรม และ การ์ด .....	67
4.38 รูปการต่ออินเตอร์เฟสระหว่างจอภาพ, บอร์ด, แรม และ การ์ด (ต่อ) .....	67
4.39 หน้าจอตัวเลือกระดับของ RAM ที่จะใช้งานในระบบ .....	68
4.40 หน้าจอการเพิ่ม Software Project ใหม่เข้าสู่ระบบ .....	73
4.41 โปรแกรม Image_tutorial หลังจากได้ทำการเพิ่มเข้ามาสู่ระบบแล้ว .....	73
4.42 หน้าจอการสร้างไฟล์ Linker Script .....	74
4.43 บล็อกไดอะแกรมของการแปลงสัญญาณจาก TV ไปออกยัง Monitor .....	74
5.1 ผลการค้นหากจากระบบในวิดีโอเกมกีฬาเบสบอล ประเทศญี่ปุ่น .....	76



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาของปัญหาและความสำคัญของปัญหา

สื่อโทรทัศน์นับว่าเป็นสื่อที่มีผู้ชมมากเป็นอันดับหนึ่ง เมื่อเทียบกับสื่ออื่นๆ สื่อโทรทัศน์มีทั้งข่าวสารบ้านเมือง บันเทิง ละคร สารคดี และอื่นๆ อีกมากมาย จะปฏิเสธเลยไม่ได้ว่า แทบทุกบ้านจะมีโทรทัศน์ ซึ่งเหตุนี้เองการพัฒนาการสื่อโทรทัศน์นั้นจะมีการพัฒนาไปเรื่อยๆ อย่างช้าๆ ไม่ว่าจะเป็นคุณภาพของการส่งสัญญาณเองก็ดี หรือ เป็นเรื่องของคุณภาพของวิดีโอก็ดี รวมถึงคุณภาพของตัวโปรแกรมทีวีที่นำเสนอในแต่ละช่องด้วยเช่นกัน ในประเทศไทยนั้น มีช่องหลัก ๆ นั้นก็คือ ช่อง 3, 5, 7, 9, 11, itv หรือแม้แต่ UBC ก็มีผู้นิยมบริโภคอย่างกว้างขวางเช่นกัน

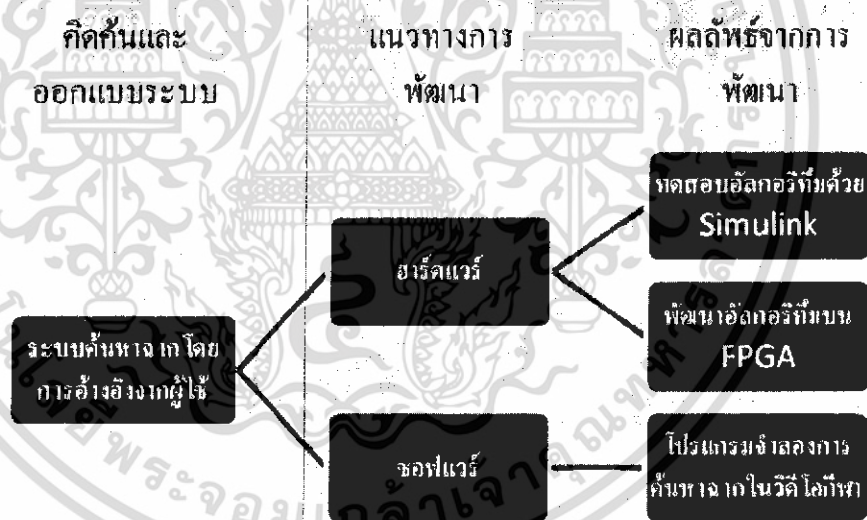
ความต้องการของผู้ชมในอดีตที่ผ่านมานอกจากจะเน้นเรื่องคุณภาพของตัววิดีโอในรายการทีวีเองแล้ว ก็ยังเน้นเรื่องการเข้า (access) ถึงรายการ หรือ จากที่ตนเองสนใจเป็นพิเศษ ซึ่งในเทคโนโลยี ปัจจุบันนั้น ก็ยังคงเป็นการค้นหาจากนั้นด้วย มือ (Manual searching) ทำให้เสียเวลาเป็นอย่างมากในการค้นหา ดังนั้นการทำดัชนีวิดีโอ (video indexing) และ บทสรุปของวิดีโอ (video summarization) จะเป็นทางออกสำหรับปัญหา โดยที่ผ่านมามีหลายงานวิจัยที่ได้ให้ความสำคัญในเรื่องของดัชนีและบทสรุป ของวิดีโอเป็นอย่างมาก [1],[2],[3] ข้อมูลที่ได้จากวิดีโอในรูปแบบตัวหนังสือ และ เสียงนั้น ได้มีการนำมาใช้ในการทำเพื่อตอบสนองจุดประสงค์นี้เป็นอย่างมาก ทั้งหมดนี้โครงการนี้จะเน้นในเรื่องของข้อความในวิดีโอที่ได้มาจากห้องส่ง หรือ ที่เรียกกันว่า (superimposed text) โดยเหตุผลหลักที่เลือก superimposed text เข้ามาวิเคราะห์ คือ

1. ข้อความที่ปรากฏบนวิดีโอ มีความสัมพันธ์กับ เนื้อหาวิดีโอ โดยตรง
2. ข้อความที่ปรากฏขึ้นนั้น จะมีลักษณะเด่นพิเศษ
3. ในการวิจัยเรื่องของ OCR นั้นมีประสิทธิภาพมากกว่า การจดจำเสียง

จากเหตุผลที่กล่าวมานั้น ทำให้งานวิจัยส่วนใหญ่ในเรื่องของ ดัชนีและบทสรุปของวิดีโอ นั้น เริ่มจาก การทำจดจำตัวหนังสือ (Text recognition) แต่ในมุมมองของผู้วิจัยเองนั้น พบว่า การจดจำตัวหนังสือ จะทำให้งานที่ออกมาไม่สามารถใช้งานจากวิดีโอ ที่มีแหล่งที่มาแตกต่างกันได้

เช่น แต่ละประเทศต่างก็มีตัวหนังสือที่แตกต่างกัน มีภาษาที่แตกต่างกัน ทำให้เกิดข้อจำกัดของระบบ

ดังนั้น ผู้พัฒนาได้ ออกแบบระบบที่สามารถทำการค้นหาจากโดยอาศัย ความต้องการของผู้ใช้เป็นหลัก โดยให้ผู้ใช้เป็นคนออกแบบรูปแบบการค้นหาได้เอง โดยระบบจะอนุญาตให้ผู้ใช้ได้ออกแบบรูปแบบการค้นหา (pattern) แล้วระบบจะทำการ ค้นหา ตามรูปแบบนั้นๆ จะสามารถจะใช้กับวิดีโอได้ทุกแบบ ทุกภาษา และใกล้กับความต้องการของผู้ใช้มากที่สุด โดยประสิทธิภาพนั้น ก็ขึ้นอยู่กับรูปแบบที่ผู้ใช้ได้ออกแบบ ซึ่งจะสามารถลบข้อจำกัดของระบบที่อาศัยวิธีการจดจำตัวหนังสือมาใช้ในการวิเคราะห์ อีกทั้งโครงการนี้ได้มีการจัดทำในรูปแบบฮาร์ดแวร์ควบคู่กับซอฟต์แวร์ โดยใช้ FPGA ในการประมวลผล โดยมีการเชื่อมต่อคอมพิวเตอร์ ตัวฮาร์ดแวร์เอง และโทรทัศน์เข้าด้วยกัน โดยแนวทางการพัฒนานั้น ได้แสดงดังรูป 1.1 ด้านล่าง



รูปที่ 1.1 แนวทางการพัฒนาระบบการค้นหาจากโดยอ้างอิงจากผู้ใช้เป็นหลัก

ในการพัฒนานั้นจะเริ่มจากการคิดค้น และ ออกแบบระบบค้นหาจากโดยการอ้างอิงจากผู้ใช้งาน ซึ่งทางที่ผู้พัฒนานั้นจะแบ่งออกเป็น 2 สองทางในการพัฒนา คือ ซอฟต์แวร์ เป็นการพัฒนา Application ตัวอย่างในการจำลองการค้นหาจาก และอีกเส้นทางคือ พัฒนาระบบฮาร์ดแวร์ โดยมีการจำลองอัลกอริทึมบน Simulink หลังจากนั้นก็พัฒนาอัลกอริทึมบน FPGA ส่วนในเรื่อง

ของการเชื่อมต่อระหว่างซอฟต์แวร์ และ ฮาร์ดแวร์นั้นเป็นส่วนในเรื่องของงานในอนาคตซึ่งอยู่นอกขอบเขตโครงการในครั้งนี้

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 ออกแบบและพัฒนาโปรแกรมค้นหาจากบนวิดีโอ โดยมีผู้ใช้เป็นคนกำหนดรูปแบบ
- 1.2.2 ศึกษาหลักการด้านการประมวลผลภาพและวิดีโอ (Digital video & image processing)
- 1.2.3 คิดค้น Algorithms ในการค้นหาจากที่ต้องการของผู้ใช้
- 1.2.4 เพื่อศึกษาการใช้งาน VC++.net กับ OpenCV library และ MATLAB
- 1.2.5 ศึกษา Simulink ใน MATLAB ในการพัฒนาบน FPGA
- 1.2.6 ศึกษา Xilinx Embedded Development Kit (Virtex-II Pro) ร่วมกับ EDK
- 1.2.7 ศึกษา System Generator ใน MATLAB
- 1.2.8 ศึกษาการใช้งาน FPGA ด้านการประมวลผลวิดีโอและภาพ และนำไปประยุกต์ใช้งาน

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 สามารถพัฒนาโปรแกรมจำลองการค้นหาจากบนวิดีโอก็ได้
- 1.3.2 เข้าใจพื้นฐานการทำงาน FPGA ด้านการประมวลผลวิดีโอ
- 1.3.3 เข้าใจการทำงาน OpenCV library ซึ่งเป็น open source ร่วมกับ VC++.net
- 1.3.4 เข้าใจการทำงาน EDK ร่วมกับ XUP Virtex-II Pro Development Board
- 1.3.5 สามารถพัฒนาด้วย Simulink, System Generator บน MATLAB ได้
- 1.3.6 ส่งเสริมความเข้าใจในการที่จะนำ FPGA ไปใช้ในงานด้านการประมวลผลวิดีโอ และภาพเพื่อให้ผู้ที่สนใจศึกษา และนำไปประยุกต์ใช้งานได้

## 1.4 ระเบียบวิธีวิจัย

- 1.4.1 ศึกษาทฤษฎีพื้นฐานทางด้านดิจิทัลที่เกี่ยวกับการคำนวณเพื่อที่จะใช้ในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ระบบการทำงานของ FPGA

- 1.4.2 ออกแบบวิธีการทำงานของระบบ จากทฤษฎี และหลักการที่ได้ศึกษามาทั้งหมด
- 1.4.3 เตรียมฐานข้อมูลวิดีโอทั้งในประเทศ และต่างประเทศ เพื่อใช้เป็นข้อมูลพื้นฐานสำหรับการวิจัย เพื่อใช้ในการตรวจสอบประสิทธิภาพของระบบที่ได้ทำการวิจัย
- 1.4.4 ทดสอบการทำงานของระบบ พร้อมการวิเคราะห์ปรับปรุงแก้ไขระบบ
- 1.4.5 ทำการพัฒนาซอฟต์แวร์จากหลักการที่ได้ออกแบบมาจากข้อ 1.4.2
- 1.4.6 ทำการพัฒนาฮาร์ดแวร์ตามที่ได้ออกแบบที่ผ่านการทดสอบเรียบร้อยแล้ว
- 1.4.7 ทำการทดลองโดยใช้ฮาร์ดแวร์ที่ได้จากงานวิจัยนี้กับฐานข้อมูลที่ได้จัดเตรียมไว้แล้ว บันทึกเก็บข้อมูลและผลลัพธ์ของทุกๆ ขั้นตอน เพื่อนำมาวิเคราะห์ปรับปรุงงานวิจัยต่อไป
- 1.4.8 นำผลลัพธ์จากการทดลองมาวิเคราะห์และประเมินงานวิจัย ทั้งในแง่ของความถูกต้อง และเวลาในการประมวลผล แล้วสรุปผลเพื่อนำเสนอผลงานวิจัย
- 1.4.9 จัดทำวิทยานิพนธ์และเตรียมตัวนำเสนอผลงาน

### 1.5 ขอบเขตของโครงการ

ในโครงการนี้ได้นำเสนอออกใน 2 รูปแบบแนวทาง คือ พัฒนาด้านซอฟต์แวร์ด้วย VC++.net และ OpenCV Library ซึ่งเป็นการจำลองการการค้นหาจากโดยอ้างอิงจากผู้ใช้งาน โดยผู้พัฒนาจะมีการจำลอง Pattern ของกีฬาชนิดต่างๆ คือ เบสบอล, ฟุตบอล และ แบดมินตัน แล้วทดลองการค้นหาตามเหตุการณ์ดังกล่าว ส่วนการทดลองในเรื่องกรณีของอัลกอริทึม จะใช้ชุดวิดีโอการทดลองคือ ฟุตบอล, บาสเกตบอล, เบสบอล, สุนัขเกอร์, เทนนิส และ แบดมินตัน เพื่อวัดประสิทธิภาพของการตรวจหา superimposed text บนวิดีโอ โดยวัดจาก Precision rate และ Recall rate ในกรณีแนวทางที่สองคือ พัฒนาด้านฮาร์ดแวร์โดยเริ่มจากการจำลองบน Simulink ของ MATLAB และมีการพัฒนาลงบนบอร์ดทดลอง Viretex-II Pro ของ Xilinx โดยมีการศึกษาเครื่องมือที่ช่วยในการพัฒนา 2 ชนิด คือ System Generator ของ MATLAB และ EDK 8.1 ของ Xilinx โดยในส่วนของการพัฒนาด้านฮาร์ดแวร์นั้นจะเป็นการนำอัลกอริทึม Superimposed Text Detection ลงบนบอร์ดทดลอง โดยชุดการทดลองนั้นจะเป็น ชุดข้อมูลภาพต่อเนื่องของ บาสเกตบอล, เบสบอล

## 1.6 ส่วนประกอบของรายงาน

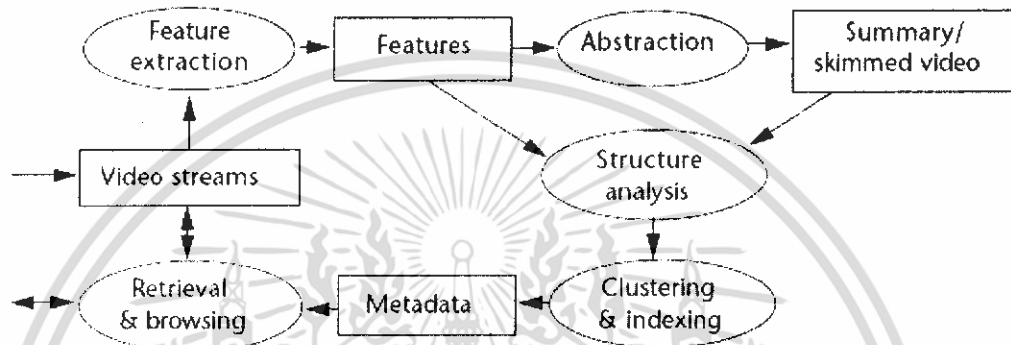
โดยรายงานจะเริ่มการอธิบายจากซอฟต์แวร์ไปยังฮาร์ดแวร์ โดยในซอฟต์แวร์นั้นจะอธิบายถึงการจำลองค้นหาจากโดยอ้างอิงจากผู้ใช้ ซึ่งอยู่ในบทที่ 3 โดยภายในนั้นจะอธิบายถึงรูปแบบเทคนิค และการทดลอง และ พัฒนาโปรแกรมขึ้นมา ต่อมาในบทที่ 4 จะอธิบายในเรื่อง Automatic Superimposed Text Detection ซึ่งจะอธิบายถึง Algorithm และการทดสอบโดยใช้ Simulink , และการพัฒนาบนบอร์ด FPGA และในบทที่ 5 จะอธิบายถึงผลจากการทดลองของบทที่ 3 และ บทที่ 4 และบทสุดท้ายเป็นการอธิบายข้อสรุปและวิจารณ์โครงการนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 พื้นฐานการค้นคืนวิดีโอ



รูปที่ 2.1 ไคอะแกรมโปรเซสการวิเคราะห์และทำดัชนีวิดีโอ

จาก รูปที่ 2.1 นั้น แสดงถึงกระบวนการการวิเคราะห์, ทำดัชนี และการค้นคืนวิดีโอ [2] ซึ่งได้แบ่งออกเป็น การดึงลักษณะ (Feature extraction), การวิเคราะห์โครงสร้าง (Structure analysis), การสรุปวิดีโอ (Video abstraction) และการทำดัชนีและการค้นคืน (Indexing and Retrieval) โดยจะอธิบายตามหัวข้อดังต่อไปนี้

#### 2.1.1 การดึงลักษณะ (Feature extraction)

ในโปรเซสนี้ ถือว่าเป็นโปรเซสที่สำคัญที่สุดในเรื่องของการค้นคืนวิดีโอ ประสิทธิภาพของการดัชนีนั้นจะขึ้นอยู่กับตัวแทนหลักที่บ่งบอกถึงเนื้อหา นั้น ๆ ซึ่งในขั้นตอนนี้ว่าเป็นโปรเซสที่ยากที่สุด โดยจะพยายามดึงลักษณะของวิดีโอออกมา ไม่ว่าจะเป็น สี, รายละเอียด, รูปร่าง, โครงสร้าง, รูปแบบ หรือ การเคลื่อนที่ของกล้อง ซึ่งนำมาสร้างเป็น แนวความคิดเชิงความหมาย (Semantic concept) ความคิดเชิงความหมายนี้เอง อาจมีการนำเรื่องของเสียงมาวิเคราะห์ร่วมด้วย เพื่อสร้างความเชื่อถือให้กับความคิดเชิงความหมาย ได้มากยิ่งขึ้น

ในการพิจารณาข้อมูลวิดีโอซึ่งประกอบด้วย วัตถุ (object) เหตุการณ์ (event) และฉาก (scene) ความสัมพันธ์ระหว่างองค์ประกอบเหล่านี้ เรียกว่า แนวความคิดเชิงความหมาย (semantic concept) ซึ่งภายในประกอบด้วยลักษณะระดับสูง (high level feature) เช่น เหตุการณ์และวัตถุ และ ลักษณะระดับล่าง (low level feature) เช่น สี, รูปร่าง, พื้นผิว, ขอบ หรือ เวกเตอร์การเคลื่อนที่ ซึ่งช่องว่างระหว่างทั้งสองระดับนี้ เรียกว่า ช่องว่างทางความหมาย (semantic gap) ซึ่งการจำแนกที่ดีนั้นควรที่จะมีช่องว่างนี้น้อยที่สุด เพราะลักษณะระดับล่างนั้นสามารถหาได้โดยใช้เทคนิคการประมวลผลภาพ แต่ในขณะที่ลักษณะระดับสูงเป็นความหมายตามความเข้าใจมนุษย์

โดยเหตุผลดังกล่าว ได้มีผู้นำเสนอวิจัยเกี่ยวกับเวกเตอร์ลักษณะเชิงความหมาย (Semantic feature) โดยนำเวกเตอร์ลักษณะระดับล่างมาสร้างแบบจำลองตรวจหาลักษณะเชิงความหมาย หรือ สร้างตัวจำแนก เพื่อนำมาอธิบายวัตถุ เหตุการณ์ หรือฉากที่เกิดขึ้นในวิดีโอเพื่อใช้ในการทำดัชนีสำหรับการค้นคืนวิดีโอ และการให้คำประกอบวิดีโอ (Video Annotation)

### 2.1.2 การวิเคราะห์โครงสร้าง (Structure analysis)

ขั้นตอนต่อมาคือการวิเคราะห์โครงสร้างของวิดีโอ หลังจากการดึงลักษณะออกมาแล้ว ซึ่งโปรเซสนี้จะเป็นการแยกส่วนวิดีโอทางเวลา (Temporal video extraction) ซึ่งจะเป็นการบ่งบอกถึงขอบเชิงเวลา และ บอกถึงความหมายในแต่ละช่วงที่ได้แบ่งแยกไว้ โดยจะมีการแบ่งไว้ในลักษณะช็อต (shot), เฟรมสำคัญ (keyframe) และ ฉากวิดีโอ (scene) ซึ่งจะถูกแบ่งและจัดประเภทคล้ายกับการสร้าง storyboard ในการแบ่งกลุ่มนั้น (ตามภาพที่ 1) จะเริ่มจากกลุ่มของฉาก และ แบ่งย่อยลงไปถึงช็อต และ ในแต่ละช็อตจะหาเฟรมตัวแทน หรือที่เรียกว่า เฟรมสำคัญ เพื่อจะนำไปสร้างดัชนีในการค้นหาวิดีโอต่อไป

การหารอยแบ่งแต่ละช็อตนั้นมีวิธีการมากมายหลายอย่าง โดยวิธีที่ธรรมดาที่สุดคือ การเปรียบเทียบลักษณะของสีในเฟรมที่ติดกัน, รายละเอียด, ขอบของวัตถุในเฟรม และ ความแตกต่างระหว่างของเฟรมที่มีแนวโน้มว่าจะเป็นรอยต่อของช็อต โดยรอยต่อของช็อตนี้เอง เรียกว่า “hard cut” ตามลักษณะทางกายภาพของรอยต่อนี้เอง ได้แบ่งประเภทออกเป็นการเปลี่ยนแปลงของช็อตแบบทันทีทันใด และ การเปลี่ยนแปลงของช็อตแบบค่อยเป็นค่อยไป เช่น fade หรือ dissolve ในประเภทหลังนับว่าการหา hard cut นั้นจะยากกว่าวิธีแรก แต่ในหลักการนั้นยังมีวิธีที่คล้ายกันแทนที่จะวิเคราะห์จากเฟรมที่อยู่ติดกัน เป็น วิเคราะห์เฟรมที่ไม่ได้อยู่ติดกันนั่นเอง

การแบ่งแต่ละฉากนั้นออกเป็นช็อตวิดีโอ นั้นับว่ามีความสำคัญในการที่จัดกลุ่มแบ่งประเภทของกลุ่มของฉากและช็อตวิดีโอ โดยการนำช็อตวิดีโอที่มีความคล้ายคลึงกันมารวมกัน เพื่อที่จะนำมาวิเคราะห์ต่อในระดับที่สูงขึ้น ซึ่งในการวิเคราะห์ระดับสูงนี้ได้แบ่งออกเป็น 2 ประเภทซึ่งทั้งสองนั้นเป็นวิธีแบบอัตโนมัติโดยอาศัยหลักการจดจำ (Recognition) ได้แก่ อาศัยกฎการผลิตวิดีโอ และ อาศัยลำดับความสำคัญจากผู้ใช้งาน ในวิธีทั้งสองนี้ยังคงมีข้อจำกัดเนื่องจากทั้งสองหลักการนี้นั้น ยังคงแสดงความหมายในระดับเฉพาะบุคคล (subjective semantic) ซึ่งอาจได้มาจากผู้พัฒนาระบบ และนับว่าการวิเคราะห์เหล่านั้นไม่เป็นสากล แต่นับว่าเป็นเรื่องที่โชคดีอย่างยิ่ง ช็อตวิดีโอที่นับว่าเป็นตัวเลือกที่ดีตัวหนึ่งในการค้นคืนสำหรับการค้นหา ซึ่งอาศัยหลักการเปรียบเทียบกับเฟรมสำคัญกับคำค้นหาจากผู้ใช้

### 2.1.3 การสรุปวิดีโอ (Video abstraction)

การสรุปวิดีโอ เป็นโปรเซสในการสร้างตัวแทนของวิดีโอและต้องมีความยาวที่น้อยกว่าวิดีโอต้นฉบับ ซึ่งโปรเซสนี้คล้ายกับการดึงเอาคำสำคัญ (key word) หรือ บทสรุปออกมา ดังนั้น การดึงเอาเฟรมสำคัญ “Key frame” หรือ high light ที่เข้าถึงในแต่ละช็อต, ฉาก หรือ เรื่องราว จากวิดีโอต้นฉบับออกมาเป็นบทสรุปของช็อตนั้นๆ โปรเซสนี้ นับว่ามีความสำคัญมากที่สุดในการสร้างการสรุปข้อมูลในวิดีโอ ไม่เพียงจะได้ตัวแทนของวิดีโอเท่านั้น แต่ยังได้สารบัญชการค้นหาของวิดีโอออกมาด้วยเช่นกัน ซึ่งสิ่งเหล่านี้เองบางครั้งสามารถสร้างตารางตัวเลือกในรูปแบบภาพแทนเนื้อหาเพื่อสำหรับการค้นหาได้ (browsing)

เฟรมสำคัญเป็นเซตของเฟรมในวิดีโอที่เลือกมาเป็นตัวแทนของชื่อวิดีโอที่มีความสำคัญมากที่สุดภายในชื่อนั้นๆ ซึ่งเฟรมสำคัญนี้เป็นภาพนิ่งที่ถูกดึงออกมาจากวิดีโอต้นฉบับ ประสิทธิภาพของเฟรมสำคัญนี้ขึ้นอยู่กับวิธีที่เลือกเฟรมสำคัญนี้ออกมาว่าสามารถบ่งบอกถึงเนื้อหาภายในแต่ละช็อตได้มากน้อยเพียงไร ดังนั้นการค้นหาเฟรมสำคัญได้มีเทคนิคออกมาต่างๆ มากมาย ในที่นี้จะยกตัวอย่างมา 4 เทคนิคดังนี้

#### 1. แนวความคิดแบบรอยต่อของช็อต (shot boundary based approach)

หลังจากวิดีโอผ่านขั้นตอนของการแยกวิดีโอทางเวลาแล้วจะได้หน่วยย่อยที่เรียกว่าช็อต วิธีการอย่างง่ายในการดึงเฟรมสำคัญออกมานั้น คือ นำเฟรมแรกของแต่ละช็อตวิดีโอมาเป็นเฟรมสำคัญ แต่วิธีการนี้เป็นวิธีที่ง่ายแต่เฟรมสำคัญในแต่ละช็อตจะถูกจำกัดเพียงแค่เฟรมเดียวเท่านั้น

#### 2. แนวความคิดแบบเนื้อหาด้วยการมองเห็น (Visual Content Based approach)

Zhang และคณะ [3] ได้นำเสนอหลายกฎเกณฑ์การมองเห็นเพื่อที่จะดึงข้อมูลเฟรมสำคัญ คือ แทนที่จะใช้เพียงเฟรมเดียวของแต่ละช็อตวิดีโอ ก็เลือกหลายเฟรมสำคัญจากช็อตวิดีโอ เรียกว่า เซตของเฟรม (Set of Key frame) โดยมีกฎดังนี้

- กฎเกณฑ์พื้นฐานของช็อต (Shot based criteria) คือการดึงเฟรมสำคัญในระดับช็อต โดยในแต่ละเฟรมแรกถือว่ามีค่าสำคัญที่สุด ส่วนเฟรมสำคัญที่เหลือขึ้นอยู่กับวิเคราะห์ในขั้นตอนต่อมา
- กฎเกณฑ์พื้นฐานของลักษณะสี (Color feature based criteria) คือ หลังจากได้เฟรมแรกของช็อตวิดีโอที่เป็นเฟรมสำคัญแล้ว จะเปรียบเทียบความเหมือนกันระหว่างเฟรมแรกกับเฟรมที่เหลือ โดยอาศัยหลักวิธีฮิสโตแกรมสี หรือ โมเมนต์สี (color moment)
- กฎเกณฑ์พื้นฐานของการเคลื่อนที่ เป็นการปรับปรุงโดยใช้ข้อมูลของการเคลื่อนที่รวม (global motion) ที่ได้จากการดำเนินการของกล้อง (camera operation) และการเคลื่อนที่ของวัตถุที่ขนาดใหญ่ในวิดีโอ เนื่องจากการดำเนินการของกล้องนั้น ใช้เพื่อหาเหตุการณ์ หรือ ต้องการเน้นวัตถุอย่างชัดเจน ซึ่งจะตรวจสอบลักษณะการเคลื่อนที่คือ การถ่าย (panning) และการซูม (zooming) ดังนั้นจำนวนของเฟรมสำคัญจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงในการมองเห็นเนื้อหาภาพ นั่นคือเฟรมสำคัญที่ได้จากการถ่ายจะต้องมีการซ้อนทับกันน้อยที่สุดโดยทั่วไปแล้วเฟรมสำคัญควรมีเนื้อหาในภาพน้อยกว่า 30%

#### 3. แนวความคิดแบบการวิเคราะห์การเคลื่อนที่ (Motion analysis based approach)

Wolf และคณะ [4] ได้นำเสนอการดึงเฟรมสำคัญโดยพิจารณาการเคลื่อนที่ กล่าวคือใช้การวิเคราะห์การไหลเชิงแสง (Optical Flow Analysis) เพื่อเป็นวัดค่าการเคลื่อนที่ที่เกิดขึ้นในช็อตวิดีโอ และเลือกเฟรมสำคัญจากช่วงที่มีค่าการเคลื่อนที่น้อยที่สุดซึ่งช่วงนี้ เรียกว่า ค่าที่น้อยที่สุดของช่วง (Local minima)

#### 4. แนวความคิดแบบการจัดกลุ่ม (Clustering based approach)

Zhuang และคณะ ได้เสนอวิธีการจัดกลุ่ม ที่ใช้ในงานการรู้จำรูปแบบ เพื่อที่จัดกลุ่มเฟรมในซีดีวิดีโอเพื่อหาเฟรมสำคัญ หลังจากการจัดกลุ่มให้แต่ละเฟรมแล้วนั้นต่อไปก็คือการเลือกหาเฟรมสำคัญ โดยที่กลุ่มที่มีขนาดใหญ่ที่สุดเพียงพอก็จะเป็นกลุ่มสำคัญ (key cluster) มากกว่าค่าขนาดเฉลี่ย จะมีเฟรมสำคัญเพียงเฟรมเดียวที่ได้จากกลุ่มที่ใหญ่ที่สุด ถ้ามีขนาดน้อยกว่าค่าเฉลี่ย เฟรมสำคัญหาได้จากเฟรมที่มีค่าใกล้เคียงกับค่าศูนย์ถ่วงของแต่ละกลุ่ม

##### 2.1.4 การทำดัชนีและการค้นคืน (Indexing & Retrieval)

หลังจากผ่านกระบวนการดึงลักษณะ, การวิเคราะห์วีดิโอ และ การสรุปวีดิโอ ข้อมูลที่ได้มานั้นจะถูกจัดเก็บไว้ในฐานข้อมูลเฉพาะด้าน (metadata) คือ ได้ทั้งข้อมูลสรุปจากการวิเคราะห์วีดิโอ และ ตัววีดิโอเอง จากข้อมูลที่ได้มานั้นสามารถนำมาสร้างเป็นดัชนีและตารางของเนื้อหาไว้สำหรับการค้นหาและการค้นคืน ซึ่งในการสร้างดัชนีนี้เอง ต้องมีการผ่านกระบวนการตัดและแยกแยะลำดับหรือจัดเป็นกลุ่ม หรือ โครงสร้างดัชนี ในระบบฐานข้อมูลนั้นด้วย เพื่อต้องการเครื่องมือสำหรับการดึง, ค้นหา ในวีดิโอที่มีขนาดใหญ่ ๆ ได้

## 2.2 ภาพรวม OpenCV Library

ไลบรารีนี้เน้นไปที่การทำงานกับการแสดงผลแบบเรียลไทม์(real time) ตัวอย่างของงานได้แก่ การตอบสนองระหว่างคนกับคอมพิวเตอร์ (Human-Computer Interaction) การวิเคราะห์วัตถุ การแบ่งและวิเคราะห์ การวิเคราะห์หน้า การวิเคราะห์การเคลื่อนไหว การติดตามความเคลื่อนไหว อัตราการเคลื่อนไหว การทำความเข้าใจเกี่ยวกับการเคลื่อนไหว การวิเคราะห์โครงสร้างจากความเคลื่อนไหว (SFM) และหุ่นยนต์ไร้สาย

### 2.2.1 Intel ® Image Processing Library

- 2.2.1.1 การสร้างภาพและการเข้าถึงเพื่อแก้ไข(Header เดียวใช้สำหรับ 2 ไลบรารี)
- 2.2.1.2 การทำงานด้านคณิตศาสตร์และตรรกศาสตร์กับภาพ
- 2.2.1.3 Images filtering
- 2.2.1.4 การแปลงภาพแบบ Linear
- 2.2.1.5 การทำงานเกี่ยวกับโครงสร้างของภาพ
- 2.2.1.6 การแปลง Color Space
- 2.2.1.7 Image histogram และ Thresholding

2.2.1.8 การเปลี่ยนแปลงในแบบเรขาคณิต ซึ่งได้แก่การซูม หมุน เบ้ สะท้อนภาพ  
บิด เปลี่ยนมุมมอง เลียนแบบ

