Assisted Media Filtering

Sam Z. Glassenberg, David Lowsky, Sam Pearlman, Sebastian Thrun
Department of Computer Science
Stanford University
Stanford, California 94305

Abstract

Media broadcasts often require personally-identifiable visual information to be obfuscated to preserve anonymity of witnesses, suspects, and minors. Currently, this process requires a manual post-processing step, incurring a significant delay that prevents content from being televised live. Using scale-invariant features and descriptors, we propose a unique clustering system that can automatically obscure an identified target object in subsequent video frames. The system maintains awareness of similar subjects, avoiding obfuscation of incorrect objects while tracking the target.

Keywords: SIFT, media filtering, object tracking, obfuscation.

1. Introduction

Live media broadcasts often require personally-identifiable visual information to be obfuscated to preserve anonymity (faces, license plates, addresses, etc.). This is especially true when video footage includes minors or witnesses. To do so for a stationary interview is simple. However, if motion relative to the camera is involved, a painstaking human post-processing step is currently required before broadcast: the subject must be manually censored. This results in a significant delay and prevents many programs from live airing.

In the live broadcast scenario, the primary objective is to enable a camera operator to rapidly locate and specify an object in the first frame, then automatically track and obfuscate it throughout the stream. This scenario presents some unique conditions that differentiate it from other face and object recognition tasks:

- No pre-training. Data about the target object must be acquired immediately prior to broadcast in the first set of frames – it cannot be pre-acquired and trained from over a period of time. Target objects are present at the start of broadcast, and can be specified by a camera-person using an integrated interface.
- Scene transformation. In live broadcast, the camera and object position and orientation are dynamic and unpredictable. Tracking must be invariant to translation, rotation, scale and lighting changes.
- Reacquisition. When the target object is temporarily occluded, out-of-focus, rotated out-of-view or out of the scene in subsequent frames, no obfuscation is required. In all cases reacquisition must occur immediately and obfuscation resumed once the object re-enters the visible scene.
- Differentiation. Similar objects can enter and leave the scene throughout this process, yet the system should consistently track only the target object.
- Temporal coherence. While movement in the scene may be rapid, reasonable temporal coherence between object features can be assumed. Data is recorded at 30 frames-per-second, and it can be assumed that the camera operator will be professional and deliberate.
- No 3D construction or location is required. The goal is the obfuscation of the element's identifiable features in image-space.

Although many solutions for object recognition and tracking have been developed, prior research has dealt with these problems in a different scope. These applications deal either with recognition of objects from a machine-learned database of templates (e.g. office locations or faces), or the tracking of a moving object without regard for global reacquisition under a wide variety of circumstances. In other words, these applications deal primarily with the fields of robotics [CHANG98], security, and human-computer interfaces [BRADSKI98] - not broadcast media. These problem spaces either incorporate significant amounts of learning and library-building, or deal with the tracking problem locally as opposed to globally.

The goal of this project is to create a proof-of-concept of an interface that will allow a camera-person to identify an element in view (such as an individual's face) and mark it for automatic obfuscation in the subsequent live video broadcast. The system will track the object automatically as it moves, concealing it via blurring or image overlay.

The algorithm will combine traditional SIFT features ([LOWE03]) with some novel modifications that track the specified object through a variety of scene transformations. A modification to the traditional Hough transform clustering method will allow objects which are similar to the target (i.e. other faces in the scene) to be simultaneously identified and tracked, preventing obfuscation of the wrong object if the primary target is occluded.

Our results thus far illustrate the robustness of SIFT feature detection and Lowe feature descriptors. As a general rule, individual features remain detectable over reasonably wide affine transformations (often with rotations in excess of 40 degrees). SIFT features have been found to be especially prominent within the objects in question (faces, license plates, etc.)

