# DOWNLOAD PDF IMAGE PROCESSING EDGE DETECTION

## Chapter 1 : Sobel edge detection | IMAGE PROCESSING

*Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques. Edge properties [ edit ] The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent.*

It helps you reduce the amount of data pixels to process and maintains the "structural" aspect of the image. Both of them work with convolutions and achieve the same end goal - finding edges. Have a look at this: The curve representing intensity The lower part is the 1-D image. The upper part is the intensity of each pixel of the 1-D image plotted as a graph. Blacks have a low intensity, so the graph curve is low. It reaches full height at the white end of the image. The first derivative of the curve above Looking for these peaks is exactly what gradient based edge detection methods do. If the change is steep enough, you mark it as an edge pixel. There has to be a certain threshold above which an edge is classified as a peak else it must be considered part of noise. On the left where the curve is rising , the slope is positive. On the right, the slope is negative. So there must exist a point where there is a zero crossing. Edge detectors that are based on this idea are called Laplacian edge detectors. The second order derivative Now, all of this is for 1-D images. It turns out that all of this holds for 2-D images as well. So we can simply use these results and try them on actual images. Another thing is - these are based on continuous images. For us, that is never the case. This is done with the help of convolutions. It works with first order derivatives. It calculates the first derivatives of the image separately for the X and Y axes. The derivatives are only approximations because the images are not continuous. To approximate them, the following kernels are used for convolution: Kernels used in the Sobel edge detection The kernel on the left approximates the derivative along the X axis. The one on the right is for the Y axis. Using this information, you can calculate the following: Magnitude or "strength" of the edge: The orientation of the edge: You can clearly see the horizontal edges highlighted. You can then threshold this result to get rid of the grey areas and get solid edges. It calculates second order derivatives in a single pass. The kernel for the laplacian operator You can use either one of these. Or if you want a better approximation, you can create a 5x5 kernel it has a 24 at the center and everything else is  The result of convolution with the laplacian operator Laplacians are computationally faster to calculate only one kernel vs two kernels and sometimes produce exceptional results! Utkarsh Sinha created AI Shack in and has since been working on computer vision and related fields. He is currently at Microsoft working on computer vision.

## Chapter 2 : Concept of Edge Detection

*There are many ways to perform edge detection. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.*