2.2.1.9 การทำงานกับภาพ ณ ช่วงเวลาหนึ่ง

## 2.2.2 ตัวอย่าง Application

2.2.2.1 การสอนเกี่ยวกับการใช้กลุ่มเครื่องมือการแบ่งแยกกล้องของ Matlab

2.2.2.2 Automatic camera calibration filter

2.2.2.3 การติดตามสี การติดตามใบหน้า

2.2.2.4 Condensation filter fracker

2.2.2.5 การวิเคราะห์หน้าโดยใช้ embedded HMMs

2.2.2.6 Kalman filter tracker

2.2.2.7 Lucas-Kanade optical flow in an image pyramid

## 2.2.3 โปรแกรมช่วยเหลือที่มีคนช่วยสร้างขึ้น(Windows)

2.2.3.1 ตัวอย่างการทำโปรแกรมติดตั้งกล้องของ Direct Show ด้วย CAMSHIFT

2.2.3.2 Matrox Meteor Direct Show capture filter

## 2.2.4 โปรแกรมช่วยเหลือที่มีคนช่วยสร้างขึ้น (Linux)

2.2.4.1 BMP to IPL file reader/writer

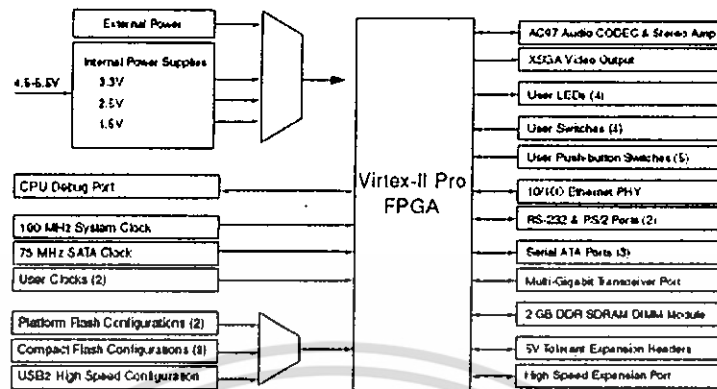
2.2.4.2 การหาค่าเฉลี่ยและค่าสูงสุดต่ำสุดของชุดข้อมูลที่อยู่บนดิส

การศึกษาการใช้งานไลบรารี OpenCV นั้นศึกษาหลักเบื้องต้น ได้จาก [5] จะอธิบายหลักการ  
ทำงานอย่างคร่าว พร้อมทั้งตัวอย่าง code

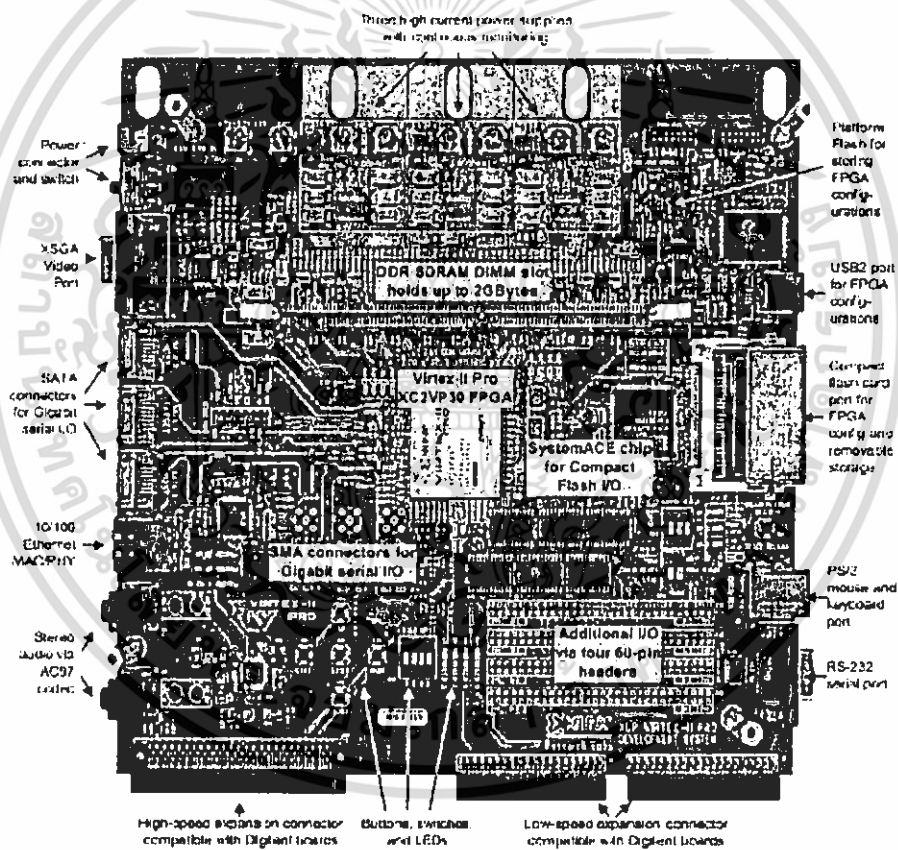
## 2.3 Xilinx Virtex-II pro FPGA

บอร์ดที่ใช้ในการพัฒนาทางด้าน Hardware คือ บอร์ดรุ่น Virtex-II pro FPGA ซึ่งนับว่าเป็น FPGA  
ที่มีประสิทธิภาพสูงในการประมวลผล อีกทั้งยังมี Interface ที่ครบสมบูรณ์แบบ ส่งผลให้สามารถ  
ทำงานได้อย่างซับซ้อน โดยเฉพาะด้าน Image & Video processing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ภาพไดอะแกรมของ Virtex-II pro FPGA



รูปที่ 2.3 ภาพบอร์ด Xilinx Virtex-II Pro FPGA

ส่วนสัญญาณวิดีโออนัน ในโครงการนี้จะรับเข้ามาจากสัญญาณทีวีธรรมดาแบบ SDTV โดยสามารถเสียบได้ทั้งสาย Component Video, S-Video และสาย Composite Video เข้ากับตัว VDEC1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Video Decoder Board) ซึ่งตัว VDEC1 นี้เป็นตัวเสริมแยกต่างหากโดยใช้ชิป ADV7183B เป็นตัวถอดรหัสสัญญาณวิดีโอที่เข้ามาจึงสามารถรองรับสัญญาณได้ทั้ง NTSC, PAL และ SECAM (ในที่นี้เราจะใช้ PAL เป็นหลัก) ออกมาเป็น ITU-R BT.656 Digital Format จากนั้นจึงติดต่อเข้ากับตัว system board ผ่านพอร์ตการใช้งานแบบ Hirose FX2 Data Connector โดยมีค่า Output ออกไปยัง system board ด้วยระบบสีแบบ YCrCb 4:2:2

เมื่อต่อ VDEC1 เข้าสู่ตัว system board แล้ว จำเป็นต้องทำการโปรแกรมชิป ADV7183B เพื่อรองรับการทำงานเสียก่อนจึงจะสามารถนำสัญญาณเข้าไปยัง system board ได้ ไม่เช่นนั้นจะทำให้เกิดการชนกันของพอร์ตข้อมูลที่เข้าสู่ชิป FPGA ในตัว system board ได้

## 2.4 การใช้งาน FPGA ควบคู่กับ Image Processing

จากการศึกษาข้อดี ข้อเสียของหลักการทำ Image Processing ของไฟล์ประเภทวิดีโอแล้ว ไฟล์วิดีโอธรรมดา จะมี frame rate อยู่ที่ 25 fps (ในกรณีการเข้าสัญญาณทีวีแบบ PAL) ซึ่งหมายความว่า ในหนึ่งวินาที จะมีภาพที่ระบบต้องนำมาใช้คำนวณทั้งหมด 25 ภาพ (ซึ่งแล้วแต่อัลกอริทึมที่นำมาใช้) และในกระบวนการทำ การเปรียบเทียบภาพแต่ละภาพ มักจะสิ้นเปลือง process และเวลาในการคำนวณ หลักการต่างๆ ยิ่งถ้าต้องการเพิ่มความแม่นยำของการเปรียบเทียบให้ได้ข้อมูลที่ถูกต้องชัดเจนยิ่งขึ้น จะต้องนำไปผ่านกระบวนการต่างๆ เพิ่มขึ้นอีกในแต่ละเฟรม ซึ่งจะทำให้เกิดภาระอย่างหนักขึ้นกับตัวเครื่องคอมพิวเตอร์ที่ใช้ทำงาน ดังนั้นทางผู้ทำโครงการจึงได้ทำการศึกษาระบบประมวลผลภายนอก แยกเฉพาะออกไป ทางนักศึกษาได้ตัดสินใจเลือกใช้ FPGA เพราะเป็นที่นิยมของตลาด และมีข้อมูลทางการวิจัยและผลิตภัณฑ์ที่ออกมาสู่ตลาดออกมามากมาย โดยทางบริษัท Xilinx ซึ่งเป็นบริษัทที่ผลิตได้แสดงผลลัพธ์ของการทำงานประมวลผลทาง Image Processing ไว้ว่าได้ผลที่เร็วกว่าถึง 6 เท่า

ในการออกแบบบน FPGA มีวิธีต่างๆ ที่น่าสนใจหลายวิธี โดยวิธีต่างๆ มีข้อดีข้อเสีย ที่ได้เรียงไว้ในตารางข้างล่างนี้ ทางนักศึกษาได้เลือกใช้วิธี System Generator หลังจากนั้นค่อยไปปรับแต่ง Code ภาษา VHDL เอาในภายหลัง

ตารางที่ 2.1 การเปรียบเทียบข้อดี ข้อเสีย แนวทางในการพัฒนา FPGA

วิธีการออกแบบ	ข้อดี	ข้อเสีย
Full VHDL/Verilog (RTL Code)	<ol style="list-style-type: none"> <li>1) มีความคล่องตัวสูง</li> <li>2) สามารถควบคุมการออกแบบงานได้อย่างสมบูรณ์</li> <li>3) สามารถแก้ไขและทำความเข้าใจใน Code ได้ง่ายกว่า</li> </ol>	<ol style="list-style-type: none"> <li>1) อาจต้องใช้เวลาในการออกแบบนาน</li> <li>2) ไม่สามารถควบคุมการทำงานของเครื่องมือสังเคราะห์วงจรได้อย่างตลอดเวลา</li> <li>3) ต้องการความเข้าใจในภาษาที่ใช้ออกแบบอยู่พอสมควร</li> <li>4) ต้องทำงานร่วมกับเครื่องมือสังเคราะห์วงจรเพื่อให้ได้มาซึ่งการออกแบบวงจรที่สมบูรณ์</li> </ol>
Full VHDL/Verilog (Instantiating Primitives)	<ol style="list-style-type: none"> <li>1) เข้าถึงโครงสร้างแกนหลักของระบบได้ลึกที่สุด</li> <li>2) สามารถเจาะลึกรายละเอียดได้มากที่สุด</li> <li>3) มีประสิทธิภาพสูงที่สุด</li> </ol>	<ol style="list-style-type: none"> <li>1) ไม่คล่องตัวเท่ากับแบบ RTL</li> <li>2) ต้องการความเข้าใจในระบบสถาปัตยกรรมของ FPGA เป็นอย่างดี</li> <li>3) ใช้เวลาในการออกแบบนาน</li> </ol>
CORE Generator	<ol style="list-style-type: none"> <li>1) สามารถเข้าถึงและใช้งานฟังก์ชันที่มีอยู่แล้วอย่างรวดเร็ว</li> <li>2) ไม่จำเป็นต้องสร้างรูปแบบวงจรขึ้นมาใหม่ถ้าตรงกับความต้องการแล้ว</li> <li>3) IP ถูกออกแบบอย่างสมบูรณ์กับบางวงจรแล้ว</li> </ol>	<ol style="list-style-type: none"> <li>1) IP อาจจะไม่ทำงานตรงตามอย่างที่ต้องการเสมอไป</li> <li>2) ต้องอาศัยความเข้าใจของสัญญาณและพารามิเตอร์ที่จะนำเข้ามาสู่ระบบการออกแบบด้วย</li> <li>3) ทำงานกับระบบปิดที่ทราบข้อมูลเพียงเล็กน้อย</li> </ol>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ) การเปรียบเทียบข้อดี ข้อเสีย แนวทางในการพัฒนา FPGA

วิธีการออกแบบ	ข้อดี	ข้อเสีย
System Generator	<ol style="list-style-type: none"> <li>1) สามารถสร้างผลิตภัณฑ์ได้มากจากระบบการออกแบบขั้นสูง</li> <li>2) มีความสามารถในการทดสอบการออกแบบในระดับขั้นวงจร</li> <li>3) เหมาะสำหรับผู้เริ่มใช้งาน FPGA รายใหม่ๆ</li> <li>4) มีความสามารถในการออกแบบ testbench ขั้นสูงที่พร้อมเพียง</li> <li>5) HDL Testbench, test vector และข้อมูลต่างๆ ถูกเขียนขึ้นให้โดยอัตโนมัติ</li> <li>6) ฮาร์ดแวร์ที่อยู่ในการทดสอบ โดยการเชื่อมต่อจำนวนรอบเพิ่ม ประสิทธิภาพของตัวผลิตภัณฑ์ และสามารถตรวจสอบความผิดปกติของวงจรได้อย่างรวดเร็วและเรียบง่าย</li> </ol>	<ol style="list-style-type: none"> <li>1) อาจจะได้ไม่ได้ให้ผลลัพธ์ที่สมบูรณ์แบบมากนักในมุมมองของการใช้งานจริง</li> <li>2) ผู้ใช้งานจำเป็นต้องศึกษาการใช้งาน Simulink ไปด้วย</li> <li>3) ไม่เหมาะสำหรับระบบที่มีสัญญาณพิกาลหลายตัวทำงานร่วมกัน</li> <li>4) ไม่สนับสนุนระบบบัสสองทาง</li> </ol>

## 2.4 เริ่มต้นการใช้งาน Simulink

Simulink เป็นซอฟต์แวร์ที่สามารถทำให้ผู้ใช้งานจำลองการทำงานของระบบ และสามารถวิเคราะห์ผลลัพธ์ที่อาจจะมีการเปลี่ยนแปลงไปได้ตามช่วงเวลาของการทำงาน Simulink สามารถนำมาใช้ตรวจสอบระบบการทำงานที่เปลี่ยนแปลงไปตามเวลาได้หลากหลายรูปแบบ ตัวอย่างเช่น วงจรไฟฟ้า ระบบขับเคลื่อนแรงกระแทก ระบบเบรก รวมไปถึง ระบบไฟฟ้า เครื่องกล และระบบความร้อน

การจำลองระบบการทำงานที่เปลี่ยนแปลงไปตามเวลา มีการประมวลผลสองระดับใน Simulink อย่างแรก ผู้ใช้ สร้าง บล็อกโคแอดแกรม ใช้ Simulink model editor ซึ่งเป็นตัวคำนวณ

ความสัมพันธ์ทางคณิตศาสตร์ของค่าที่รับเข้ามาของระบบ สถานะของระบบ และผลลัพธ์ของระบบ โดยผู้ใช้สามารถกำหนดระยะเวลาที่จะให้ระบบจำลองการทำงานได้

#### 2.4.1 Creating Models

Simulink นำเสนอ editor ที่อนุญาตให้ผู้ใช้สร้างและต่อบล็อกที่เลือกมาจากไลบรารี ในไลบรารีรวบรวมเซตไว้ทันทีที่ Simulink มีไลบรารีของบล็อกที่จำลองระบบขั้นพื้นฐานเพื่อการใช้งานไว้มากมาย ผู้ใช้ Simulink สามารถสร้างบล็อกของตัวเองขึ้นมาและนำมาจำลองในระบบได้เองด้วย

#### 2.4.2 Custom Blocks

Simulink อนุญาตให้ผู้ใช้สร้างไลบรารีของบล็อกที่สร้างขึ้นเองโดยผู้ใช้ สามารถนำมาใช้จำลองในระบบได้ ผู้ใช้สามารถสร้างบล็อกได้จากทั้งในรูปแบบกราฟิกหรือโดยการเขียนโปรแกรมลงไปได้ การสร้างบล็อกโดยการเขียนโปรแกรมนั้น สามารถสร้างได้จาก M-file หรือ MEX-file ที่เก็บคำสั่งการทำงานของบล็อกเอาไว้ โดยเราจะเรียกไฟล์นี้ว่า S-function เราสามารถนำ S-function เข้ามาใช้งานร่วมกับ Simulink ในบล็อกที่เราสร้างขึ้นได้ทันที โดยสามารถเพิ่มค่าตัวแปรจาก parameter dialog เข้าไปใช้ใน S-function ของเรา โดยเขียนฟังก์ชันขึ้นมารองรับเพิ่มเติมตามจำนวนค่าตัวแปรได้

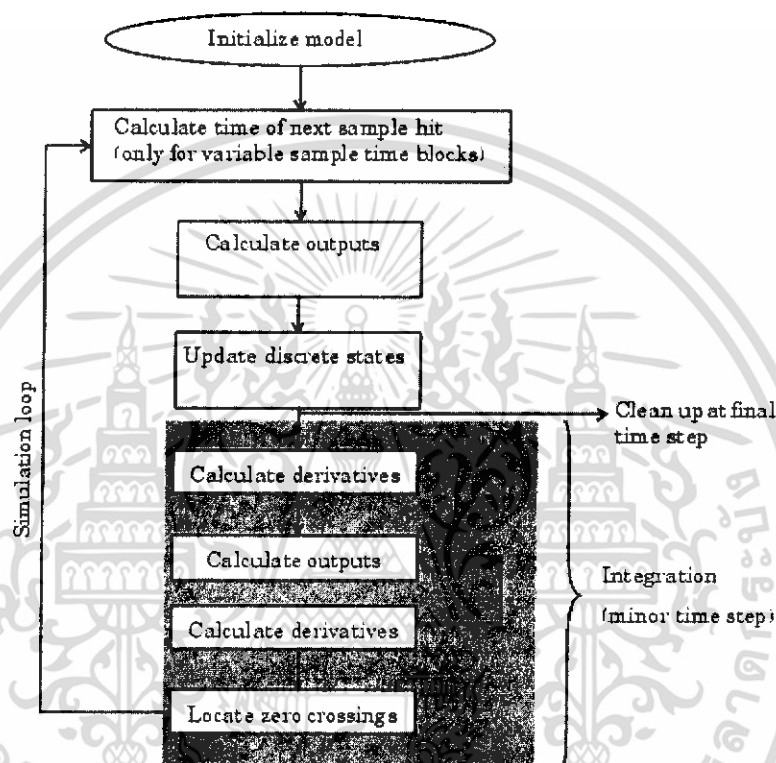
#### 2.4.3 Systems and Subsystems

บล็อกโคแอดแกรมของ Simulink สามารถรองรับงานเป็นชั้นๆ ได้ ในแต่ละชั้นจะเรียกว่า “ระบบย่อย” (subsystem) ระบบย่อยเป็นส่วนหนึ่งของระบบทั้งหมด ซึ่งมีไว้เพื่อการจัดการรูปแบบของบล็อกโคแอดแกรมให้เป็นหมวดหมู่มากขึ้น โดย Simulink แยกความแตกต่างของระบบย่อยไว้สองชนิดคือ เสมือนจริง (virtual) กับ ไม่เสมือนจริง (nonvirtual) สิ่งที่แตกต่างกันนั้นคือใน ระบบย่อยแบบ nonvirtual นั้นสามารถที่จะเข้าไปควบคุมว่าจะให้ รับหรือส่งผลลัพธ์ออกไปได้เมื่อไหร่ ซึ่งจากการศึกษาระบบย่อยและการทำงานแล้ว ทางผู้จัดทำได้เลือกใช้ระบบย่อยในรูปแบบไม่เสมือนจริง เพราะสามารถควบคุมผลลัพธ์ที่จะออกมาตามเวลาที่ต้องการได้

#### 2.4.4 Simulation Stages

การทำงานของ Simulink model กระทำเป็นทีละขั้นตอน เริ่มจากการตั้งค่า ในกระบวนการนี้ Simulink มองบล็อกออกมาเป็น model กำหนดขนาดความกว้าง ชนิดของข้อมูล และจังหวะของเวลา ตรวจสอบค่าพารามิเตอร์ของแต่ละบล็อก จัดเตรียมลำดับการคำนวณของทุกบล็อก และเตรียมขนาดของหน่วยความจำเพื่อรองรับการทำงานทั้งหมดไว้ จากนั้น Simulink จะเข้าสู่การจำลองระบบงาน โดยการทำงานเสร็จในแต่ละรอบจะเรียกว่าลำดับขั้นการจำลอง ในแต่ละลำดับขั้นการจำลอง Simulink จะ

คำนวณผลลัพธ์ที่จะออกมาของแต่ละบล็อก ที่ได้จากการตั้งค่าไว้แล้วในตอนต้น โดยในแต่ละบล็อก Simulink จะเรียกใช้ function ที่คำนวณสถานะของบล็อก ค่าที่เปลี่ยนแปลง และผลลัพธ์ในจังหวะเวลานั้น และวนทำเช่นนี้ต่อไปจนจบการจำลอง ภาพข้างล่างแสดงขั้นตอนการทำงานของการทำงานของการจำลองระบบใน Simulink



รูปที่ 2.4 แสดงการทำงานของ Simulink ในแต่ละรอบจังหวะเวลา

#### 2.4.5 Writing Level-2 M-File S-Functions

Level-2 M-file S-function API ทำให้ผู้ใช้งานสามารถใช้ภาษา M ของ MATLAB ในการสร้างบล็อกของตัวเองขึ้นมาได้โดยไม่จำกัดจำนวนอินพุตและเอาต์พุตและยังรองรับสัญญาณใดๆที่เกิดขึ้นในระบบ Simulink ได้ทั้งหมด รวมทั้งค่าเมตริกซ์ของทุกชนิดข้อมูล

Level-2 M-file S-function คือ M-file ที่เก็บค่าคำสั่งและลักษณะการทำงานของบล็อกที่ผู้ใช้สร้างขึ้นมานำมาใช้ใน Simulink model โดยผู้ใช้งานสามารถปรับเปลี่ยนโค้ดได้ตามใจชอบ เพื่อรองรับการทำงานจากการจำลองได้ ซึ่งในการเขียนแบบ M-file นี้ ผู้ใช้งานสามารถนำฟังก์ชันต่างๆใน MATLAB มาใช้งานได้ทันทีที่เหมือนกับการเขียน M-file ทั่วไป นอกจากนี้ S-function ยังสามารถเขียนได้ในภาษาอื่นๆอีก เช่น ภาษา C, Pascal, Ada, Fortran โดยสามารถแปลงมาเป็นไฟล์ที่ใช้งานใน MATLAB

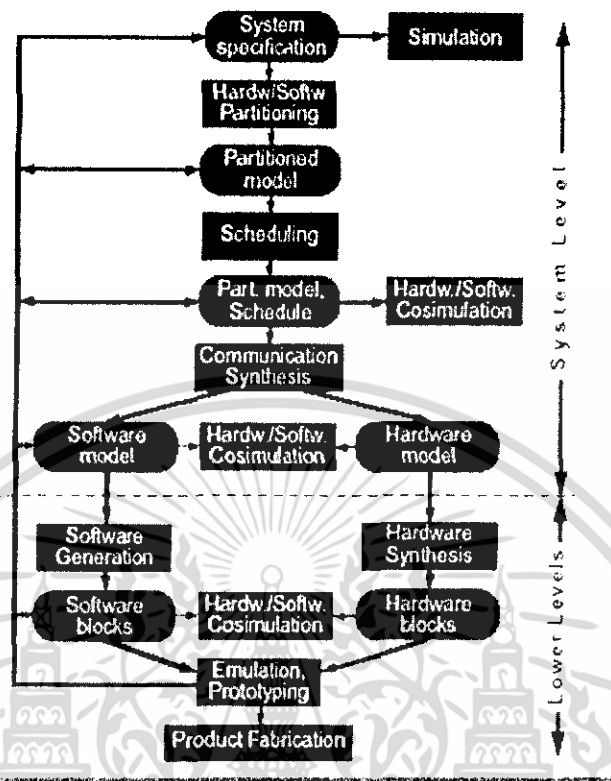
ได้ ในโครงการนี้จะมุ่งไปที่การเขียน Level-2 M-file S-function ในแบบ M-file เพราะความง่ายของการทำงานและประยุกต์ใช้ร่วมกับระบบ Simulink

## 2.5 การออกแบบร่วมกันระหว่าง ฮาร์ดแวร์ และ ซอฟต์แวร์ (Hardware-Software Codesign)

นิยามของ Codesign ที่ให้โดย Franke และ Purvis ในบทความในปี 1991 ว่า “เป็นการออกแบบระบบที่มีทั้งฮาร์ดแวร์ และซอฟต์แวร์ พร้อม ๆ กันตั้งแต่จุดเริ่มต้นของการออกแบบ เพื่อความยืดหยุ่นในการออกแบบ และประสิทธิภาพของการจัดการฟังก์ชันการทำงานในระบบอย่างเหมาะสม” Wayne Wolf ให้นิยามเพิ่มเติมว่า “ ระบบจะต้องออกแบบร่วมกันทั้งในส่วนของฮาร์ดแวร์ และซอฟต์แวร์ เพื่อเป็นการประกันว่าสามารถทำงานได้ในตอนท้ายที่สุด ซึ่งมีประสิทธิภาพ เชื่อถือได้ และราคาตามที่ต้องการ ”

ดังนั้นการออกแบบร่วมกันระหว่างฮาร์ดแวร์-ซอฟต์แวร์เป็นสร้างระบบที่มีความสมดุลกันระหว่างประสิทธิภาพ (System performance) ที่ได้มาจากฮาร์ดแวร์ และราคา (System cost) ที่ได้มาจากซอฟต์แวร์ หลักการของCodesign คือการหลีกเลี่ยงการแยกกันของการออกแบบฮาร์ดแวร์ และการออกแบบซอฟต์แวร์ โดยได้ผลลัพธ์เป็นระบบที่มีประสิทธิภาพ

Hardware software codesign ในภาพที่ 2.5 นั้น ผู้ออกแบบเริ่มต้นออกแบบจากภาษาชั้นสูง (High-level design language) ในการกำหนดความสามารถของระบบ (System specification) หลังจากนั้นก็จะพยายามออกแบบทั้งในส่วนของ ฮาร์ดแวร์และซอฟต์แวร์ไปพร้อม ๆ กัน โดย codesign tools เช่น *Partitioner* ที่แบ่งส่วนงานให้ไปทำที่ฮาร์ดแวร์หรือซอฟต์แวร์ *Scheduler* ที่เรียงลำดับการทำงานในส่วนต่าง ๆ ของระบบทั้งที่อยู่ในฮาร์ดแวร์ และซอฟต์แวร์ โดยมีการกระทำที่เรียกว่า *Cosimulation* เพื่อทดสอบการทำงานร่วมกันตลอดเวลา



รูปที่ 2.5 แสดงการพัฒนาแบบ Reconfigurable computing system

2.5.1 ประเภทของการออกแบบร่วมกันระหว่างฮาร์ดแวร์-ซอฟต์แวร์

การประยุกต์ใช้งานในลักษณะของการออกแบบร่วมกันระหว่างฮาร์ดแวร์-ซอฟต์แวร์มีได้หลากหลาย โดยมีรากฐานมาจากระบบฝังตัว หรือ Embedded system ซึ่งเป็นระบบที่ประกอบด้วย ไมโครโปรเซสเซอร์ และอุปกรณ์ฮาร์ดแวร์ เพื่อช่วยเพิ่มประสิทธิภาพของระบบ ยิ่งถ้าเราสามารถที่จะทำฮาร์ดแวร์ในระบบมีความยืดหยุ่นได้โดยใช้ FPGA ก็จะทำให้มีความน่าสนใจในการใช้งานเพิ่มขึ้นอย่างมากทีเดียว แบ่งลักษณะของระบบ Codesign ให้เป็นสองลักษณะใหญ่ๆ ดังนี้คือ

- Embedded system: ระบบจะประกอบด้วย CPUs และทรัพยากรในระบบต่าง ๆ เช่น ASIC, FPGA, หน่วยความจำ (Memory), การอินเตอร์เฟสกับผู้ใช้งาน (Human interface), พอร์ตสำหรับการวิเคราะห์ (Diagnostic port), ตัวแปลงสัญญาณ ADC/DAC, อุปกรณ์ตรวจจับ (Sensor) และอุปกรณ์ทางกล (Actuator) ระบบนี้มักจะออกแบบขึ้นมาเพื่อจุดประสงค์เฉพาะอย่าง เช่น บอร์ดควบคุมในเครื่องซักผ้า ระบบเบรคในรถยนต์ หรือระบบบังคับการบินในเครื่องบิน เป็นต้น ถ้าจะแยกย่อยลงไปอีกเรา

72927

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแบ่งเป็นระบบที่มีการตอบสนอง (Interact) สอดคล้องสถานะในการใช้งานจริง โดยมีตัวอุปกรณ์ตรวจจับและอุปกรณ์ทางกล ในการใช้งานกับส่วนต่าง ๆ ของระบบให้มีการตอบสนองเป็นลักษณะของ Real-time reactive ภายใต้งานที่กำหนดให้ทำงาน ที่กำหนดไว้ในตัว CPU อีกลักษณะหนึ่งจะเป็นระบบที่ไม่มีมีการตอบสนองกับสภาวะรอบด้าน แต่ใช้ในการประมวลผลทางข้อมูล หรือการสื่อสาร เพื่อที่จะรองรับการประยุกต์ใช้งานแบบ Real-time เช่น การบีบอัดสัญญาณการเข้ารหัสสัญญาณ ซึ่งจะต้องมีฮาร์ดแวร์มาช่วยในการคำนวณให้สามารถทำงานได้ทันตามที่ต้องการ

• Reconfigurable computing system: ระบบจะประกอบด้วย CPU และฮาร์ดแวร์ที่สามารถโปรแกรมได้นั้นคือ FPGAรวมทั้งหน่วยความจำ ซึ่งสามารถใช้ร่วมกับคอมพิวเตอร์ หรือมี CPU เป็นของตัวเอง สิ่งที่กำหนดความแตกต่างกับ Embedded system ก็คือ ระบบนี้สามารถทำการปรับเปลี่ยนวงจรฮาร์ดแวร์ของตัวเองได้ ตามลักษณะของการประยุกต์ใช้งานที่ต่างกัน ซึ่งการปรับเปลี่ยนนี้สามารถทำได้ก่อนการเริ่มทำงานของระบบ (Compile-time configuration) หรือทำในขณะที่ระบบมีการทำงาน (Run-time reconfiguration) ดังนั้นจะเห็นได้ว่า ระบบนี้มีความยืดหยุ่นสูงมาก มีการใช้ทรัพยากรฮาร์ดแวร์อย่างคุ้มค่าโดยสามารถนำใช้ใหม่ (Reuse) ได้โดยการโปรแกรมในอุปกรณ์ฮาร์ดแวร์ตัวเดิม ปัญหาก็คือ การออกแบบก็จะมีความซับซ้อนขึ้นเป็นทวีคูณ

## 2.6 ขั้นตอนการออกแบบ FPGA

### 2.6.1 ขั้นตอนที่ 1 การสร้างข้อกำหนดของการออกแบบ (Design Specification)

เป็นขั้นตอนการสร้างข้อกำหนดต่างๆ ของวงจร เช่นวงจรทำงานที่ความถี่เท่าไร ฟังก์ชันมีอะไรบ้าง ซึ่งเป็นรายละเอียดของวงจรที่ต้องการออกแบบและเขียนฟังก์ชันการทำงานของวงจรตามที่ถูกออกแบบกำหนดด้วยภาษา VHDL ในระดับ RTL

### 2.6.2 ขั้นตอนที่ 2 จำลองการทำงานโมเดลวงจรระดับ RTL (RTL Simulation)

เป็นขั้นตอนตรวจสอบการทำงานของโมเดลวงจรในระดับ RTL หรือฟังก์ชันโดยการจำลองการทำงาน จะถูกทดสอบด้วย Testbench ซึ่งการจำลองการทำงานในขั้นตอนนี้ จะจำลองการทำงานเพียงฟังก์ชัน ไม่คำนึงถึงค่าดีเลย์ (Delay) ของวงจรแต่อย่างใด

### 2.6.3 ขั้นตอนที่ 3 สังเคราะห์และออปติไมซ์วงจร (Synthesis & Optimization)

เป็นขั้นตอนการสร้างแผนภาพวงจร (Schematic) จากโมเดลวงจรระดับ RTL ให้อยู่ในรูปแบบของลอจิกเกต โดยอาศัยซอฟต์แวร์ช่วยในการสังเคราะห์วงจร โดยในขั้นตอนนี้จะต้องมีการเลือกใช้

เทคโนโลยี FPGA ที่ผู้ออกแบบต้องการเลือกใช้ ซึ่งบริษัทผู้ผลิต FPGA จะมีเทคโนโลยีไลบรารี (Technology library) เตรียมไว้ให้ผู้ออกแบบไว้ในซอฟต์แวร์ที่ใช้พัฒนาชิป FPGA ของแต่ละบริษัทผู้ผลิตไว้เรียบร้อยแล้ว โดยเมื่อทำการสังเคราะห์ได้ผังวงจร จากนั้นซอฟต์แวร์จะทำการออปติไมซ์วงจรตามข้อกำหนดหรือเงื่อนไขของการสังเคราะห์ ที่ผู้ออกแบบกำหนดขึ้นตามขั้นตอนแรกของการออกแบบ ซึ่งผลลัพธ์ที่ได้จากการสังเคราะห์จะอยู่ในรูปแบบไฟล์ VHDL แบบโครงสร้างลอจิก (Structured VHDL) และไฟล์เน็ตลิสต์มาตรฐาน (Netlist) ประเภท EDIF ( Electronic Design Interchange Format) ที่จะนำไปใช้ในขั้นตอน Place & route ต่อไป

#### 2.6.4 ขั้นตอนที่ 4 การจำลองการทำงานของวงจรระดับลอจิกเกต (Gate level Simulation)

เป็นขั้นตอนที่ผู้ออกแบบจะต้องทดสอบไฟล์เน็ตลิสต์ที่เป็นโมเดลของวงจรระดับ

ลอจิก โดยใช้ Testbench ตัวเดิมที่ใช้จำลองการทำงานระดับ RTL มาแล้วซึ่งในการจำลองการทำงานในระดับนี้ จะมีเรื่องของเกตดีเลย์ (Gate delay) เข้ามาเกี่ยวข้องในผลการจำลองการทำงาน ซึ่งจะแตกต่างจากการจำลองการทำงานในระดับ RTL เนื่องจากโมเดลของวงจรระดับเกตนี้ จะมีข้อมูลเรื่องดีเลย์ของเกตภายในเทคโนโลยีที่ผู้ออกแบบเลือกมาใช้ ดังนั้นผู้ออกแบบจะต้องทำการจำลองการทำงานเพื่อตรวจสอบให้มากยิ่งขึ้นว่ายังถูกต้องตามข้อกำหนดของวงจรหรือไม่ ถ้าไม่ตรงตามข้อกำหนดจะต้องกลับไปขั้นตอนที่ 3 เพื่อทำการสังเคราะห์และออปติไมซ์วงจรใหม่ เพื่อให้ผลการจำลองการทำงานถูกต้องตาม Design specification

