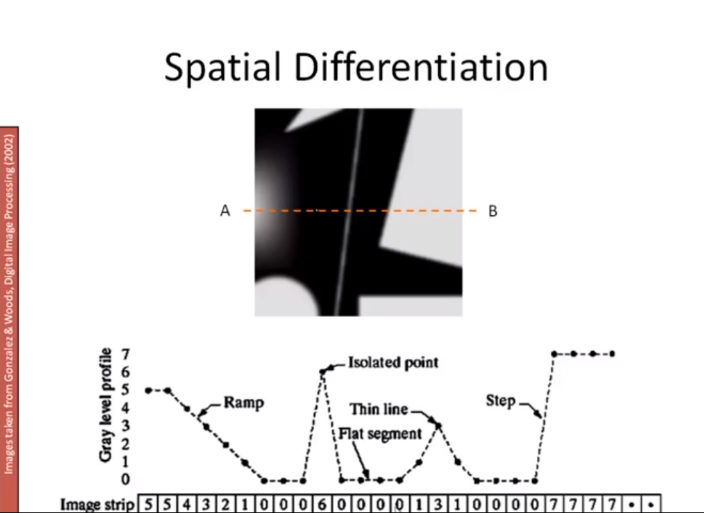
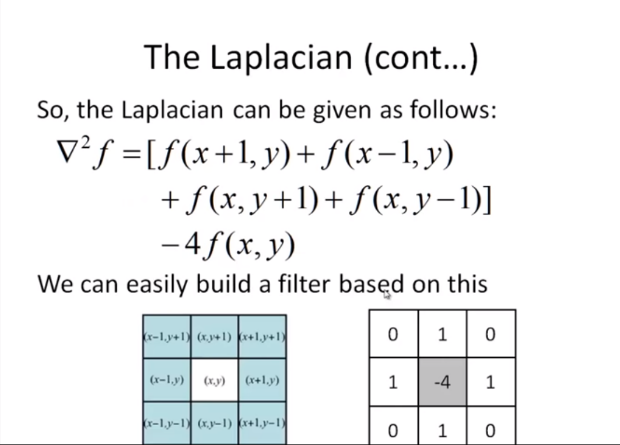
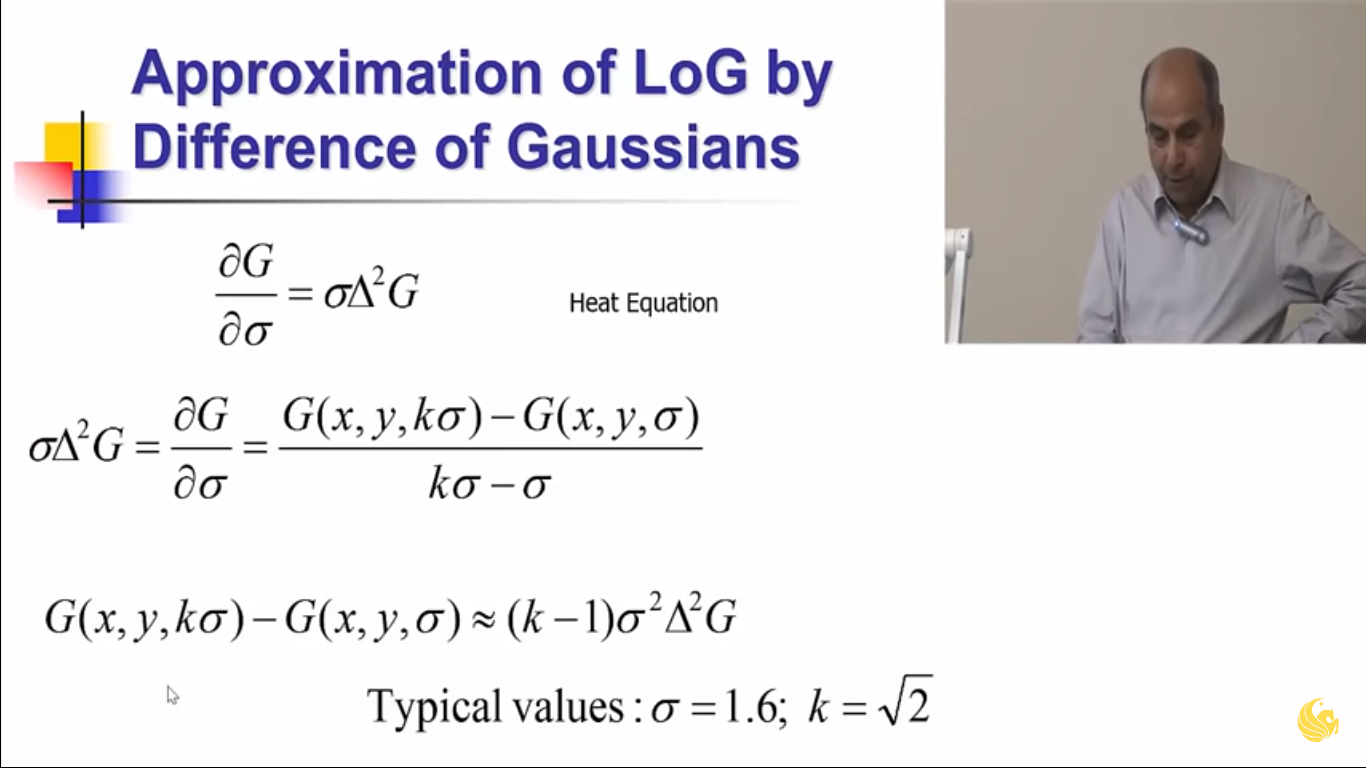
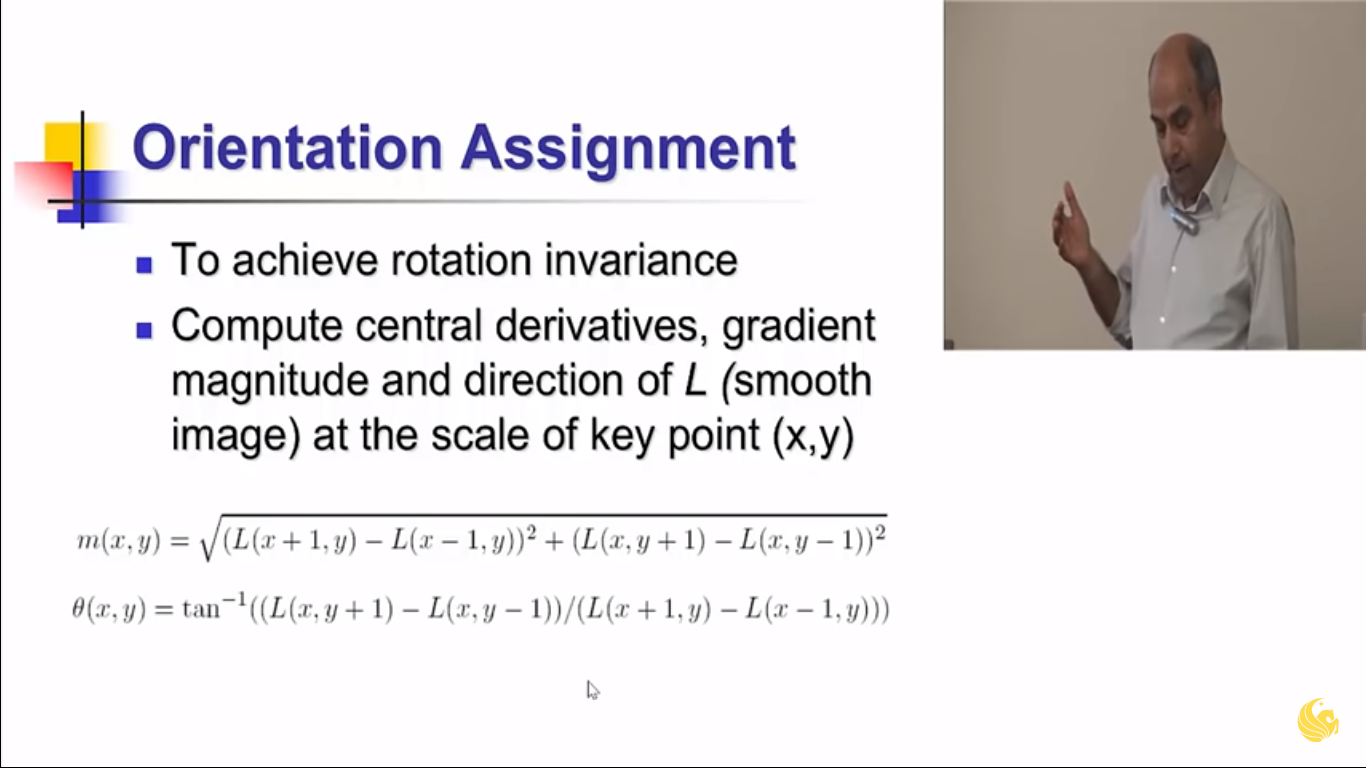
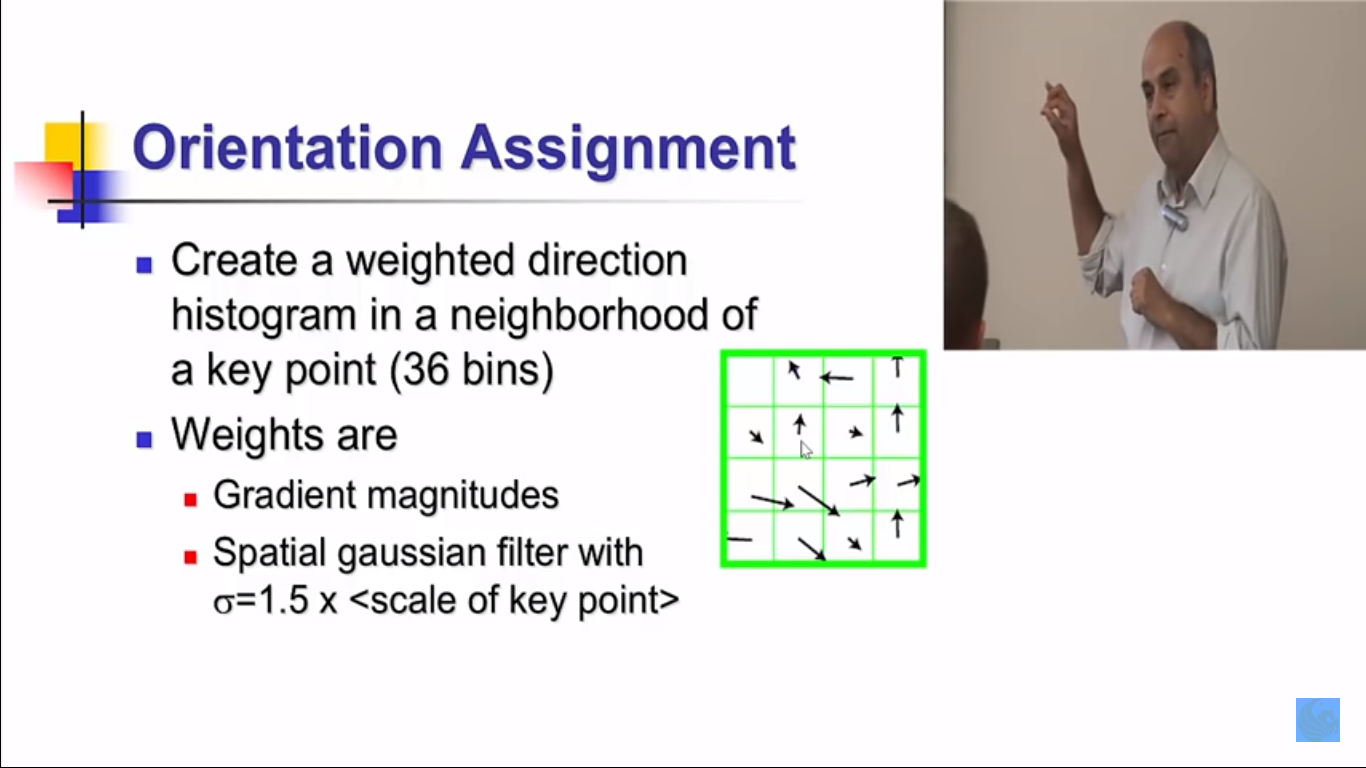
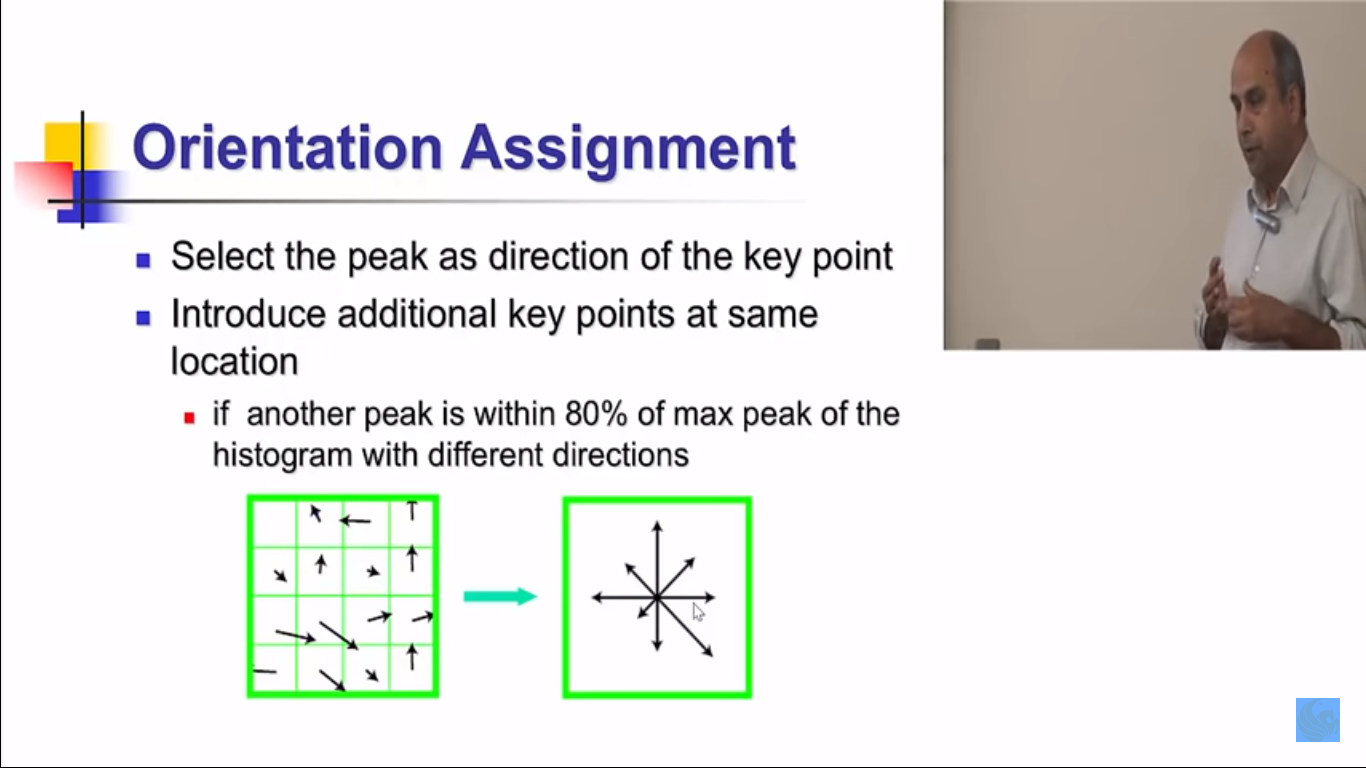