It is possible to extend filters dimension to avoid the issue of recognizing edge in low SNR image. The cost of this operation is loss in terms of resolution. Examples are Extended Prewitt 7x7. Thresholding and linking[ edit ] Once we have computed a measure of edge strength typically the gradient magnitude , the next stage is to apply a threshold, to decide whether edges are present or not at an image point. The lower the threshold, the more edges will be detected, and the result will be increasingly susceptible to noise and detecting edges of irrelevant features in the image. Conversely a high threshold may miss subtle edges, or result in fragmented edges. If the edge is applied to just the gradient magnitude image, the resulting edges will in general be thick and some type of edge thinning post-processing is necessary. For edges detected with non-maximum suppression however, the edge curves are thin by definition and the edge pixels can be linked into edge polygon by an edge linking edge tracking procedure. On a discrete grid, the non-maximum suppression stage can be implemented by estimating the gradient direction using first-order derivatives, then rounding off the gradient direction to multiples of 45 degrees, and finally comparing the values of the gradient magnitude in the estimated gradient direction. A commonly used approach to handle the problem of appropriate thresholds for thresholding is by using thresholding with hysteresis. This method uses multiple thresholds to find edges. We begin by using the upper threshold to find the start of an edge. Once we have a start point, we then trace the path of the edge through the image pixel by pixel, marking an edge whenever we are above the lower threshold. We stop marking our edge only when the value falls below our lower threshold. This approach makes the assumption that edges are likely to be in continuous curves, and allows us to follow a faint section of an edge we have previously seen, without meaning that every noisy pixel in the image is marked down as an edge. Still, however, we have the problem of choosing appropriate thresholding parameters, and suitable thresholding values may vary over the image. Edge thinning[ edit ] Edge thinning is a technique used to remove the unwanted spurious points on the edges in an image. This technique is employed after the image has been filtered for noise using median, Gaussian filter etc. This removes all the unwanted points and if applied carefully, results in one pixel thick edge elements. Sharp and thin edges lead to greater efficiency in object recognition. If Hough transforms are used to detect lines and ellipses, then thinning could give much better results. If the edge happens to be the boundary of a region, then thinning could easily give the image parameters like perimeter without much algebra. There are many popular algorithms used to do this, one such is described below: Choose a type of connectivity, like 8, 6 or 4. Remove points from North, south, east and west. Do this in multiple passes, i. Remove a point if: The point has no neighbors in the North if you are in the north pass, and respective directions for other passes. The point is not the end of a line. The point is isolated. Removing the points will not cause to disconnect its neighbors in any way. Else keep the point. Second-order approaches[ edit ] Some edge-detection operators are instead based upon second-order derivatives of the intensity. This essentially captures the rate of change in the intensity gradient. Thus, in the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient. The early Marrâ€”Hildreth operator is based on the detection of zero-crossings of the Laplacian operator applied to a Gaussian-smoothed image. It can be shown, however, that this operator will also return false edges corresponding to local minima of the gradient magnitude. Moreover, this operator will give poor localization at curved edges. Hence, this operator is today mainly of historical interest. Differential[ edit ] A more refined second-order edge detection approach which automatically detects edges with sub-pixel accuracy, uses the following differential approach of detecting zero-crossings of the second-order directional derivative in the gradient direction: Following the differential geometric way of expressing the requirement of non-maximum suppression proposed by Lindeberg, [4] [15] let us introduce at every image point a local coordinate system.

## Chapter 3 : C# How to: Image Edge Detection | Software by Default

*Edge Detection in Digital Image Processing Debosmit Ray Thursday, June 06, 1. Introduction In this paper, I discuss the mathematical theorems and algorithms.*

Non-maximum suppression[ edit ] Non-maximum suppression is an edge thinning technique. Non-maximum suppression is applied to find "the largest" edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. With respect to criterion 3, there should only be one accurate response to the edge. Thus non-maximum suppression can help to suppress all the gradient values by setting them to 0 except the local maxima, which indicate locations with the sharpest change of intensity value. The algorithm for each pixel in the gradient image is: Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions. If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction i. Otherwise, the value will be suppressed. In some implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step that is, the edge strength and gradient directions. At every pixel, it suppresses the edge strength of the center pixel by setting its value to 0 if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction. In more accurate implementations, linear interpolation is used between the two neighbouring pixels that straddle the gradient direction. The gradient magnitude at the central pixel must be greater than both of these for it to be marked as an edge. Note that the sign of the direction is irrelevant, i. Double threshold[ edit ] After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. The two threshold values are empirically determined and their definition will depend on the content of a given input image. Edge tracking by hysteresis[ edit ] Canny edge detection applied to a photograph So far, the strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. To achieve an accurate result, the weak edges caused by the latter reasons should be removed. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved. Improvement on Canny edge detection[ edit ] While traditional Canny edge detection provides relatively simple but precise methodology for edge detection problem, with more demanding requirements on the accuracy and robustness on the detection, the traditional algorithm can no longer handle the challenging edge detection task. The main defects of the traditional algorithm can be summarized as follows: This will increase the possibility of missing weak edges, and the appearance of isolated edges in the result. This method is sensitive to noise and can easily detect false edges and lose real edges. In the traditional Canny edge detection algorithm, there will be two fixed global threshold values to filter out the false edges. However, as the image gets complex, different local areas will need very different threshold values to accurately find the real edges. In addition, the global threshold values are determined manually through experiments in the traditional method, which leads to complexity of calculation when a large number of different images need to be dealt with. The result of the traditional detection cannot reach a satisfactory high accuracy of single response for each edge - multi-point responses will appear. In order to address these defects, improvement for the canny edge algorithm is added in the fields below. Replace Gaussian filter[ edit ] As both edge and noise will be identified as high frequency signal, simple Gaussian filter will add smooth effect on both of them. However, in order to reach high accuracy of detection of the real edge, it is expected that more smooth effect should be added to noise and less smooth effect should be added to the edge. Bing Wang and Shaosheng Fan from Changsha University of Science and Technology developed an adaptive filter, where the filter will evaluate discontinuity between