#### 2.6.5 ขั้นตอนที่ 5 การวางและเชื่อมต่อเซลล์ภายในของ FPGA (Place & Route)

เมื่อตรวจสอบการทำงานในระดับลอจิกเกตเป็นที่เรียบร้อยแล้ว เราจะนำไฟล์เน็ตลิสต์ที่อยู่ในรูปแบบ EDIF มาทำการแปลงสู่เทคโนโลยีเซลล์ภายในของ FPGA และทำการเชื่อมต่อเซลล์ภายในเข้าด้วยกัน ตามรูปแบบการเชื่อมต่ออุปกรณ์ต่างๆ ภายในเน็ตลิสต์ โดยขั้นตอนนี้ จะมีการเรียกใช้เทคโนโลยีเซลล์ของ FPGA เนื่องจากเทคโนโลยีเซลล์ของ FPGA อาจมีมาโครเซลล์ (Macro cells) สำหรับสร้างฟังก์ชันต่างๆ ให้กับผู้ออกแบบใช้งานเพราะมาโครเซลล์ต่างๆ พวกนี้ถูกทางผู้ผลิตผู้ออกแบบไว้สำหรับชิป FPGA แต่ละตัว โดยผลลัพธ์จากขั้นตอนนี้ จะอยู่ในรูปของไฟล์ VHDL โครงสร้างของเซลล์ภายในของ FPGA และไฟล์ประเภท SDF ( Standard Delay Format) ซึ่งเป็นไฟล์รูปแบบมาตรฐานที่มีข้อมูลเกี่ยวกับค่าดีเลย์ของเส้นทางการเชื่อมต่อภายใน (Routing delay) เซลล์ FPGA และไฟล์ที่ใช้สำหรับไปโปรแกรมชิป FPGA ซึ่งส่วนใหญ่จะอยู่ในรูปแบบของ Configuration file

#### 2.6.6 ขั้นตอนที่ 6 การจำลองการทำงานวงจรระดับฐานเวลาจริง (Timing Simulation)

เป็นขั้นตอนสุดท้ายของการตรวจสอบความถูกต้อง ก่อนจะนำวงจรที่ออกแบบไปโปรแกรมหรือบางครั้งเรียกว่า Download ลงสู่ชิปจริงต่อไป โดยขั้นตอนการจำลองการทำงานในระดับไทม์มิง (Timing model) โดยผลลัพธ์จากการจำลองการทำงานที่ได้จะมีความใกล้เคียงกับไทม์มิงการทำงานจริงบนชิป FPGA เนื่องจากในการจำลองการทำงานในขั้นตอนนี้มีข้อมูลเกี่ยวกับดีเลย์ของเซลล์ภายใน FPGA และดีเลย์ของการเชื่อมต่อเซลล์เข้ามาเกี่ยวข้อง ทำให้ผลการจำลองการทำงานในระดับนี้ใกล้เคียงกับไทม์มิงการทำงานจริงบนฮาร์ดแวร์

#### 2.6.7 ขั้นตอนที่ 7 โปรแกรมลงสู่ชิปจริง (Download to device)

เป็นขั้นตอนสุดท้ายสำหรับการออกแบบวงจรเพื่อใช้งานชิป FPGA คือขั้นตอนโปรแกรม Configuration file ลงสู่ชิป FPGA บนบอร์ดเพื่อทดสอบการทำงานจริงต่อไป



### บทที่ 3

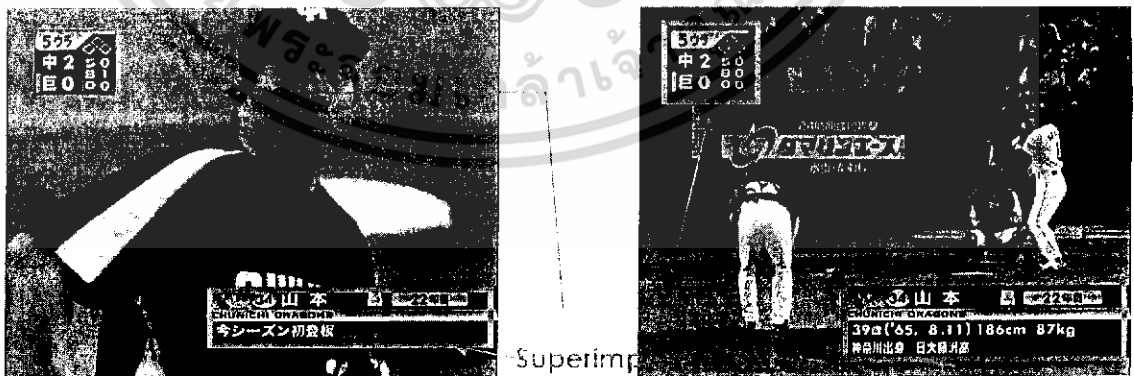
## การค้นหาจากวิดีโอ โดยอ้างอิงจากผู้ใช้

### 3.1 การค้นหาจากวิดีโอ อาศัย Superimposed text เป็นหลักในการค้นหา

#### 3.1.1 แนวความคิด

การทำดัชนีของวิดีโอ นั้น ส่วนใหญ่จะเริ่มจากการจดจำข้อความบนวิดีโอ (Superimposed Text) ซึ่งนับว่าเป็นข้อจำกัดอย่างยิ่งในการทำระบบออกไปใช้งานที่ที่ภาษานั้นต่างกัน เช่น ระบบที่ออกแบบในประเทศไทย ก็ยากจะนำไปใช้ในประเทศญี่ปุ่น ซึ่งจุดนี้เองทำให้ผู้พัฒนาคิดถึงระบบแบบ World Wide (ทั่วโลก) ที่สามารถใช้ร่วมกันได้

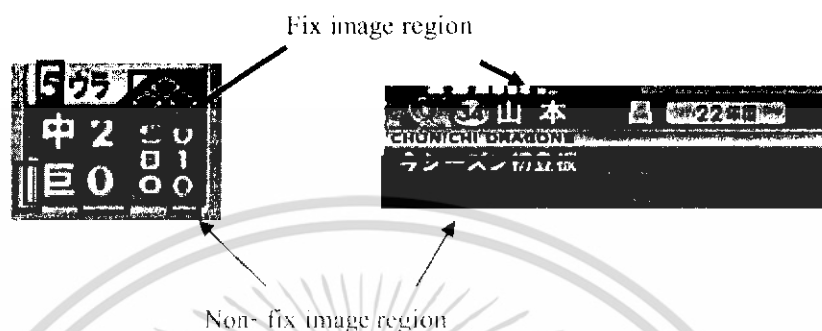
Superimposed Text เป็นข้อความที่ใช้อธิบายเหตุการณ์ หรือ สรุปเหตุการณ์ในเฟรมวิดีโอ ณ เวลานั้นๆ ซึ่งจะถูกสร้างมาจากผู้กำกับ หรือผู้จัดทำในสตูดิโอของห้องส่ง ซึ่งข้อความเหล่านี้จะสั้น และกระชับ มีความหมายชัดเจน เช่น สกอร์ของการแข่งขันฟุตบอล การแสดงชื่อผู้เล่นเมื่อกล้องจับหน้าผู้เล่นคนนั้น หรืออธิบายเกม ก็ตาม ซึ่งจะเป็นส่วนดึงดูดผู้ใช้ให้สนใจมากขึ้น ซึ่งจากการสังเกตรูปแบบวิดีโอ นั้น Superimposed Text นับเป็นกีย์หลัก เพราะว่า พื้นที่ของ Superimposed Text (ภาพที่ 3.1) นั้นจะมีตำแหน่งคงที่ และภายในพื้นที่นี้นั้นจะมี พื้นที่ Fix-image region โดยยกตัวอย่างเช่นภาพที่ 3.2 การนำจุดนี้มาตัดแปลงเป็นดัชนีให้กับผู้ใช้เห็น พบว่า “ระบบสามารถใช้ได้แบบ World Wide โดยไม่ต้องอาศัยการจดจำ”



รูปที่ 3.1 แสดง Superimposed Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง ณ ตำแหน่งเดียวกัน Superimposed Text จะมีเนื้อหาเป็นเรื่องเดียวกัน เช่น คะแนนในการแข่งขันเบสบอล ณ ประเทศญี่ปุ่น จะปรากฏอยู่ทางซ้ายมือด้านบนเสมอ และ จะมีข้อความอธิบายผู้เล่น ในด้านซ้ายมือล่าง ที่แสดงให้เห็นในภาพที่ 3.1



รูปที่ 3.2 แสดง Non – fix image region

จากการสังเกต พบว่าความสัมพันธ์ระหว่าง พื้นที่คงที่ กลับ พื้นที่ ไม่คงที่ นั้นสามารถนำมาทำดัชนีได้ โดยมีความสัมพันธ์ดังนี้

1. พื้นที่คงที่ และ พื้นที่ไม่คงที่ จะวิเคราะห์ภายใน Superimposed Text เท่านั้น
2. พื้นที่คงที่ จะปรากฏ ณ ตำแหน่งเดียวกัน มีลิสต์เหมือนกันทั้งตลอดรายการ หรือ ทั้งเกมกีฬา โดยพื้นที่ ที่ไม่คงที่จะเปลี่ยนแปลง เช่น คะแนน ชื่อ รายละเอียด
3. Superimposed Text จะเป็นตัวอธิบายเหตุการณ์ ณ ขณะเฟรมนั้น ปรากฏอยู่และมักจะขึ้น เมื่อเกิดเหตุการณ์สำคัญๆ เช่น ในเกมกีฬา; มีการเปลี่ยนตัวผู้เล่น มีอธิบายลักษณะของแต่ละผู้เล่น เกม หรือคะแนนในการแข่งขัน เป็นต้น

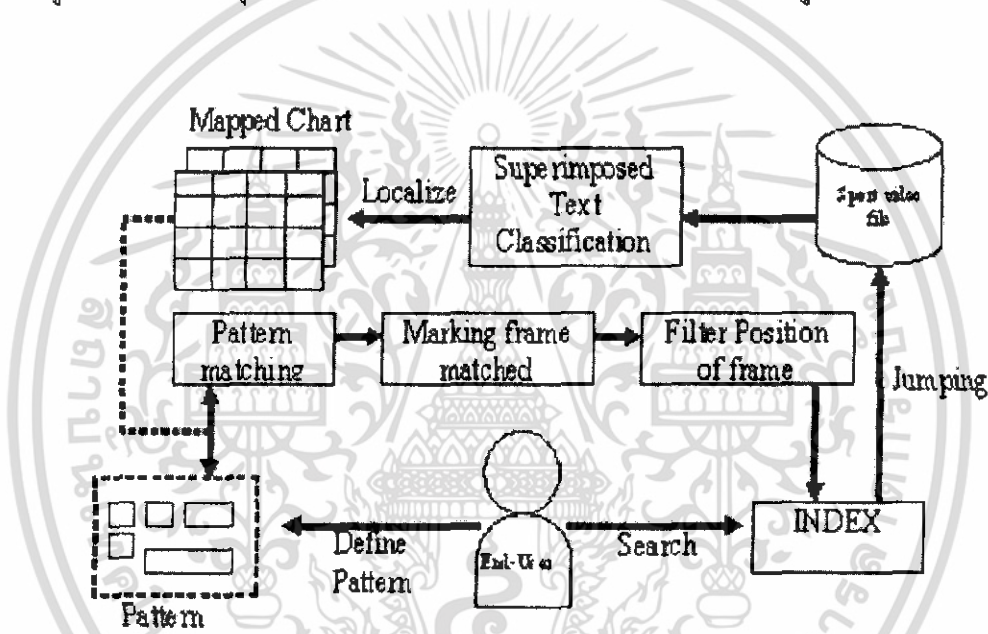
จากข้อสันนิษฐานทั้งสามข้อนี้ ผู้จัดทำได้ออกแบบ ลักษณะตัวโปรแกรม โดยระบบที่ออกแบบมานั้น จะอนุญาตให้ผู้ใช้ นั้น สามารถกำหนดรูปแบบขึ้นมาเองได้ (Pattern) ซึ่งพบว่า “การกำหนดรูปแบบโดยผู้ใช้นั้น จะตรงกับความต้องการของตนเองมากที่สุด [4]”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 โครงสร้างของระบบ

การออกแบบ ผู้จัดทำได้ออกแบบ ระบบการค้นหาจากแบบ World Wide และอนุญาตให้ผู้ใช้ได้กำหนดรูปแบบในการค้นหาเอง โดยทำการพัฒนาโดยใช้เครื่องมือ VC++.net กับ OpenCV library โดยการทดลองจะทดลองกับกีฬา เบสบอล ซึ่งจะเห็นภาพชัดมากที่สุด เพราะ กีฬาเบสบอลมีรายละเอียดในการเล่นมากมาย

ระบบนั้น จะเห็นว่า ทำอย่างไรให้ติดต่อกับผู้ใช้งานได้ง่ายที่สุด โดยให้ผู้ใช้งานได้กำหนดรูปแบบได้ง่ายที่สุด และ ผลลัพธ์ออกมาได้ตรงกับความต้องการของผู้ใช้งานจริง



รูปที่ 3.3 โครงสร้างของระบบค้นหาในไฟล์วิดีโอ

ในการออกแบบให้ผู้ใช้เลือก Pattern จะทำการจำลองว่า ผู้ใช้งานนั้น ได้ใช้รีโมท เป็นตัวออกแบบ ไม่ใช่ในเมาส์หรือคีย์บอร์ดเป็นตัวออกแบบ โดยวิธีการใช้รีโมทนั้น จะไม่สามารถที่จะกำหนดรูปแบบได้โดยละเอียด โดยผู้จัดทำได้ออกแบบให้ผู้ใช้งานนั้นออกแบบรูปแบบ ได้สองวิธี คือ

1. เลือกรูปแบบในการค้นหาโดยกำหนด พื้นที่ คงที่ไว้
2. เลือกรูปแบบ โดยใช้เครื่องมือช่วยเหลือ หาพื้นที่ คงที่



รูปที่ 3.4 การเลือกรูปแบบการค้นหาวีธีที่ 1



รูปที่ 3.5 การเปรียบเทียบหาพื้นที่คงที่

วิธีที่สอง นั้นเป็นการใช้ตัวช่วยในการกำหนด พื้นที่คงที่ โดยเราจะเลือกเพียงแค่ Superimposed Text จากหลายๆ จาก แล้วนำมารวมกัน เพื่อหา พื้นที่คงที่ภายในโดยอาศัยหาพิกเซลที่เคลื่อนที่ หรือ พิกเซลที่เปลี่ยน ไป เช่น ภาพที่ 3.5 ซึ่งจะเป็นการหาพื้นที่ คงที่ จากจากที่สามโดย พื้นที่ ในกรอบสีน้ำเงินนั้น จะเป็นพื้นที่ ที่เราไม่ต้องการ ใช้ในการค้นหา พิกเซลที่มีการเปลี่ยนแปลงนั้น จะใช้ หลักการของ Standard Deviation

$$M_{(x,y)} = \frac{S_{(x,y)}}{N} \tag{3.1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sigma_{(x,y)} = \sqrt{\frac{Sq_{(x,y)}}{N} - \left(\frac{S_{(x,y)}}{N}\right)^2} \quad (3.2)$$

$$\left| M_{(x,y)} - P_{(x,y)} \right| > C[\sigma_{(x,y)}] \quad (3.3)$$

- $S_{(x,y)}$  คือค่าผลบวกของค่า pixel ในแต่ละตำแหน่ง
- $Sq_{(x,y)}$  คือค่าผลบวกของกำลัง 2 ของค่าในแต่ละตำแหน่ง
- $N$  คือค่าตัวเลขของ frame ที่รวบรวมไว้
- $C$  คือค่าคงตัว
- $C_{(x,y)}$  คือค่าการเบี่ยงเบนมาตรฐาน
- $P_{(x,y)}$  คือค่าในแต่ละ pixel

สาเหตุของการใช้ Standard Deviation นั้นก็ ค่าพิกเซลนั้นในแต่ละตำแหน่งจะมีการกระจายของค่าความสว่างอยู่ ซึ่งเราไม่สามารถนำมาลบกันได้ ก็คือ แม้ว่าจะเป็นพิกเซลที่ตำแหน่งเดียวกัน รูปเหมือน แต่ค่าความสว่างของพิกเซลอาจจะต่างกัน ดังนั้น เราไม่สามารถนำรูปแต่ละรูปมาลบกันได้โดยตรง เราต้องการหา Standard Deviation เสียก่อน ซึ่งจะนำมาแทนได้ว่า พิกเซลนั้นๆ ได้เปลี่ยนแปลงไปหรือไม่ ตาม สมการที่ (3.1), (3.2) และ (3.3) โดยที่ พิกเซลที่อยู่ในเงื่อนไขที่ (3.3) นั้นถือว่าเป็นพิกเซลที่เคลื่อนที่ และ เป็นพิกเซลที่เปลี่ยนแปลง โดยผู้จัดทำได้ทำการทดลอง กับ สกอร์บอร์ด ใน กีฬาเบสบอล ดังภาพที่ 3.6



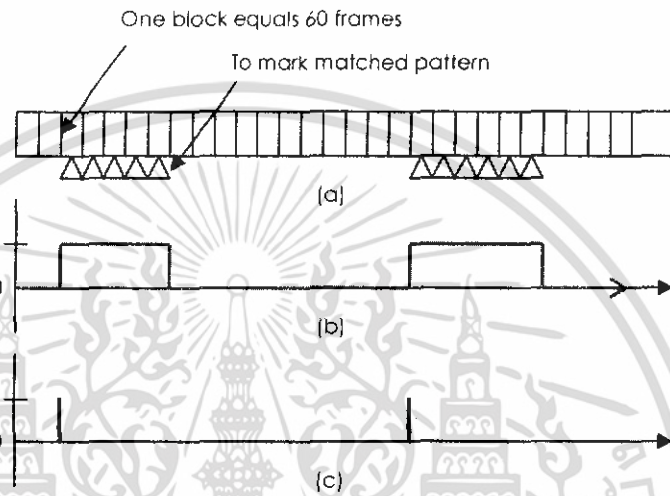
รูปที่ 3.6 การหาพิกเซลที่เคลื่อนที่

ระบบจะทำการนำรูปแบบของผู้ใช้ที่ได้ออกแบบไว้ มาเปรียบเทียบกับวิดีโอ ซึ่งการเปรียบเทียบนั้น จะเป็นไปตามสมการ ที่ (3.4)

$$I = \begin{cases} 1; & \sum_{(i,j) \in R} (f - g)^2 < Threshold \\ 0; & otherwise \end{cases} \quad (3.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $f$  เป็นภาพของ Pattern และ ภาพ  $g$  เป็นภาพที่จะทำการเปรียบเทียบในวิดีโอไฟล์ ซึ่งได้กำหนดไว้ว่า ถ้าภาพที่ผ่าน threshold จะกำหนดไว้เป็น 1 และภาพที่ไม่ผ่านจะกำหนดไว้เป็น 2 โดยการเปรียบเทียบนั้น จะไม่ได้เปรียบเทียบ pattern กับภาพทั้งเฟรม จะเปรียบเทียบเฉพาะตำแหน่งเดียวกันกับ pattern



รูปที่ 3.7 การระบุตำแหน่งเฟรมที่ตรงกับ Pattern

เฟรมที่ตรงกัน กับ สมการที่ (4) จะทำการกำหนดตำแหน่งไว้ (Marking) ซึ่งพบว่า เราจะไม่ทำการเปรียบเทียบทุก ๆ เฟรม ซึ่งจะทำการเปรียบเทียบ ทุก ๆ 30 เฟรมแทน เพราะว่า ไม่มีประโยชน์ที่จะทำการเปรียบเทียบทุก ๆ เฟรม ยิ่งทำให้เสียเวลามากขึ้น เพราะ ในหนึ่งวินาที ภาพนั้นเกิดการเปลี่ยนแปลงทางเนื้อหาบ่อยมาก จากนั้น จะทำการสังเกตลักษณะ กราฟออกแบบ ดังรูป ที่ 3.7 (b) และ จะกำหนดฉากที่ตรงกับรูปแบบของผู้ใช้ โดยกำหนดไว้ในขอบข้างขึ้นของกราฟ ดังภาพที่ 3.7 (c)

## 3.2 การพัฒนาโปรแกรมจำลองการค้นหาคอนวิดิโอ

### 3.2.1 วิธีการพัฒนาโปรแกรม

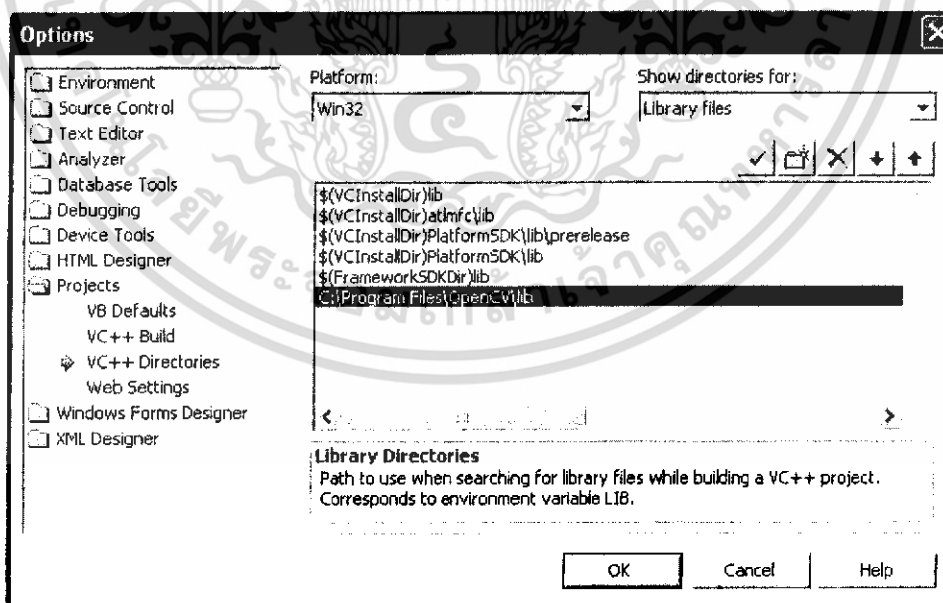
การพัฒนาโปรแกรมได้ใช้เครื่องมือ คือ Visual C++.net ร่วมกับ OpenCV library ในการพัฒนาโปรแกรมจำลองการค้นหาคอนวิดิโอ โดยขั้นตอนการพัฒนาเริ่มต้นจาก

1. ติดตั้งค่า Global เพื่อเชื่อมการใช้งาน OpenCV และ VC++.net

หลังจากการติดตั้ง OpenCV ที่ได้ดาวน์โหลดมาจากเว็บ source force แล้วนั้น ก็ต้องทำการตั้งค่าก่อนโดยได้อธิบายเป็นลำดับดังต่อไปนี้

- เปิด Visual C++.Net เลือก Tools-> Options ในเมนูบาร์
- เลือก Project -> VC++ Directories ในลิสต์
- ลำดับแรกนั้น เลือก Library files จาก "Show Directories for" ใน list box
- คลิกไอคอน Insert New และเลือก folder ที่ได้ทำการติดตั้ง OpenCV ในไลบรารี ให้เพิ่ม

"C:\Program Files\OpenCV\lib"

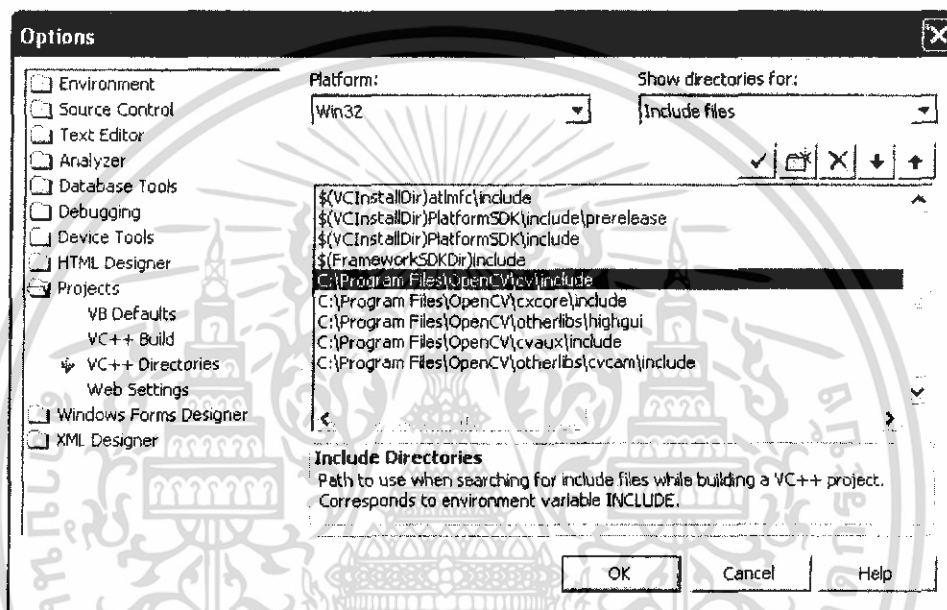


รูปที่ 3.8 การตั้ง path ไลบรารีให้สามารถใช้งานร่วมกับ OpenCV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปทำการเลือกหัวข้อต่อมา คือ Include file ภายใน list box และกำหนด path เช่นเดียวกับไลบรารี โดยเลือกไดเรกทอรีดังต่อไปนี้

```
"C:\Program Files\OpenCV\cv\include"
"C:\Program Files\OpenCV\cxcore\include"
"C:\Program Files\OpenCV\otherlibs\highgui"
"C:\Program Files\OpenCV\cvaux\include"
"C:\Program Files\OpenCV\otherlibs\cvcam\include"
```

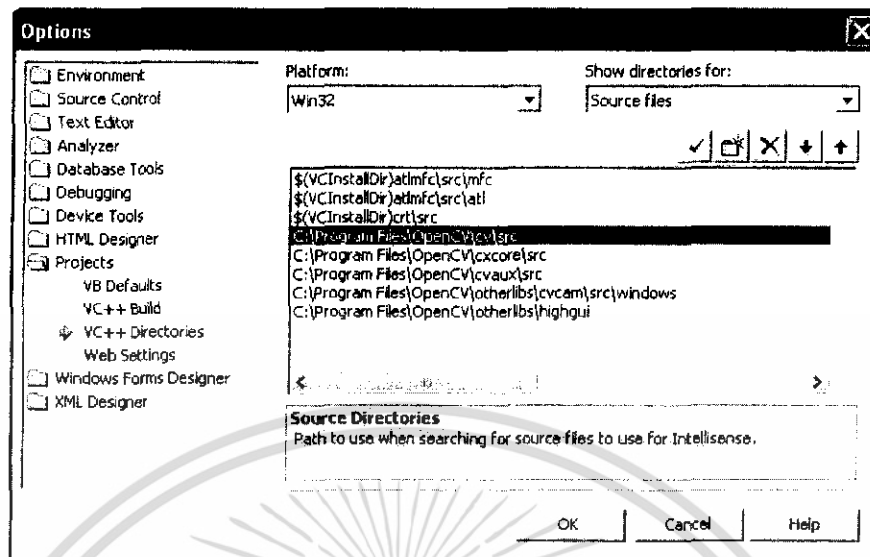


รูปที่ 3.9 การตั้ง path Include file ให้สามารถใช้งานร่วมกับ OpenCV

ต่อมาเลือก Source files ใน list box และทำการตั้งค่า path ไดเรกทอรี เช่นเดียวกับที่ทำมาในขั้นต้น

```
"C:\Program Files\OpenCV\cv\src"
"C:\Program Files\OpenCV\cxcore\src"
"C:\Program Files\OpenCV\cvaux\src"
"C:\Program Files\OpenCV\otherlibs\highgui"
"C:\Program Files\OpenCV\otherlibs\cvcam\src\windows"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

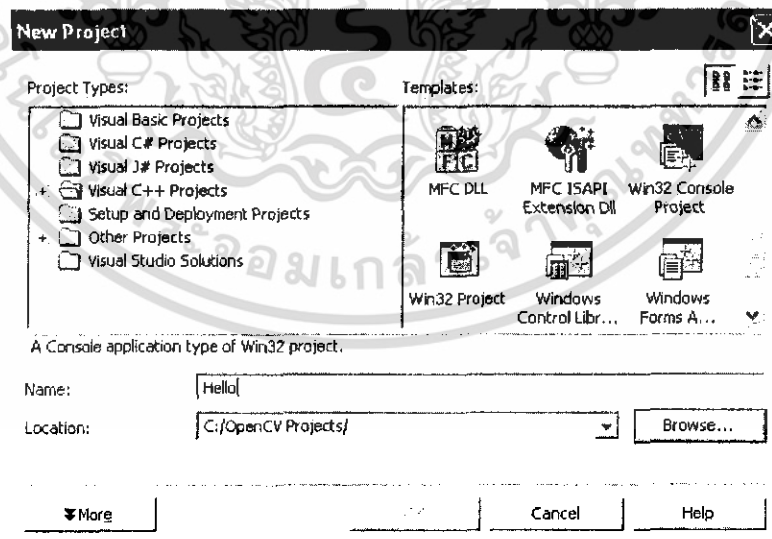


รูปที่ 3.10 การตั้ง path source file ให้สามารถใช้งานร่วมกับ OpenCV

จากนั้นเลือก Ok ทั้งหมดนี้เป็นขั้นตอนการติดต่อกำหนดขั้นชนิด Global Setting เพื่อลิงก์การใช้งานร่วมกันระหว่าง OpneCV และ Visual C++ .net

## 2. สร้างโปรเจกใหม่

ในเครื่องมือการพัฒนาของ Studio นั้น ให้สร้างโปรเจกใหม่ โดยการเลือกเมนู File -> New... -> "Project" โดยทำการเลือก "Win32 Application" หรือ "Win32 console application"



รูปที่ 3.11 การสร้างโปรเจกใหม่

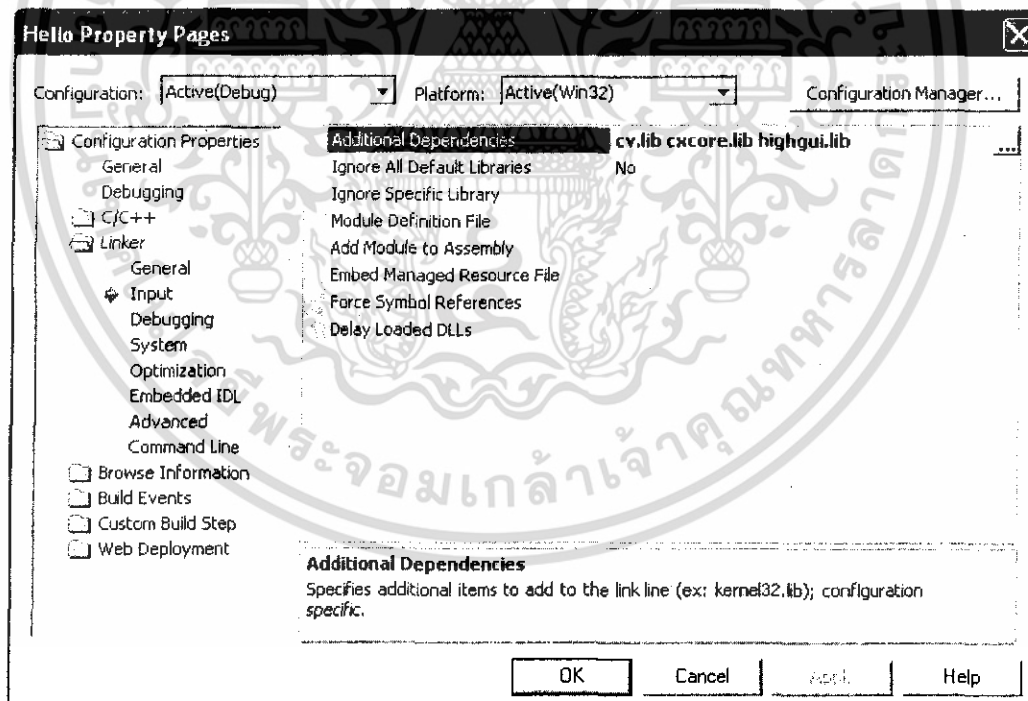
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเสร็จสิ้นขั้นตอนเบื้องต้นนั้น Developer Studio จะสร้างขึ้นมา 5 ไฟล์ นั่นก็คือ <project name>.vcproj , <project name>.sln, <project name>.cpp, stdafx.cpp และ stdafx.h ซึ่ง StdAfx นี้เองนั้นเป็นไฟล์ header ก่อนที่จะเกิดการคอมไพล์ ซึ่งเป็นประโยชน์มากในกรณีลดเวลาในการคอมไพล์ตัวโปรแกรม

### 3. วิธีการเขียนโปรแกรมร่วมกัน OpenCV

เปิดไฟล์ <project name>.cpp จากนั้นเพิ่มคำสั่งที่เกี่ยวข้องกับ Open คือ #include<cv.h>, #include<cxcore.h>, #include<highgui.h> ควรจะเพิ่มหลังจาก stdafx.h ไม่งั้นจะเกิดการ build error ได้ จากนั้นก็ต้องมีการติดตั้งค่าเฉพาะในโปรเจก คือ เลือกเมนู “Project”-> “Properties” เลือก “Linker” ในแท็บ -> “Input” และหัวข้อ -> “Additional Dependencies:” จากนั้นเพิ่ม path ซึ่ง path ที่จำเป็นในโครงการนี้ คือมีการโหลดไลบรารี คือ

cxcore.lib, cv.lib, highgui.lib, cvaux.lib และ cvcam.lib



รูปที่ 3.11 ตัวอย่างการเพิ่มไลบรารีในโปรเจก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. การเข้าถึงภาพในวิดีโอ ( Image Access )

ในการเข้าถึงภาพนั้น จะใช้ IplImage class ของไลบรารี ซึ่งเป็นชนิดของภาพที่ใช้ใน IPL และ OpenCV API นับว่าเป็นฟังก์ชันพื้นฐานที่ใช้สำหรับการเก็บชนิดของข้อมูลภาพ ถูกเก็บไว้ใน ipl.h ในไลบรารี OpenCV นั้นเสนอให้ผู้พัฒนาได้สามารถสร้างฟังก์ชันได้โดยง่ายและสะดวก ซึ่งโครงสร้างของ IplImage มีรูปแบบดังนี้คือ

