Interactive Lossy Compression for Images and Video

Nathan Brewer
and Lei Wang
College of Engineering
and Computer Science
The Australian National University
Canberra, Australia, 0200
Email: nbrewer@cecs.anu.edu.au
lei.wang@anu.edu.au

Nianjun Liu
NICTA
Canberra, Australia
Email: nianjun.liu@nicta.com.au

Li Cheng
Toyota Technological Institute at
Chicago, USA
Email: licheng.atwork@ieee.org

***Abstract*—In any given scene, a human observer is typically more interested in some objects than others, and will pay more attention to those objects they are interested in. This paper aims to capture this attention focusing behavior by selectively merging a fine-scale oversegmentation of a frame so that interesting regions are segmented into smaller regions than uninteresting regions. This results in a new type of image partitioning which reflects in the image the amount of attention we pay to a particular image region. This is done using a novel, interactive method for learning merging rules for images and videos based on defining a weighted distance metric between adjacent oversegments. We present as an example application of this technique a new lossy image and video stream compression method which attempts to minimize the loss in areas of interest.**

I. Introduction

It is natural for a human being to focus on 'interesting' objects when scanning a scene, and ignore 'boring', or background information. Existing image segmentation methods tend to either segment an image into pieces of the same class or split the scene into areas of roughly equal size, regardless of how interesing the user finds each a particular region of an image. In this paper, we present a method which works with a user to determine how interesting parts of an image are to them and split the image into partitions which reflect both the interestingness of the partition and the differences between imaged objects.

The way humans perceive the environment is the subject of significant research in cognitive psychology. It is believed that the human perceptual system has a visual spatial attention property, which causes us to focus on a region of interest, extracting more information about this region than other parts of the scene [1], [2].

In this paper, we quantify this behavior by first fragmenting the image into small, similarly sized homogeneous regions. We then allow the user to define which of these segments are to be merged, as they belong to a 'boring' object, and which are to remain separate, either because they contain different objects, or an 'interesting' object. We use this information to learn a model which can merge these segments into a partitioned image with fine partitioning over interesting regions and coarse partitioning over boring regions, as shown in Fig. 1.

Fig. 1. (a) An image from the Berkeley image segmentation data set showing (b) The attention focussed partitioning generated with the ostrich as the interesting object and a 'boring' background.

The merging method that we present determines the distance between oversegments and merges them if this distance is below some threshold. Simply using Euclidian distances is insufficient for our goal, as the oversegments of an interesting object may have a similar distance between them as boring regions. As such, we define a Mahalanobis distance based metric similar to that proposed by Xing et al. [3] which makes use of user input to learn which features in an image are important to produce an attention focussed segmentation.

By retaining a fine 'mesh' of segments over interesting regions, we are able to discard pixel-level information about the image in favor of region averages while retaining a significant amount of information about the regions that are of interest. This allows us to generate a lossy compression which concentrates the loss in regions which are uninteresting to the user.

For the purposes of this paper, we want to generate a fine, relatively homogeneous segmentation of the image into superpixels, or superpixel-like oversegments. It is also important

that each of these oversegments contains an image portion that contains only one image object. As these oversegments will be used to interactively train a distance metric later in our process, it is also important that these oversegments are able to be easily recognized and separated by a human observer.

There are many possible methods for producing oversegments for an image, including Constrained Delaunay Triangulation (CDT) [4], Superpixels [5] and Superpixel Lattices [6]. CDT was ruled out for our purposes as it fails to achieve the required oversegment properties.

The Superpixel method by Mori [5], based on the Normalized Cut algorithm [7], produces similarly sized, typically homogeneous segments across the image. Superpixels are not constrained to any particular distribution across the image, and this lends itself well to ensuring that multiple image objects do not fall into the same superpixel. However, the irregularity in the oversegmentation makes it difficult to define a neighborhood for superpixels, and superpixel generation is quite slow.

The superpixel lattice presented by Moore et al. in [6] is a technique which produces a defined number of semi-regular superpixels arranged in a grid-like pattern across the image. This provides well-defined neighborhoods for all oversegments, and generation time is quite fast. Unfortunately, the predefined structure makes it difficult for this technique to resolve fine detail, and in a number of images the oversegments generated were not as useful for our purposes as Mori's superpixels.