Several factors have been confirmed or uncovered that have a significant effect on the SIFT algorithm's ability to discriminate features. Independent of the magnitude of transformation, SIFT features have a tendency to be temporally erratic. This mandates a flexible clustering model that allows for recognition based on an incomplete set of features, some of which may be incorrectly correlated.

2. Background

The right and desire of the viewing public to see events broadcast in real-time must be balanced with an individual's right to privacy. This conflict is most pronounced in broadcasts of sensitive events involving legal culpability such as police actions and court proceedings. Viewers demand live broadcast of these events, yet the privacy of suspects, victims and witnesses must be preserved. It has been concluded that "Broadcasting the identity of a crime victim most often only adds to the person's grief, anguish and trauma" [CBC03], while broadcasting the identity of a suspect can jeopardize the fairness of criminal proceedings. Governments have sought reasonable compromise [TEXAS02], but the conflict remains, exacerbated by the fact that preservation of anonymity demands a broadcast delay of minutes to hours.

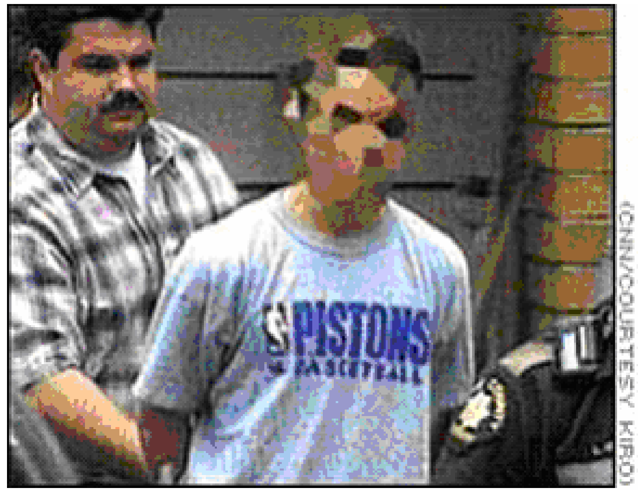

Figure 1: Pixilated image of a 13-year-old murder suspect turning himself in to the police (the youth's face has been obscured because he is a juvenile)

Preservation of privacy is not necessarily guaranteed by a system limited to facial occlusion. Additional scenes may require obfuscation of other personally-identifiable information (PII), such as license plates or addresses. By enabling a camera-operator to identify PII while filming, automatic object obfuscation could begin, allowing the output of such a system to be broadcast live. A courtroom scene could be broadcast live without the risk of privacy violation if the camera or protected subject moved.

Existing solutions deal separately with variations of two basic problems: Tracking and Object Recognition. Tracking is traditionally performed using gradient descent techniques to compute optical flow, such as those presented by Lukas and Kanade. These methods are fast and provide an excellent solution to the local tracking problem. These methods fail, however, when objects rotate beyond a threshold or temporarily leave the local scene-space. In the case of target loss, [CHANG98] stops camera motion and performs a continual search for the object based on its most recent visual template. This works well if a stationary object is temporarily occluded by an intermediate object, but fails if the camera or object has changed orientation when the object returns to the view.

Object recognition is traditionally dealt with as a separate problem. Many methods exist for the recognition of objects, ranging from broad 3D object recognition to specific applications for human faces, etc. Several approaches to facial recognition use statistical methods to train on specific faces. [SCHNEID00] requires training on images of each facial orientation. [WISKOTT97] describes a method using Elastic Bunch Graphs that does not require per-face training but is not illumination invariant. In these cases, either the algorithm is designed to detect a generic object type, requires an extensive database of pre-acquired data, or does not enjoy the transformation invariance of SIFT features.

Once features are recognized, many robust fitting methods are available to cluster them into objects. Techniques such as RANSAC or Least Median of Squares are potential candidates, but have been found to perform poorly when the ratio of cluster inliers to outliners falls below 0.5. The Hough transform cluster method described in [LOWE03] was shown to provide better performance in this case.