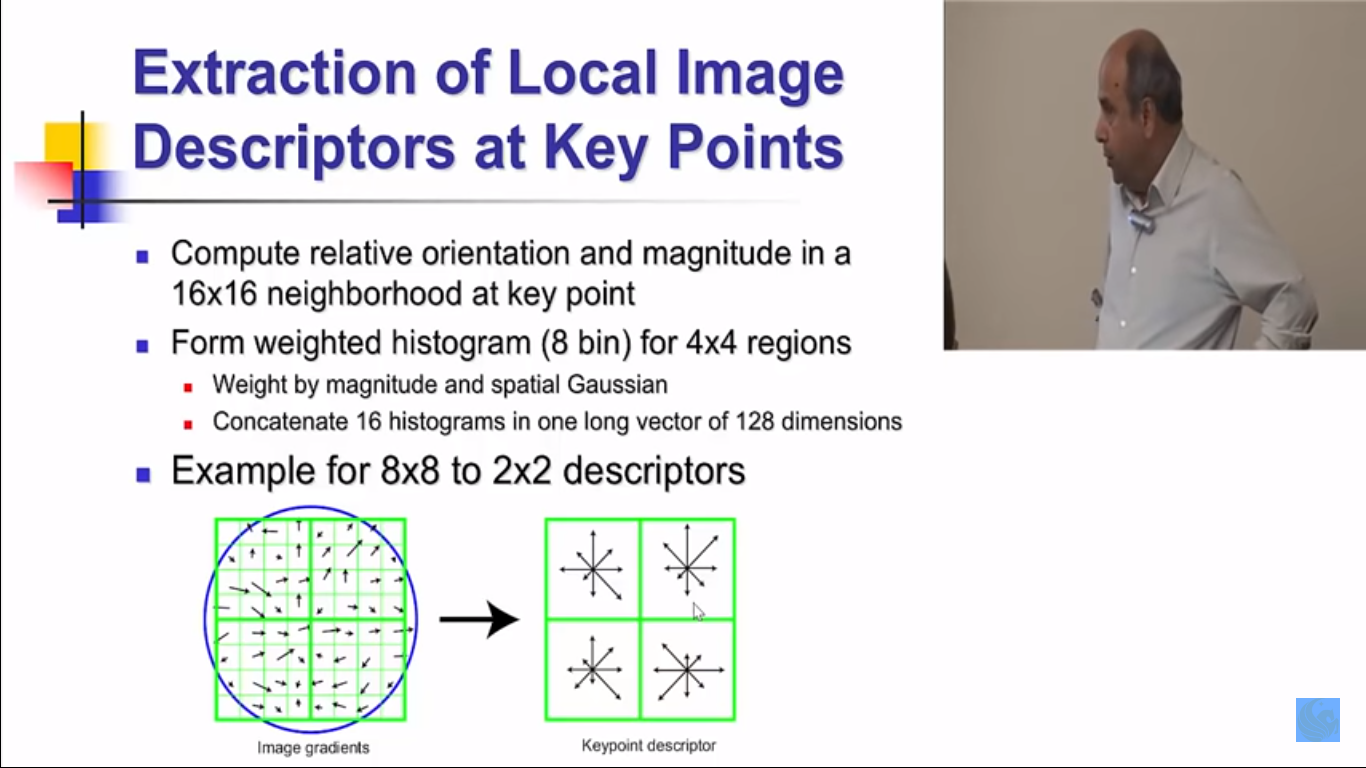
שלבים SIFT:

1. מציאת נקודות עניין.
2. המרה של נקודות העניין לDESCRIPTOR.
3. השוואה בין הDESCRIPTOS למציאת דמיון.
4. החזרת תוצאה לפי TRESHOLD (רמת דמיון).

שלבים למציאת נקודות עניין:

* נקודות עניין פוטנציאליות נמצאות לפי אזורים בתמונה עם השינויים הכי קיצוניים בצבעים.
* על מנת למצוא את האזורים עם השינויים הכי קיצוניים, SIFT משתמשים בשיטת Laplacian of Gaussian (LoG).
* השלבים לכך הם:

1. המרה של הפיקסלים לערכים שמייצגים את הפיקסל, כך ככל שההפרש בין המספרים גדול יותר, כך הקיצוניות ביניהן גדולה יותר.
   1. צריך לברר כיצד לבצע את ההמרה הזאת:
   2. 
   3. בדוגמא לעיל, אנחנו רואים כיצד ממירים שורה אחת של פיקסלים.
   4. נראה שהם מבצעים את ההמרה של לפי ה"GRAY LEVEL" שלו, כלומר מה המיקום שלו בטווח בין שחור - 0 ללבן - 7.
   5. התהליך מבוצע על כל השורות בתמונה.
2. מהטבלה שיצרנו, עוברים על כל ערך, מבצעים השוואה לערכים הקרובים אליו ומחשבים את רמת הקיצוניות בשינויים באמצעות LAPLACIAN:
   1. 
   2. בתמונה לעיל מתואר החישוב.
   3. אנחנו נקבל מטריצה של רמות הקיצון של השינויים בין הפיקסלים, בהתאמה לחישוב הLAPLACIAN.