```
typedef struct _IplImage
{
    int nSize;      /* sizeof(IplImage) */
    int ID;        /* version (=0)*/
    int nChannels;  /* Most of OpenCV functions support 1,2,3 or 4 channels */
    int alphaChannel; /* ignored by OpenCV */
    int depth;     /* pixel depth in bits: IPL_DEPTH_8U, IPL_DEPTH_16U,
                    IPL_DEPTH_16S, IPL_DEPTH_32S, IPL_DEPTH_32F and IPL_DEPTH_64F are supported */
    char colorModel[4]; /* ignored by OpenCV */
    char channelSeq[4]; /* ditto */
    int dataOrder;    /* 0 - interleaved color channels, 1 - separate color channels.
                    cvCreateImage can only create interleaved images */
    int origin;       /* 0 - top-left origin,
                    1 - bottom-left origin (Windows bitmaps style) */
    int align;        /* Alignment of image rows (4 or 8).
                    OpenCV ignores it and uses widthStep instead */
    int width;        /* image width in pixels */
    int height;       /* image height in pixels */
    struct _IplROI *roi; /* image ROI. when it is not NULL, this specifies image region to process */
    struct _IplImage *maskROI; /* must be NULL in OpenCV */
    void *imageId;    /* ditto */
    struct _IplTileInfo *tileInfo; /* ditto */
    int imageSize;    /* image data size in bytes
                    (=image->height*image->widthStep
                    in case of interleaved data)*/
    char *imageData; /* pointer to aligned image data */
    int widthStep;   /* size of aligned image row in bytes */
    int BorderMode[4]; /* border completion mode, ignored by OpenCV */
    int BorderConst[4]; /* ditto */
    char *imageDataOrigin; /* pointer to a very origin of image data
                    (not necessarily aligned) -
                    it is needed for correct image deallocation */
}
IplImage;
```

#### รูปที่ 3.12 โครงสร้างของ IplImage ในไลบรารี OpenCV

ในการเข้าถึงวิดีโอ นั้น ทาง OpenCV จะมีฟังก์ชัน โดยเริ่มจากการเปิดไฟล์วิดีโอคือ

```
CvCapture *input_video = cvCaptureFromFile("filename.avi");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้น ในการ access เข้าถึงเฟรมวิดีโอและการ get property ของวิดีโอ ประกอบด้วย หมายเลขเฟรม , ขนาดของเฟรม เป็นต้น

```
cvQueryFrame( input_video );
IplImage *Getframe = cvQueryFrame( input_video );
long Number_fame = cvSetCaptureProperty
                    (input_video,CV_CAP_PROP_FRAME_COUNT);
CvSize frame_size;
Frame_size.height =
cvGetCaptureProperty(input_video,CV_CAP_PROP_FRAME_HEIGHT);
```

### 5. การแสดงผลวิดีโอ

เนื่องจากทางผู้พัฒนานั้นต้องการที่จะให้ภาพวิดีอนั้นขึ้นในตัว Application โดยไม่ใช่ฟังก์ชันการสร้างหน้าต่างใหม่ แต่จะเกิดปัญหาคือ ในการเปิดไฟล์แต่ละไฟล์นั้น ขนาดของวิดีโอ มีขนาดที่แตกต่างกัน แต่ต้องการแสดงผลในขนาดในกรอบที่ถูกกำหนดไว้บน Application ซึ่งขนาดของวิดีโอที่มีขนาดใหญ่เกินไปนั้น ก็ไม่สามารถแสดงผลได้ และ ขนาดที่เล็กก็เกินไป ก็ภาพที่ออกมา ก็จะเล็กมาก

ดังนั้นผู้พัฒนาโปรแกรมจึงเสนอแนวทางเป็นการสร้างฟังก์ชันขึ้นมาปรับขนาดภาพที่ได้ get มาจากวิดีโอเทียบอัตราส่วนแล้วนำมาแสดงผลบนหน้าต่างของโปรแกรม ได้ตั้งชื่อว่า DrawFunction(); โดยมี code แสดงไว้ด้านล่างนี้

```
void CMKAppDlg::DrawFunction ( HDC hDC, BYTE *pImage, LONG lDestWidth, LONG lDestHeight,
LONG lSrcWidth, LONG lSrcHeight, WORD sBitCount, DWORD biCompression)
{
```

```
    HBITMAP          hDIB;
    HDC              hMemDC;
    HPEN            hPen;
    LPBITMAPINFO    pBitmapInfo;
    if ( sBitCount >= 16 ) {
        pBitmapInfo = ( LPBITMAPINFO ) new unsigned char [ sizeof ( BITMAPINFO ) ];
    } else
    {
        pBitmapInfo = ( LPBITMAPINFO ) new char[ sizeof ( BITMAPINFOHEADER ) +
            ( 1 << sBitCount ) * sizeof ( RGBQUAD ) ]; //GrayTalbe[256]
        ::memcpy ( pBitmapInfo->bmiColors, &GreyTable4[0], sizeof ( GreyTable4 ) );
    }
    //pBitmapInfo->bmiHeader = BitmapInfo.bmiHeader;
    pBitmapInfo->bmiHeader.biSize          = sizeof ( BITMAPINFOHEADER );
    pBitmapInfo->bmiHeader.biWidth       = lSrcWidth;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pBitmapInfo->bmiHeader.biHeight          = lSrcHeight;
// change it when the display invert (-lSrcHeight)
pBitmapInfo->bmiHeader.biPlanes          = 1;
pBitmapInfo->bmiHeader.biBitCount        = sBitCount;
pBitmapInfo->bmiHeader.biCompression     = biCompression;
pBitmapInfo->bmiHeader.biSizeImage       = 0;
pBitmapInfo->bmiHeader.biXPelsPerMeter   = 0;
pBitmapInfo->bmiHeader.biYPelsPerMeter   = 0;
pBitmapInfo->bmiHeader.biClrUsed         = 0;
pBitmapInfo->bmiHeader.biClrImportant    = 0;

DOUBLE dDestRatio      = ( double ) lDestWidth / ( double ) lDestHeight;
DOUBLE dSrcRatio       = ( double ) lSrcWidth / ( double ) lSrcHeight;

DOUBLE dZoom;
LONG   lNewDestWidth;
LONG   lNewDestHeight;

if ( dDestRatio > dSrcRatio )
{
    //shrink by height
    dZoom = ( double ) lDestHeight / ( double ) lSrcHeight;
}
else
{
    //shrink by width
    dZoom = ( double ) lDestWidth / ( double ) lSrcWidth;
}

lNewDestWidth = ( LONG ) ( 0.98 * dZoom * lSrcWidth );
lNewDestHeight = ( LONG ) ( 0.98 * dZoom * lSrcHeight );

INT dx = ( lDestWidth - lNewDestWidth ) / 2;
INT dy = ( lDestHeight - lNewDestHeight ) / 2;

hMemDC = ::CreateCompatibleDC ( hDC );
hDIB = ::CreateDIBitmap( hDC, &pBitmapInfo->bmiHeader, CBM_INIT,
    pImage, pBitmapInfo, DIB_RGB_COLORS );
::SelectObject ( hMemDC, hDIB );
SetStretchBltMode ( hDC, HALFTONE );
StretchBlt ( hDC, dx, dy, lNewDestWidth, lNewDestHeight,
    hMemDC, 0, 0, lSrcWidth, lSrcHeight,
    SRCCOPY );

hPen = ::CreatePen ( PS_SOLID, 1, RGB ( 0, 0, 255 ) );
::SetBkMode ( hDC, TRANSPARENT );
::SelectObject ( hDC, ::GetStockObject ( NULL_BRUSH ) );
::SelectObject ( hDC, hPen );
::Rectangle ( hDC, 0, 0, lDestWidth, lDestHeight );

::DeleteObject ( hPen );
::DeleteObject ( hDIB );
::DeleteObject ( hMemDC );

delete [ ] pBitmapInfo;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. การพัฒนาโปรแกรมในรูปแบบ Multithread

ชนิดของเธรด ซึ่งเราสามารถแบ่งเป็นสองชนิดภายใต้สถาปัตยกรรม MFC ได้แก่ เธรดงาน (Worker Thread) เป็นเธรดที่ใช้ในการคำนวณ หรือทำงานในลักษณะเป็นงานรองจากโปรแกรมหลักซึ่งจะไม่มีเมสเสจวินโดว์เข้ามาเกี่ยวข้อง แต่สามารถใช้เมสเสจที่เรากำหนดขึ้นมาได้ และเธรดที่ใช้สร้างส่วนติดต่อกับผู้ใช้ (User Interface Thread) เป็นเธรดที่ใช้ในการสร้างวินโดว์เพื่อสร้างส่วนติดต่อกับผู้ใช้งาน ดังนั้นจึงมีเมสเสจวินโดว์มากมาย โดยในโครงงานนี้ ได้แบ่งเธรดดังรูปด้านล่าง (รูปที่ 3.13)



รูปที่ 3.13 การแตกเธรดของโปรแกรมค้นหาภาพโดยอ้างอิงจากผู้ใช้

การประกาศ Handle ของเธรด โดยในโครงงานมีการประกาศ 2 ตัว คือ

hGrabImageThreadHandle และ hThreadDoneEvent

```

DWORD WINAPI
HANDLE
HANDLE
DWORD
GrabImageThread( LPVOID lpParameter);
hGrabImageThreadHandle;
hThreadDoneEvent;
dwGrabImageThreadHandleID;

```

จากนั้นก็ใช้ฟังก์ชันการสร้างเธรด คือ CreateThread(); ขึ้นมา

```

if ( capture )
{
    hGrabImageThreadHandle = ::CreateThread( NULL, 0 , GrabImageThread, NULL, 0,
&dwGrabImageThreadHandleID );
    if ( hGrabImageThreadHandle == NULL )
    {
        ::MessageBox( NULL, "Could not make a thread", NULL, MB_OK);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

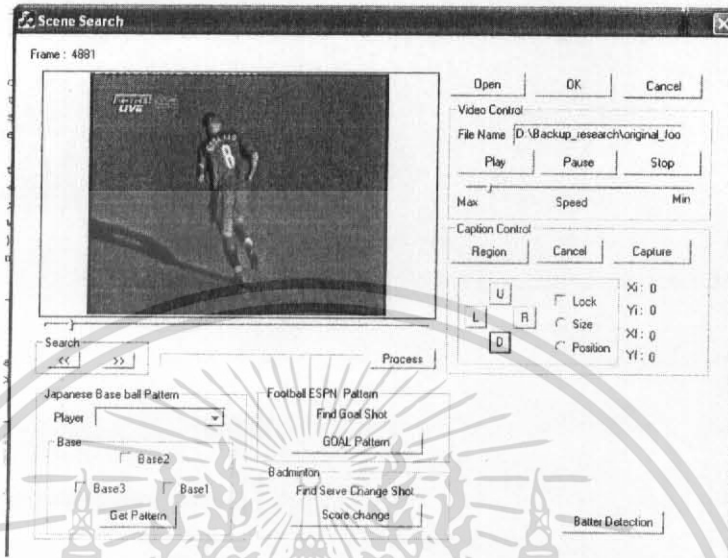
```

DWORD WINAPI GrabImageThread( LPVOID lpParameter)
{
    BOOL Loop = TRUE;
    for (;;)
    {
        // display
        while ( !bDataPause && ( NumFrame > nCurrentFrame ) )
        {
            cvSetCaptureProperty( capture,
CV_CAP_PROP_POS_FRAMES, (double) nCurrentFrame );
            bDataOK = TRUE;
            frame = cvQueryFrame( capture );
            if( !frame )
                break;
            ::InvalidateRect ( hAppDlg, &DisplayRect, FALSE );
            int a = (int) cvGetCaptureProperty( capture,
CV_CAP_PROP_POS_FRAMES );
            nCurrentFrame++;
            CString text;
            text.Format ( "%d", a );
            ::SetWindowText ( hSlideNumber, text );
            pslider->SetPos ( a );
            ::Sleep ( Speedvalue );
        }
        if ( NumFrame == nCurrentFrame ) nCurrentFrame--;
        if ( bDataPause ) {
            cvSetCaptureProperty( capture,
CV_CAP_PROP_POS_FRAMES, (double) nCurrentFrame );
            frame = cvQueryFrame( capture );
            if( !frame )
                break;
            ::InvalidateRect ( hAppDlg, &DisplayRect, FALSE );
            ::Sleep ( Speedvalue );
        }
    }
    cvReleaseCapture( &capture );
    return TRUE;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7. หน้าตา User Interface ของตัวโปรแกรม



รูปที่ 3.14 การพัฒนาโปรแกรมค้นหาฉาก

1. ตัวแสดงผลวิดีโอ (Display) การแสดงผลนั้น จะสามารถแสดงผลได้เฉพาะ ไฟล์ \*.AVI ได้เท่านั้น และ ตัวแสดงผลจะทำการปรับขนาดเป็นอัตราส่วนของขนาดไฟล์จริงๆ แล้วนำมาแสดงผลบนหน้าจอ ทำให้ไม่ว่าไฟล์จะมีขนาดใหญ่ หรือเล็ก ก็สามารถที่แสดงผลได้ และมี Time Line ด้านล่างตัวแสดงผล
2. เครื่องมือควบคุมวิดีโอ (Video Control) เป็นเครื่องมือ ควบคุมความเร็วในการแสดงผล (Speed) การเล่น(Play) การหยุด (Stop & Pause) โดยความเร็ว นั้น จะขึ้นอยู่กับประสิทธิภาพของเครื่อง โดยตรง ดังนั้นเครื่องที่มีความเร็วสูง ก็จะ สามารถ เร่งสปีด ได้มากขึ้น
3. เครื่องมือออกแบบรูปแบบ โดยผู้ใช้ (เฉพาะวิธีที่ 1) (Caption Control) & (Pattern) โดยจะจำลองการใช้รีโมท กับ โทรทัศน์ไว้ในส่วนนี้ และตัวโปรแกรมสามารถใช้ได้เพียงวิธีที่ 1 เท่านั้น ในวิธีที่สองนั้น ยังอยู่ในขั้นตอนการติดตั้งอยู่
4. เครื่องมือการค้นหา (Search) หลักการผ่านกระบวนการ การค้นหาฉาก ผู้ใช้สามารถที่จะกระโดด ไปฉากที่ต้องการได้เลขทันที โดยใช้เครื่องมือค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# การค้นหา Superimposed Text บนวิดีโอ

### 4.1 เทคนิคการค้นหา Superimposed text บนวิดีโอแบบอัตโนมัติ

จุดประสงค์หลักของการทำบทสรุปของวิดีโอ คือ การบีบอัดหัวข้อที่ไม่สำคัญทิ้งไปที่มีผลต่อการจัดเก็บไฟล์วิดีโอ นั้น จะเห็นได้ชัดอย่างยิ่ง ในข่าว และ เกมกีฬา เพราะผู้ใช้นั้นจะให้ความสำคัญกับเฉพาะเหตุการณ์ที่ตนสนใจเท่านั้น โดยจะไม่ต้องการที่จะใช้เวลาไปกับการรอกอยฉากที่ตนเองอยากชม ดังนั้นการทำบทสรุปของวิดีโอ ( Video summarization ) นั้นจะมีผลต่อการค้นหา ซึ่งพบว่า ข้อความที่ขึ้นมา นั้น ก็นับว่าเป็นการอธิบายเหตุการณ์ของวิดีโอในช่วงเวลานั้นเป็นอย่างดี เช่น เกมกีฬาฟุตบอล จะมี Superimposed text บอกคะแนนการทำประตู เหตุการณ์ต่างๆ เพื่อจะดึงดูดให้เกมกีฬานั้นดูน่าสนใจมากยิ่งขึ้น



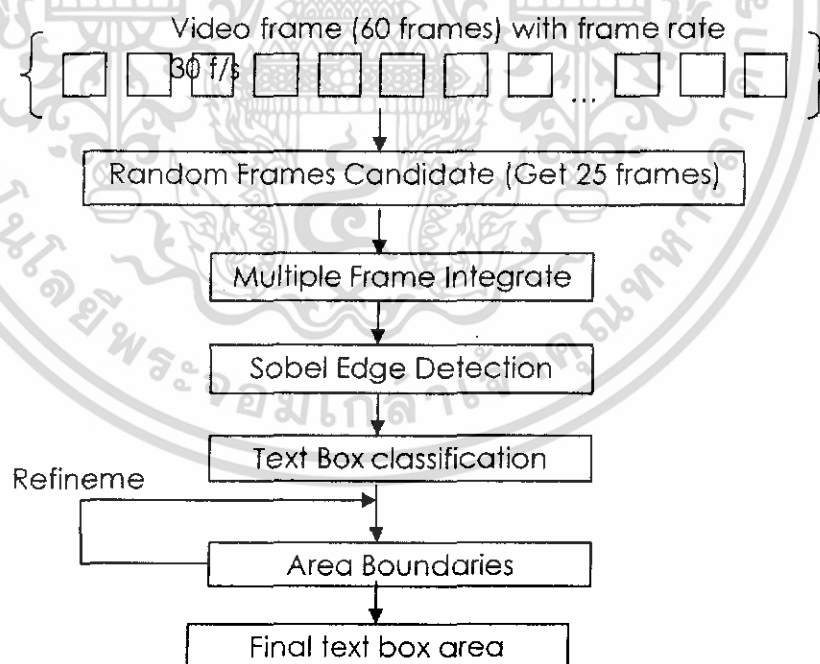
รูป 4.1 ภาพฉากต่างๆ ที่มี Superimposed Text ปรากฏ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในขั้นตอนการค้นหา Superimposed Text นั้น จะเริ่มจากการ ค้นหา (Detection) พิกัด ตำแหน่ง (Localization) และ ตัดทอน (Extraction) ซึ่ง การค้นหานี้ จะเป็นการแบ่งอย่างคร่าวๆ ว่า พิกเซลไหนคือพิกเซลที่อยู่ในขอบข่าย การบอกพิกัดตำแหน่ง เป็นการพิจารณาเส้นขอบของตัวหนังสือ ในการตัดทอนนั้น เป็นขั้นตอนการฟิลเตอร์เอาพิกเซลพื้นหลังจากออก ทำให้เหลือเพียงแค่ พื้นที่ที่มีแต่ ตัวหนังสือ จะสังเกตได้ว่า การตรวจหา superimposed text ในวิดีโอ นั้น จะมีสิ่งที่เป็นผลกระทบหลาย อย่าง เช่น คุณภาพของวิดีโอเอง ใน การส่งสัญญาณทีวี นั้นอุดมไปด้วย noise ซึ่งยากแก่การทำ image processing แม้ว่าจะใช้วิธีการลด noise ก็จะทำให้ภาพนั้น เบลอได้ , การที่มีพื้นหลัง กับ superimposed text ที่ยากจะแยกออกจากกัน , ความละเอียดต่ำ, ขนาดของตัวอักษร ซึ่งสิ่งเหล่านี้ ทำให้ยากแก่การ ตรวจหา superimposed text

#### 4.1.1 ภาพรวมเทคนิค การค้นหา Superimposed Text

จากการสังเกตของผู้จัดทำ พบว่า “Superimposed text นั้นจะมีการแสดงบนวิดีโออย่างน้อย 2 วินาที” จากการสังเกตนี้ ทางผู้จัดทำได้ออกแบบ Algorithm สำหรับการค้นหา superimposed text โดยเขียนออกเป็นแผนภาพ ดังรูปที่ 4.2



รูปที่ 4.2 ขั้นตอนการค้นหา Superimposed Text

#### 4.1.2 Random Frame Candidate

จากข้อสังเกตนั้น จะวิเคราะห์เป็นช่วง ๆ ของวิดีโอ โดยวิเคราะห์ทีละ 2 วินาที โดยมี frame rate 30 frames/sec ดังนั้น จะวิเคราะห์ทีละ 60 เฟรม จากนั้น จะผ่านกระบวนการ Random Frames Candidate เป็นการเลือกมาเฟรมแบบ Random มาเพียง 25 เฟรมมาเท่านั้น เพราะถ้าวิเคราะห์ทั้งหมด 60 เฟรม จะทำให้เปลืองทรัพยากรเครื่องโดยเปล่าประโยชน์ และ ผลที่ออกมานั้นก็ไม่ต่างกัน

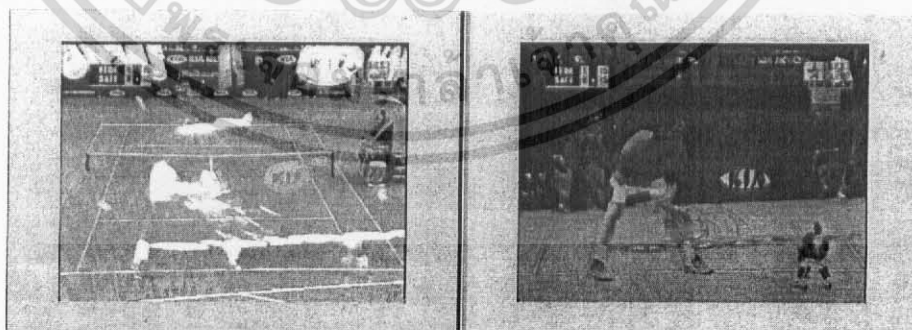
#### 4.1.3 Multiple Frame Integration

ขั้นตอนนี้ เป็นการรวมเฟรมจากทั้งหมด 25 เฟรม เหลือเพียง 2 เฟรม ซึ่งเฟรมที่จะวิเคราะห์ นั้น จะต้องเป็นภาพ gray scale ทั้งหมด โดยทำการแปลงจากภาพสี เป็นภาพ gray scale จากนั้น นำเฟรมทั้งหมดมาคำนวณด้วยสมการ (4.1) และ (4.2) ซึ่งผลลัพธ์ ออกมาจะเหลือภาพเพียง 2 ภาพ คือ ภาพค่าสูงสุด (Max Image) และ ภาพค่าต่ำสุด (Min Image)

$$MinIMAGE_i(x, y) = \min_{j \in C_i} (p_j(x, y)) \quad (4.1)$$

$$MaxIMAGE_i(x, y) = \max_{j \in C_i} (p_j(x, y)) \quad (4.2)$$

โดยที่  $C_i$  เป็นเฟรมที่ได้เลือกไว้ (Candidate frame =  $i, \dots, i+24$ )  $P(x, y)$  คือ ค่าของภาพ gray scale จากสมการ นั้นแสดงให้เห็นว่า มีการเข้าถึงภาพทีละ 25 เฟรม โดยจะทำการวิเคราะห์ แต่ละพิกเซลของแต่ละเฟรม โดยจะหาค่าสูงสุด และ ค่าต่ำสุดของแต่ละตำแหน่ง แล้ว ออกมาสร้างภาพใหม่ ดังภาพที่ 4.3 ด้านล่าง



รูปที่ 4.3 ภาพค่าสูงสุด และ ภาพค่าต่ำสุด หลังจากผ่าน Multiple Frame Integration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.4 Sobel Edge Detection

การหาขอบภาพ เป็นการหาขอบเขตของวัตถุภายในภาพหรือเป็นการดึงลักษณะโครงร่างที่เด่นของวัตถุออกมา การหาขอบภาพโดยใช้ข้อมูลพิกัดอันดับที่หนึ่งเป็นการแปลงเกรเดียนต์แบบไม่ต่อเนื่องบนข้อมูลภาพเชิงตัวเลข เนื่องจากการหาขอบภาพเป็นการประมวลผลแบบไม่ต่อเนื่อง ดังนั้นจึงต้องใช้ข้อมูลพิกัดย่อยแบบไม่ต่อเนื่องตามทิศทางที่ตั้งฉากกับแกน  $x$  และแกน  $y$  ซึ่งสามารถกำหนดได้ดังนี้

$$\begin{aligned}\nabla_x g(x, y) &= g(x, y) - g(x-1, y) \\ \nabla_y g(x, y) &= g(x, y) - g(x, y-1)\end{aligned}\quad (4.3)$$

ส่วนขนาดโดยประมาณของเกรเดียนต์  $g(x, y)$  สามารถกำหนดได้ดังนี้

$$|\nabla g(x, y)| = (\nabla_x g(x, y))^2 + (\nabla_y g(x, y))^2 \quad (4.4)$$

เพื่อให้ง่ายต่อการคำนวณในบางครั้งจะให้ขนาดของเกรเดียนต์มีค่าประมาณขนาดของเกรเดียนต์ตามทิศทางที่ตั้งฉากกับแกน  $x$  และแกน  $y$  รวมกัน คือ

$$|\nabla g(x, y)| = \nabla_x g(x, y) + \nabla_y g(x, y) \quad (4.5)$$

การหาขอบภาพโดยใช้เกรเดียนต์ในทางปฏิบัติจะมีลักษณะที่แตกต่างกันไป แต่ได้เลือกวิธีของ Sobel สามารถเขียนเป็นสมการได้ดังนี้

$$|\nabla g(x, y)| = \left( \begin{array}{l} \left[ \begin{array}{l} g(x, y) + 2g(x, y+1) + g(x, y+2) - \\ (g(x+2, y) + 2g(x+2, y+1) + g(x+2, y+2)) \end{array} \right] + \\ \left[ \begin{array}{l} g(x, y) + 2g(x+1, y) + g(x+2, y) - \\ (g(x, y+2) + 2g(x+1, y+2) + g(x+2, y+2)) \end{array} \right] \end{array} \right) \quad (4.6)$$

จากสมการของ (7) นั้นสามารถนำมาเขียนในรูปของวินโดว์ ได้ คือ

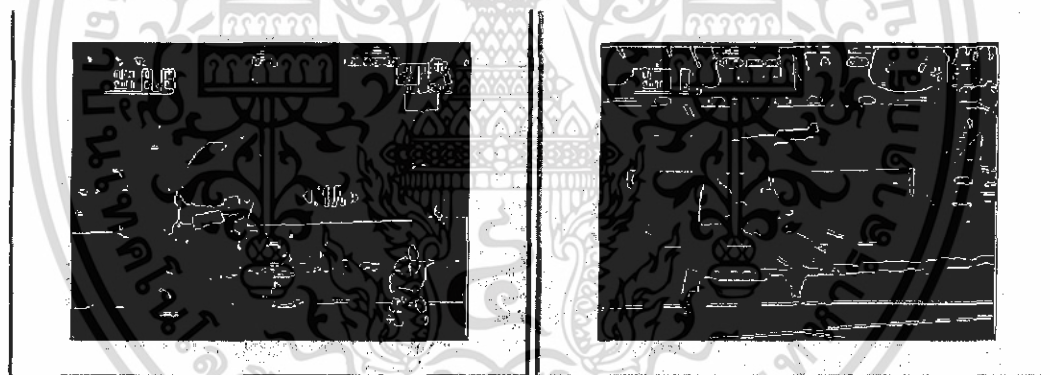
$$W_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำวินโดว์  $w$  ของแต่ละวิธีมาหาขอบภาพ จะต้องนำวินโดว์ทุกตัวของวิธีนั้นๆ มาทำการคอนโวลูชัน (Convolution) หรือการประสานกับข้อมูลภาพ ซึ่งสามารถแสดงในรูปของสมการได้ดังนี้

$$h_k(x, y) = \sum_i \sum_j w_k(i, j)g(x+i-1, y+j-1) \quad (4.8)$$

โดย  $h_k(x, y)$  คือจุดภาพ ณ ตำแหน่ง  $x, y$  ที่ผ่าน การคอนโวลูชันแล้วด้วยวินโดว์ที่  $k$   $w_k(i, j)$  คือสมาชิกตัวที่  $i, j$  ของวินโดว์ตัวที่  $k$  เมื่อดำเนินการคอนโวลูชันระหว่างข้อมูลภาพกับวินโดว์ของวิธีที่จะนำมาหาขอบภาพครบทุกวินโดว์แล้วก็นำ  $h_k(x, y)$  สำหรับ  $k$  ทุกตัวมาทำการเปรียบเทียบกัน ณ พิกัด  $x, y$  เดียวกันเพื่อเลือกค่าที่สูงที่สุด และนำค่าที่ได้มาเทียบกับค่าขีดเริ่มเปลี่ยน (Threshold) คือ ถ้าน้อยกว่าค่าขีดเริ่มเปลี่ยนจุดภาพนั้นก็ไม่ใช่ขอบภาพ



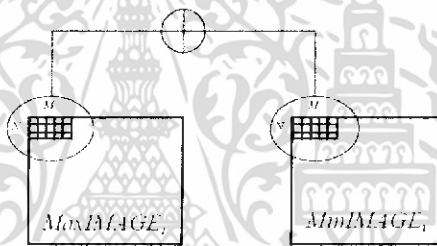
รูปที่ 4.4 ภาพหลังจากการผ่าน Sobel Edge Detection ของ MaxIMAGE & MinIMAGE

หลังจากสร้างภาพขึ้นมาได้ 2 ภาพ : MinIMAGE and MaxIMAGE จะใช้ Sobel edge detection กับภาพ MinIMAGE and MaxIMAGE และพิจารณาภาพที่มี Sobel edge point น้อยๆ จะเลือก Sobel edge operator 4 ทิศทางในการกำหนดขอบ ถ้าค่า maximal gradient ณ จุดซึ่งมีค่ามากกว่าค่า threshold มันจะสามารถพิจารณา ณ จุดขอบนั้น ในรูป 4.4 เป็นการแสดงตัวอย่างหาขอบด้วยวิธีของ Sobel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

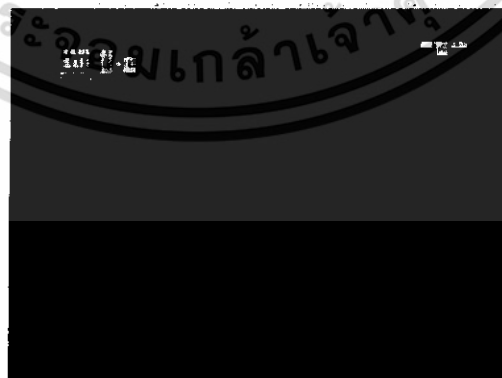
#### 4.1.5 Text Box Classification

ในขั้นตอนการจัดแบ่งประเภทของข้อความ จะใช้ขนาด  $M \times N$  ( $20 \times 10$  pixel) เป็นวินโดว์ สำหรับตรวจรายละเอียดของการรวมกันของภาพ และจัดกลุ่มในแต่ละวินโดว์ว่าเป็น text หรือ non-text ซึ่งในการกำหนดค่า  $M$  และ  $N$  นั้นจะกำหนดให้ค่า  $M > N$  เพราะว่าคุณลักษณะของวินโดว์นั้นจะมีค่าแนวอนมากกว่าเสมอ ซึ่งถ้าค่าความเข้มของขอบในวินโดว์ ( Edge density ) มีขนาดใหญ่กว่าค่า threshold ที่กำหนดไว้ล่วงหน้าซึ่งจะมีการแบ่งประเภทของ text มิฉะนั้นจะมีการแบ่งประเภทของ non-text ซึ่งในการทดลองค่า threshold ที่ใช้คือค่า 0.70 โดยจะไม่ย้ายตำแหน่งของวินโดว์ทุก ๆ หนึ่งพิกเซล เพราะว่ามันจะสิ้นเปลืองเวลาและไม่จำเป็นจะต้องพิจารณาการแลกเปลี่ยนค่าระหว่างความแม่นยำกับความเร็ว เลื่อนวินโดว์ใหม่แนวตั้ง 10 พิกเซล และ แนวอน 5 พิกเซล ณ เวลานั้นๆ



รูปที่ 4.5 วิธีการ Text box classification

จากภาพที่ 4.5 นั้นจะมีการใช้เชื่อมความสัมพันธ์ของ MaxIMAGE และ MinIMAGE ด้วยการกระทำแบบ AND ซึ่งจะช่วยลดข้อผิดพลาดในแต่ละวินโดว์ด้วย และ อีกทั้งยังเป็นการรวมภาพเข้าด้วยกันให้เหลือเพียงภาพเดียว ซึ่งผลลัพธ์ออกมา ได้ดังภาพ 4.6

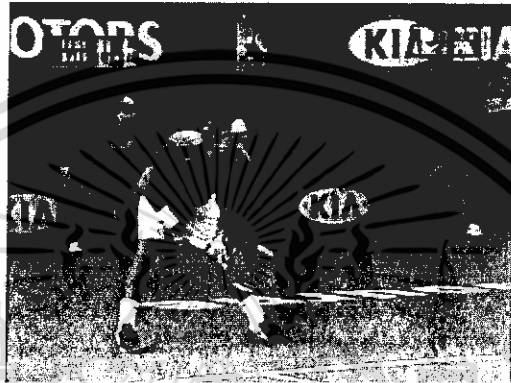


รูปที่ 4.6 ผลลัพธ์จากการทำ Text box classification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.6 Area boundaries

เป็นการขยายขนาดของพื้นที่ ที่กำหนดเป็น text เพราะว่า พื้นที่ที่กำหนดได้นั้น ที่ขอบ จะไม่สามารถตรวจหาได้ เพราะ พื้นหลังนั้นมีสีที่ใกล้เคียงกับตัวพื้นที่ที่เป็น text มาก ทำให้ต้องขยาย ขอบ แล้วทำกับ กำหนดใส่เส้นแสดงว่า เป็นพื้นที่ของ superimposed text ดังภาพที่ 4.7



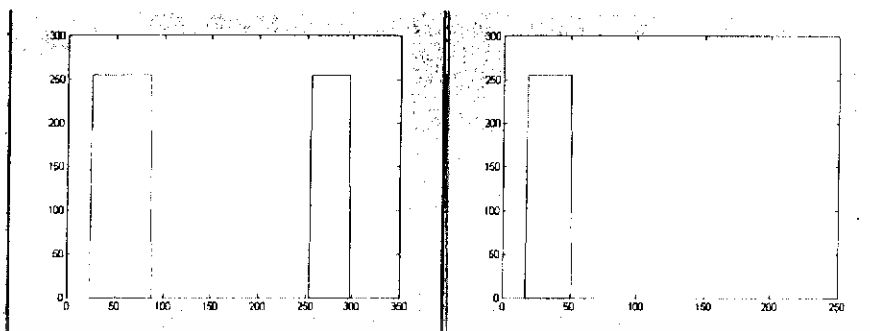
รูปที่ 4.7 ผลลัพธ์จากการทำ Area boundaries



รูปที่ 4.8 ผลลัพธ์จากการทำ Area boundaries ในวิดีโออื่นๆ

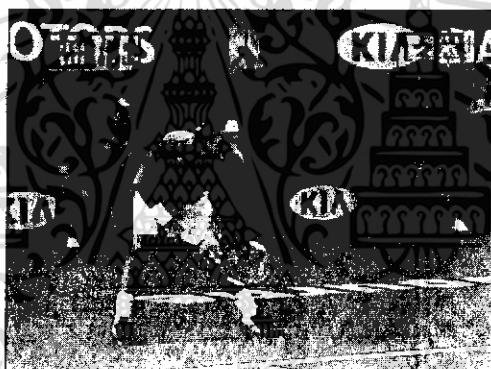
ขอบเขตของพื้นที่ที่เป็น text นั้น ไม่ใช่ภาพสี่เหลี่ยม ดังนั้น จะต้องทำการ Refinement หรือ ปรับปรุงกรอบของพื้นที่ที่ต้องการ โดยใช้วิธีการ Projection ดังกราฟที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 กราฟที่เกิดจากการทำ projection

โดยจะทำการฉายในแนวตั้ง และ แนวนอน โดยนำค่าแต่ละพิกเซลมาบวกกันในแต่ละแกน ทำให้สามารถหาจุดตัดของ ดังภาพที่ 4.10



รูปที่ 4.10 text box หลังจากการทำ refinement

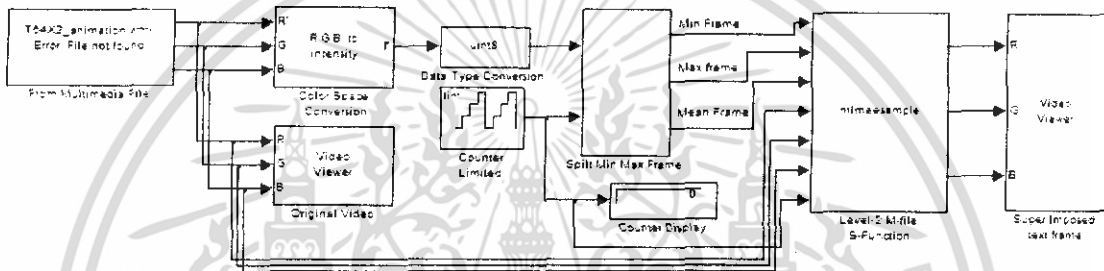
#### 4.2 การจำลองอัลกอริทึมบน Simulink ของ MATLAB

ในทั้งข้อที่ผ่านมานั้น ได้มีการอธิบายหลักการค้นหา Superimposed text และได้แสดงผลลัพธ์ของการทำงานในลำดับขั้นต่างๆ ไปแล้ว ทางผู้พัฒนาจะนำผลลัพธ์ต่างๆเหล่านี้ มาแสดงการจำลองการทำงานใน Simulink ซึ่งจะสามารถแสดงผลลัพธ์ในระยะเวลาที่ต้องการทดสอบได้ จากการออกแบบและทดสอบผลในส่วน Simulink นี้ จะสามารถนำไปออกแบบระบบฮาร์ดแวร์บน FPGA ด้วย Xilinx System Generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการสร้างโมเดลขึ้นมาตามรูปที่ 4.11 โดยจะใช้โปรแกรม Simulink ซึ่งเป็นส่วนหนึ่งของโปรแกรม MATLAB เข้ามาทำการออกแบบระบบนี้

ในการออกแบบนั้น ได้มีการตั้งค่า sampling time ของทั้งระบบไว้ที่ 1/25 หรือ 25 ครั้งต่อวินาที (ค่าเทียบเท่ากับจำนวนเฟรมต่อวินาทีของระบบ PAL ) เพื่อให้เข้ากันกับ ไฟล์วิดีโอที่นำมาทดสอบได้ (เฟรมเรท 25 เฟรมต่อวินาที) และง่ายต่อการคำนวณเวลาโดยรวม และได้ตั้งชนิดของข้อมูลที่จะใช้ส่งรับในระบบทั้งหมดให้เป็น UNSIGNED INT 8 BITS ทำให้การรับส่งค่าในระบบเป็นไปในแนวทางเดียวกัน เพื่อที่จะสามารถหลีกเลี่ยงปัญหาการขัดแย้งกันของชนิดข้อมูลที่จะเกิดขึ้นใน Simulink ได้



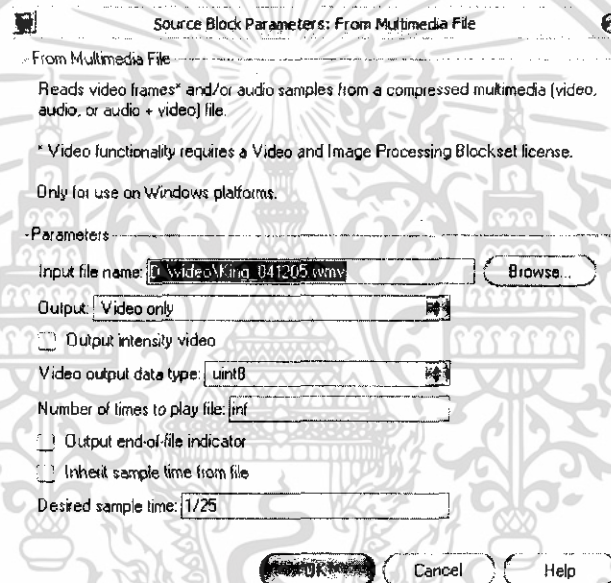
รูปที่ 4.11 Model การจำลองระบบค้นหาพื้นที่ Superimposed text บน Simulink

ตารางที่ 4.1 เขตค่าพื้นฐานของโมเดลใน Simulink

Block	Library	Quantity
From Multimedia File	Video and Image Processing Blockset / Sources	1
Color Space Conversion	Video and Image Processing Blockset / Conversions	1
Video Viewer	Video and Image Processing Blockset / Sinks	2
Data Type Conversion	Simulink / Signal Attributes	1
Counter Limited	Simulink / Sources	1
In1	Simulink / Ports & Subsystems	4
Out1	Simulink / Ports & Subsystems	6
Memory	Simulink / Discrete	50
Display	Simulink / Sinks	1
Subsystem	Simulink / Ports & Subsystems	2
Level-2 M-file S-function	Simulink / User-Defined Functions	1

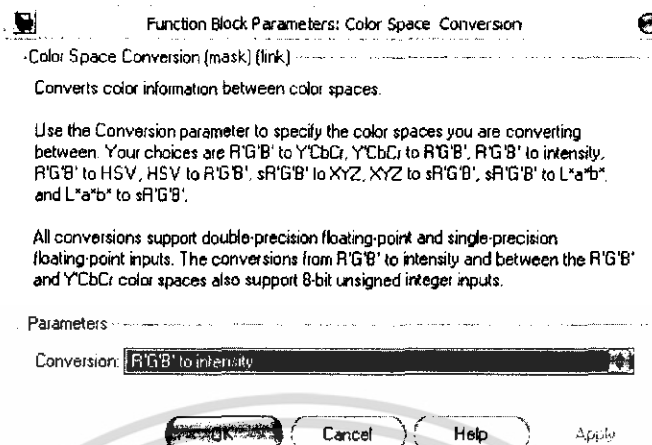
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนถัดมาเลือกไฟล์ที่ต้องการในบล็อกพารามิเตอร์ From Multimedia File ในช่อง Input file name โดยสามารถเลือกไฟล์วิดีโอใดๆที่ตัวเครื่องสนับสนุน ออกมาคำนวณใน Simulink ได้ ทางผู้พัฒนานั้นได้ทำการลงโปรแกรม “K-Lite Codec Pack” ซึ่งเป็นโปรแกรมที่รวบรวม encoder สำหรับการรับชมไฟล์วิดีโอรูปแบบต่างๆที่มีอยู่ในอินเทอร์เน็ต หลังจากนั้น ให้เลือกช่อง Output: เป็น Video only เพราะต้องการเฉพาะข้อมูลภาพมาใช้ในการคำนวณเท่านั้น จากนั้นให้ตั้งค่า Video output data type: เป็น UINT8 หรือก็คือ unsigned integer 8 bit ที่ได้กล่าวไว้ในตอนต้นของบทนั้นเอง ต่อมาในช่อง Number of times to play file: ในตั้งไว้เป็น inf (ไม่มีที่สิ้นสุด) และในช่องสุดท้าย Desired sample time: ให้ตั้งเป็น 1/25 ดังภาพที่ 4.12



รูปที่ 4.12 แสดงการตั้งค่าเริ่มต้นของ Block From Multimedia File

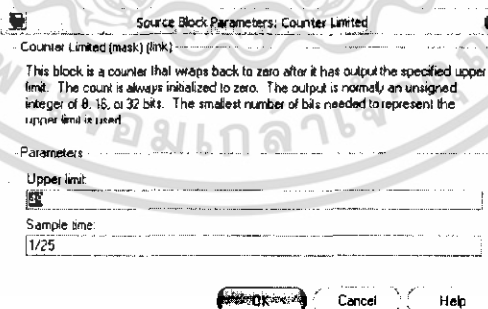
เมื่อเตรียมค่าไฟล์วิดีโอที่ต้องการจะทดสอบพร้อมแล้ว ต่อไปจะเป็นส่วนแปลงค่าของไฟล์วิดีโอ ที่ตอนนี้ยังแสดงอยู่ในรูปแบบ RGB อยู่ ให้กลายเป็นรูปแบบ grayscale โดยใช้บล็อก Color Space Conversion โดยให้ตั้งค่าในพารามิเตอร์เป็น R'G'B to intensity ตามรูปที่ 4.13



รูปที่ 4.13 แสดงการตั้งค่าเริ่มต้นของ Block Color Space Conversion

เมื่อได้ค่าของพัลส์วิดีโอออกมาเป็น grayscale แล้วนำไปเข้า subsystem “Split Min Max Frame” ในพอร์ตอินพุตที่ 1 ส่วนในพอร์ตอินพุตที่ 2 จะรับค่าจำลองเวลาขึ้นมา ในการทำ superimposed text นั้น ทฤษฎีได้ตั้งเวลาที่เหมาะสมของการที่ super imposed text จะปรากฏบนไฟส์ไว้ที่ 2 วินาที ในขณะที่ได้กำหนดให้พัลส์วิดีโอส่งข้อมูลออกมาทีละ 25 เฟรมต่อวินาที ดังนั้น ถ้าต้องการจะให้มีการทำงานในทุกๆ 2 วินาทีแล้ว จำเป็นจะต้องตั้งค่าไว้ที่ 2\*25 เฟรม นั่นคือ 50 เฟรม ซึ่งจะใช้บล็อก Counter Limited เป็นตัวคอยนับจำนวนเฟรมไปเรื่อยๆจนถึง 50 เฟรม จากนั้นจึงเริ่มต้นนับใหม่ ทำให้ได้รอบการทำงานในทุกๆ 2 วินาทีขึ้นมา จากบล็อก Counter Limited ให้ตั้งค่าพารามิเตอร์ไว้ดังนี้

- Upper limit: ให้ตั้งไว้เท่ากับ 49 (เพราะบล็อกนี้เริ่มนับขึ้นมาจากค่าศูนย์)
- Sample time: ให้ตั้งค่าไว้เท่ากับ 1/25 เหมือนกับบล็อกอื่นๆ



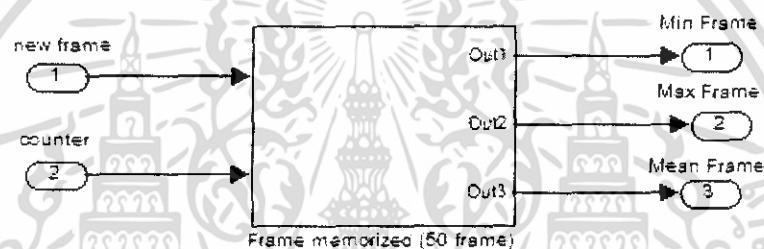
รูปที่ 4.14 แสดงการตั้งค่าเริ่มต้นของ block Counter Limited

จากนั้นให้สร้าง subsystem “Split Min Max Frame” ขึ้นมาโดยให้มีพอร์ตอินพุตเข้ามา 2 พอร์ต และมีพอร์ตเอาต์พุตออกมา 3 พอร์ต โดยกำหนดให้แต่ละพอร์ตรับค่าเข้ามามีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

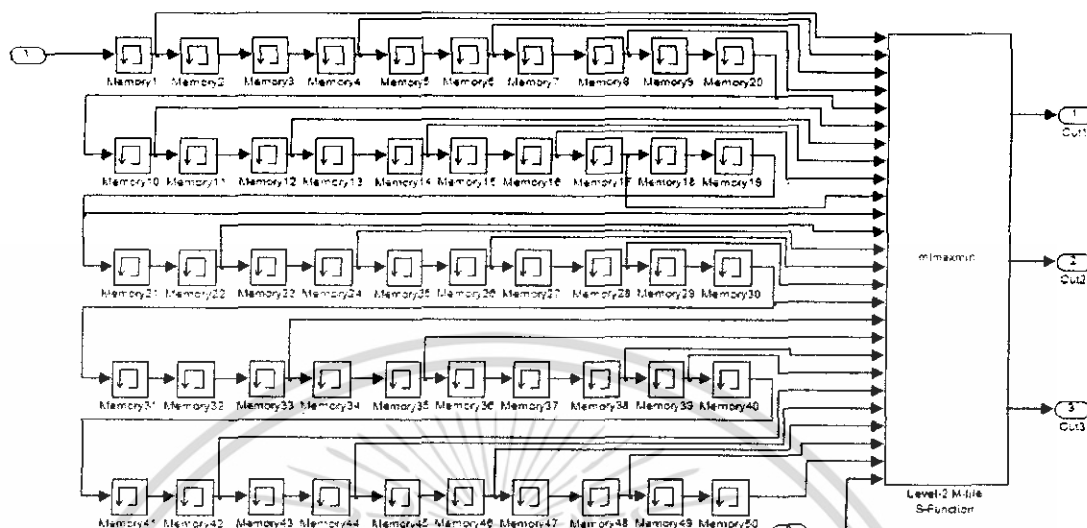
- พอร์ตอินพุต 1 : รับค่าเฟรมใหม่ที่มาจากไฟส์วิดีโอซึ่งได้แปลงเป็นสีแบบ grayscale ไว้
- พอร์ตอินพุต 2 : รับค่าจาก Counter Limited เข้ามาเพื่อคอยเช็คเวลาที่จะการคำนวณโดยรวม
- พอร์ตเอาต์พุต 1 : นำผลลัพธ์ของการรวมภาพด้วยการหา MinFRAME ออกมา
- พอร์ตเอาต์พุต 2 : นำผลลัพธ์ของการรวมภาพด้วยการหา MaxFRAME ออกมา
- พอร์ตเอาต์พุต 3 : นำผลลัพธ์ของการรวมภาพด้วยการหา MeanFRAME ออกมา

ให้สร้าง subsystem เข้าไปอีกชั้นหนึ่งใน subsystem ชั้นนี้ ชื่อ “Frame memorized (50 frame)” โดยสร้างให้มีพอร์ตอินพุตและเอาต์พุตจำนวนเท่ากับ subsystem “Split Min Max Frame” เพื่อความรวดเร็วในการแก้ไขเมื่อเกิดมีข้อผิดพลาดขึ้นมาใน subsystem “Frame memorized” ดังที่ได้แสดงไว้ในรูปที่ 4.11



รูปที่ 4.15 model sub-layer ย่อยของบล็อก Split Min/Max Frame

จากนั้นเข้าไปสู่ sub-layer “Frame memorized (50 frame)” ให้จัด model เป็นไปตามรูปที่ 3 โดยใช้บล็อก Memory ทั้งหมด 50 บล็อกเพื่อเก็บเฟรม “ทุกเฟรม” ในรอบ 2 วินาทีที่ออกมาจากไฟส์วิดีโอ จากนั้นเลือกเฟรมออกมาจากกลุ่มนี้ 25 เฟรม (ตามทฤษฎีควรจะใช่วิธีสุ่มเลือก แต่ผู้จัดทำพบว่าเป็นการสิ้นเปลืองเวลาการคำนวณของเครื่องมากเกินไป จึงตัดสินใจวิธีจิ้มเลือกออกมา 25 เฟรมแทน) นำมาเข้า S-function ที่ชื่อว่า mlmaxmin โดยคุณสมบัติของฟังก์ชันนี้มีดังนี้



รูปที่ 4.16 model sub-layer ย่อยของบล็อก Frame memorized (50 frame)

มีพอร์ตอินพุตทั้งหมด 26 พอร์ต โดยรับค่าเฟรม ณ เวลาต่างๆ 25 พอร์ต และ รับค่า counter อีก 1 พอร์ต มีพอร์ตเอาต์พุตทั้งหมด 3 พอร์ต โดยนำค่า min max mean ออกไปในแต่ละพอร์ตตามลำดับ โดย s-function นี้ทำงานตาม pseudo-code ดังต่อไปนี้