Merging superpixels has been the subject of some previous exploration, however it has not been used to investigate compression in this body of work. Dunlop et al. [8] make use of multi-scale superpixel merging together with an appearance model for rocks to develop a robust method for identifying rocks in martian landscapes. Merging is also mentioned in Moore et al.'s work [6] on superpixel lattices, which differs from our approach in a number of key ways. First, they define a specific number of segments to merge to, not allowing the process to work its way to a natural conclusion, and second he does not allow the user to specify interest, but instead greedily removes the superpixel boundary with the lowest cost. The work of Ren and Malik [9] also deals with merging and splitting superpixels with the goal of generating a 'good' image segmentation, which differs from the goal of this paper which is to produce a segmentation which reflects the interestingness of a region.

II. Learning a Distance Metric for Superpixel Merging

The crux of our attention-focussing procedure is determining which features for a particular image or class are important when deciding to merge or separate two oversegments. This problem has been of significant importance in the clustering community for many years, and we take inspiration from their methods [3], [10]. We do this in a partially supervised process which allows the user to define similarity for the image class.

A. Oversegment Features

In order to determine whether oversegments should be merged, we must extract information about each oversegment in a consistent and sensible manner. For each oversegment, we determine a set of N features, which we define as a feature vector:

$$O_i = \{o_{i1}, o_{i2}, ..., o_{iN}\} \tag{1}$$

While this vector can contain arbitrary information, for this paper we have restricted it to features that are invariant to both rotation and scale. As oversegment size, shape and orientation are largely arbitrary using Mori's superpixels, we are unable to use this information when deciding to merge oversegments. For compression purposes, we want to perform this merging using single channel images, which further limits the amount of information that we are able to extract.

We use a 10-bin intensity histogram with constant bin widths for each oversegment in the image, defined by the range of intensities in the image as a whole. The histogram for each oversegment is normalized by the number of pixels present in the oversegment. Due to the way we define distance, we are able to simply store each bin of the intensity histogram as a separate entry in the feature vector. We also utilize a simple set of eight texture features, derived using the method presented by Varma and Zisserman in [11]. This texture set features rotational invariance and low texture dimensionality. We take the average response of each pixel in an oversegment to each of these eight features as a description of the texture of the oversegment.

B. Defining Pairs to Merge or Separate

A number of neighboring oversegment pairs are selected for both merging and separation in a training input image. It is important that this user input includes non-merge pairs for all differing adjacent object classes to avoid excessive undesirable merging and sufficient merge and non-merge pairs that all desired interesting and boring regions are annotated. We define a set of pairs to merge, S, as a set of observation couples O_i, O_j from user input. We similarly define a set of pairs to keep separate as D.

C. Distance Metric Learning

For any particular image or image class, some features will be more important than others for determining whether a pair of oversegments represent the same object. In order to do this, we require a method for learning the importance of each oversegment feature using the information that the user provides.

We can consider this problem as determining the Mahalanobis distance, given by (2), between two oversegments.

$$D_{ij} = \sqrt{(O_i - O_j)^T A^{-1} (O_i - O_j)} \tag{2}$$

We want this metric to weight particular features in object feature vectors so that oversegments to merge are closer under this metric than oversegments to keep separate. To do this,

we must learn a covariance-like matrix A with elements that correspond to the importance of that particular vector or combination of vectors.

Finding an A which does this is a difficult problem. Xing et al. [3] pose it as an optimization problem. This optimization problem is posed in two ways, a relatively simple method for finding a diagonal A given by (5) and an equivalent, but more difficult, problem for finding full rank A, given by (9).

$$\min_{A} \quad \sum_{(O_{i\neq j})\in S} \|O_i - O_j\|_A^2 \tag{3}$$

$$s.t. \quad \sum_{(O_{i\neq j})\in D} \|O_i - O_j\|_A \geq 1 \tag{4}$$

$$A \succeq 0. \tag{5}$$

Using 5, Xing et al. define (6) as an equivalent problem which can be efficiently minimized using the Newton-Raphson method.