Pre-and post-processing methods also exist to improve recognition accuracy. Algorithms exist to increase lighting independence, such as [ROSS00] which is based on the biological model of human vision.

Methods exist to compensate for the effects of motion blur on motion estimation. The method described in [TIMONER01] uses intrinsic properties of a video camera to anticipate the impact of motion blur on optical flow and compensate accordingly. Unfortunately, these methods clarify optical flow but do not improve the quality of the original image, yielding no benefits for feature detection. Generalized blurring can be remedied using a sharpening filter. Motion blur effects can traditionally be compensated for only by more expensive camera hardware with a faster simulated shutter speed.

SIFT features can maintain prominence over a wide range of transformations and lighting conditions, and their gradient-based descriptors have shown to be highly discriminatory over other methods [Mikolajczyk03]. SIFT features and descriptors form the basis of our solution.

3. Methods

3.1 Implementation

All new algorithms developed for this project are implemented in MATLAB. They could eventually be ported to a real-time implementation that runs on camera hardware. This long-term goal is kept in mind in design decisions: tracking/recognition is dependent only on data found in the current frame and previous frames. No pre-acquired training data is necessary.

3.2 Interface

A prototype interface is provided to allow the user to target an object in the center of view in the first frame. A throttle allows the region to enlarge or shrink. Once this region is determined, the tracking process commences.

3.3 Object Recognition

In each frame, a SIFT feature detection pass is conducted over the entire image using [LOWE03B]. 4x4x8 SIFT descriptors are constructed for features in the image region with a prominence above a certain threshold. Features within the target range are catalogued in a database (MATLAB array) for the target object.

SIFT features detected inside the user-specified range while outside of the face silhouette could accidentally be included in the databases and unwittingly tracked. To prevent this, the gradient-based method proposed in [BIRCHFIELD97] can be used to optionally fit an ellipse to the object's silhouette. The algorithm identifies the ellipse state (**s**) that maximizes gradient magnitude ($\mathbf{g}_i$) over the perimeter of the ellipse in a small search space (**S**). **Nσ** represents the number of pixels in the perimeter.

$$\mathbf{s}^* = \arg\max_{\mathbf{s}\in S} \left\{ \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} |\mathbf{g}_i| \right\}$$

The ellipse's aspect ratio will be equal to the aspect ratio of the targeted image region. Thus, only features detected inside the silhouette will be added to the feature database.

In the following frames, SIFT feature detection is performed on the entire image, and detected features are compared to the existing database of target features using a Euclidean cost function. A Hough transform is used to detect the object elsewhere in the scene through feature clusters as described in [LOWE03]. Large Hough orientation bins are used to accommodate rigid and non-rigid transforms.

The Hough Orientation bin with the most votes is chosen as the orientation of the primary object in the scene. The corresponding affine transformation is then computed from the agreeable feature locations in the bin using the least-squares approach described in [LOWE03]:

A: **x:** **b:**

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\mathbf{x} = [\mathbf{A}^{\mathbf{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{b}$$

The transformation is applied to all detected features, and outliers are removed.

If the transformation is agreeable with a sufficient number of features, this transformation is used to transform the primary target ellipse. If the affine-transformed ellipse does not provide a close enough match, the [BIRCHFIELD97] method could be re-used at every frame.