```

if InputPort(26).Data = 49
    for i = 1:25
        maxframe = max(maxframe , InputPort(i).Data)
        minframe = min(minframe , InputPort(i).Data)
        multimatrixmean(:,i) = InputPort(i).Data
    end
    for m = 1:res_m
        for n = 1:res_n
            meanframe(m,n)= mean(multimatrixmean(m,n,:))
        end
    end
end
    OutputPort(1).Data = minframe
    OutputPort(2).Data = maxframe
    OutputPort(3).Data = meanframe
end

```

หลังจากนั้น นำค่าเอาต์พุตออกไปยังตามพอร์ตที่ได้ตั้งไว้ในโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกระบวนการ superimposed text ตามที่ได้กล่าวไว้ จะยังเหลือการคำนวณ sobel edge detection, text box classification, area boundaries สามารถคำนวณกระบวนการทั้งสามนี้พร้อมกันในบล็อก S-function “mlmeesample” โดยนำค่าทั้งสามที่ได้จากข้อที่แล้ว ไปเข้าสู่ S-function “mlmeesample” ซึ่งมีคุณสมบัติการทำงานดังต่อไปนี้

มีพอร์ตอินพุตทั้งหมด 7 พอร์ต โดยรับค่าภาพสูงสุด ภาพต่ำสุด ภาพเฉลี่ย อย่างละพอร์ตเป็น 3 พอร์ต และค่า RGB จากไฟสวิตช์โอดั้งเดิมอีก 3 พอร์ต และค่า counter อีก 1 พอร์ตและมีพอร์ตเอาต์พุตทั้งหมด 3 พอร์ต โดยนำค่า RGB ของเฟรม area boundaries โดย s-function นี้ทำงานตาม pseudo-code ดังต่อไปนี้

```

if InputPort(7).Data == 49
    % do a sobel edge detection
    edgemax = sobeledgedetection(InputPort(1).Data)
    edgemin = sobeledgedetection (InputPort(2).Data)
    edgemean = sobeledgedetection (InputPort(3).Data)

    % do text box classification
    textboxred = settextbox(edgemax,edgemin, edgemean,InputPort(4).Data)
    % make rectangle
    recttextboxred = makerectangle(textboxred)

    finalframed = max(InputPort(4).Data, recttextboxred)
    finalframegreen = block.InputPort(5).Data
    finalframeblue = block.InputPort(6).Data