$$g(A) = \sum_{O_i,O_j)\in S} \|O_i - O_j\|_A^2 - \log(\sum_{(O_i,O_j)\in D} \|O_i - O_j\|_A) \tag{6}$$

Solving this optimization is equivalent to finding A to within a multiplication of A, which does not affect the eventual result of the distance weighting.

$$\max_{A} g(A) \quad = \quad \sum_{(O_{i\neq j})\in D} \|O_i - O_j\|_A \tag{7}$$

$$s.t. f(A) \quad = \quad \sum_{(O_{i\neq j})\in S} \|O_i - O_j\|_A^2 \geq 1 \tag{8}$$

$$A \quad \succeq \quad 0. \tag{9}$$

In this case, gradient ascent on (7) and projection of A onto first the space constrained by (8) and then the space of all positive semi definite matrices. These steps must be performed in an iterative fashion to find the optimal A which does not violate either constraint.

As we are working with histogram features, it is important that we take into account the cross-correlation between histogram bins [12]. While we do not want to explicitly define color similarity, we want to learn a full-rank A based on user input that will implicitly account for the similarity of colors within each histogram. Additionally, learning a full-rank matrix allows our distance metric to capture richer information on the interaction of features in a particular image object.

We use the method proposed by Xing et al. rather than the Relevant Component Analysis method of Bar-Hillel et al. [10] as it allows us to explicitly select pairs as different, which enables more natural user interaction.

Increasing the number of features tends to improve the performance of the merging algorithm. It does, however, increase the complexity and duration of learning, and also increases the likelihood that the algorithm fails to locate the optimal solution within a reasonable number of iterations.

D. Determining a Distance Threshold for Merging

After determining the weighting matrix A, we want to determine a threshold for determining whether to merge or separate a pair of oversegments.

We do this by finding the distance between all oversegments that have been marked by the user. We define: D_M as the set of distances between oversegments marked to merge and D_S as the set of distances between those marked to separate. We then find the lowest value in D_S and the highest value in D_M. We set the merging threshold T_m at the mean of these two values. In the event that $\max D_M > \min D_S$, we alternate between discarding the highest value in D_M and the lowest value in D_S until there is no longer an overlap. This discards outliers in the training set which may be the result of input error in a simple fashion which results in improved performance.

Oversegment merging is performed iteratively. After an oversegmentation is merged, we repeat the distance calculation on the returned segmentation, treating this as a new oversegmentation to be merged. This allows for oversegments to grow based on their new properties, which generates larger segments in the image. The merging threshold is decreased with each iteration in order to limit the growth of the segmentation, and prevent the 'gradual shift' problem from allowing the entire image to merge into a single segment. We repeat this step until the image segmentation converges.

An advantage to this partially supervised method is that it does not require complex user mark-up of entire images to learn the distance metric, instead allowing the user to annotate only a relatively small amount of oversegment pairs in order to achieve the desired focussing behaviour. Additionally, the learned metrics can be used for several images of the same class, as seen in Section III. We have found experimentaly that a good focussed segmentation is returned from a variety of images when trained with between twenty and forty user input pairs, depending on the complexity of the scene. The number of merge and separate pairs that are required will differ based on the complexity of the image, and there do not need to be the same number of pairs in both sets.

E. Iterative Oversegment Merging

Given the Mahalanobis metric A and the maximum threshold for merging T, we can simply find the distance between each oversegment i and each of its neighbors j using equation (2), which we define as D_{ij}. If $D_{ij} < T$, then oversegments i and j are merged, otherwise, they are left as they were.

When merging it is common for many pairs to be merged together, resulting in non-neighbors being grouped into a single segment. While this is often desirable, as segments become larger errors occasionally occur in which oversegments which should remain separate become merged. This can lead to large sections of the image which contain different image objects being merged due to a single error. To account for this, we add an additional consistency check to the merging process. By ensuring that the distance between any segment that is to be added to an altered segment remains below the threshold

before adding it, we allow the algorithm to avoid merging very different oversegments together through gradual change.