If an insufficient number of features remain after outliers are removed, the selected object is determined to be a false-positive. The true target is assumed to be occluded or off-screen. In this case, the algorithm continues to

```
'Good' and 'Bad' feature databases initialized
In Frame 1:
{
        Target image region is selected by user;
                Region is optionally fitted to ellipse/quad;
        Detect SIFT features over whole image;
        FOR (all SIFT features above prominence threshold):
                Create SIFT descriptors;
        Add Feature descriptors in target region to 'good' feature database;
        FOR (all features detected outside of target region):
                Compute matches with features in 'good' feature database;
        FOR (all matched features):
                Perform Hough transform to detect clusters;
        FOR (all Hough transform bins with >3 votes):
        {
                WHILE(# of inliers>threshold AND transformed points are within an error threshold)
                {
                        Compute affine transformation for cluster;
                        Remove outliers;
                }
                IF (cluster has # of features > threshold):
                {
                        Transform a temporary target region;
                        Add all features within the temporary region to 'bad' feature database;
                        DRAW the obfuscating shape on the output frame;
                        DISPLAY the output frame.
                }
        }
}
FOR (All Subsequent Frames):
{
        Detect SIFT features over whole image;
        FOR (all SIFT features above prominence threshold):
                Create SIFT descriptors;
        Match descriptors with features in 'good' and 'bad' databases
        FOR (all matched features):
                Perform Hough transform with positive/negative voting to detect primary (target)
        cluster;
        Select the Hough transform bin with largest # of votes:
                WHILE(# of inliers>threshold AND transformed points are within an error threshold)
                {
                        Compute affine transformation for cluster;
                        Remove outliers;
                }
                IF (cluster has # features > threshold)
                {
                        Transform the target region;
                        Add all features within the region to 'good' feature database;
                        DRAW the obfuscating shape on the output frame;
                        DISPLAY the output frame;
                }

        FOR (all matched features):
                Perform Hough transform with positive/positive voting to detect all other clusters;
        FOR (all Hough transform bins with >3 votes):
        {
                Compute affine transformation for cluster;
                Remove outliers;
                IF (cluster has # of features > threshold):
                {
                        Transform a temporary target region;
                        Add all features within the temporary region to 'bad' feature database;
                }
        }
```

Figure 2: Basic Program Flow
(No Pre- and Post-processing steps)

search for the primary object in subsequent frames, and continues to track similar-looking objects to prevent incorrect obfuscation as described below.

3.4 Dynamic Learning

Since some rotation/illumination/expression change may have occurred, a new frame provides an additional 'view' with which to enhance the target object feature database *dynamically*. Prominent SIFT features found in this slightly-shifted region are added to the database accordingly. Their descriptor location and orientation is stored relative to the frame-of-reference of the initial video frame, generated using the computed affine transform. This novel approach allows the system to dynamically 'learn' about features in new orientations as they arrive.

Many usage scenarios require the system to discriminate between multiple similar objects in the scene. In the most trying cases, the target will leave the scene, and several false-subjects will remain. The ability to track these potential false-positives throughout recording is a unique component of our algorithm. At every frame, once the most prominent Hough cluster is selected, all other orientations with a large number of matching features will be considered a 'potentially hazardous' cluster. These clusters may be incorrectly censored if the primary target is occluded or leaves the frame. If such a cluster is sufficiently far from the primary target, it is treated as a 'bad cluster'. Just as with the 'good cluster', an affine transformation is computed for the 'bad cluster', and an elliptical window around this cluster is used to identify undiscovered 'bad' features. These features are necessary to discriminate the similar but wrong subjects from the primary target. They are therefore stored in a separate database of 'bad' features. In subsequent frames, detected features are compared to both 'good' and 'bad' databases and participate in the Hough voting scheme. Unlike the 'good' features, features that match with the 'bad' features vote with a *negative weight.* They will vote *against* a certain orientation, preventing incorrect clusters from being obfuscated.

This unique approach allows the system to remain aware of all potentially hazardous objects in the scene. Combined with the innate ability of SIFT features and descriptors to discriminate between similar objects, this system should effectively avoid incorrect obfuscation.

4. Results

Preliminary investigation focused on determining the effectiveness of using SIFT features in this domain, as well as testing other feature detection methods.

Initial data was collected from relevant cable TV-broadcasts, using an ATI 9700 All-in-Wonder TV Tuner Card at an input resolution of 640x480. As part of the acquisition process, input data was automatically MPEG-2 compressed by the video capture card as it was digitized.