    OutputPort(1).Data = finalframed
    OutputPort(2).Data = finalframegreen
    OutputPort(3).Data = finalframeblue
end

```

จาก pseudo-code ข้างต้น แสดงการทำงานของ mlmeesample ใช้อย่างคร่าวๆ หลังเสร็จสิ้นการโปรแกรมในส่วนนี้แล้ว นำเอาต์พุตออกไปยังบล็อก Video Viewer เพื่อแสดงผลลัพธ์สุดท้ายของการจำลองระบบตรวจสอบหาพื้นที่ superimposed text นี้

### 4.3 การพัฒนาอัลกอริทึมบนบอร์ดทดลอง Virtex-II Pro

ก่อนอื่นจะอธิบายถึงบอร์ดทดลอง XUP Virtex-II Pro Development System ตัวนี้เป็นรุ่นที่ทางบริษัท Digilent ได้ทำการออกแบบขึ้นมาเพื่อเจาะกลุ่มตลาดระดับนักศึกษามหาวิทยาลัยที่สนใจศึกษาเทคโนโลยี FPGA เพื่อนำไปใช้งานในระบบที่ต้องการความรวดเร็วในการคำนวณสูง และด้วยความยืดหยุ่นของ component ที่ประกอบมาให้กับตัวบอร์ด จึงเหมาะสมต่อนักศึกษาที่สนใจในงานด้านนี้เป็นอย่างมาก แต่เนื่องด้วยบอร์ดตัวนี้นั้นยังคงมีข้อมูลการใช้งาน รวมถึงนักวิจัยในประเทศไทยที่ได้ทำการพัฒนามีอยู่น้อยมาก ซึ่งนับว่าเป็นอุปสรรคอย่างใหญ่หลวงสำหรับโครงการนี้

#### การทดสอบบอร์ดการทดลองเบื้องต้น

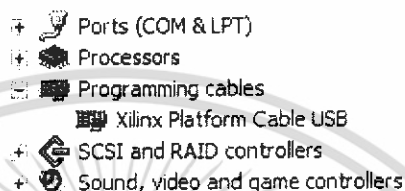
1. Xilinx University Virtex II Pro Development Board
2. DDR SDRAM ขนาด 256~512MB มีประโยชน์มากสำหรับงานคำนวณที่ต้องใช้พื้นที่เก็บข้อมูลที่ได้จากการคำนวณในปริมาณมาก อย่างเช่น Image Data, Video Data เป็นต้น เพราะภาพที่จะเอามาคำนวณนั้น Block-Ram ที่มีมาให้ของตัวชิป Virtex-II Pro (ขนาด 128K) ไม่สามารถรองรับได้พอกับความต้องการในโครงการนี้ได้
3. สาย USB ที่มาพร้อมกับบอร์ด (หรือจะเป็น Parallel Port ซึ่งสามารถนำมาใช้ได้เหมือนกัน แต่ในที่นี้จะใช้สาย USB ที่ให้มากับตัวบอร์ดทดลอง)
4. หม้อแปลงแรงดันไฟฟ้า
5. สาย RS-232 หรือที่รู้จักกันในนามว่า Serial Port นั้นเอง (ในโครงการนี้ได้นำ USB 2 Serial เพื่อแปลงหัวจาก Serial พอร์ตมาเป็น USB แทนนับว่าเหมาะสำหรับคนที่จะใช้ Notebook เป็นเครื่องทำงาน)
6. เครื่อง Host PC 1 เครื่อง
7. Device อื่นๆ สำหรับการทดสอบด้วย Build-In Self Test

#### วิธีติดตั้งโปรแกรมและข้อเสนอแนะ

ทำการ install โปรแกรม Xilinx ISE Foundation, Xilinx EDK 8.1 (ต้องเป็นซอฟต์แวร์ของ Xilinx เวอร์ชันที่ตรงกันกับบอร์ดทดลองนี้) เมื่อติดตั้งโปรแกรมสำเร็จ จากนั้นก็ทำการรีบูตเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำการลง Driver สาย USB กับตัวเครื่อง โดยเสียบสาย USB เข้าตัวเครื่อง PC และตัวบอร์ด เปิดสวิตช์บอร์ด ระบบจะทำการ Initialize ตัวมันเอง ให้สอดคล้องตามระบบที่แนะนำ ซึ่งเมื่อติดตั้งเสร็จเรียบร้อยแล้ว ก็สามารถเข้าไปเช็คใน Device Manager ถ้าทำการติดตั้งได้อย่างถูกต้อง จะมีฮาร์ดแวร์ปรากฏในลิสต์คือ Programming cables – Xilinx Platform Cable USB ขึ้นมาดังรูป (4.17)



รูปที่ 4.17 เมื่อสามารถลง Driver Programming Cables สำหรับบอร์ดทดลองได้อย่างถูกต้อง

สำหรับกรณีที่มีปัญหาลงไม่ได้ อาจมีสาเหตุมาจากการลงไฟล์ .sys แบบผิดวิธี ทางผู้พัฒนาขอแนะนำ ให้ทำการลบโปรแกรม Xilinx ISE ออกไปเสียก่อน แล้วตามหา Driver ชื่อ .sys ในโฟลเดอร์ Windows/System32/Driver ในไดร์ฟที่ท่านได้ลง OS Windows ไว้แล้วทำการลบไฟล์นั้นออกไป แล้วจากนั้นก็ย้อนกลับมาสู่ขั้นตอนการติดตั้งใหม่ หากไม่ใช้วิธีนี้ ก็ไม่สามารถที่จะติดตั้งโปรแกรมใหม่ได้อีกครั้ง อาจจะต้องมีการทำ Format เครื่องใหม่ก่อนลงโปรแกรม

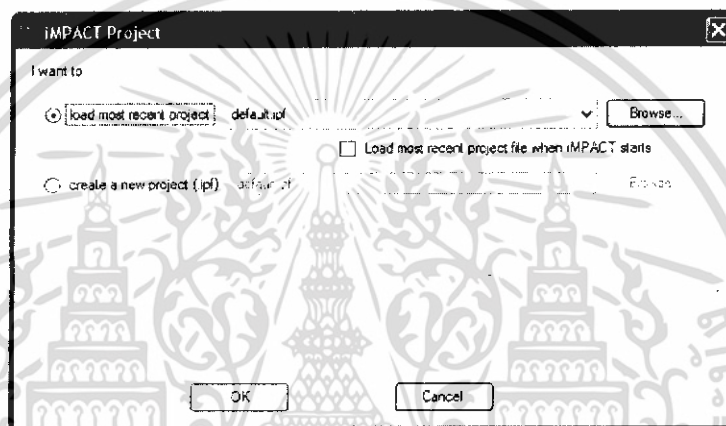
#### ทดสอบการใช้งานบอร์ดทดลอง

สำหรับสิ่งแรกที่ผู้ใช้งานควรจะต้องรู้เกี่ยวกับบอร์ดทดลองตัวนี้คือ “ความสามารถของบอร์ดทดลอง Virtex-II Pro” ดังนั้นจึงต้องมาทำการทดสอบก่อนว่าบอร์ดของนี้มี feature ใช้งานที่ครบหรือไม่ ภายในตัวบอร์ด XUP2VP นี้มีระบบทดสอบตัวเองที่เรียกว่า Built-In Self Test โดยสามารถดาวน์โหลดได้จาก website ของ Xilinx ในส่วนของ University Program / Hardware Section ที่ <http://www.xilinx.com/univ/xupv2p.html> โดยในเว็บไซต์นี้มีข้อมูลพื้นฐานเบื้องต้นการใช้งานบอร์ดทดลอง XUPV2P หรือสามารถค้นหาได้จากใน CD ประกอบที่ให้มากับตัวบอร์ดทดลอง

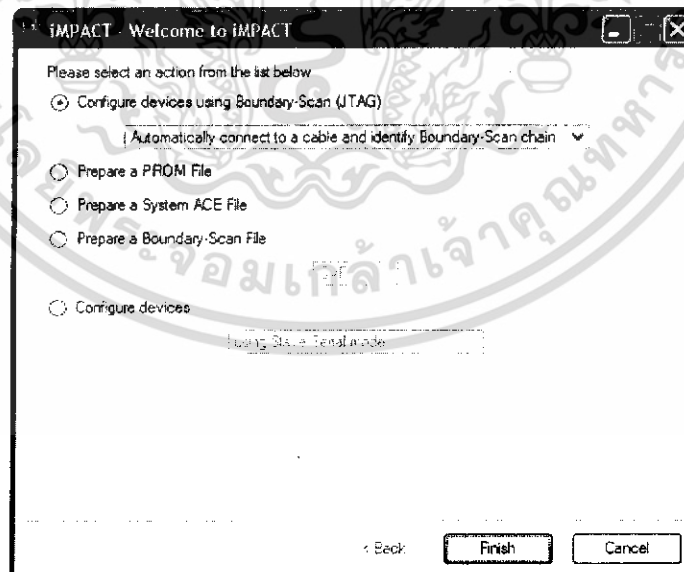
ภายในไฟล์ที่ได้ทำการดาวน์โหลดในข้างต้น จะมีโปรแกรม IMPACT โปรแกรมสำหรับการ download file ลงสู่ตัวบอร์ดทดลอง FPGA ผ่าน Programming Cables ประเภท Parallel กับ

USB หรืออาจจะให้ FPGA โหลดจาก System ACE โปรแกรมนี้ก็แปลง stream ไปลงสู่ compact flash ได้ด้วยเช่นกัน

เมื่อเปิดโปรแกรมขึ้นมาแล้ว โปรแกรมจะขึ้นภาพดังต่อไปนี้ขึ้นมา ให้ทำการเลือกไปที่ create a new project (.ipf) แล้วกด OK แล้วจะขึ้นหน้า Welcome to iMPACT ดังรูปที่ 4.18 ให้เลือกว่าจะให้โปรแกรมทำงานในรูปแบบไหน สำหรับตอนนี้ต้องการดาวน์โหลดไฟล์ Bitstream ลงสู่ตัวบอร์ดทดลอง ให้เลือก Configure devices using Boundary-Scan (JTAG) แล้วกด Finish

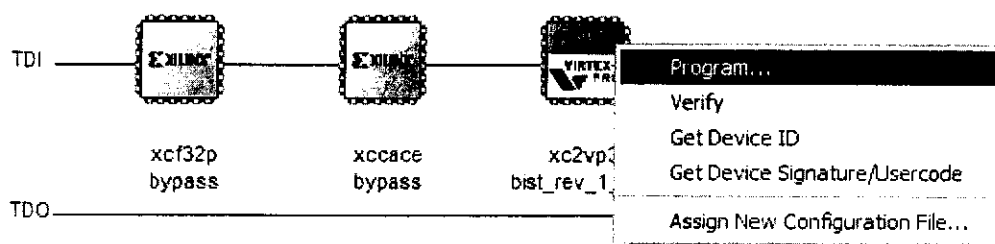


รูปที่ 4.18 Dialog Box แรกเมื่อเปิดโปรแกรม Impact



รูปที่ 4.19 หน้า Welcome to iMPACT เพื่อเลือกการใช้งานที่ต้องการจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 เมื่อโปรแกรมทำการ initialize chain เสร็จแล้ว

โปรแกรมจะทำการ auto initialize chain ขึ้นมา ซึ่งจะแสดงให้เห็นชิป 3 ตัวคือ xc32p bypass, xccace bypass และ xc2vp30 bist\_rev\_1\_6b.bit พร้อมกับ dialog ซึ่งให้โหลดโปรแกรม bitstream ผ่านลงไปในตัวชิป สำหรับ 2 ตัวแรกนั้น ให้ตั้งค่าพื้นฐานเป็น Bypass

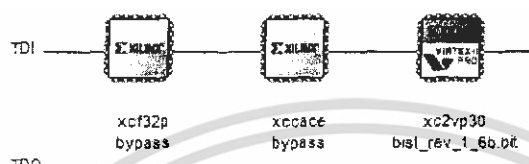
ส่วนตัวที่สามนี้เป็นชิป Virtex-II Pro สำหรับบอร์ดทดลอง ในกรณีที่ไม่ขึ้นว่า xc2vp30 ให้ทำการตรวจสอบถึงสาย USB นั้นเข้าระหว่างตัวบอร์ดกับเครื่อง ถ้าได้ทำการตรวจสอบเรียบร้อยแล้ว ลองทำการ Initialize chain ใหม่อีกครั้ง ถ้ายังคงเกิดปัญหาในลักษณะเดิมอยู่นั้น แสดงว่า ได้ทำการติดตั้งโปรแกรม ISE ผิดรุ่น ในกรณีที่ต้องการติดตั้ง ISE webpack ร่วมด้วยนั้น ทางผู้พัฒนา แนะนำให้ติดตั้ง ISE Foundation เสียก่อนเพื่อป้องกันการเกิดปัญหาดังที่กล่าวมา



รูปที่ 4.21 ให้เลือกไฟล์ .bit เพื่อนำมาดาวน์โหลดลงสู่บอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการค้นหาไฟล์ bist\_rev\_1\_6.bit จากในไฟล์ BIST ที่ Download มา แล้วกด OK จากนั้น ให้คลิกขวาที่ตัวชิป xc2vp30 แล้วเลือก Program ดังรูปที่ 4.20 ระบบจะทำการ Download โปรแกรมลงสู่บอร์ดให้ จากนั้นจะมี Dialog Box ขึ้นมาถามการ Verify จากผู้ใช้งาน หลังจากนั้นระบบก็จะทำการเบิร์นลง บอร์ด โปรแกรมจะแสดงผลดังภาพที่ 4.22



Program Succeeded

รูปที่ 4.22 เมื่อทำการดาวน์โหลดไฟล์ .bit ลงสู่ตัวบอร์ดทดลองได้สำเร็จ

ภายใน Build-In Self Test ที่ได้ทำการเบิร์นลงไปนั้น เป็นตัวทดสอบประสิทธิภาพของบอร์ด XUPV2P ว่าสามารถทำอะไรได้บ้าง โดยมี feature ในระบบ Build in Self Test นี้จะรองรับการทดสอบบอร์ดทดลองใน 2 รูปแบบ คือ Non Processor Based test Design กับ Processor Based test design โดยที่แบบแรกนั้นครอบคลุมการทำงานของบอร์ดดังนี้

- 1) Power Supply และ RESET generation
- 2) 100MHz system clock, 75MHz MGT clock, 32MHz SystemACE controller clock
- 3) LEDs
- 4) Push buttons กับ DIP switches
- 5) การแสดงออกทาง VGA มอนิเตอร์
- 6) Audio CODEC beep tone passthrough and power amplifier
- 7) RS232 serial port interface
- 8) PS/2 serial port interface
- 9) Silicon Serial Number
- 10) พอร์ตเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11) Configuration Source and mode selection, Platform FLASH, USB, PC4 and SystemACE Compact FLASH

สำหรับระบบที่จำเป็นต้องใช้ Processor ในการควบคุมมีได้แก่

- 1) SystemACE controller processor interface
- 2) Audio CODEC PCM data path
- 3) 10/100 Ethernet interface
- 4) MGT ports
- 5) DDR SDRAM module and Serial Presence Detect PROM

เมื่อได้ติดตั้ง Serial Port เข้ากับตัวเครื่องเป็นที่เรียบร้อยแล้ว ให้เปิดโปรแกรม Terminal ขึ้นมา (ใน Window-> Hyper Terminal -> Accessories/Connection) เลือกสร้างงานชุดใหม่ ขึ้นมา ตั้งชื่อว่า “9600Test” เลือก Connect Using เป็น ตำแหน่งที่ติดตั้ง Serial Port (ในกรณีของ Usb2Serial เลือกค่าเป็น Com 3 แต่ในกรณีที่พัฒนาด้วย Serial Port โดยตรงให้กำหนดค่าเป็น Com 1) แล้วตั้งค่าดังต่อไปนี้

Bits per Second= 9600  
 Data Bits = 8  
 Parity = None  
 Stop Bits = 1  
 Flow Control = None

ต่อไปทำการเปิด Switch Reset บนบอร์ดทดลองณ บริเวณตรงมุมขวาของบอร์ดทดลองกดเพียง 1 ครั้ง (ในกรณีที่ทำการกดค้าง 1~2 วินาที จะเป็นการล้างโปรแกรมของตัวบอร์ดออกไป) เมื่อกด Reset แล้วทีนี้ก็จะเห็น LED ติดไฟขึ้นซึ่งแสดงว่าถึงตัวโปรแกรม Initial ที่ทาง Xilinx ใ้เข้ามาแสดงการทดสอบตัว LED ในกรณีที่ถ้าไม่มีข้อผิดพลาดระหว่างการทดสอบ ข้อความดังรูป 4.23 ใน terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### XUP-V2Pro BuiltIn Self Test Main Menu Rev. 1.6 Sept. 2005

- 1 - Test SATA port with Aurora loopback.
- 2 - Test Ethernet with WEB example.
- 3 - Test AC97 audio codec.
- 4 - Test System ACE.
- 5 - Test DDR SDRAM.
- 6 - Test Expansion connectors.
- q - Quit

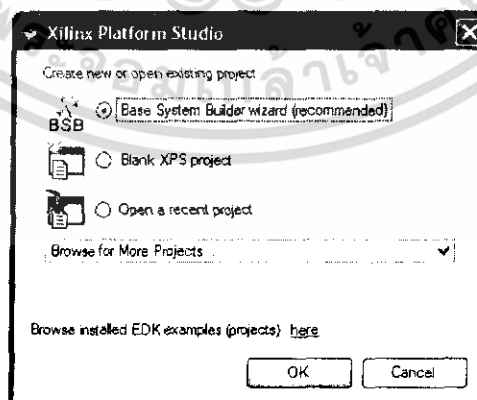
#### รูปที่ 4.23 หน้าจอต้อนรับของระบบ Build-In Self Test ผ่านโปรแกรม Hyper Terminal

##### เริ่มต้นการใช้งาน EDK 8.1

บทนี้จะเริ่มการใช้งาน Xilinx EDK แล้วครับ ขอเกริ่นเล็กน้อยก่อน EDK คือ Embedded Development Kit ที่ทาง Xilinx พัฒนาขึ้นมา โดยที่บอร์ด XUP2VP นี้มีความสามารถรองรับระบบ PowerPC กับ MicroBlaze ได้ เป็น Soft Processor Core สำหรับใช้งานใน FPGA โดยเฉพาะ

ก่อนจะพัฒนาโปรแกรมด้วยเครื่องมือ EDK นั้นนั้น ต้องทำการดาวน์โหลด Package [http://www.xilinx.com/univ/XUPV2P/lib/lib\\_rev\\_1\\_1.zip](http://www.xilinx.com/univ/XUPV2P/lib/lib_rev_1_1.zip) นี้มาแล้วทำการ Unzip ออกแล้วค้นหาไฟล์เคอร์ซีชื่อ boards จากนั้นก็อปปี Xilinx\_XUP\_V2P ไปวางไว้ใน ...AEDK\board\Xilinx\boards ในไฟล์ Xilinx\_XUP\_V2P นั้นเป็นไฟล์อธิบายลักษณะของตัวบอร์ดทดลองในโครงการนี้

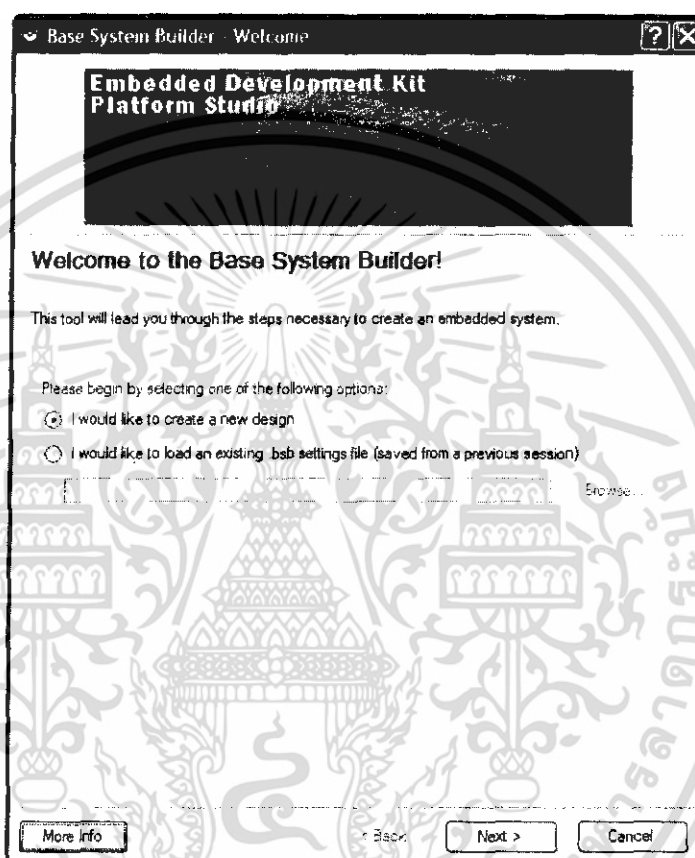
เปิดโปรแกรม EDK ขึ้นมา ตัวโปรแกรมจะถามว่า ให้เลือก BSB Builder แล้ว OK จากนั้น ให้เลือก Folder ที่จะเก็บ Project นี้ ทางผู้พัฒนาแนะนำให้ตั้ง Folder ขึ้นมาใหม่สำหรับงาน โดยเฉพาะเนื่องจากใน Project XPS นี้จะมีไฟล์ย่อยอยู่เยอะอาจก่อให้เกิดความสับสนระหว่างการพัฒนาได้



รูปที่ 4.24 หน้าแรกของ Base System Builder Wizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

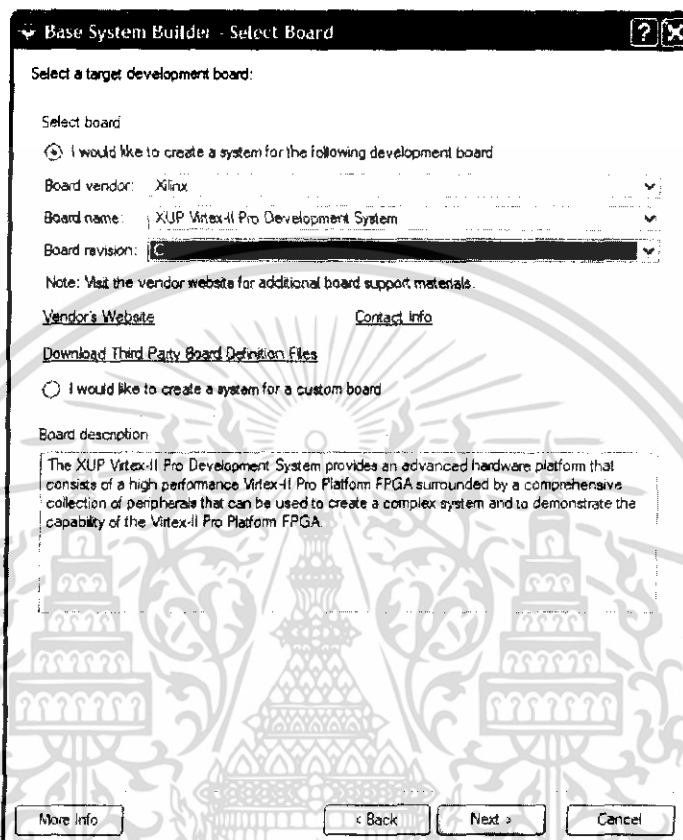
ต่อมาให้จะเข้าสู่ Base System Builder Wizard ในหน้าแรกนั้นให้ทำการเลือก “I would Like to create a new Design” ในส่วนอีกตัวเลือกนั้น จะเป็นการนำ bsb file ที่ setting ไว้เป็นประจำแล้วมาใช้งานใน Project ใหม่ซึ่งในขั้นตอนแรกนั้น ทางผู้พัฒนาได้สร้างโปรเจกใหม่ขึ้นมา ก่อน



รูปที่ 4.25 หน้าจอต้อนรับผู้ใช้งานของ Base System Builder

ต่อมาจะเป็นการเลือกบอร์ดทดลองที่จะใช้ทำงาน ให้เลือก “I would like to create a system for the following development board” จะเห็นว่าในส่วนของ Board Name นั้นมีบอร์ดของทาง Xilinx รุ่นต่างๆ แต่เนื่องจากได้พัฒนาบนบอร์ดรุ่น Virtex-II Pro ให้เลือก “XUP Virtex-II Pro Development Board” ที่อยู่ท้ายสุดของตัวเลือก และเลือก Board Revision เป็น “C” ในกรณีที่ไม่พบพบชื่อบอร์ด XUPV2P ก็แสดงว่ายังไม่ได้ก๊อปปี้ Package ของตัวบอร์ด XUPV2P ลงใน

...\EDK\board\Xilinx\boards ซึ่งให้ cancel กระบวนการ Base System Builder Wizard นี้ไปก่อนแล้ว  
ไปก๊อปปี้ Package ลงในโฟลเดอร์ในขั้นตอนที่กล่าวมาในข้างต้น



รูปที่ 4.26 เลือกบอร์ดทดลองที่ต้องการจะทำการออกแบบระบบ

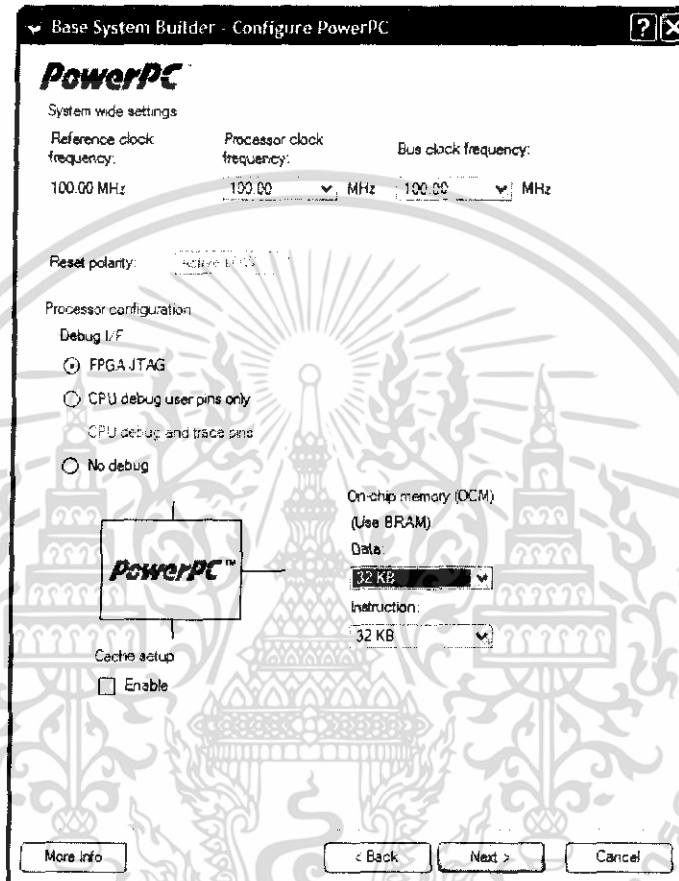
ต่อมาก็ให้เลือก Processor Core ที่จะใช้โครงการ ซึ่งในที่นี้จะมาเลือกใช้บริการของ PowerPC  
Core ดังภาพที่ 4.27



รูปที่ 4.27 ทำการเลือก Processor Core ที่จะใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาให้ทำการ Configure ตัว PowerPC ซึ่งในส่วนนี้ ทางผู้พัฒนาจะอธิบายไปที่ละตัวเลือกเพราะต้องอาศัยความเข้าใจในการตั้งค่าต่างๆ โดยจะอธิบายเฉพาะส่วนที่ใช้ในโครงการนี้เท่านั้น



รูปที่ 4.28 หน้าจอ Configure PowerPC ในส่วนแรก

System Wide Setting จะมี Processor Clock Frequency ความถี่สัญญาณนาฬิกาสำหรับการทำงานของตัว Processor ให้ตั้งไว้ที่ 100 MHz นะครับ Bus Clock Frequency ความถี่ของระบบ Bus ในการส่งผ่านข้อมูลไปมาระหว่าง Device ต่างๆกับตัว Core Processor ทางผู้พัฒนานั้นตั้งค่าไว้ที่ 100 MHz ซึ่งค่านี้จะนำไปประยุกต์ใช้ร่วมกับ SDRAM ในภายหลัง

Processor Configuration สำหรับ Debug I/F ให้ตั้งค่าไปที่ FPGA JTAG หมายความว่า จะทำการ Debug ระบบผ่านทางสาย USB ส่วนค่า On-Chip Memory ซึ่งมีไว้ใช้เก็บ Data กับ Instruction ในโครงการนี้ มีความต้องการจะเก็บคำสั่งโปรแกรมและข้อมูลที่เกิดขึ้นระหว่างทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาไว้ในส่วนของ BlockRam ดังนั้นตั้งค่าสำหรับทั้งสองช่องไว้ที่ 32KB สำหรับ Cache ปล่อยให้ว่างไว้ เพราะ ยังไม่ใช้งานในคำสั่งที่ต้องการ Feed ไปกลับระหว่าง Memory และตัว Processor แต่จะทำการ initialize ค่า Cache เอาจาก Software ของผู้พัฒนาได้ในภายหลังแทน

RS232\_Uart\_1

Peripheral: OPB UARTLITE

Baudrate (bits per seconds): 115200

Data bits: 8

Parity: NONE

Use interrupt

#### รูปที่ 4.29 ตั้งค่าสำหรับการใช้งาน Serial Port

ในส่วนของ RS232\_Uart\_1 ให้ตั้งค่าดังต่อไปนี้ ตามรูปที่ 4.29 โดยให้ Baud rate = 115200 (ในส่วนของ Baud rate นี้ ส่วนมากจะใช้งานกันที่ 9600 แต่ทางผู้พัฒนาเลือกจะใช้ในระดับ 115200 ซึ่งเร็วกว่า 12 เท่าเพื่อที่จะมีประโยชน์ในเรื่องการ access ความเร็วที่ต้องส่งผ่านมาทาง terminal เป็นการลดปัญหาคอขวดของระบบลงไปได้มาก โดยเฉพาะในระบบที่จะต้องข้อมูลส่งค่าไปมาระหว่าง PC กับ FPGA ) Data Bits = 8 Parity = None

SysACE\_CompactFlash

LEDs\_4Bit

Peripheral: OPB GPIO

Use interrupt

DIPSWs\_4Bit

Peripheral: OPB GPIO

Use interrupt

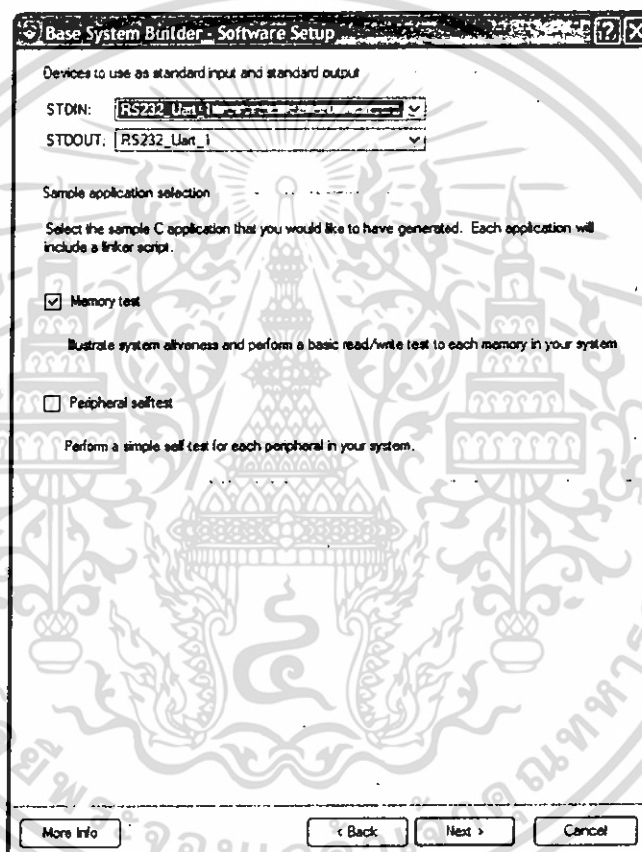
PushButtons\_50A

#### รูปที่ 4.30 ตั้งค่าสำหรับการใช้งาน LED ขนาด 4 บิต และ DipSwitch ขนาด 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ LED กับ Dip Switch ทั้งคู่ให้ตั้ง Peripheral ไว้ที่ OPB GPIO เพราะเป็น device ที่ไม่จำเป็นต้อง access ให้เร็วแต่อย่างใดนัก

ในส่วนของ Software Setup นั้น ที่ตัวเลือกของ STDIN,STDOUT ให้เลือกเป็น RS232\_Uart\_1 ทั้งคู่ ผู้พัฒนาต้องการส่งข้อมูลไปมาผ่านทาง Serial Port ส่วน Sample Application Selection ให้เลือกเฉพาะ memory test



รูปที่ 4.31 ตั้งค่าในส่วนของ Software Setup

จากที่กล่าวมานั้น ผู้พัฒนาเลือกให้โปรแกรม Generate Software ชุด Memory Test ขึ้นมา ก็จะมีการให้ตั้งค่าว่าจะเก็บ Instruction, Data, Stack/Heap ไว้ที่ไหนบ้าง โดยตั้งไปว่า iocm\_cntrl สำหรับเก็บส่วนของ Instruction และ docm\_cntrl สำหรับเก็บข้อมูลส่วน Data กับ Stack/Heap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MemoryTest

Select the memory devices which will be used to hold the following program sections:

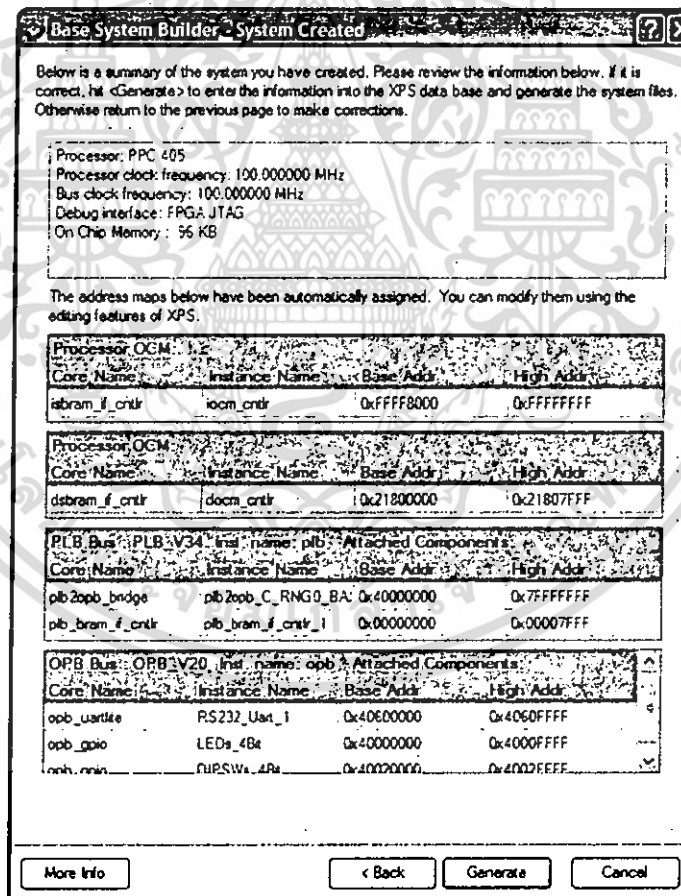
Instruction:

Data:

Stack/Heap:

รูปที่ 4.32 ตั้งค่าระบบที่จะจัดเก็บข้อมูล Data, Instruction และ Stack/Heap ไว้

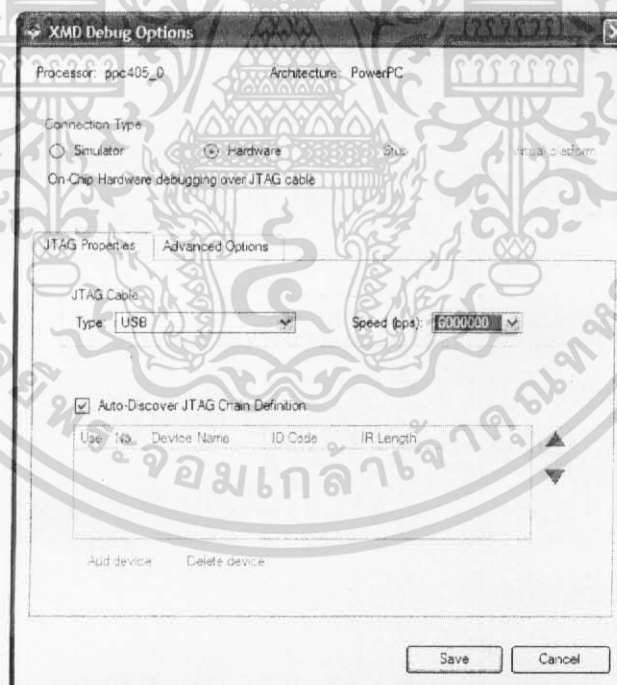
เมื่อทำการตั้งค่าทั้งหมดเรียบร้อยแล้ว โปรแกรมจะแสดงสรุปทั้งหมดของระบบที่สร้างขึ้นมาให้กด Generate เป็นอันเสร็จสิ้นการตั้งค่า Base System Builder ของ โปรเจ็ค



รูปที่ 4.33 หน้าจอสรุปการตั้งค่าทั้งหมดที่จะเกิดขึ้นในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อมาทำการ Generate Hardware กับ Software เพื่อใช้ในงานของโครงการ สำหรับในส่วนของ Hardware นั้นให้เลือก Hardware -> Generate Bitstream เพื่อทำการ synthesis ระบบทาง Hardware ขึ้นมา ในขั้นตอนจะต้องใช้เวลาประมาณ 4-6 นาทีในการ Generate ระบบทั้งหมด ออกมา (ความเร็วขึ้นอยู่กับประสิทธิภาพของเครื่อง) เมื่อเสร็จแล้ว ต่อไปจะคอมไพล์ Software ตัว AppMemory ที่ได้ทำการเขียนเอาไว้ก่อนหน้านี้ เตรียมเอาไว้ โดยการพัฒนา software นั้นจะพัฒนาโดย ภาษา C ซึ่งจะต้องมีการพัฒนาโปรแกรมก่อนการ Map กับฮาร์ดแวร์ เมื่อโหลดเข้าไปแล้ว ทำการ Build Project ภายใน Software -> Build All User Application ซึ่งจะทำการสร้าง Library ขึ้นมารองรับกับ Hardware ที่ได้สร้างไปก่อนหน้านี้ เมื่อ Build เสร็จแล้ว ให้ทำการ Download ไฟล์ Bitstream ของโปรเจกต์ใหม่นี้ไปลงยังบอร์ด FPGA ที่เตรียมไว้ สำหรับการ Download แบบนี้จะเป็นการเตรียมระบบให้พร้อมไว้ก่อน แต่ยังคงขาด Software ที่จะมาทำงานในระบบ หลังจากนั้นทำการเปิดระบบ Xilinx Microprocessor Debugger (XMD) เพื่อดาวน์โหลด Software ลงสู่ตัว FPGA ให้กดปุ่ม Launch XMD หรือเลือกได้จาก Debug -> Launch XMD...



รูปที่ 4.34 ตั้งค่าสำหรับ XMD ก่อนเริ่มใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นระบบจะแสดง Dialog Box ว่าต้องการใช้ Processor ตัวไหนในการทำงาน ให้เลือก PPC405\_0 (มีสองตัวคือ 0 กับ 1 ตัว 0 จะเป็น default ของระบบ) จากนั้นใน tab JTAG Properties ให้เลือกสายสัญญาณที่จะส่งข้อมูลเป็น type USB และเลือกความเร็วเป็น 6000000 กด save แล้วระบบจะทำการเริ่ม XMD ขึ้นมา

ในระบบ XMD นี้เป็นการติดต่อไปยัง PowerPC Core ผ่านสาย USB ฉะนั้นระบบที่ต้องการติดต่ออยู่นี้จะมาจากตัวบอร์ดทดลองโดยตรงซึ่ง PowerPC ใช้ Embedded Linux เป็นระบบปฏิบัติการฝั่งตัวเองไว้

```

G:\EDK\bin\nt\xmd.exe
INFO:MDT - Assumption: Selected Device 3 for debugging.
JTAG chain configuration
-----
Device  ID Code      IR Length  Part Name
-----  -
1       05059093         16        XCF32P
2       0a001093          8        System_nCE
3       0127e093         14        XC20P30
XMD: Connected to PowerPC target. Processor Version No : 0x200100a0
Address mapping for accessing special PowerPC features from XMD/GDB:
I-Cache <Data> : Disabled
I-Cache <Tag>  : Disabled
D-Cache <Data> : Disabled
D-Cache <Tag>  : Disabled
ISOCM        : Start Address - 0xffff8000, Size - 32768 bytes
TLB          : Disabled
DCR         : Disabled
Connected to "ppc" target. id = 0
Starting GDB server for "ppc" target (id = 0) at TCP port no 1234
XMD% ls
TestApp_Memory  implementation  ppc405_0      system.log      system_incl.make
__xps          libgen.log      ppc405_1      system.make
data          pcores         synthesis     system.mhs
etc          platgen.log     system.bsh    system.ncs
hdl          platgen.opt     system.gui    system.xmp
XMD% cd TestApp_Memory
XMD% ls
executable.elf  src
XMD% doe executable.elf
  
```

รูปที่ 4.35 หน้าต่าง XMD แสดงสถานะ shell prompt

เมื่อพบ Shell Prompt แล้วให้พิมพ์ cd TestApp\_Memory ตามด้วย dow executable.elf

เมื่อดำเนินโหลดเสร็จแล้วให้กด run เพื่อทำการรันโปรแกรม

```

G:\EDK\bin\nt\xmd.exe
invalid command name "doe"
XMD: dow executable.elf
WARNING: Attempted to read location: 0xffffffffc. Reading ISOCM memory not supported

section, .text: 0xffffffff00-0xffffffff49c
section, .init: 0xffffffff49c-0xffffffff84c0
section, .fini: 0xffffffff84c0-0xffffffff84e0
section, .boot0: 0xffffffff84e0-0xffffffff84f0
section, .boot: 0xfffffffffc-0x00000000
section, .rodata: 0x21800000-0x21800032
section, .sdata2: 0x21800034-0x21800034
section, .shss2: 0x21800034-0x21800034
section, .data: 0x21800038-0x21800030
section, .got: 0x21800330-0x21800330
section, .got1: 0x21800330-0x21800330
section, .got2: 0x21800330-0x2180034c
section, .ctors: 0x2180034c-0x21800354
section, .dtors: 0x21800354-0x2180035c
section, .fixup: 0x2180035c-0x2180035c
section, .eh_frame: 0x2180035c-0x21800364
section, .jcr: 0x21800364-0x21800368
section, .gcc_except_table: 0x21800368-0x21800368
section, .sdata: 0x21800368-0x2180036c
section, .shss: 0x2180036c-0x2180036c
section, .bss: 0x2180036c-0x21800388
section, .bss_stack: 0x21800388-0x21800390
section, .bss_heap: 0x21800390-0x21800390
Downloaded Program executable.elf
Setting PC with program start addr = 0xffffffffc
PC reset to 0xffffffffc. Clearing MSR Register
XMD: run
PC reset to 0xffffffffc. Clearing MSR Register
Processor started. Type "stop" to stop processor
RUNNING>

```

รูปที่ 4.36 หน้าต่าง XMD แสดงสถานะการ run ของ program อยู่

ในตอนนี้ให้เปิด Hyper Terminal ขึ้นมาใหม่ ให้ตั้งค่าดังนี้

```

Bits per Second= 115200
Data Bits = 8
Parity = None
Stop Bits = 1
Flow Control = None