greyscale values of each pixel. Contrarily, the lower the discontinuity between the greyscale values, the higher the weight value is set to the filter. The process to implement this adaptive filter can be summarized in five steps: Calculate the gradient value G.

## Chapter 4 : Edge Detection - MATLAB & Simulink

*Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision. Common.*

Above mentioned all the filters are Linear filters or smoothing filters. Prewitt Operator Prewitt operator is used for detecting edges horizontally and vertically. Sobel Operator The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. It also calculates edges in both horizontal and vertical direction. Robinson Compass Masks This operator is also known as direction mask. In this operator we take one mask and rotate it in all the 8 compass major directions to calculate edges of each direction. Kirsch mask is also used for calculating edges in all the directions. Laplacian Operator is also a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask. It can be further divided into positive laplacian and negative laplacian. All these masks find edges. Some find horizontally and vertically, some find in one direction only and some find in all the directions. The next concept that comes after this is sharpening which can be done once the edges are extracted from the image Sharpening: Sharpening is opposite to the blurring. In blurring, we reduce the edge content and in sharpneng , we increase the edge content. So in order to increase the edge content in an image , we have to find edges first. Edges can be find by one of the any method described above by using any operator. After finding edges , we will add those edges on an image and thus the image would have more edges , and it would look sharpen. This is one way of sharpening an image. The sharpen image is shown below.

## Chapter 5 : The Sobel and Laplacian Edge Detectors - AI Shack

*We will cover both image and video recognition, including image classification and annotation, object recognition and image search, various object detection techniques, motion estimation, object tracking in video, human action recognition, and finally image stylization, editing and new image generation.*

One of the most useful tools which allow engineers to design vision systems detecting or recognizing objects in images is subpixel edge detection. This article explains the concept and these which lay the foundation of it. Image representation The base for many measurement applications with optical methods is intensity images. The intensity which is perceived as brightness in the image is mapped to a digital gray scale image. Therefore these images are called grayscale images. The image is a grid that is composed of individual picture elements, so-called pixels. Each pixel represents a numerical value which represents the gray value. In a camera with a resolution of 8 bit grayscale differs from 0 for black to for white, with bit resolution there are gray levels. Grayscale images can be displayed as a matrix for processing and storing with software Figure 1. Computer based representation of grayscale images as matrix There are different formats for storing digital images. For use in metrology, only image formats are possible, which are suitable for lossless transfer of image data. An involving loss transfer, as it is used for example in image compression to reduce image size, changes the image and may affect the location of edges and thus the measurement result. A distinction is made between point operators, local, global, and morphological operators. Image processing operations that affect a pixel only depending on its value and its current position in the image without considering the neighborhood of the pixel are called point operations. Examples for point operators are brightness correction and the inversion of a grayscale image. By potentiating the gray values, a non-linear stretching in one part of the image and a non-linear compression in another part of the image is performed. With values for gamma larger than one, the image is darker, and for values less than one, the image is brighter. Figure 2 shows the use of two other point operators. For contrast enhancement that is also called histogram stretching, the gray values are changed so that the entire available gray scale is used. For image segmentation often a global thresholding is used. Here, a binary image is created black-white image by displaying pixels below the threshold as black and above as white. This method is also known as binarization. A suitable threshold value can be determined from the histogram of the gray values when a bi-modal distribution of the gray values is available. A known computational method for thresholding is represented in [3]. Contrast enhancement for histogram stretching, binary image with threshold from bimodal histogram and edge image derived from binary image For local operators, the new gray value of a pixel depends not only on its previous value but also on the gray values of the pixels in its environment. The environment is defined by a so-called neighborhood. A typical neighborhood is the 8-neighborhood 3 x 3 pixel. Figure 3 shows the use of two operators considering the pixel itself and its eight neighbors, which are referred in this context as filters for eliminating image distortions. Local operators for eliminating image distortion: The underlying mathematical procedure is a so-called convolution. There are many different linear filters [4]. Filters, such as the average filter described above or the Gaussian filter, in which the weighting factors depend on the distance to the subject pixel according to the shape of the Gaussian curve, are used to smooth the image. Thus they represent a low-pass filter. Also, the median filter, in which the median of the surrounding pixels determines the filtered pixel, is a low-pass filter. Edge detection In contrast to the low-pass filters, the high-pass filters are used for highlighting edges. Given the image captured by the camera, in this example first preprocessing is done to remove distortion with the above described low-pass filters. Subsequently, edges are highlighted in two directions by two filter masks of the Sobel filter. The superposition of the images provides the edge image. This type of edge filters is based on the discrete differentiation of the image and is therefore also referred as a gradient filter. Gradient filters have high-pass properties and increase the image noise. Therefore, the filters are designed so that they result is averaged over multiple rows or columns. Another representative of this kind of edge filters is the Prewitt filter [4, 5]. There are moreover gradient filters for edges that combine various filters such as the Canny edge detector [6]. Edge detection using Sobel filter Also a binary image Figure 2 is suitable for edge determination.