Oversegment merging is performed iteratively. After an oversegmentation is merged, we repeat the distance calculation on the returned segmentation, treating this as a new oversegmentation to be merged. This allows for oversegments to grow based on their new properties, generating larger segments in boring parts of the image. The merging threshold is decreased with each iteration in order to limit the growth of the segmentation, and prevent the a gradual shift of segments towards the image mean from allowing the entire image to merge into a single segment. We repeat this step until the image segmentation converges.

III. Oversegment Merging for Video Stream Compression

Assuming we have a deterministic way of extracting oversegments using a single channel, we can transmit a single channel from the input video stream, perform the oversegment merging at both sides and then transmit the color information of each segment in terms of a segment mean. We make use of the fact that the user defines the segmentation, retaining finer detail in important or interesting regions, such as faces, which improves the visual results of this compression. The compression process works on a per-frame basis, transmitting a grayscale video stream and a set of recolouration parameters in place of the full color video stream. For this paper, we use H.264 to compress the greyscale video stream. Oversegments are generated from the compressed greyscale video stream using Mori's method at both the host and the client. The host transmits the merging parameters for the video sequence and a codebook containing the mean colour values of the merged oversegments for each frame.

On the transmitting computer (HOST), we extract the intensity channel of the NTSC image which we then transmit to the receiving computer (CLIENT) along with either the parameters A and T or the training pairs for this video stream. We then compute the oversegmentation of this frame using the method described in Section I on both machines. We then use the oversegment merging method detailed above to generate a focussed partition of the image on both machines. On HOST, we then find the color channel averages for each segment, which we transmit as a lookup table from HOST to CLIENT. Finally, we recolor the image on CLIENT by setting the color channels of each segment in the image from the lookup table. This process results in a full-color image for each frame at CLIENT which can then be reconstructed into a full-color video stream.

For many video sequences, we need to learn the merging parameters only for the first frame, which we are then able to use for subsequent frames in the video. In the event that the video stream contains disparate scenes which require different merging parameters, we are able to slightly modify the transmission procedure and send the required A and T parameters together with a list of which frames each should be used for.

The compression ratio depends largely on the complexity of the scene imaged and the amount of motion in the sequence. As our method utilisers the H.264 video compression method to compress the greyscale video stream, our method is also dependant on the factors which influence this, such as scene motion and complexity.

We measure compression accuracy quantitatively in two ways, with the Peak Signal to Noise Ratio (PSNR) [13] and the Explained Variance (EV), as used in [6]. The Explained Variance formulation has been modified slightly for this paper, as the intensity is known for the reconstructed image. Moore et al. present explained variation as:

$$R^2 = \frac{\sum_i (\mu_i - \mu)^2}{\sum_i (\mathbf{x}_i - \mu)^2} \tag{10}$$

where μ is the global pixel mean, x_i is the actual pixel value and μ_i is the average pixel value for the segment containing pixel i. We have adjusted the definition of μ_i to be the reconstructed pixel value at i, which includes the true intensity value rather than the segment average. We calculate R^2 in NTSC space.

As mentioned previously, the HSV intensity value for the image is transmitted exactly. To get a measure of the color accuracy without artificially boosting the result, we have modified both PSNR and EV to only measure the disparity between the two NTSC color channel values. As expected, we see lower PSNR and Explained Variance scores from these modified results, but the values are all acceptably high. Table I shows the average PSNR and EV values for a number of webcam streams across their duration, for both the full and color only formulations.

	H.264 Full Color	Unmerged Oversegment Compression	Merged Oversegment Compression
A	0.2663	0.2201	0.1747
B	0.2667	0.2445	0.1952
C	0.4612	0.3885	0.3446
D	0.8032	0.6411	0.6011

TABLE II
Bits per Pixel for each of the four video sequences under various compression schemes

Qualitatively, the recovered images look quite good. We lose information about small colored regions in the image, such as eye color, as the iris is usually contained in an oversegment with a large amount of skin even before merging. The majority of errors appear in background areas, particularly in areas toward the edge of the scene which are subject to camera vignetting. Colour quality in people's faces and expressions is quite high, and compression noise is rarely distracting to the observer.