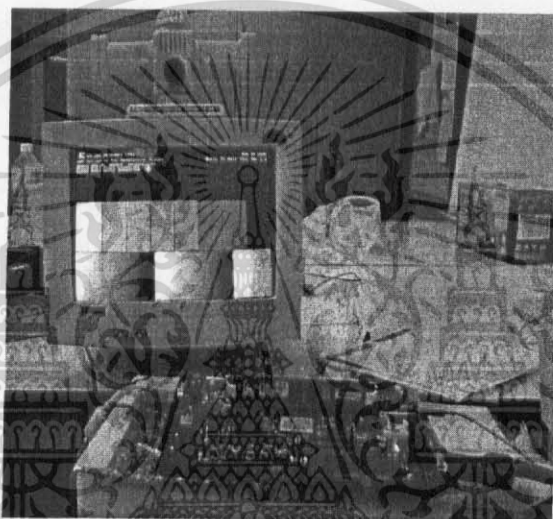
```

แล้วกดปุ่ม reset ที่บอร์ดทดลองจากนั้น ก็จะเห็นข้อความว่า “Enter Main, Exit Main” ขึ้นที่หน้าจอ terminal เป็นอันเสร็จสิ้น ซึ่งสามารถสร้าง Embedded Processor ขึ้นมาควบคุมการทำงาน FPGA ได้

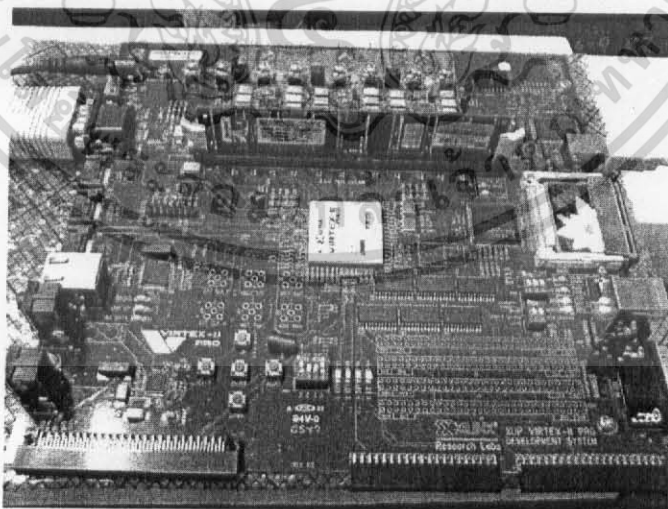
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การพัฒนาอัลกอริทึม Superimposed Text detection ลงบนบอร์ด FPGA

เริ่มต้นนั้นจะเป็นการแสดงผลภาพ Bitmap ออกมาหน้าจอมอนิเตอร์ คือในขั้นตอนแรกนั้นจะทดสอบโดยการโหลดเอาไฟล์ภาพ Bitmap เข้าไปเก็บไว้ใน DDR-SDRAM แล้วให้แสดงผลภาพออกทางหน้าจอมอนิเตอร์ ซึ่งจะต้องมีการเพิ่มอุปกรณ์คือ DDR SDRAM ขนาด 512 MB และ Compact Flash ขนาด 1GB ของ Kingston จอมอนิเตอร์ที่รองรับการแสดงผลได้ 800\*600 เป็นอย่างน้อย



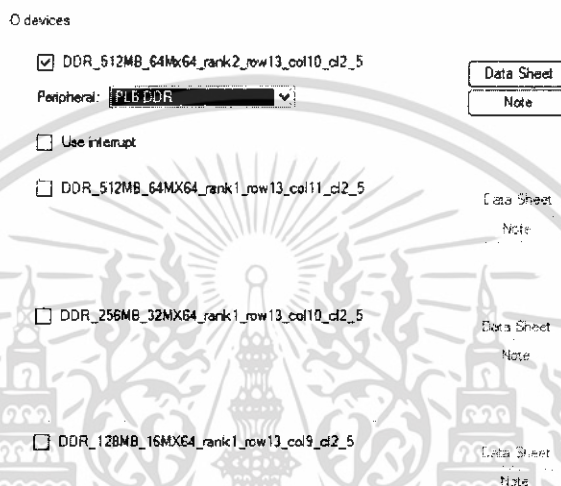
รูปที่ 4.37 รูปการต่ออินเตอร์เฟสระหว่าง จอภาพ, บอร์ด, แรม และ การ์ด



รูปที่ 4.38 รูปการต่ออินเตอร์เฟสระหว่าง จอภาพ, บอร์ด, แรม และ การ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจากการสร้าง Project Base System Builder กันใหม่ครับ ทำตามขั้นตอนเดิมในบทที่แล้วมาจนถึงขั้นตอน Processor Configuration ให้ท่านเลือก Device ดังต่อไปนี้มาใช้งาน RS232, SysACE Compact Flash, DDR\_512MB\_64Mx64\_rank2\_row13\_col10\_cl2\_5, Plb\_bram\_if\_cntrl\_1 (ให้เลือกขนาดของ BlockRam ไว้ 64 MB) จากนั้นให้เลือก StdIO ด้วย RS232 ส่วนโปรแกรมที่ไว้ทดสอบนั้น ในคราวนี้จะทำการเว้นว่างไว้ทั้งสองช่อง เพราะต้องการจะสร้าง Software Project



### รูปที่ 4.39 หน้าจอตัวเลือกขนาดของ RAM ที่จะใช้งานในระบบ

สำหรับการที่จะแสดงภาพไปออกยังหน้าจอมนิเตอร์นั้น จำเป็นต้องใช้ Library Core เพื่อทำการควบคุมสัญญาณที่จะนำไปออกยังมนิเตอร์ ที่มีชื่อว่า plb\_tft\_cntrl\_ref\_v1\_00\_d ซึ่งใน Library ชุดนี้ประกอบไปด้วยไฟล์ Verilog ทั้งหมดที่ควบคุมในส่วนการแสดงผลภาพให้ไปออกยังหน้าจอ โดยสามารถหาได้จาก โปรเจ็คตัวอย่างที่ได้ดาวน์โหลดมาก่อนในโฟลเดอร์ PCores จะมีไฟล์ชุด plb\_tft\_cntrl\_ref\_v1\_00\_d อยู่ ให้ทำการก๊อปปี้ทั้ง folder มาลงไว้ใน PCores ของโปรเจ็ค และทำการก๊อปปี้ โฟลเดอร์ Drivers ที่อยู่ในโฟลเดอร์หลักของ Project Slider มาด้วยเพราะในนั้นจะมีโฟลเดอร์ tft\_ref\_v1\_00\_a ซึ่งเป็น reference สำหรับการใช้งานคำสั่งที่เกี่ยวกับการแสดงผลออกหน้าจอภาพที่ทาง Xilinx จัดมาให้ในภาษา C

ในขั้นตอนต่อไปจะทำการดัดแปลงไฟล์ 3 ไฟล์ คือ System.mhs = “Microprocessor Hardware Specification (MHS)” which describes the instantiations and connections of hardware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

components. โดยที่ระบบ Simulation Model Generator จะอ่านค่าจากไฟล์ดังกล่าวเพื่อไปทำการ generate ชุด Netlist และ ไฟล์ Bistream สำหรับส่วนของ Hardware Core System.mss = ไฟล์ mss จำกัดความหมายของไคร์เวอร์ที่ทำงานร่วมกับ peripheral , Standard I/O devices interrupt handler routines และซอฟต์แวร์ที่เกี่ยวข้องอื่นๆ Libgen จะทำการตรวจสอบไลบรารีและไคร์เวอร์ได้จากไฟล์นี้

สำหรับการใช้งานไฟล์ mhs, mss สามารถอ่านรายละเอียดได้เพิ่มจาก Platform Specification Format Reference Manual ซึ่งอยู่ในโฟลเดอร์ Doc ของ EDK Installation Path, System.ucf = User Constraint Files ที่ขึ้นกับตัวบอร์ดทดลองที่จะใช้งาน เป็นไฟล์ที่เก็บค่าการ assign port และ netlist ที่จำเป็นจะต้องเชื่อมต่อกับตำแหน่ง port address ในบอร์ดทดลอง

ในขั้นตอนนี้ต่อไป จะอธิบายถึงไฟล์ system.ucf เมื่อตรวจสอบ Modules ที่มีการประกาศค่าเอาไว้ จะมี Module RS232\_Uart\_1 constraints, Module SysACE\_CompactFlash constraints, Module DDR\_512MB\_64Mx64\_rank2\_row13\_col10\_cl2\_5 constraints แต่ยังคง Modules ในส่วนของ TFT LCD VGA framebuffer อยู่ ให้เติมค่า assign ต่อไปนี้ลงต่อไปได้เลย

```
# Module for VGA FrameBuffer TFT LCD
Net fpga_0_VGA_FrameBuffer_TFT_LCD_CLK_pin LOC=H12;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_CLK_pin IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_CLK_pin SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_CLK_pin DRIVE = 12;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC_pin LOC=B8;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC_pin IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC_pin SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC_pin DRIVE = 12;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC_pin LOC=D11;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC_pin IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC_pin SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC_pin DRIVE = 12;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<0> LOC=H15;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<0> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<0> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<0> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<1> LOC=J15;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<1> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<1> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<1> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<2> LOC=C13;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<2> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<2> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<2> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<3> LOC=D13;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<3> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<3> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<3> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<4> LOC=D14;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<4> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<4> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<4> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<5> LOC=E14;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<5> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<5> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin<5> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<0> LOC=D10;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<0> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<0> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<0> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<1> LOC=D8;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<1> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<1> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<1> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<2> LOC=C8;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<2> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<2> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<2> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<3> LOC=H11;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<3> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<3> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<3> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<4> LOC=G11;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<4> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<4> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<4> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<5> LOC=E11;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<5> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<5> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin<5> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<0> LOC=G9;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<0> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<0> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<0> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<1> LOC=F9;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<1> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<1> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<1> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<2> LOC=F10;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<2> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<2> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<2> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<3> LOC=D7;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<3> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<3> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<3> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<4> LOC=C7;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<4> IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<4> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<4> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<5> LOC=H10;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<5> IOSTANDARD = LVTTTL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<5> SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin<5> DRIVE = 6;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK_pin LOC=A8;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK_pin IOSTANDARD = LVTTTL;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK_pin SLEW = SLOW;
Net fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK_pin DRIVE = 6;

```

ตั้งค่าไฟล์ mss

```

BEGIN DRIVER
PARAMETER DRIVER_NAME = tft_ref
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = VGA_FrameBuffer
END

```

```

BEGIN DRIVER
PARAMETER DRIVER_NAME = generic
PARAMETER DRIVER_VER = 1.00.a
PARAMETER HW_INSTANCE = opb2dcr_bridge_0
END

```

```

BEGIN LIBRARY
PARAMETER LIBRARY_NAME = xilfatfs
PARAMETER LIBRARY_VER = 1.00.a
PARAMETER PROC_INSTANCE = ppc405_0
END

```

ในขั้นตอนถัดมานั้น ผู้พัฒนาจะอธิบายถึงการทำงานในแต่ละตัวแปรที่ได้ประกาศไว้ ในตอนต้น ประกอบด้วย tft\_ref จะเป็น Driver Reference สำหรับการทำงานของ VGA\_FrameBuffer ตัวต่อมาก็คือสายเชื่อมส่งต่อสัญญาณจากชุด opb ไปยัง dcr\_bridge ส่วนตัวที่ 3 คือ Library “xilfatfs” เป็นไลบรารีที่ควบคุมการ อ่าน/เขียนของไฟล์ที่ถูกเก็บอยู่ใน Xilinx System ACE compact flash

โค้ดภายในไฟล์ mhs ที่ได้สร้างขึ้น

```

PORT fpga_0_VGA_FrameBuffer_TFT_LCD_CLK_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_CLK, DIR = 0
PORT fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC, DIR = 0
PORT fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC, DIR = 0
PORT fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK, DIR = 0
PORT fpga_0_VGA_FrameBuffer_TFT_LCD_B_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_B, VEC = [5:0], DIR = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PORT fpga_0_VGA_FrameBuffer_TFT_LCD_G_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_G, VEC = [5:0], DIR = 0
PORT fpga_0_VGA_FrameBuffer_TFT_LCD_R_pin =
fpga_0_VGA_FrameBuffer_TFT_LCD_R, VEC = [5:0], DIR = 0

```

```

BEGIN opb2dcr_bridge
PARAMETER INSTANCE = opb2dcr_bridge_0
PARAMETER HW_VER = 1.00.a
PARAMETER C_BASEADDR = 0xD0000000
PARAMETER C_HIGHADDR = 0xD00003FF
BUS_INTERFACE SOPB = opb
BUS_INTERFACE MDCR = dcr_v29_0
END

```

```

BEGIN plb_tft_cntrl_ref
PARAMETER INSTANCE = VGA_FrameBuffer
PARAMETER HW_VER = 1.00.d
PARAMETER C_DEFAULT_TFT_BASE_ADDR = 0b000000000000
PARAMETER C_PIXCLK_IS_BUSCLK_DIVBY4 = 0b1
# start with display off
PARAMETER C_ON_INIT = 0b0
PARAMETER C_DCR_BASEADDR = 0b0000010000
PARAMETER C_DCR_HIGHADDR = 0b0000010001
BUS_INTERFACE MPLB = plb
BUS_INTERFACE SDCR = dcr_v29_0
PORT SYS_dcrClk = sys_clk_s
PORT TFT_LCD_CLK = fpga_0_VGA_FrameBuffer_TFT_LCD_CLK
PORT TFT_LCD_HSYNC = fpga_0_VGA_FrameBuffer_TFT_LCD_HSYNC
PORT TFT_LCD_VSYNC = fpga_0_VGA_FrameBuffer_TFT_LCD_VSYNC
PORT TFT_LCD_B = fpga_0_VGA_FrameBuffer_TFT_LCD_B
PORT TFT_LCD_G = fpga_0_VGA_FrameBuffer_TFT_LCD_G
PORT TFT_LCD_R = fpga_0_VGA_FrameBuffer_TFT_LCD_R
PORT TFT_LCD_BLNK = fpga_0_VGA_FrameBuffer_TFT_LCD_BLNK
END

```

```

BEGIN dcr_v29
PARAMETER INSTANCE = dcr_v29_0
PARAMETER HW_VER = 1.00.a
PARAMETER C_DCR_NUM_SLAVES = 1
END

```

ทำการเปลี่ยนแปลงค่า Address ใน Module ต่อไปนี้เพื่อให้ทำงานสอดคล้องกันได้ทั้งระบบ  
ใน plb2opb\_bridge ให้เพิ่ม code เข้าไปดังนี้

```

PARAMETER C_RNG0_BASEADDR = 0xD0000000
PARAMETER C_RNG0_HIGHADDR = 0xD0000FFF
PARAMETER C_RNG1_BASEADDR = 0x78000000
PARAMETER C_RNG1_HIGHADDR = 0x7807ffff
PARAMETER C_NUM_ADDR_RNG = 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

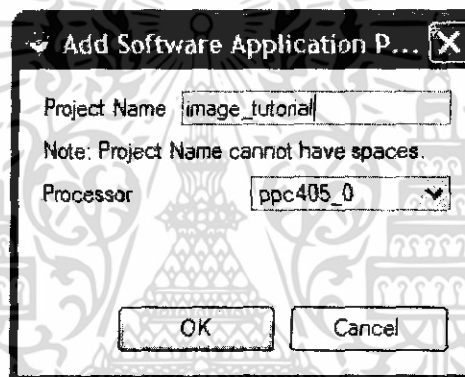
ใน opb\_uartlite ให้เปลี่ยนค่าเป็นดังนี้

```
PARAMETER C_BASEADDR = 0x78050000
PARAMETER C_HIGHADDR = 0x7805ffff
```

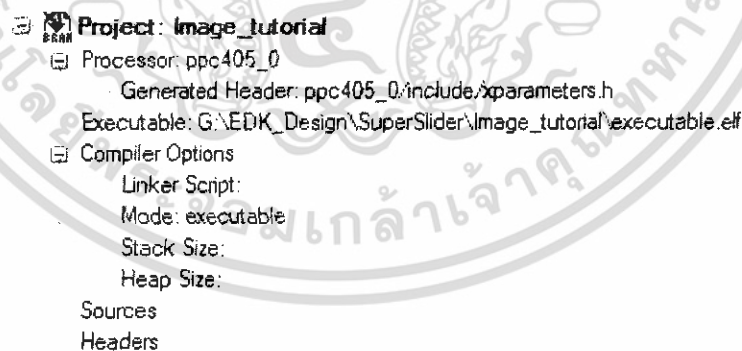
ใน opb\_sysace ให้เปลี่ยนค่าเป็นดังนี้

```
PARAMETER C_BASEADDR = 0x78060000
PARAMETER C_HIGHADDR = 0x7806ffff
```

หลังจากนั้นให้ทำการสร้างไฟล์ Bitstream จากคำสั่ง Hardware->Generate Bitstream ซึ่งหลังจากที่ทำการสร้าง Bitstream สำเร็จแล้ว ให้ทำการสร้าง Software Project ขึ้นมา ในแท็บ Application ตัวเลือกแรกที่ท่านจะเห็นก็คือ Add Software Application Project... (ดังรูปที่ 4.40)

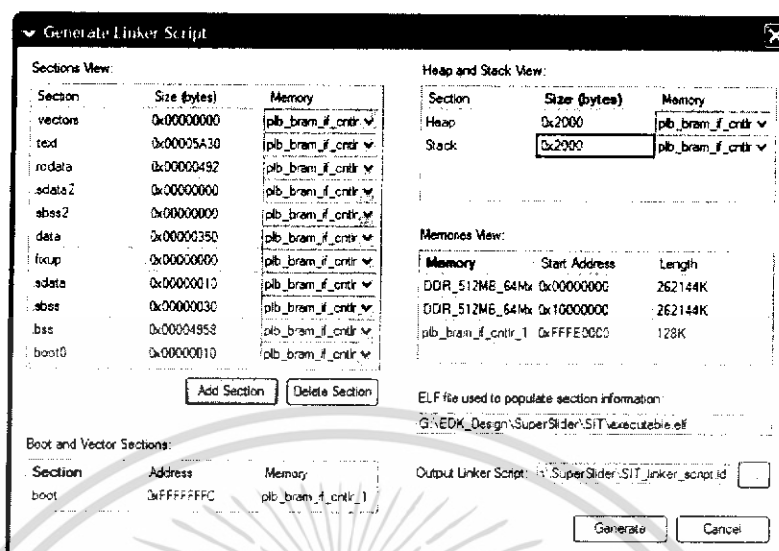


รูปที่ 4.40 หน้าจอการเพิ่ม Software Project ใหม่เข้าสู่ระบบ



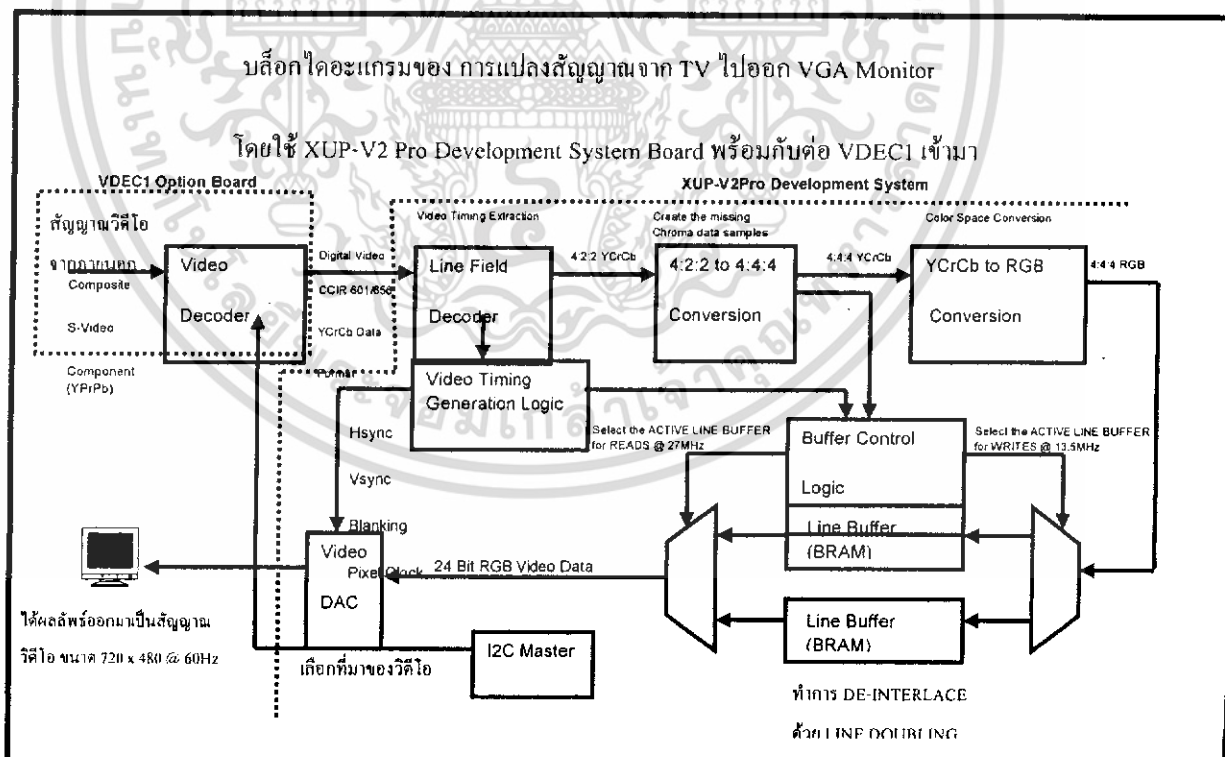
รูปที่ 4.41 โปรเจ็ค Image\_tutorial หลังจากได้ทำการเพิ่มเข้ามาสู่ระบบแล้ว

ให้คลิกขวาแล้วทำการสร้าง Software Project ชื่อว่า image\_tutorial ขึ้นมา และให้เลือก Processor ที่จะใช้งานเป็น ppc405\_0 เมื่อเสร็จแล้วให้ทำการ generate Linker Script โดยกำหนดให้ขนาดของ Stack กับ Heap ไว้เท่ากับ 0x2000 (เป็นค่าที่ใช้ในโครงการนี้)



รูปที่ 4.42 หน้าจอการสร้างไฟล์ Linker Script

ส่วนตัวเลือกอื่นนั้นให้รอเก็บไว้ที่ plb\_bram\_if\_cntlr\_1 ทั้งหมด ในความเป็นจริงแล้ว ส่วนอื่นจะเก็บไว้ที่ DDRAM แต่ต้องมีการตั้งค่าก็คือ .boot0 กับ .boot เมื่อเสร็จการสร้างส่วน Linker Script แล้ว ให้ทำการ Add new file เข้าไปที่ส่วน source เพื่อเขียนโปรแกรมแสดงผลในภาษา C



รูปที่ 4.43 บล็อกโคะแกรมของ การแปลงสัญญาณจาก TV ไปออกยัง VGA Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้โค้ดแล้วก็ให้ทำการคอมไพล์ตัว Software ไปด้วยคำสั่ง Software->Build all user application เมื่อคอมไพล์เสร็จแล้วให้ทดสอบโปรแกรมที่ได้พัฒนาขึ้นมา

ในโครงการนี้ ได้ทำการอิมพลีเมนต์ อัลกอริทึม Superimposed Text Detection ที่ทีมผู้พัฒนาในคิดค้นขึ้น ( กล่าวไว้ในตอนต้นของบท ) ลงในบอร์ดทดลอง โดยยังคงใช้ภาพนิ่งเป็นภาพอินพุตจาก compact flash เนื่องจากการติดต่อกับวิดีโอไฟสวิตช์ไอกับตัว FPGA ของบอร์ดทดลอง ต้องมีการสร้างตัวเล่นไฟสวิตช์ไอในบอร์ดทดลองดังกล่าว ซึ่งมีขั้นตอนที่ซับซ้อน และ ยากต่อการพัฒนาเป็นอย่างมาก ดังนั้นทีมพัฒนาจึงเลือกส่วนของอัลกอริทึมลงบอร์ดทดลองเพียงอย่างเดียว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### ผลการทดลอง

#### 5.1 ผลการทดลองจากการพัฒนาระบบค้นหาโดยอ้างอิงจากผู้ใช้โดยตรง

การทดลองนั้น ได้ใช้เครื่องคอมพิวเตอร์ Pentium(M) processor 1.6 GHz มีขนาด memory 512 MB โดยทดสอบกับวิดีโอไฟล์ที่เป็นกีฬาเบสบอลที่ถ่ายทอดทางโทรทัศน์ญี่ปุ่น (ช่อง NTV) โดยวิดีโอเป็นไฟล์ AVI มีความเร็ว 110 Kbps ด้วยขนาดภาพ 512X348 และเฟรมเรท 30 fps

ประสิทธิภาพของอัลกอริทึม โดยวิดีโอไฟล์มีความยาว 52.22 นาที ด้วยเครื่องคอมพิวเตอร์ที่อ้างอิงไว้ ใช้เวลาในการประมวลผล 3.41 นาที โดยเฉลี่ย โดยผลจากการทดลองนั้น ได้เสนอไว้ในรูปที่ 3.8 โดยได้ยกตัวอย่าง การค้นหาผู้เล่นที่ชื่อ ยามะโมโตะ (山本)

山本

a รูปแบบการค้นหา



b ผลจากการค้นหาด้วยรูปแบบ a

รูปที่ 5.1 ผลการค้นหาจากระบบในวิดีโอเกมกีฬาเบสบอล ประเทศญี่ปุ่น

#### 5.2 ผลการทดสอบประสิทธิภาพของอัลกอริทึมการค้นหา Superimposed Text บนวิดีโอ

การหาวิธีที่ดีที่สุดในการแบ่งแยกประเภทพื้นที่ของ Superimposed text นั้น ได้ทดสอบโดยวัด Precision และ Recall ซึ่งปกติใช้ใน information retrieval (IR) ในการสรุปผลการตรวจหา Superimposed text ในไฟล์วิดีโอที่ได้นำมาทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{Recall rate} = \frac{N_c}{N_c + M_m} \times 100\%$$

$$\text{Precision rate} = \frac{N_c}{N_c + M_f} \times 100\% \quad (5.1)$$

โดยกำหนดให้  $N_c$  เป็นจำนวนตำแหน่งที่ตรวจสอบ superimposed text ได้ถูกต้อง,  $N_m$  เป็นจำนวนตำแหน่งที่พลาดการค้นหา และ  $N_f$  เป็นจำนวนตำแหน่งที่ค้นหาผิดพลาดของ superimposed text

ตารางที่ 5.1 ตารางเปรียบเทียบ PR กับ RE ในขั้นตอนการหาขอบภายในอัลกอริทึม

A method of Edge Detectious	Recall	Precision
Sobel	93.55%	69.18%
Laplacian of Gaussian	46.77%	24.15%
Robert	80.65%	62.08%
Prewitt	82.26%	55.53%
Cann	33.87%	17.13%

จากผลการทดลองพบว่า เปอร์เซนต์ของ Recall ที่ออกมาสูงสุดคือ Sobel > Canny > Prewitt > Robert > Laplacian of Gaussian และ Precision สูงสุด ก็เป็น Sobel เช่นกัน การวัดผลต่อไปคือ การวัดประสิทธิภาพของ Algorithm ที่ได้คิดค้นขึ้นมา โดยใช้การวัดผลแบบ Recall และ Precision เหมือนกับตารางที่ 5.1

ตารางที่ 5.2 ตารางเปรียบเทียบ PR กับ RE ในวิดีโอกีฬาต่างๆ กัน

Spors	Recall	Precision
Soccer	91.3%	71.2%
Basketball	93.5%	69.2%
Baseball	95.1%	72.3%
snooker	95.7 %	70.5%
Tennis	87.7%	73.5%
Badminton	90.2%	60.9%

โดยวัดผลจาก รายการทีวีกีฬาชนิดต่างๆ คือ ฟุตบอล, บาสเกตบอล, เบสบอล, สนุกเกอร์, เทนนิส และ แบดมินตัน โดยเป็นเทปบันทึกจากรายการทีวีในประเทศ และ ต่างประเทศ โดยเปอร์เซนต์ Recall ค่อนข้างสูง แต่ พบว่า Precision นั้น ยังคงอยู่ระดับปานกลาง

การเพิ่มประสิทธิภาพด้วยการเพิ่ม Adaptive Threshold ภายในขั้นตอน Sobel Edge Detection

จากการทดลองในตารางที่ 5.2 นั้นพบว่า ความผิดพลาดของ Superimposed Text Detection ยังมีค่าค่อนข้างสูงซึ่งสังเกตได้จาก Precision ซึ่งมีค่าเฉลี่ย อยู่ประมาณ 69.6% และมีค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ Recall เท่ากับ 92.5% ซึ่งทางผู้พัฒนาได้มีการทดลองโดยปรับแก้ขั้นตอนภายใน Sobel Edge Detection เพื่อเลือกเฉพาะขอบที่อยู่เป็นกลุ่มก้อนมากยิ่งขึ้น โดยเข้าไปแทรกในหลังจากการทำ gradient ทั้งแกน x และ แกน y โดยการกรองค่าที่คาดว่าจะพื้นหลังออกไป

จากขั้นตอนภายใน Sobel นั้น หลังจากการรวมกันของ gradient แกน x และแกน y แล้วจะมีการทำ Thinning เพื่อบ่งบอกให้เห็นเส้นขอบที่ชัดเจน แต่จากพบว่าวิธีการค้นหา Superimposed text นั้นไม่จำเป็น แต่ผู้พัฒนาได้สร้างฟิสเตอร์กรองค่าที่มีความน่าจะเป็นว่าเป็น superimposed text เพื่อแยกออกจากพื้นหลัง โดยค่า Threshold นั้นมีค่าเท่ากับ

$$GxThreshold = \frac{Gx_{Max} - Gx_{Min}}{2} \quad (5.2)$$

$$GyThreshold = \frac{Gy_{Max} - Gy_{Min}}{2} \quad (5.3)$$

จากสมการ 5.2 และ 5.3 นั้นเป็นการหา Threshold ซึ่งอยู่ในลักษณะ Adaptive threshold ทำให้กรองค่าที่คาดว่าจะจะเป็นเฉพาะ superimposed text เท่านั้น โดยได้ทดลองกับ 3 วิดีโอ คือ ฟุตบอล, บาสเกตบอล และ เบสบอล

ตารางที่ 5.3 ตารางเปรียบเทียบการใช้ Adaptive Threshold ภายในขั้นตอน Sobel Edge Detection

วิดีโอ	Recall rate		Precision rate	
	Th	non-Th	Th	non-Th
เบสบอล	96.7%	95.1%	68.1%	62.2%
ฟุตบอล	98.8%	91.3%	88.5%	71.2%
บาสเกตบอล	97.52	93.5%	79.5%	69.2%

จากตารางที่ 5.3 นั้น หลังจากทำฟิวเจอร์ด้วย Adaptive Threshold ในขั้นตอนระหว่างการหาขอบด้วย Sobel edge นั้นพบว่า ทำให้มีประสิทธิภาพการค้นหาสูงมากยิ่งขึ้น โดยมี Recall rate เฉลี่ย คือ 97.67% และ precision rate คือ 78.72%

## บทที่ 6

# บทวิจารณ์และสรุป

### 6.1 บทสรุป

การแนวทางการพัฒนานั้น ได้แบ่งออกเป็น 2 แนวทางคือ พัฒนาการทางซอฟต์แวร์ และ พัฒนาการทางฮาร์ดแวร์ ซึ่งการพัฒนาและการทดลอง รวมถึง Algorithm นั้นมีความแตกต่างกัน และมีจุดประสงค์ที่แตกต่างกันออกไป ซึ่งการพัฒนาทางซอฟต์แวร์ จะเป็นการพัฒนาโปรแกรมจำลองการค้นหาลากบนวิดีโอโดยอ้างอิงจากผู้ใช้งานโดยตรง ซึ่งเป็นการตอบสนองปัญหาของผู้ใช้งาน ในการค้นหาที่ต้องการ เพื่อลดเวลาในการชมรายการโทรทัศน์ต่าง ๆ โดยเฉพาะเกมกีฬา ซึ่งแฟนคลับนั้น จะอยากดูเฉพาะฉากที่สำคัญ ๆ เท่านั้น โปรแกรมค้นหาลากนั้น จะถูกกำหนดการค้นหา โดยมีผู้ใช้นั้น ได้ออกแบบเอง ซึ่งเหตุผลที่ให้ผู้ใช้ออกแบบเองนั้น ก็คือ เพื่อให้ได้กับความต้องการของตนเองมากที่สุด ซึ่งบางครั้ง ความต้องการนั้น ค่อนข้างที่จะซับซ้อนและหลายเงื่อนไข ที่แตกต่างกันออกไป โดยในที่นี้ได้มีการออกแบบ Algorithm ในการจับคู่ pattern กับ วิดีโอไฟล์ และมีการกำหนดจุดที่ตรงตาม pattern ไว้บนเฟรมนั้น ๆ ซึ่งทำให้ผู้ใช้งานสามารถที่จะเข้าถึงได้โดยตรง หรือ เป็นการกระโดดไปยังฉากที่เราต้องการได้ทันที โดยปราศจากการกด forward และ backward ในรีโมท ซึ่งการจำลองการออกแบบ pattern ของผู้ใช้นั้น จะจำลองคล้ายกับ ผู้ใช้ได้ออกแบบด้วยรีโมทคอนโทรลโดยการทดลองนั้น ได้ทดลองกับวิดีโอการแข่งขันกีฬาเบสบอลในประเทศญี่ปุ่น, วิดีโอฟุตบอลจาก ESPN และ วิดีโอเบดมินตัน ซึ่งผลจากการทดสอบนั้น ยังมีตรวจจับผิดพลาด เนื่องจาก อัลกอริทึมของ Pattern Matching นั้นยังคงเกิด error โดยเฉพาะในกรณีตรวจจับวัตถุเล็ก แต่เมื่อเทียบกับภาพที่หาได้นั้น ถือว่า ยังอยู่ในเกณฑ์ดี

ในส่วนของฮาร์ดแวร์นั้น เริ่มจากการตั้งข้อสันนิษฐานถึงว่า Superimposed Text นั้น เป็นวัตถุกราฟิกตัวอักษรที่มีประโยชน์ในการค้นหาเป็นอย่างมาก เพื่อสนับสนุนการพัฒนาของซอฟต์แวร์ ก็ได้มีการคิดค้นอัลกอริทึมตรวจจับวัตถุนี้อย่างอัตโนมัติ โดยมีขั้นตอน คือ การค้นหา (Detection) การบอกพิกัดตำแหน่ง (Localization) และ การตัดทอนพื้นที่หลัง (Extraction) ซึ่ง Algorithm ที่ได้ออกแบบไปนั้น ได้จากการสังเกตของผู้จัดทำ และได้มีการทดสอบประสิทธิภาพด้วย Simulink โดยเปอร์เซ็นต์ที่ออกมา พบว่าจากแหล่งวิดีโอที่ใช้ทดสอบ จะมี Recall ที่สูง แต่ มี Precision ปานกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก ข้อสันนิษฐานของอัลกอริทึมนี้ไม่สามารถใช้ได้กับวิดีโอ ลักษณะกล้องนิ่งได้ เช่น การ สัมภาษณ์ข่าว, ห้องส่ง หรือ กีฬาจ่าพวก เทนนิส, สนุกเกอร์ เป็นต้น ดังนั้น ทางทีมพัฒนาจึงมีการคิดค้น การพัฒนาเปอร์เซ็นต์ของ Precision ให้สูงมากยิ่งขึ้น โดยการเสริม Threshold ในส่วนภายในของ Sobel Edge Detection ซึ่งค่าของ Threshold ได้จากการทดลอง ทำให้ประสิทธิภาพของอัลกอริทึมสูงมากขึ้น คือ ค่า Precision สูงมากขึ้น ซึ่งหมายความว่า มีข้อผิดพลาดน้อยลง