Here the definition of a global threshold value, which is used for segmentation of the image into foreground and background, determines the edge position. This approach is beneficial when only one edge in an image with several edges must be identified e. In this area, the search beams are generated. Along each search beam, an edge point is determined. The maximum of the first derivative along the search path or a threshold value are used as edge criteria. The first criteria corresponds to the previously described edge detection with a gradient. The second criteria corresponds to the edge detection based on a binary image, as shown above. When threshold criteria is used you distinguish between a global threshold, which applies to the entire edge region, and a local threshold which is determined individually for each search area or search path. Subpixel interpolation [8] Figure 7. Grey value line and its 1st derivative along a search path A correct determination of the edge position requires that light intensity is always below signal saturation of the camera because the edge position might be shifted due to the saturation. Multi sensor systems for dimensional quality inspection, in: Leitfaden zur industriellen Bildverarbeitung, Vision Leitfaden 13 1. Auflage Vision Leitfaden 1, English: Machine vision â€" Basics, terms, and definitions , April

## Chapter 6 : Image Processing : Edge Detection

*In this episode, we will learn how to use OpenCV functions to apply edge detection to an image. In edge detection, we find the boundaries or edges of objects in an image, by determining where the brightness of the image changes dramatically.*

The types of edge detection discussed are: All instances are implemented by means of Image Convolution. Sample source code This article is accompanied by a sample source code Visual Studio project which is available for download here. Using the Sample Application The concepts explored in this article can be easily replicated by making use of the Sample Application, which forms part of the associated sample source code accompanying this article. The dropdown combobox towards the bottom middle part of the screen relates the various edge detection methods discussed. If desired a user can save the resulting edge detection image to the local file system by clicking the Save Image button. The following image is screenshot of the Image Edge Detection sample application in action: Edge Detection A good description of edge detection forms part of the main Edge Detection article on Wikipedia: Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing , machine vision and computer vision , particularly in the areas of feature detection and feature extraction. From the article we learn the following: Convolution is a simple mathematical operation which is fundamental to many common image processing operators. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values. In an image processing context, one of the input arrays is normally just a graylevel image. The second array is usually much smaller, and is also two-dimensional although it may be just a single pixel thick , and is known as the kernel. Single Matrix Convolution The sample source code implements the ConvolutionFilter method, an extension method targeting the Bitmap class. The ConvolutionFilter method is intended to apply a user defined matrix and optionally covert an image to grayscale. The implementation as follows: LockBits new Rectangle 0, 0, sourceBitmap. Scan0, pixelBuffer, 0, pixelBuffer. LockBits new Rectangle 0, 0, resultBitmap. Copy resultBuffer, 0, resultData. The original image is attributed to Kenneth Dwain Harrelson and can be downloaded from Wikipedia. Laplacian Edge Detection The Laplacian method of edge detection counts as one of the commonly used edge detection implementations. From Wikipedia we gain the following definition: Discrete Laplace operator is often used in image processing e. The discrete Laplacian is defined as the sum of the second derivatives Laplace operator Coordinate expressions and calculated as sum of differences over the nearest neighbours of the central pixel. The detected edges are expressed in a fair amount of fine detail, although the Laplacian matrix has a tendency