To test the level of compression, a number of short (37-123 frame) webcam video sequences were taken with backgrounds of varying complexity. These videos were captured at 640x480 pixels in WMV format using an inexpensive Microsoft^{tm}

	A (123 frames)		B (73 frames)		C (65 frames)		D (36 frames)	
	Unmerged	Merged	Unmerged	Merged	Unmerged	Merged	Unmerged	Merged
PSNR	43.9263	39.7123	40.1235	35.1125	38.1745	36.2811	35.9390	34.5874
Color PSNR	42.2606	37.9662	38.3796	33.3579	36.4290	34.5311	34.1907	32.8342
EV	0.9942	0.9822	0.9951	0.9836	0.9917	0.9877	0.9874	0.9834
Color EV	0.9485	0.8524	0.9871	0.9553	0.9448	0.9123	0.7390	0.6436

TABLE I
THE AVERAGE PSNR, COLOR PSNR, EV AND COLOR EV ACCURACY METRICS FOR FOUR COMPRESSED VIDEO SEQUENCES. THE COLUMNS CORRESPOND TO FIGURE 2A-D RESPECTIVELY. WE COMPARE THE UNMERGED SUPERPIXEL COMPRESSION AND OUR MERGED SUPERPIXEL COMPRESSION PERFORMANCE

LifeCam webcam. The Bits per Pixel (BPP) values of these is shown in Table II. As can be seen, we achieve a BPP rate that is between 25.16% and 34.40% lower than full color H.264 encoding.

IV. SUMMARY AND FUTURE WORK

Our method is able to generate an image segmentation which produces fine segments over areas of interest while producing larger segments over boring regions using a simple superpixel merging technique inspired by the Mahalanobis distance.

We achieve high compression and a good quality decompression using a focussed lossy compression technique which makes use of the 'interestingness' of image areas as defined by user input to focus compression error on 'boring' regions.

Our present compression method treats each frame of the video as entirely independent when transmitting color information. Clearly, if we can track parts of the video stream which have the same color properties as a previous frame, we can reduce the codebook transmission from one per frame to one for the video sequence, or at least for a section of the sequence, which will increase the compression ratio.

Improving the underlying oversegment merging algorithm can also improve the performance of the algorithm, both in terms of the compression ratio and the quality of the reconstruction.

REFERENCES

[1] A. Martinez, W. Teder-Salejarvi, M. Vazquez, S. Molholm, J. J. Foxe, D. C. Javitt, F. Di Russo, M. S. Worden, and S. A. Hillyard, "Objects Are Highlighted by Spatial Attention," *J. Cogn. Neurosci.*, vol. 18, no. 2, pp. 298–310, 2006. [Online]. Available: http://jocn.mitpress.org/cgi/content/abstract/18/2/298

[2] J. Duncan, "Selective attention and the organization of visual information," *Journal of Experimental Psychology: General*, vol. 113, no. 4, pp. 501 – 517, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/B6X07-4NRKDG4-2/2/4c6b429f2300fc43bf74c278e0bbb09c

[3] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 505–512.

[4] X. Ren, C. C. Fowlkes, and J. Malik, "Scale-invariant contour completion using conditional random fields," *ICCV 2005*, vol. 2, pp. 1214–1221, 2005.

[5] G. Mori, "Guiding model search using segmentation," *ICCV 2005.*, vol. 2, pp. 1417–1423 Vol. 2, Oct. 2005.

[6] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," *CVPR 2008*, pp. 1–8, June 2008.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[8] H. Dunlop, D. Thompson, and D. Wettergreen, "Multi-scale features for detection and segmentation of rocks in mars images," *CVPR '07*, pp. 1–7, June 2007.

[9] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV 2003*, 2003, pp. 10–17.

[10] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *J. Mach. Learn. Res.*, vol. 6, pp. 937–965, 2005.

[11] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *ECCV*. Springer-Verlag, 2002, pp. 255–271.

[12] J. Z. Wang, *Integrated Region-Based Image Retrieval*. Springer, May 2001. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0792373502

[13] Y. Fisher, Ed., *Fractal image compression: theory and application*. London, UK: Springer-Verlag, 1995.