A variety of non-SIFT tracking algorithms were tested, including OpenCV implementations of Lukas-Kanade, Haar-based face detection, and Kalman filtering. All of these methods showed promising local tracking results, but appeared insufficient for the global tracking/reacquisition problem with multiple similar subjects. A solution providing improved lighting and transform invariance was required.

Initial SIFT algorithm tests were performed using two implementations: MATLAB [ETTINGER02] and the pre-compiled Linux implementation provided by Lowe [LOWE03B]. A performance comparison demonstrated an order-of-magnitude speed advantage for the precompiled version. The Lowe implementation took 7 seconds to detect SIFT features and generate normalized descriptors on a 720x480 video frame. The MATLAB implementation took 59 seconds for detection only. These tests were conducted on the same linux workstation running at 1 GHz, with 512 MB of RAM. Lowe's implementation also includes a rapid Best-Bin-First (BBF) comparison algorithm [BEIS97] for feature matching.

These tests illustrated several SIFT failure scenarios. The SIFT detection algorithm showed a distinct tendency to latch onto signal static and compression artifacts. This significantly reduced temporal and spatial coherence between detected features. Detection quality was further reduced by the fact that the image was a scaled result of a low-resolution analog television signal. These tests indicated that higher-quality input data was necessary.

Additional control data was collected using a Sony DCR-TC120 Digital Video Camera. Two gigabytes of DV-Compressed data was collected at 720x480 resolution. Collected data includes clips with varying:

- Indoor and outdoor lighting conditions, including transitions between them.
- Subject motions into and around camera view
- Subject rotations
- Camera motion/zoom
- Full/partial occlusion by objects and other subjects

Data was separated into clips and converted into uncompressed AVI format for import into MATLAB and subsequent SIFT processing. This data produced more promising results. The higher-resolution images yielded a larger set of more prominent SIFT features. Although DV compression caused some subtle artifacting, detected SIFT features had much stronger temporal coherence.

Nonetheless, results indicate that SIFT detection capability is reduced in the following scenarios:

- Changing lighting conditions
- Motion Blur

Controlled comparisons between video frames were used to determine how well the SIFT algorithm distinguishes between facial features of different subjects. SIFT generated positive matches when comparing two images of the same subject in different orientations (Figure 4).

In order to test the possibility of mismatches due to the exit of the target and subsequent entry of a similar subject, we performed a match test of two similar subjects in nearly identical poses (Figure 6). Few facial matches were produced, suggesting that SIFT may not be easily confused between different human subjects in a scene.

In additional tests, SIFT performed well in matching two different facial expressions by the same subject, as well as two images displaying the subject at a different scale (Figure 5).

A preliminary interface was also developed in MATLAB to allow a target region to be selected with the mouse (Figure 3). The interface enables a user to click and drag a rectangle around a target region. Upon releasing the mouse, a Gaussian blur is applied to the region to demonstrate the obfuscation effect, and the region coordinates are displayed.

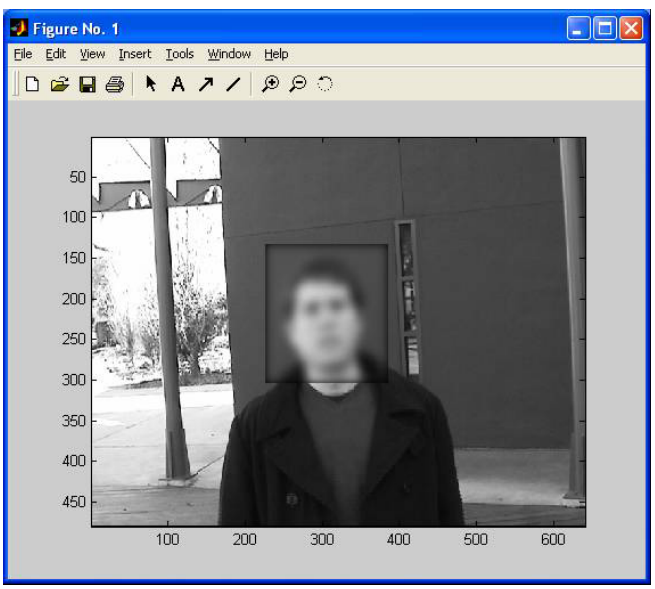

Figure 3: Screenshot of the preliminary user interface.