3. נבצע "החלקה" של אזורי הקיצון, זאת על מנת לייצג את האזור הכללי של נקודת העניין. זאת על מנת שבמידה אותו אזור מצולם מזווית אחרת, ה"החלקה" תקרב בין נקודות הקיצון שלהם ויהיה יותר קל לראות אזורים דומים.
4. ההחלקה מבוצעת באמצעות Gaussian:
   1. 
      1. <https://www.youtube.com/watch?v=L77m5xuDSKw>
   2. נשים לב, ככל שנקטין את הסיגמא; כך נזהה אזורים בתמונה עם נקודות קיצון שמייצגות אזורים קטנים יותר, וכן ההפך נכון.
   3. לפי "מובארק שה" (מהסרטון), בדרך כלל הערך של סיגמא יהיה 1.6, כאשר הK יהיה שורש של 2.
   4. נקודות המינימום ומקסימום המקומיות יהיו נקודות העניין הפוטנציאליות שלנו.
5. יש לעבור שוב על מציאת OUTLIERS והוצאתם מהנקודות העניין הפוטנציאליות.
6. נמצא את האוריינטציה של כל נקודת עניין.
   1. 
   2. נגדיר לאזור של כל נקודת עניין את האוריינטציה הפרטית שלו, כאשר בנוסף לכיוון, נשמור את המשקל (רמת השינוי מנקודת הקיצון לסביבה שלו – קיצון יותר גבוהה הוא בכל משקל גדול יותר).
   3. 
   4. 
   5. נמיר את האוריינטציות הספציפיות של כל נקודת עניין לאוריינטציה כללית של אזור.
   6. נגדיר את האוריינטציה הכללית של אזור לפי חלוקת הכיוון לאזורים כלליים (בדוגמא שניתנה, הכיוונים חולקו ל4 כיוונים של 45 מעלות).
   7. עבור כל כיוון כללי, נוסיף את המרחק שכל החצים שנעים לאותו כיוון שלו (הכיוון הכי קרוב).
   8. בדוגמא, מכיוון שהחץ השמאלי-ימיני הוא הארוך ביותר, הוא מייצג את האוריינטציה בעלת המשקל הכי גדול.
7. נגדיר DESCRIPTOR עבור נקודות העניין (האזור והאוריינטציה הכללית שלו).
   1. 
   2. עבור כל אזור, נשמור את האוריינטציה הכללית שלו.
   3. הDESCRIPTOR שומר בווקטורים את הנקודות עניין (אזורים בפיסקלים) ועל ידי נורמליזציה, נמיר את האוריינטציה הכללית להתאמה לאזורים בתמונה.
   4. בדוגמא שלנו, המרנו מ8X8 כיוונים של נקודות מקסימה (מינימום / מקסימום) ל2X2 אזורים כלליים.
   5. מבנה הDESCRIPTOR:
      1. וקטור המייצג נקודות עניין, כאשר לכל נקודת עניין יש את המימדים:
         1. 4X4 – האוריינטציות הכלליות (כפי שמוצגות בדוגמא לעיל), אלו האזורים שהומרו מ16X16 פיקסלים סביב נקודת.
         2. 8 – הכיוונים שחולקו כל הווקטורים (בדוגמא שלנו, ניתן לראות שיש 8 כיוונים אפשריים לכל איחוד 4X4 האזורים של ה16X16 פיקסלים המקוריים).
8. נבצע התאמה בין תמונות.
   1. נחפש באמצעות KNN את הנקודות עניין עם האוריינטציות הכלליות הכי קרובות.
   2. בדוגמא של ההרצאה מהווידאו, אם האוריינטציה בין התמונה הראשונה הכי קרובה לתמונה השנייה הכי קרובה גדול מ0.8, אז ההתאמה היא בעלת הסתברות לTRUE POSITIVE טוב יותר.
   3. ההתאמה בין הDESCRIPTOS של תמונות נעשית על ידי השוואה של הווקטורים של נקודות העניין.
   4. לפי THRESHOLD, נקבע כמה נקודות עניין צריכות להיות דומות כדי לקבוע האם התמונות דומות.
   5. בדרך כלל משתמשים בKNN – משמש כTHRESHOLD לקירוב בין המימדים באופן נאיבי.