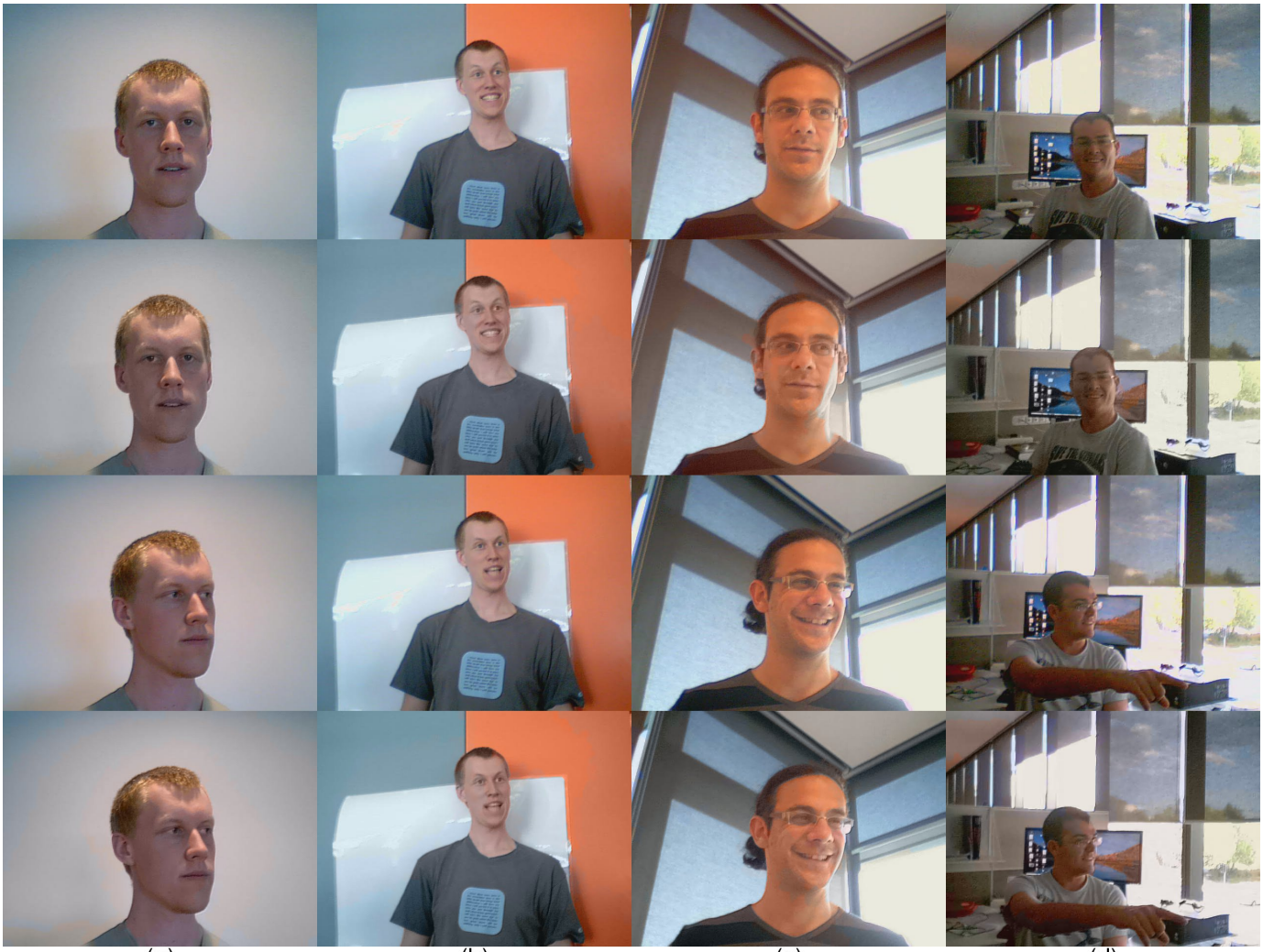

Fig. 2. (a-d) Images taken from a webcam with varying actors and backgrounds, showing Top: Training frame actual color, Second Top: Decompressed training frame, Second Bottom: Original non-training frame, Bottom: Decompressed non-training frame