Figure 4: SIFT performs well when matching features across object rotation in depth at video resolution.

Figure 5: SIFT's invariance to non-rigid deformation is demonstrated by accurate facial feature matching. SIFT also correctly matches features across changes in scale.

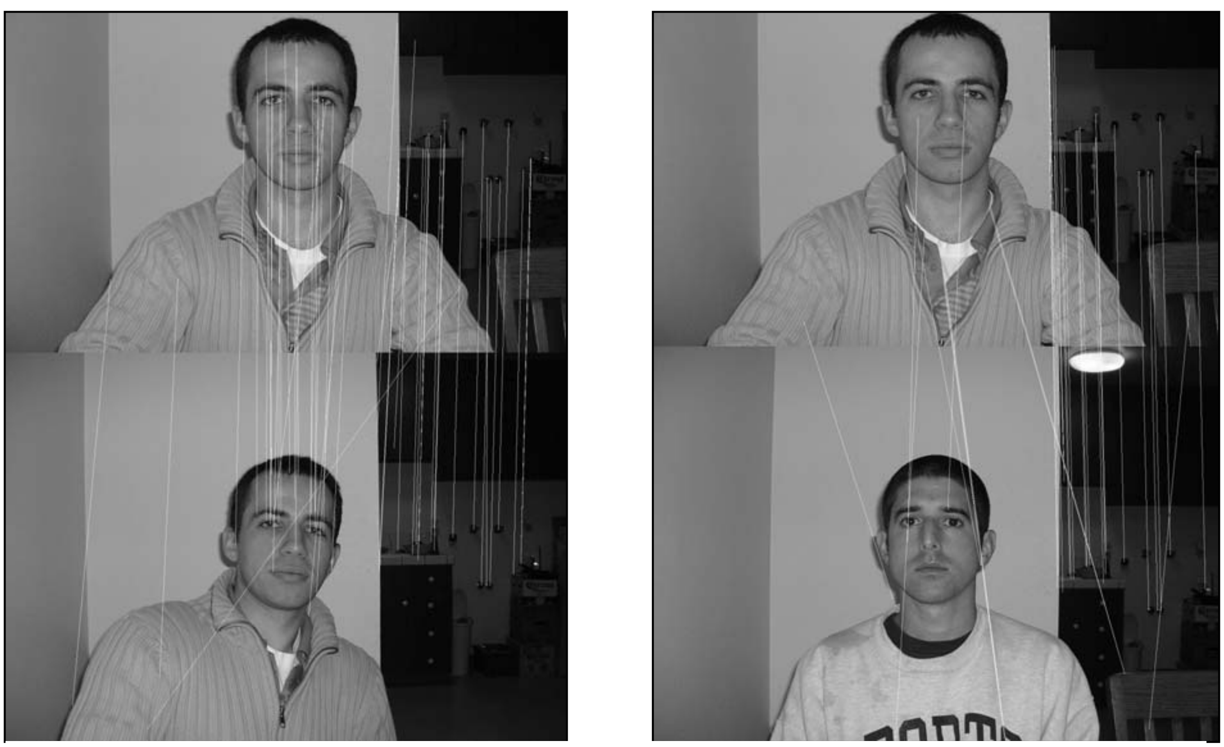

Figure 6: SIFT generates many matches between different poses of the same subject, but very few matches between different, similar subjects in the same pose.

5. Next Steps

The immediate next step involves writing an automated UNIX/MATLAB script that separates image frames from video, executes the [LOWE03B] feature detection program, and places the resulting matches and descriptors into MATLAB arrays. Once this data is readily available, algorithms can be rapidly tested as they are developed. A MATLAB script to interface with the [LOWE03B] comparison algorithm would also be useful. Utilization of these faster, precompiled utilities will allow development time to be devoted to novel aspects of the algorithms.