to be sensitive to image noise. Laplacian of Gaussian is intended to counter the noise sensitivity of the regular Laplacian filter. Laplacian of Gaussian attempts to remove image noise by implementing image smoothing by means of a Gaussian blur. In order to optimize performance we can calculate a single matrix representing a Gaussian blur and Laplacian matrix. In this scenario the first variations Type 1 appears to result in less image noise. We gain the following quote from Wikipedia: The Sobel operator is used in image processing , particularly within edge detection algorithms. Technically, it is a discrete differentiation operator , computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image. Unlike the Laplacian filters discussed earlier, Sobel filter results differ significantly when comparing colour and grayscale images. The Sobel filter tends to be less sensitive to image noise compared to the Laplacian

filter. From Wikipedia we gain the following quote: The Prewitt operator is used in image processing , particularly within edge detection algorithms. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. The Prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. The Prewitt operator was developed by Judith M. In simple terms, the operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The result therefore shows how "abruptly" or "smoothly" the image changes at that point, and therefore how likely it is that that part of the image represents an edge, as well as how that edge is likely to be oriented. In practice, the magnitude likelihood of an edge calculation is more reliable and easier to interpret than the direction calculation. Similar to the Sobel filter, resulting images express a significant difference when comparing colour and grayscale images. In the following scenario we only implement two components: Resulting images tend to have a high level of brightness. If you know of an alternative implementation or have ideas on a more efficient implementation please share in the comments section.

## Chapter 7 : Canny edge detector - Wikipedia

*We have discussed briefly about edge detection in our tutorial of introduction to masks. We will formally discuss edge detection here. We can also say that sudden changes of discontinuities in an image are called as edges. Significant transitions in an image are called as edges. Most of the shape.*

BW â€" Output binary image logical array gpuArray Output binary image, returned as a logical array of the same size as I , with 1s where the function finds edges in I and 0s elsewhere. If you use a GPU to find the edges, then threshOut is returned as a gpuArray that contains a numeric scalar. Gv â€" Vertical gradient Vertical gradient, returned as a numeric array of the same size as I. Gh â€" Horizontal gradient Horizontal gradient, returned as a numeric array of the same size as I. Tips The function edge changed in Version 7. If you need the same results produced by the previous implementation, use the following syntax: For the zero-crossing methods, including Laplacian of Gaussian, edge uses threshold as a threshold for the zero-crossings. In other words, a large jump across zero is an edge, while a small jump is not. The Canny method applies two thresholds to the gradient: This helps fill in gaps in the detected edges. In all cases, edge chooses the default threshold heuristically, depending on the input data. The best way to vary the threshold is to run edge once, capturing the calculated threshold as the second output argument. Then, starting from the value calculated by edge, adjust the threshold higher to detect fewer edge pixels, or lower to detect more edge pixels. Usage notes and limitations: Note that if you choose the generic MATLAB Host Computer target platform, the function generates code that uses a precompiled, platform-specific shared library. Use of a shared library preserves performance optimizations but limits the target platforms for which code can be generated. The method, direction, and sigma arguments must be compile-time constants. Nonprogrammatic syntaxes are not supported. For example, if you do not specify a return value, then edge displays an image. This syntax is not supported with code generation.