หลังจากที่ทดสอบอัลกอริทึมกับ Simulink แล้วจึงมีการศึกษาเพื่อที่จะทดลองกับบอร์ด ทดลองรุ่น XUP Virtex-II Pro Develop ของ Xilinx ซึ่งเครื่องมือที่นำมาพัฒนาร่วมด้วยนั้น ทางผู้พัฒนา ได้เลือกไว้คือ System Generator และ EDK 8.1 แต่จากการศึกษา และ ทดสอบ พบว่า ในบอร์ด XUP Virtex-II Pro นั้นไม่รองรับกับ System Generator (ไม่สามารถจัดการกับ DDR Ram ซึ่งระบบแชนร์ หน่วยความจำของ Microblaze Processor นั้นไม่เพียงพอที่จะใช้พัฒนางานในโครงการนี้) ซึ่งทำให้ทาง ผู้พัฒนาหันมาใช้เครื่องมือ EDK 8.1 ในการพัฒนาแทน จากการออกแบบระบบผ่านทาง EDK โดยใช้ PowerPC Processor เป็น Core หลักในการทำงาน ซึ่งเป็นการทำงานในรูปแบบ Hardware Software co-design ผลลัพธ์สามารถ implement super imposed text detection algorithm ลงไปในบอร์ดทดลองได้ สำเร็จ และสามารถแสดงหน้าจอผลลัพธ์ออกมายังหน้าจอคอมพิวเตอร์ซึ่งได้ผลลัพธ์ตามที่ต้องการ เหมือนกับการทดลองใน Simulink แต่ทั้งนี้ทั้งนั้น ระบบยังไม่สามารถ Detect ได้จากไฟสวิตช์โอโดยตรง เพราะในการทดลองนี้ ผู้พัฒนาได้ทำการ implement algorithm ลงไปเพื่อใช้งานกับภาพ Bitmap จำนวน 50 รูป ซึ่งได้ผลลัพธ์ออกมาเหมือนกับทำใน Simulink แต่ไม่สามารถค้นหาได้หลายๆเฟรมดังเช่นที่ Simulink ทำได้ ส่วนเรื่องความเร็วของการทำงานจากบอร์ดทดลองนั้นช้ากว่าการทำงานของ Simulink ถึง 25-30 เท่า ซึ่งผลลัพธ์ในระดับไม่สามารถเรียกว่า Real-Time ได้เลย จึงสรุปได้ว่าการ implement ลง บอร์ดทดลองเพื่อเพิ่มความเร็วนั้น ไม่ประสบผลสำเร็จ

## 6.2 วิจารณ์สิ่งที่ได้จากโครงการ

จากการจัดทำโครงการนี้นั้น สิ่งที่ได้จากโครงการนี้ มีลำดับขั้นตอนดังนี้

- 6.2.1 การศึกษาลักษณะของรายการโทรทัศน์ ข่าว กีฬา โดยสังเกตสิ่งที่สามารถนำมาเป็น ภัยหลักในการวิจัย รวมถึงตั้งสมมุติฐานจากปัญหาของการวิจัย
- 6.2.2 ศึกษาแนวทางการทำวิจัย และ คิดค้น Algorithm ได้
- 6.2.3 ศึกษาพื้นฐานของ Image processing และ Video processing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6.2.4 ศึกษาการเขียนโปรแกรม ด้วย MATLAB และ Simulink
- 6.2.5 ศึกษาการเขียนโปรแกรมด้วย VC++.net ร่วมกับ OpenCV Library (open source)
- 6.2.6 ศึกษาแนวทางการออกแบบ User Interface
- 6.2.7 ศึกษาพื้นฐานของ FPGA
- 6.2.8 ศึกษา System Generator เพื่อออกแบบและพัฒนาบน FPGA
- 6.2.9 ศึกษา EDK เพื่อออกแบบและพัฒนาบน FPGA
- 6.2.10 ตีพิมพ์บทความงานวิจัยได้

จากที่กล่าวมาทั้งหมดนี้ เป็นสิ่งที่ได้รับการวิจัยในโครงการ โดยจะแบ่งออกเป็น 2 แนวทาง คือ ด้านฮาร์ดแวร์ ประกอบด้วย MATLAB, Simulink, System Generator, EDK และ FPGA ส่วนด้านซอฟต์แวร์นั้นประกอบด้วย VC++.net และ OpenCV ซึ่งนำมาพัฒนาโปรแกรมการค้นหาฉากโดยอ้างอิงจากผู้ใช้ในรายการโทรทัศน์ ส่วนเทคนิคการค้นหา Superimposed Text ในวิดีโอ ซึ่งนับว่าสิ่งที่ได้ทำขึ้นมา นั้น เป็นสิ่งที่คิดค้นถึงมาเองและแตกต่างกับงานวิจัยของคนอื่น ๆ ที่ได้ทำในอดีตที่ผ่านมา ซึ่งจุดนี้เป็นการฝึกการทำงานวิจัย ฝึกคิด ฝึกสังเกต แล้วนำมาพัฒนาใช้งาน

### 6.3 ปัญหาอุปสรรค และ แนวทางการแก้ไข

ปัญหาหระหว่างที่จัดทำโครงการ โดยจะแบ่งแยกออกในส่วนของการคิดค้นและออกแบบ, การพัฒนาซอฟต์แวร์, การพัฒนาทางด้านฮาร์ดแวร์ และการค้นหาแหล่งข้อมูลสำหรับการทดสอบ

การคิดค้นและออกแบบ เป็นขั้นตอนที่ต้องเกิดการลองผิดลองถูก และต้องมีการสืบค้นเอกสารงานวิจัยต่างๆ ที่ผ่านมา เนื่องจากโครงการนี้นำเสนอในรูปแบบการวิจัยที่ไม่ได้มุ่งเน้นในการสร้างผลิตภัณฑ์ขึ้นมา ความแตกต่างของงานวิจัยถือเป็นสิ่งสำคัญ และ สิ่งนี้เองนับว่าเป็นสิ่งที่ยากในการคิดค้นและออกแบบ การออกแบบนั้นจะต้องมีการทดสอบ ในที่นี้ ได้ใช้เครื่องมือในการทดสอบคือ VC++.net และ MATLAB รวมไปถึงโปรแกรมสำเร็จรูปอื่นๆ นอกจากนั้นยังคงต้องศึกษาเรื่อง Image & Video processing ซึ่งปัญหาที่สำคัญที่ยังคงเป็นปัญหาของทุกๆ งานวิจัยคือ การหาค่า Threshold ที่ดีที่สุด ซึ่งค่าที่ได้มานั้น ได้จากการทดลอง นับว่าต้องเสียเวลาเป็นอย่างมากที่จะต้องใช้เวลาในการทดสอบเพื่อหาค่า Threshold ที่ดีที่สุดเพื่อนำมาพัฒนาในส่วนอื่นๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์ ในขั้นตอนนี้ปัญหาส่วนใหญ่จะเกิดขึ้นในเรื่องของเทคนิค ซึ่งในที่นี้ได้ใช้ VC++.net กับไลบรารี OpenCV ต้องอาศัยการเรียนรู้เรื่องการแตกเรดเข้ามาช่วย เพื่อให้ผู้ใช้ติดต่อกับ UI ได้งาน ปัญหาที่สำคัญที่สุดในขั้นตอนนี้ คือ การนำวิดีโอมาแสดงผล ซึ่งโดยปกติแล้วนั้นวิดีโอก็มีขนาดแตกต่างกันออกไป แต่โปรแกรมต้องการสร้างกรอบการแสดงผลแบบคงที่ ดังนั้น จะต้องมีการสร้างตัวแปลงขนาดของวิดีโอให้พอดีกับกรอบคงที่ของโปรแกรมโดยอาศัยเทคนิคเทียบอัตราส่วน ซึ่งนับว่าเป็นขั้นตอนที่ค่อนข้างใช้เวลาในการพัฒนาในส่วนนี้มากที่สุด

การพัฒนาฮาร์ดแวร์ นับว่าเป็นขั้นตอนที่มีความยากที่สุดในระหว่างการทำโครงการนี้ เนื่องจากบอร์ดทดลองนั้นได้นำเข้ามาจากประเทศสหรัฐอเมริกา ซึ่งไม่มีขายในประเทศไทย ทำให้ผู้ที่ศึกษาและเอกสารต่างๆ หาได้ยาก และไม่มีผู้ให้คำแนะนำเกี่ยวกับการพัฒนาโดยใช้บอร์ดนี้ได้ ดังนั้นผู้พัฒนาต้องมีการลองผิดลองถูก โดยเริ่มจากการใช้ System Generator เข้ามาพัฒนา แต่กลับพบว่าทางบอร์ดไม่รองรับการใช้งานร่วมกับ DDR RAM ซึ่งในโครงการเป็นการประมวลผลภาพ ต้องการหน่วยความจำมาก ทำให้ทางผู้พัฒนาเองเสียเวลาไปกับการศึกษา System Generator เป็นอย่างมาก แต่ในท้ายที่สุดนั้นก็ก็ได้ศึกษา EDK เข้ามาแทน แต่ปัญหาที่พบคือ ไม่มีเอกสารให้คำแนะนำวิธีใช้อย่างละเอียด ซึ่งทำให้ผู้พัฒนาต้องใช้ความพยายามเป็นอย่างมากในการศึกษาวิธีใช้งานบอร์ดทดลองนี้

การหาแหล่งข้อมูลเพื่อทำการทดลอง ปัญหานี้เป็นปัญหาเรื่องของ source ที่จะนำมาทดสอบงานวิจัย ซึ่งความต้องการนั้นต้องการวิดีโอที่มีความหลากหลายและมาจากหลายๆ ที่ ซึ่งแน่นอนว่ายากต่อการค้นหา หรือ การอัดจากรายการทีวีนั้นก็มีความยุ่งยากค่อนข้างมาก

#### 6.4 แนวทางการพัฒนาต่อ

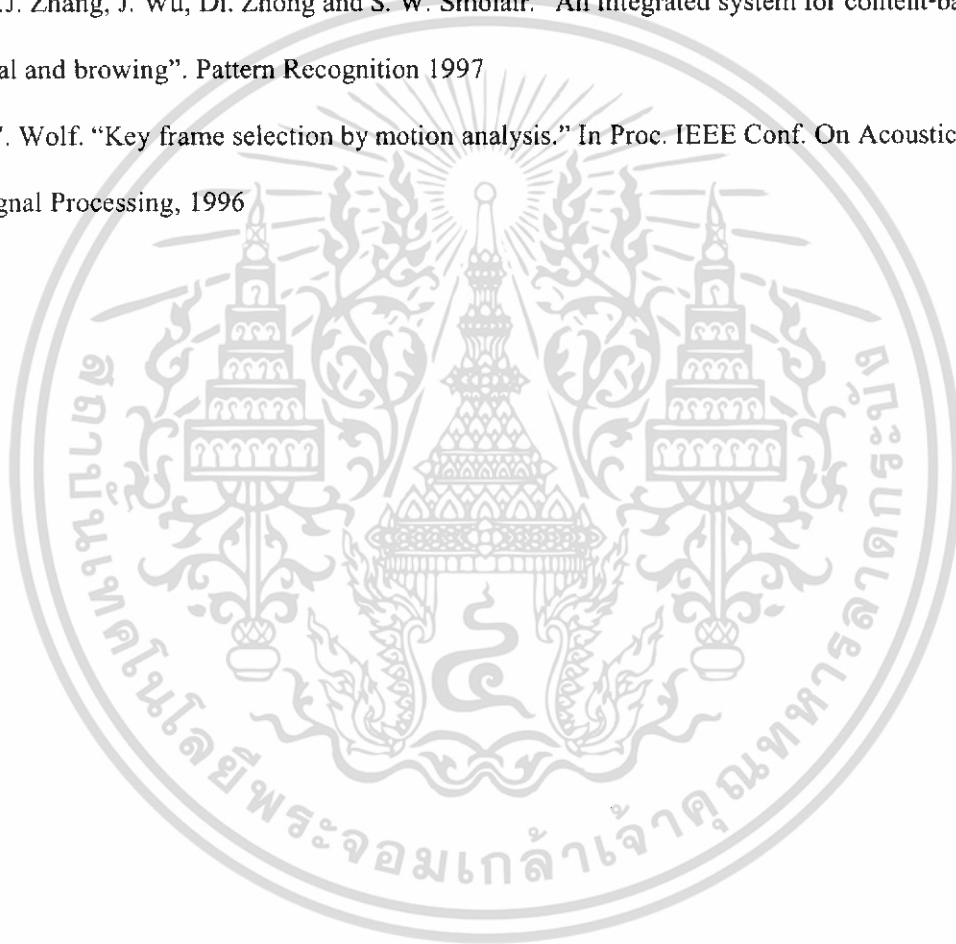
การพัฒนาต่อนั้น ผู้พัฒนามีแนวคิดที่จะนำทั้งซอฟต์แวร์และฮาร์ดแวร์เข้ามาใช้งานร่วมกัน โดยสามารถติดต่อกับเครื่องเล่นวิดีโอโดยตรง หรือ ต่อตรงจากทีวี และมีการจัดเก็บสู่ฐานข้อมูลในเครื่องคอมพิวเตอร์ โดยผ่านกระบวนการตัด Shot และการดึงลักษณะ (Feature Extraction) จากวิดีโอออกมาเพื่อมาใช้ในการค้นหา อีกทั้งเป็นการเพิ่มประสิทธิภาพในการค้นหาของตัวโปรแกรมได้ทำงานได้อย่างรวดเร็วมากยิ่งขึ้น และการทำงานของฮาร์ดแวร์นั้นต้องมีการพัฒนาให้สามารถใช้เวลาในการประมวลผลได้เร็วมากยิ่งขึ้น จนสามารถใช้งานในรูปแบบ Real time ได้

## เอกสารอ้างอิง

- [1] N. Babaguchi, Y.Kawai and T. Kitahashi. "Event Based Indexing of Broadcasted Sport Video by Intermodal Collaboration" IEEE Transactions on Multimedia, Vol 4, NO.1, March d2002
- [2] R.Leonardi, P. Miglisorati and M. Prandini. "Semantic Indexing of Soccer Audio-Visual Sequences: A Multimodal Approach Based on Controlled Markov Chins" IEEE Transactions on Circuits and System for video technology, VOL.14, NO. 5, May 2004
- [3] D.Tijondronegoro , Yi-Ping P. Chen and B. Pham. "Integrating Highlights for More Complete Sports Video Summarization" IEEE Multimedia [Article] October – December 2004, P.22 – 37
- [4] F.Todsaporn, K. Nakayama, Y. Kobayashi and M. Maekawa. "End-user Customizable Scene Search for TV Programs" Multimedia technology, ISCIT 2005, China, October 2005
- [5] OpenCV Tutorial by Robert Laganier, VIVA lab, University of Ottawa [Website]Available: <http://www.site.uottawa.ca/~laganier/tutorial/opencv+directshow/cvision.htm>
- [6] Rafael C. Gonzalez , Richard E. Woods and Steven L. Eddins. 2004 "Digital Image Processing using MATLAB" Pearson Education, ISBN 0-13-008519-7.
- [7] D.Chen, H. Bourlard and Jean-Philippe T. "Text Identification in Complex Background Using SVM" IEEE computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01) 2001
- [8] Mike Blaszcak 1999. "Professional MFC with Visual C++ 6" Wrox Press Ltd Published, ISBN 1-861000-15-4
- [9] Xilinx. "Xilinx University Program Virtex-II Pro Development System UG069". Version 1.0. March 2005.
- [10] Xilinx. "System Generator for DSP". Version 8.2. User's Guide. July 2006.
- [11] Xilinx. "Embedded System Tools Reference Manual: Embedded Development Kit EDK 8.1i UG111". Version 5.0. October 2005.
- [12] Xilinx. "Platform Specification Format Reference Manual : Embedded Development Kit EDK 8.1i UG131". Version .3.0. October 2005.
- [13] Xilinx. "LibXil FATFile System(FATfs)". October 2005.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] The MathWorks, Inc. "Simulink Simulation and Model-Based Design". March 2006.
- [15] The MathWorks, Inc. "Video and Image Processing Blockset For Use with Simulink". March 2006.
- [16] Alan F. SMEATON. "Technique Used and Open Challenges to Analysis, Indexing and Retrieval of Digital Video". Journal of Elsevier Science. 2006
- [17] N. Diamitrova, H.J. Zhang, B. Shahraray, I. Sezan, T. Huang and A. Zakhor. "Application of Video-Content Analysis and Retrieval" IEEE multimedia journal. 2002.
- [18] H.J. Zhang, J. Wu, Di. Zhong and S. W. Smolair. "An integrated system for content-based video retrieval and browsing". Pattern Recognition 1997
- [19] W. Wolf. "Key frame selection by motion analysis." In Proc. IEEE Conf. On Acoustics, Speech, and Signal Processing, 1996



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# A Quick Scene Search with Constructed Mapped Charts for TV Sport Programs

Todsaporn Fuangrod

Department of Computer Engineering  
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang  
Bangkok, Thailand 10520. Email: s5010298@kmitl.ac.th

Amnach Khawne

Department of Computer Engineering  
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang  
Bangkok, Thailand 10520. Email: kkamnach@kmitl.ac.th

**Abstract**— This paper presents the fast scene search algorithm using fixed-image regions of the superimposed-text in a video. Our scheme focuses on the fixed-image region of the superimposed-text which is often an intuitive clue for an end-user for specifying a certain frame in a sport video. The video scene, consisting of superimposed-text detection using MFI algorithm, and Mapped Chart (MC) which constructed the index of a pattern matching, will be searched by the user. Our method found that the searching method with MC was faster (reduced the time consuming more than 50%) and more robust.

## I. INTRODUCTION

In recent year, the video technology has been developing for multimedia contents, especially in the sport video. The technology for the fast search, in video scenes, is crucial demand of sport fan clubs and sport coaches. Sport videos that contain some fixed background scenes are well planned and organized, and have similar meaning information. However, the most viewers, watching the sport video, usually want to skip unwanted scenes in the video. The video summarization [1] and Video Indexing [2] are the existing ways of this research fields. In the literature, the video summarization and the video indexing, however, still have a limitation of languages, huge storage and difficult to implement. Our survey techniques, e.g., text recognition [3][4], Face recognition [5], and speech recognition [6] are often used in this area of the video searching. Thus, this paper focuses on the system that can be used in the world-wide, and that can be easily implemented in the real world. The superimposed-text is crucial information for the scene summarization. Thus, we are interested in fixed-image regions of the superimposed-text which is often intuitive for an end-user for specifying the certain frame in the sport video. This paper is firstly described our proposed method in section 2. Definable patterns will describe in section 3. The superimposed-text classification (section 4) will briefly introduce Multi-Frame Integration (MFI), mostly found from [7] by R.Wang. It is not only importance of fixed-image regions but also crucial plays a major and fundamental role. The fast searching algorithm is proposed by using the Mapped Chart. The rest of this paper is organized as follows. The pattern matching is shown in the section 5. Experiments and discussions described in the section 6. Conclusions and future work described in the section 7.

## II. PROPOSED METHOD

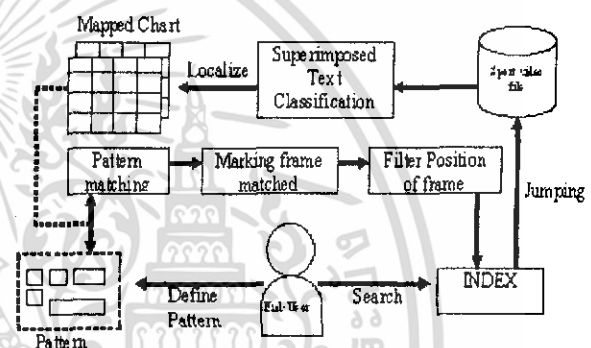


Fig. 1. Block diagram of system

Following figure 1, the end-user can define the patterns for searching the interested scenes. Then, the pattern is defined is desirable by the user, and the pattern depends up on the knowledge of each sport category. Some sport types have complicated regulation and environment, such as baseball, tennis, golf, etc. In addition, the definable pattern has to be the majority referred to the dominant knowledge of the end-user to define the pattern. Moreover, access every frame in the video file made a laden method for searching. We offer the superimposed-text classification method for cluttering the sport video shot which contains the superimposed-text, and we create the Mapped Chart for updating the superimposed-text position. The Mapped Chart is used to reduce the consumed time for searching. The pattern matching stage is used to find the matching scene by using the template matching technique. The filtering marking stage which is used to change the sequence of match scenes become a position for the user who can be searched that scene. The position called key frame is used for the user who can search the scene by "jumping".

## III. DEFINE PATTERNS

From our observation, the fixed-image region of the superimposed-text can be suitable pattern for using in the sport video searching. Our interested areas are information on the superimposed-text which is crucial information. Format of the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

superimposed-text will be occurred in the same position during the game, including positions, colors, and styles as well. In the different frame, some fixed regions of the superimposed-text describe the meaning of the scene. The example of the superimposed-text has shown in Figures. 2, (a) soccer (b) baseball (c) tennis

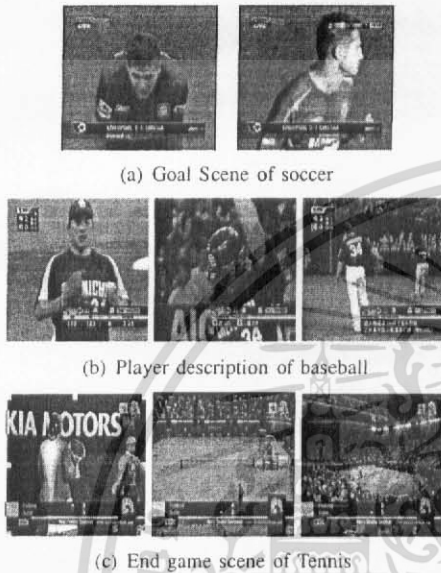


Fig. 2. Example of scenes which have fixed-image regions of the superimposed-text is in the video files. (a) Soccer (b) Baseball (c) Tennis

The define pattern is an important stage as the depth of details of the search, depending on pattern. Domain knowledge of a sport will help the user to understand how defined pattern are. In our previous research, we recommended two options of defining the pattern. The first method is manually selected texts or symbols as a pattern by searching the desired pattern using a rectangular search box. The second method [8] is directly to select a pattern using a tool which automatically assists users to find their specific patterns. When the user search for a proper pattern in a TV program, the second method is recommended that it can be advantageous over the first method when the user selects an optimal pattern.

#### IV. SUPERIMPOSED-TEXT CLASSIFICATION

##### A. Random Frame Candidate

In a common real-life video, the super-imposed text will approximately appear for 2 seconds, i.e., 60 consecutive frames in 2 seconds. Thus we will process every 2 seconds for the candidate frame. Normally, 30 frames per second have similar to the frame inside. If we process all of 60 frames, the time consuming takes a long time. The optimal process exists before we find the superimposed-text called Random Frame Candidate (RFC), so we randomly choose 25 frames from 60 frames. In our experiment shown that 25 frames process and 60 frames process are the same and this stage will reduce the time consuming. Block diagram of the Superimposed-Text Detection shown in Figure 3.

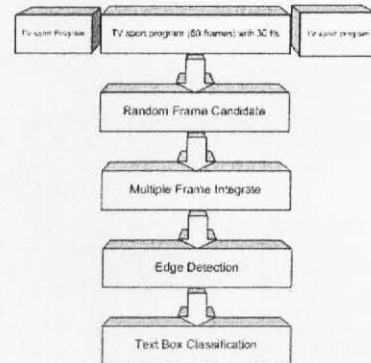


Fig. 3. Block diagram of the Superimposed-Text Detection

##### B. Multi-Frame Integration (MFI) [7]

After we are random frames from the candidate stage, the initially selected frame is converted into grayscale. Then, the 25 frames grayscale frames will be integrated into a minimum and maximum grayscale image using the equation in (1) and (2). The result of this stage has shown in Figure 4.

$$\text{MinIMAGE}_i(x, y) = \min_{j \in C_i} (p_j(x, y)) \quad (1)$$

$$\text{MaxIMAGE}_i(x, y) = \max_{j \in C_i} (p_j(x, y)) \quad (2)$$

Where  $p_j(x, y)$  is the pixel value at the same location, and  $C_i$  is amount of collected frames.



Fig. 4. Example of  $\text{MaxIMAGE}$  and  $\text{MinIMAGE}$  in tennis games

##### C. Sobel Edge Detection

After generating  $\text{MinIMAGE}_i$  and  $\text{MaxIMAGE}_i$ , we will employ edge detection on both of them. There are many ways to perform the edge detection. However, the well-known methods are Sobel, Prewitt, Robert, Canny and Laplacian of Gaussian. To find the best method of the superimposed text classification, we have to proof by using these methods in our test. We use precision and recall which commonly used in information retrieval (IR) to summarize the superimposed text detection results:

$$\text{Recall rate} = \frac{N_c}{N_c + M_m} \times 100\% \quad (3)$$

$$\text{Precision rate} = \frac{N_c}{N_c + M_f} \times 100\% \quad (4)$$

where  $N_c$  is a number of correctly detected Superimposed Texts,  $N_m$  is a number of miss detected Superimposed Texts, and  $N_f$  is a number of false detected Superimposed Texts respectively.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### D. Text box Classification

For the step of text classifications shown in Fig. 5, we use the  $M \times N$  ( $20 \times 10$  pixels in our system) window to scan the integrated image and classify each window as a text or a non-text. We choose  $M > N$  because video captions will always align horizontally. If the edge density of the window is larger than the predetermined threshold, it will be classified as a text; otherwise, it will be classified as a non-text. In our experiment, the threshold is empirically assigned to 0.70. We do not move the window at 1 pixel step, because the process will take more time consuming and it is not necessary. To consider the trade-off between the precision and the speed, we move the window 10 pixels horizontally and 5 pixels vertically at a time referred to [7] which used only  $MaxIMAGE_i$ ,

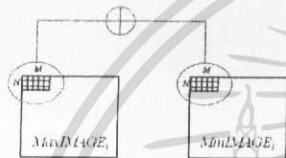


Fig. 5. Test box classification procedure

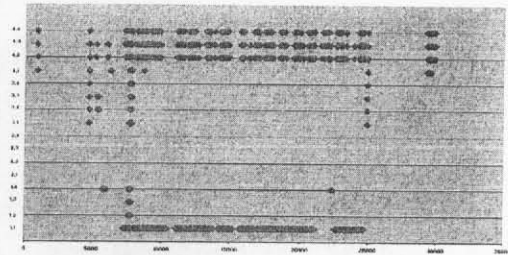
but our purposed algorithm used both of  $MaxIMAGE_i$  and  $MinIMAGE_i$  to calculate by using the AND operation. The results give us superlative more efficient than the previous algorithms. The reason for choosing the  $MaxIMAGE_i$  and  $MinIMAGE_i$  is that the result from Sobel edge detection in  $MaxIMAGE_i$  and  $MinIMAGE_i$  will be different but it should be almost the same in the superimposed area. The AND operation is used to reduce the error in the process of classification in the window scan ( $M \times N$ ). The result of the superimposed-text detection step is shown in Figure 6.



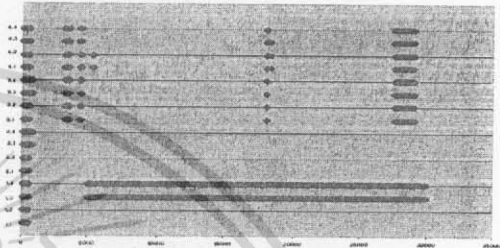
Fig. 6. Result of Superimposed-Text Detection.

#### E. Mapped Chart

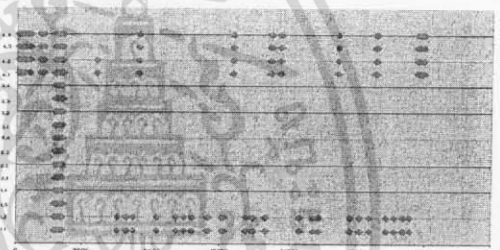
Each frame is divided into blocks of size  $4 \times 4$ . The objective of this stage is for a faster yet precise search. We construct the Mapped Chart for updating the position of the superimposed-text that is detected from the previous stage. The Mapped Chart consists of 2D graph which has timeline (horizontal) and block numbers (vertical). Block numbers are divide into 16 values ( $B_{1 \times 1}, B_{1 \times 2}, B_{1 \times 3}, \dots, B_{4 \times 4}$ ). The examples of mapping chart are shown in Figure 7.



(a) Mapped Chart of Baseball (20 Minutes)



(b) Mapped Chart of Soccer (20 Minutes)



(c) Mapped Chart of Tennis (20 Minutes)

Fig. 7. Mapped Chart of sport video

#### V. PATTERN MATCHING

Positions of the pattern defined by the end-user will be analyzed with corresponding Mapped Chart, and then positions of the pattern of blocks of size  $1 \times 1$  and  $1 \times 2$  are processed by the pattern matching. The process will be accessed only block  $1 \times 1$  and  $1 \times 2$  of the appeared superimposed-text. This approach has been reduced the time consuming and has correctly performed matching. The frames that do not have the superimposed-text were omitted. The system will compare a frame with an adjacent frame in a sequence and then compare with a set of frames of given target frames. We have a template  $g[i, j]$  and we wish to detect its instances in an image  $f[i, j]$  using (5). An obvious thing to do is to place the template at a location in an image and to detect its presence at that point by comparing intensity values in the template with the corresponding values in the image.

$$I = \begin{cases} 1 & ; \sum_{[i,j] \in R} (f_{i,j} - g_{i,j})^2 < Threshold \\ 0 & ; otherwise \end{cases} \quad (5)$$

If the value of  $I$  becomes 1, the position of the frame is positioned to the right. Afterwards, a template frame is used to compare and to locate the position of the frame. If a frame

เอกสารนี้เป็นเอกสารที่สวชนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

matches with template frame, then it produces a binary number of 1 or a mismatch produces 0 or off. It is processed every 60 frames per a time for the same reasons as in MFI. Sometimes an error, however, can occur when a text appears more than 2 seconds. Therefore, the position which allows the user to jump between frames will be marked when a mismatch (0) and a match (1) becomes.

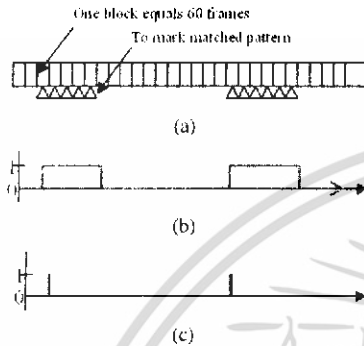


Fig. 8. Example of pattern matching (a) TV program movie file, (b) sequence of match, not match and (c) the result of filtering sequence.

## VI. EXPERIMENTS AND DISCUSSIONS

Our experiments were accurately performed to investigate of the proposed method. Video files consist of six types of sports such as soccer, baseball, tennis, basketball, badminton, and snooker. The videos, encoding in AVI format, were captured from broadcasting sport on television. Baseball has image size of  $512 \times 384$  and has frame rate 29 fps. Soccer and Tennis have image size of  $320 \times 256$  and have frame rate 30 fps. All simulations were on a standard Pentium (M) processor 1.60 GHz, memory 512 MB. We implemented the proposed algorithm using VC++.NET and OpenCV library (open source).

TABLE I  
RECALL AND PRECISION RATES OF EDGE DETECTION PHASE WITH TV BASKETBALL PROGRAM

A method of Edge Detections	Recall	Precision
Sobel	93.55%	69.18%
Laplacian of Gaussian	46.77%	24.15%
Robert	80.65%	62.08%
Prewitt	82.26%	55.53%
Cann	83.87%	17.13%

The result in Table I shown that the Sobel method, which has 93.55% recall rate and 69.18% precision rate, was the best method for our superimposed text classification. Even though Robert, Prewitt and Canny also have high recall rates but they gave low precision rate which means that false detected appear extremely. Thus, we choose the Sobel method in edge detection phase in superimposed text classification.

Table III shows the performance when the system implemented with the Mapped Chart. Refer to the result; the use of MC will be reduced the time consumed more than 50%, depending on each block in MC.

TABLE II

THE RESULT OF SUPERIMPOSED-TEXT DETECTIONS USING MFI

Spots	Recall	Precision
Soccer	91.3%	71.2%
Basketball	93.5%	69.2%
Baseball	95.1%	72.3%
snooker	95.7%	70.5%
Tennis	87.7%	73.5%
Badminton	90.2%	60.9%

TABLE III

MAPPED CHART(MC) PERFORMANCE WITH BASEBALL GAME (52.2 MINUTES)

Baseball	Non-MC	MC
Pattern1 (B1x1)	3.41	1.28
Pattern2 (B4x3)	3.41	1.34
Pattern3 (B3x2)	3.41	0.26

## VII. CONCLUSIONS AND FUTURE WORK

We have presented the novel system that can be used in the world-wide and also our method omits the recognition phase that spent a long time to process. The customizable scene search for sport video for the end-user [8] was easier and more alternative way. Following [8], we designed the system allowed the user to define the pattern by them. Unfortunately, we met the problem about the time-consumed and the performance when the user searches the desired frame every time. In this paper, we used the Mapped Chart to construct the index of the pattern matching and found that the searching with the MC was faster (reduce the time consuming more than 50%) and more robust. In the future, we would like to implement our proposed method into hardware using FPGA as the top box over the real programs on the television.

## REFERENCES

- [1] D. Tjondronegoro, Y. Phoebe Chem and B. Pham, "Integrating Highlights for More Complete Sports Video Summarizations," *Article of IEEE Society*, 2004.
- [2] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event Based Indexing of Broadcasted Sports Video by Intermodal Collaboration," *IEEE Transactions on Multimedia*, Vol.4, No. 1, March 2002.
- [3] R. Lienhart, and A. Wernicke, "Localizing and Segmenting Text in Images and Videos," *IEEE Trans. on Circuit and System for video Technology* Vol. 12, Nr. 4, S. 256-258, 2002.
- [4] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Transaction on Image Processing*, vol. 9, no. 1, pp. 147-156, January 2000.
- [5] D. Ya-Feng, S. Gunag-Da, Z. Jun and B. FU "Fast and Robust Face Detection in Video," *Conference on Machine Learning and Cybernetics*, Vol. 7, pp. 4577- 4582, Aug. 2005.
- [6] D. A. Sadlier and N. E. O'Connor, "Event Detection in Field Sport Video Using Audio-Visual Features and a Support Vector Machine," *IEEE Trans. on Circuits and System for Video Technology*, Volume 15, Issue 10, pp. 1225-1233, Oct. 2005.
- [7] R. Wang, W. Jin, L. Wu, "A Novel Video Caption Detection Approach Using Multi-Frame Integration," *Proceedings of the 17th International Conference on Pattern Recognition(ICPR'04)*, pp. 449-452, Volume 1, 2004.
- [8] T. Fuangrod, K. Nakayama, Y. Kobayashi and M. Maekawa, "End-user customizable scene search for TV programs," *International Symposium on Communications and Information Technologies 2005 (ISCIT 2005)*, October 12-14, 2005.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้