The following step will be to implement our modified affine warped Hough transform object-tracking algorithm with the positive/negative bin voting scheme as described in Section 3.

The new obfuscation-shape will be drawn and an implicit function calculated to determine whether newly-discovered features lie within its limits. A method to dynamically add new detected features to the two feature databases would be developed. The ability of the algorithm to detect prominent, recurring 'bad' features, will impact the negative weights needed for these features to prevent false-positive results. A slew of tests on all of the input data could be used to optimize these and other parameters:

- Feature prominence threshold
- Organization/range of Hough Transform bins; larger will be needed to account for non-rigid deformations
- 'Bad' feature negative Hough Transform weights
- Number of Hough Transform inliers below which object is considered out-of-view

If the affine-transformed ellipse does not consistently obscure the entire object, other methods will be needed. The gradient-based method proposed in [BIRCHFIELD97] could be implemented to fit an ellipse to the object's silhouette. An alternative approach would be to record the ratio of feature distances from the center of the ellipse and outline, respectively, for each new feature detected. Thus, after the target object is detected in a frame, the features that yielded this detection could automatically establish the dimensions of the new ellipse based on a best-fit algorithm which maintained the original distance ratios. This ellipse could be further optimized using the implementation described in [BIRCHFIELD97].

It is possible that the negative-voting scheme may cause 'bad' objects to be quickly ignored, but then intermittently reconsidered as the object sufficiently changes view. To prevent this, two voting passes may be needed: One using the positive/negative scheme to determine the primary target, and another using a positive/positive scheme to determine competitive clusters once the primary target has been identified. This will allow a more continuous tracking of bad clusters.

Initial results have shown that only a small percentage of detected SIFT features tend to be continually detected in subsequent frames. It can be assumed that the others are not sufficiently resilient to image variance. As such, an additional voting scheme could be implemented to 'retire' features in both databases that have only appeared in one frame. In addition to reducing computational complexity and storage, this may also reduce the chances of false positives due to noise.

A possible problem could arise if the target subject leaves the scene, and a similar-looking but different subject subsequently enters. Initial results indicate that few SIFT features (if any) are matched between two different faces with similar visual features; therefore, a heuristic which takes into account the number of positive matches achieved on the target subject in previous frames could be used to determine how many Hough inliers are needed to confirm a correct object match.

Several potential factors may still adversely affect the SIFT algorithm's ability to discriminate features. These include:

- Lighting changes
- Motion blur

If varying lighting conditions are not sufficiently compensated-for by the algorithm, a biologically-inspired method described in [ROSS00] could be used. By dividing Difference of Gaussian (DOG) points with a Sum of Gaussian (SOG), greater lighting robustness can be achieved. This would likely require reversion to the [ETTINGER02] SIFT framework, which does not currently build descriptors (this code would also need to be written). Motion blur can be reduced by increasing shutter speed via more expensive hardware. We propose the following potential solution as a pre-processing step, in software, if SIFT detection fails due to camera motion blur:

1. Compute optical flow using existing image pyramids and gradient-descent methods
2. Correct for motion blur in the optical flow using camera's intrinsic parameters as described in [TIMONER01]
3. Use a high-pass sharpening filter scaled along the direction of optical flow across the image.

This method can potentially provide far better results than a generalized high-pass filter over the entire image. An unscaled filter would likely cause sharpening artifacts in unblurred sections of the image, resulting in similar SIFT errors as found with compression. This filter guarantees that only moving areas in the image are sharpened. The sharpening is based upon a Gaussian model for motion blur.