## Chapter 8 : Edge detection using Local Variance | IMAGE PROCESSING

*The edge detected image can be obtained from the sobel gradient by using a threshold value. If the sobel gradient values are lesser than the threshold value then replace it with the threshold value.*

Above mentioned all the filters are Linear filters or smoothing filters. Prewitt Operator Prewitt operator is used for detecting edges horizontally and vertically. Sobel Operator The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. It also calculates edges in both horizontal and vertical direction. Robinson Compass Masks This operator is also known as direction mask. In this operator we take one mask and rotate it in all the 8 compass major directions to calculate edges of each direction. Kirsch mask is also used for calculating edges in all the directions. Laplacian Operator Laplacian Operator is also a derivative operator which is used to find edges in an image. Laplacian is a second order derivative mask. It can be further divided into positive laplacian and negative laplacian. All these masks find edges. Some find horizontally and vertically, some find in one direction only and some find in all the directions. The next concept that comes after this is sharpening which can be done once the edges are extracted from the image Sharpening Sharpening is opposite to the blurring. In blurring, we reduce the edge content and in Sharpening, we increase the edge content. So in order to increase the edge content in an image, we have to find edges first. Edges can be find by one of the any method described above by using any operator. After finding edges, we will add those edges on an image and thus the image would have more edges, and it would look sharpen. This is one way of sharpening an image. The sharpen image is shown below.

## Chapter 9 : Subpixel edge detection and image processing -OptiNav

*Original Sample Image. The original source image used to create all of the edge detection sample images in this article has been licensed under the Creative Commons Attribution-Share Alike Unported, Generic, Generic and Generic license.*

Other Methods of Edge Detection There are many ways to perform edge detection. However, the most may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zerocrossings in the second derivative of the image to find edges. This first figure shows the edges of an image detected using the gradient method Roberts, Prewitt, Sobel and the Laplacian method Marrs-Hildreth. Various Edge Detection Filters Notice that the facial features eyes, nose, mouth have very sharp edges. These also happen to be the best reference points for morphing between two images. Notice also that the Marr-Hildreth not only has a lot more noise than the other methods, the low-pass filtering it uses distorts the actual position of the facial features. Due to the nature of the Sobel and Prewitt filters we can select out only vertical and horizontal edges of the image as shown below. This is very useful since we do not want to morph a vertical edge in the initial image to a horizontal edge in the final image. This would cause a lot of warping in the transition image and thus a bad morph. Vertical and Horizontal Edges The next pair of images show the horizontal and vertical edges selected out of the group members images with the Sobel method of edge detection. You will notice the difficulty it had with certain facial features, such as the hairline of Sri and Jim. This is essentially due to the lack of contrast between their hair and their foreheads. Vertical Sobel Filter Horizonatal Sobel Filter We can then compare the feature extraction using the Sobel edge detection to the feature extraction using the Laplacian. Sobel Filtered Common Edges: Jim Sobel Filtered Common Edges: Roger We see that although it does do better for some features ie. A morph constructed using individually selected points would still work better. It should also be noted that this method suffers the same drawbacks as the previous page; difficulties due to large contrast between images and the inability to handle large translations of features. Another method of detecting edges is using wavelets. Specifically a two-dimensional Haar wavelet transform of the image produces essentially edge maps of the vertical, horizontal, and diagonal edges in an image. This can be seen in the figure of the transform below, and the following figure where we have combined them to see the edges of the entire face. Haar Filtered Common Edges: Jim Haar Filtered Common Edges: Roger Although the Haar filter is nearly equivalent to the gradient and Laplacian edge detection methods, it does offer the ability to easily extend our edge detection to multiscales as demonstrated in this figure. Extended Haar Wavelet Transform.