6. Conclusion

SIFT features show promise as the basis of a solution for live video obfuscation, and are a preferred choice over other matching algorithms for their robustness and transformation invariance. In our tests, SIFT accurately matched numerous features between the target face in the initial image and transformed views of the target in subsequent frames, while producing few or no matches to other faces or objects in the scene. These results suggest that an implementation in which new features are identified and continuously added to the match database will be able to smoothly track and obfuscate a target object as it undergoes unpredictable transformations. In addition, a variation of the Hough transform clustering approach described by Lowe has been devised that will track both the primary target and similar objects, as a measure to prevent mismatches in future frames. Simultaneously, several potential algorithm failure scenarios have been identified, along with appropriate solutions.

7. References

[BEIS97] Beis J, Lowe DG. Shape indexing using approximate nearest- neighbour search in high-dimensional spaces. In: Conference on Computer Vision and Pattern Recognition, Puerto Rico 1997. pp. 1000-1006.

[BIRCHFIELD97] Birchfield, S. An Elliptical Head Tracker. 31st Asilomar Conference on Signals, Systems, and Computers. November 1997. Online: http://vision.stanford.edu/~birch/publications/headtracker_asilomar1997.pdf

[BRADSKI98] Bradski GR. Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal. Q2 1998.

[CAUWENBERGHS98] Cauwenberghs G, et al. Center-Surround Image Preprocessor for Contrast Enhancement. ONR/DARPA MURI N00014-95-1-0409. 1998. online: http://bach.ece.jhu.edu/~gert/muri/iuw98f.html

[CBC03] Canadian Broadasting Corporation. Journalistic Standards and Practices. 2003. Online: http://www.presscouncils.org/library/Canada_CBC.doc

[CHANG98] Chang P, Hebert M. Omni-Directional Visual Servoing for Human-Robot Interaction. Robotics Institute, Carnegie Mellon University, Pittsburgh 1998.

[ETTINGER02] Ettinger S. SIFT MATLAB Implementation. Intel 2002.

[LOWE03] Lowe D. Distinctive Image Features from Scale-Invariant Keypoints. Computer Science Department, University of British Columbia. Vancouver, B.C., Canada 2003.

[LOWE03B] Lowe D. Demo Software: Invariant Keypoint Detector Version 2.
Online: http://www.cs.ubc.ca/~lowe/keypoints/

[MIKOLAJCZYK03] Mikolajczyk K, Schmid C. A performance evaluation of local descriptors. 2003 Conference on Computer Vision and Pattern Recognition (CVPR '03) - Volume II.

[OPENCV03] Open Source Computer Vision Library (OpenCV). Intel Corporation, 2003.

[ROSS00] Ross WD, Grossberg S, Mingolla E. Visual cortical mechanisms of perceptual grouping. Neural Networks 13 (2000) 571–588.

[SCHNEID00] Schneiderman H, Kanade T. A Histogram-Based Method for Detection of Faces and Cars. Robotics Institute, Carnegie Mellon University, Pittsburgh 2000.

[TEXAS02] Texas Supreme Court Rules Advisory Committee. Report on Cameras in the Courtroom and Media Guidelines. 2002.
Online:
http://www.supreme.courts.state.tx.us/rules/committee/Sep-2002/2.2%20report.PDF

[TIMONER01] Timoner SJ, Freeman DM. Multi-Image Gradient-Based Algorithms for Motion Estimation. Department of Electrical Engineering and Computer Science. Massachusetts Institute of Technology. Cambridge, 2001. Online:
http://www.ai.mit.edu/people/samson/papers/Timoner_Freeman_MultiImageGradient.pdf

[VENTOURAS03] Ventouras, Elli. A cinematographic representation of the crimes of Sexual Assault and Domestic Violence. Masters Thesis. Parsons School of Design and Technology, 2003.
http://dt.parsons.edu/thesis_archive/m2003/Ventouras_Elli/thesis%20document.pdf

[WISKOTT97] Wiskott L, Fellous J, Krüger N, von der Malsburg C. Face Recognition by Elastic Bunch Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997.